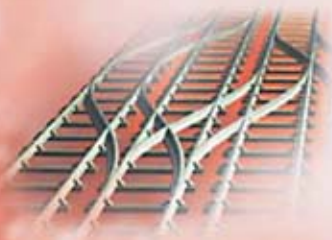
Solutions Réseaux

Innokenty Rudenko

# Configuration IP des routeurs Cisco

Plus de 400 scripts de configuration pour IOS



**Au sommaire**

- Pontage et routage statique
- Routage dynamique : RIP, IGRP, EIGRP, OSPF
- Contrôle de flux, filtrage et listes d'accès, redistribution
- Multicast IP (IGMP, PIM-DM/SM)
- Routage sélectif, NAT, routage à la demande (DDR), tolérance aux pannes (HSRP)

www.editions-eyrolles.com

Eyrolles

# Configuration IP des routeurs Cisco

# Configuration IP des routeurs Cisco

Innokenty Rudenko

Traduit de l'anglais par Alain Tamby et Gérard Mamou

Eyrolles

EDITIONS EYROLLES
61, Bld Saint-Germain
75240 Paris cedex 05
www.editions-eyrolles.com

Traduction autorisée de l'ouvrage en langue anglaise intitulé
*Cisco Routers for IP Routing*, de Innokenty Rudenko
The Coriolis Group LLC, ISBN 1-57610-421-4

Traduit et adapté de l'anglais par Alain Tamby et Gérard Mamou


Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissement d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'Éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands-Augustins, 75006 Paris.

© 2000, The Coriolis Group LLC. All rights reserved.

© Éditions Eyrolles 2001, pour la présente édition, ISBN 2-212-09238-5

ISBN édition Adobe eBook Reader : 2-212-28128-5, 2002

Distribution numérique par GiantChair, Inc.

*À ma femme Adelya, pour son amour et son soutien*

Innokenty Rudenko

# Remerciements

Je souhaite remercier les nombreuses personnes qui ont collaboré à cet ouvrage.

Tout d'abord, ma gratitude va à Elliot Goykhman, président de Tsunami Computing – c'est lui qui a pensé à l'écriture de ce livre. Il m'a soutenu tout au long de la rédaction de cet ouvrage en mettant à ma disposition tous les matériaux nécessaires.

Je remercie l'équipe de Coriolis qui a rendu ce projet possible, en particulier Michelle Stroup, éditeur en charge de ce projet. Le travail avec elle et Colleen Brosnan fut un plaisir ; je suis impressionné par le travail qu'elle a accompli et les progrès linguistiques que je lui dois. Je remercie également Stephanie Wall, Jennifer Watson et Meg Turecek.

Je souhaite remercier mes collègues de Tsunami Computing et J .P. Morgan. Howard Poznansky m'a abondamment conseillé sur les aspects linguistiques de la rédaction de l'ouvrage. Je remercie également Frank Kettles, sans qui la rédaction de cet ouvrage aurait été bien moins amusante.

Je remercie Gregg Messina et Mike Strumpf pour leurs conseils avisés sur l'équipement du laboratoire de test qui a servi à rédiger ce livre, et Boris Guzman pour son excellente relecture critique de certaines parties de l'ouvrage.

Je souhaite remercier Cornelius Hull, Anuj Kumar, Pat Coen, Roger Hampar, George Young, Carl Vitale, Mike Andrascik, Reginald Dancy, Ronnie Sun, Albert Mui, Julie Yip, Bill Hammill, Walter Sacharok, Dmitri Tcheverik, Valery Tsyplenkov, Artem Letko et tous les autres qui m'ont éclairé et encouragé.

Enfin, je remercie mon épouse Adelya, à qui ce livre est dédié, pour son amour, son soutien et sa patience tout au long de l'écriture de ce livre.

# Auteur

Innokenty Rudenko, diplômé en sciences informatiques, CCIE 3805, MCSE, est consultant senior chez Tsunami Computing, Inc. Spécialisé dans les réseaux basés sur les routeurs et les commutateurs Cisco, il travaille aujourd'hui en mission chez J.P. Morgan à New York.

# Table des matières



© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

## Annexes



© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

# Introduction

« Configuration IP des routeurs Cisco, encore un livre parmi tant d'autres ! », vous-êtes-vous dit en prenant en main ce livre. Détrompez-vous ; ce livre s'adresse tout d'abord aux praticiens qui, grâce à la lecture de cet ouvrage, mettront en œuvre plus facilement les routeurs Cisco dans un environnement opérationnel, par une meilleure connaissance des principes de fonctionnement d'un réseau moderne. Ils y trouveront des solutions innovantes et des exemples de problèmes rarement mentionnés ailleurs.

Les problèmes liés aux réseaux, du fait de leur nature distribuée, semblent plus complexes que ceux liés à un ordinateur isolé. De surcroît, les réseaux s'articulent autour de matériels dédiés tels que les routeurs, qui pour être des ordinateurs n'en sont pas moins très différents des ordinateurs usuels que sont les Macintosh, les PC voire les systèmes Unix. Ainsi, les routeurs Cisco, dotés d'un système d'exploitation et d'une interface de ligne de commande propriétaires, apparaissent-ils de prime abord comme des créatures difficiles à maîtriser.

Mais, les réseaux sont-ils vraiment complexes ? La réponse est non. Les principes sous-jacents sont en général abordables. Ils peuvent être explicités et le seront dans cet ouvrage. Déjà mis en pratique depuis quelques années, ils sont soumis depuis, à un processus constant de recherches et d'améliorations. De nos jours, prévoir le comportement d'un réseau n'est plus un problème dans bien des cas. Les solutions existent, *a fortiori* si l'on recourt à une approche systématique.

Les routeurs Cisco eux-mêmes ne sont pas si complexes. Ils le sont même moins que les PC sous Windows 98 ou NT, les Macintosh sous Mac OS, sans parler des systèmes fonctionnant sous Unix. Le nombre de commandes d'un routeur Cisco est limité et la plupart s'expliquent d'elles-mêmes. Si nous faisons une comparaison fonctionnelle entre ce que peut faire un PC et un routeur Cisco, ce dernier peut sembler un appareil très rustique. Et pourtant, les administrateurs de Windows 98 n'ont pas grand mal à maîtriser leur sujet.

Cet ouvrage traite de cas de configuration assez pointus que l'on ne peut trouver dans la plupart des autres ouvrages. Dans chaque exemple de configuration, il sera expliqué pourquoi

© copyright Éditions Eyrolles

chaque action doit être faite et comment celle-ci se justifie au regard des principes de fonctionnement des réseaux.

Nombreux sont ceux qui pensent que l'administration d'un réseau relève plus de l'art que de la technique – la technique étant vue comme une méthode d'investigation rationnelle, par opposition à un art où les solutions apparaîtraient comme par magie. Si tant est que le réseau fasse appel à l'art, ce ne peut être que d'une façon limitée.

Prenons l'exemple du schéma d'adressage qui utilise des masques de sous-réseau de taille variable, connu aussi sous le nom de VLSM (*Variable Length Subnet Mask*). Dans un tel cas, les protocoles de routage envoient en même temps que l'adresse réseau ou sous-réseau, le masque associé ; ce qui permet la coexistence de masques différents au sein d'une même adresse réseau.

Mais le VLSM n'a jamais été considéré comme une méthode d'implémentation aisée, en raison des problèmes de chevauchement et de gaspillage d'adresses qu'il peut entraîner. C'est pourquoi l'on tend à dire qu'il relève plus de l'art que de la technique. Or, c'est bien dans cet ouvrage une *technique* d'implémentation du VLSM qui est proposée pour minimiser les problèmes évoqués plus haut. Cet exemple parmi tant d'autres sert l'ambition de ce livre : rendre à l'administration de réseau un aspect plus technique qu'artistique.

Ce livre est aussi destiné à ceux qui voudraient obtenir la certification CCIE, car il présente des exemples de configuration fort utiles dans le cadre de la préparation des épreuves pratiques de cet examen.

## Organisation de l'ouvrage

Cet ouvrage est organisé selon la même structure que ceux de la série intitulée « Black Book » de Coriolis. Chaque chapitre comprend une partie théorique et une partie pratique intitulée « Solutions de configuration ».

La première partie tient lieu de bref rappel théorique sur le sujet traité, en se bornant à expliquer les points indispensables à la compréhension de la partie « Solutions de configuration ». Celle-ci propose des tâches à exécuter sur des cas pratiques qui servent d'illustration concrète du sujet exposé. Ces tâches varient en difficulté, allant du plus simple au plus complexe, pour donner une palette d'hypothèses aussi large que possible.

Cet ouvrage contient huit chapitres et six annexes allant de A à F pour un complément d'informations.

**Chapitre 1 : Le modèle de communication organisé en couches et le protocole Internet**

Le titre de ce chapitre n'est pas celui qui suscitera le plus d'enthousiasme dans l'esprit du lecteur. Cependant, il constitue un passage obligé car les concepts de base des réseaux auxquels il sera fait référence tout au long de cet ouvrage y sont décrits.

**Chapitre 2 : Le pontage avec les routeurs Cisco**

Ce chapitre traite de deux types de pontage que sont le pontage transparent et le pontage par la source. Bien qu'il s'agisse d'une digression par rapport au routage qui est le sujet principal de cet ouvrage, ce chapitre sert à dissiper la confusion sur le pontage en expliquant son fonctionnement et ses implications quand un routeur doit exécuter les deux fonctions (pontage et routage) simultanément. À ce titre, ce chapitre constitue une lecture indispensable.

© copyright Éditions Eyrolles

**Chapitre 3 : Routage statique**

Ce chapitre introduit les différents types de routage statique et contient une section consacrée aux algorithmes de routage et leur logique décisionnelle. Dans la partie « Solutions de configuration », on abordera plusieurs cas pratiques.

**Chapitre 4 : Routage dynamique avec protocoles à vecteur de distance**

Ce chapitre décrit les protocoles de routage dynamique les plus anciens et les plus répandus que sont ceux à vecteur de distance. On y aborde aussi la technique du VLSM. Dans la partie « Solutions de configuration », différents cas pratiques impliquant les protocoles RIP v1 et v2, ainsi que IGRP et EIGRP, sont exposés.

**Chapitre 5 : Protocoles de routage dynamique à état des liens**

Ce chapitre consacré au protocole OSPF présente différents cas de configuration, y compris le plus « honni » de Frame Relay sur des réseaux maillés, intégralement ou non. Certains cas particuliers de topologies à aires multiples, à aire dorsale sectionnée avec circuits virtuels, sont également expliqués.

**Chapitre 6 : Maîtrise du flux de données et des mises à jour de routage**

Le filtrage et la redistribution sont deux éléments primordiaux du routage. Ce chapitre présente différents scénarios avec pour chacun les risques associés tel que le phénomène de boucle.

**Chapitre 7 : Cas spéciaux de routage**

Ce chapitre traite du protocole de tolérance aux pannes ou HSRP (*Hot Standby Routing Protocol*), de la politique de routage, de la traduction d'adresses ou NAT (*Network Address Translation*), et du routage à la demande par ligne téléphonique ou DDR (*Dial on Demand Routing*).

**Chapitre 8 : Routage IP multicast**

Ce chapitre donne des indications sur la configuration de base pour la diffusion multicast (multidestinataire) IP qui comprend le protocole multicast indépendant ou PIM (*Protocol Independent Multicast*) en mode épars et dense.

## Comment utiliser cet ouvrage

Cet ouvrage est avant tout un guide pratique sur la configuration des routeurs Cisco : il peut être lu dans n'importe quel ordre.

Vos commentaires et suggestions sont les bienvenus. L'auteur peut être contacté par courrier électronique à l'adresse irudenko@hugewave.com. Tous les exemples de configuration, ainsi que les utilitaires tels que l'application de test sur la diffusion multicast IP, appelé MCASTER, sont disponibles gratuitement sur le site web de Tsunami Computing : **www.hugewave.com/blackbook**.

© copyright Éditions Eyrolles

# 1

# Le modèle de communication organisé en couches et le protocole Internet

**Solutions de configuration présentées dans ce chapitre**



Dans la première partie de ce chapitre, nous décrirons deux modèles de communication organisés en couches : le modèle de référence de l'OSI (*Open Systems Interconnection*) et le modèle Internet. Comme tout professionnel des réseaux, vous avez eu à étudier le modèle OSI et vous vous rappelez qu'il comprend sept couches, ayant chacune une fonction spécifique. Les modèles Internet et OSI se ressemblent, à quelques exceptions près. Nous allons donc, sans entrer dans le détail de chaque couche de ces deux modèles, nous intéresser aux aspects les plus importants. Nous les reverrons dans les chapitres suivants.

La seconde partie sera consacrée à la description du protocole Internet, également appelé IP (*Internet Protocol*). La manipulation d'un routeur Cisco nécessite une connaissance approfondie de IP, dont nous allons faire une présentation complète. Les lecteurs bien versés dans IP pourront considérer cette partie comme une référence à consulter. Les autres y trouveront tous les éléments essentiels pour bien comprendre IP et les sujets connexes.

Une grande partie du contenu de ce livre consacré à IP vient des documents Internet appelés RFC (*Request for Comments*). Les RFC sont une source précieuse d'informations, hélas trop

© copyright Éditions Eyrolles

souvent négligée. Ainsi, toutes les normes Internet relatives aux protocoles IP, TCP parmi d'autres, sont publiées en tant que RFC dans leurs moindres détails. De surcroît, les RFC sont gratuites et mises à la disposition du public. Quand la référence à une RFC s'avère nécessaire, il en sera fait mention sous la forme RFC XXXX , où XXXX correspond au numéro de la RFC. Sauf nécessité absolue, on n'utilisera pas d'autres sources d'informations.

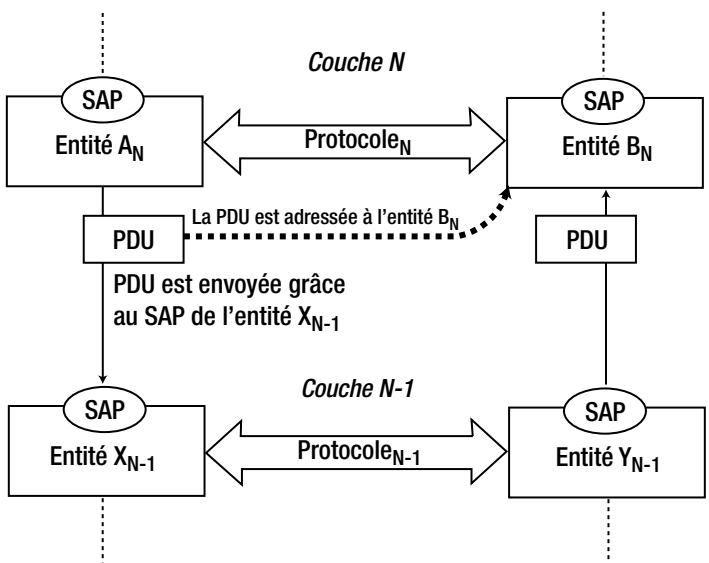
À ceux désireux d'en savoir plus sur les RFC, nous conseillons, pour commencer, de visiter le site web *www.rfc-editor.org.*

## Modèle de communication organisé en couches

Un tel modèle de communication comprend sept couches remplissant chacune des fonctions bien précises, complémentaires entre elles. L'implémentation d'une couche dans un nœud de réseau est désignée sous le nom d'entité. Lors d'une communication, cette entité communique avec d'autres entités de la même couche, qualifiées alors d'entités homologues. Les entités homologues, au lieu de se parler directement, utilisent les services fournis par les entités de la couche immédiatement inférieure. En d'autres termes, une entité de la couche (N) communique avec ses autres homologues en utilisant les services mis à disposition par l'entité de la couche (N-1). Les entités de la couche la plus basse communiquent directement au travers du support physique.

Un protocole est un ensemble de règles qui définissent le format des données échangées entre entités homologues et la façon dont se déroule cet échange. Chaque protocole définit l'unité de transmission des données, ou PDU (*Protocol Data Unit*), qui lui est propre. Une entité donnée ne peut être associée qu'à un seul protocole. Le point d'accès au service, ou SAP (*Service Access Point*), est le moyen par lequel une entité de la couche (N) utilise les services de la couche (N-1). Le modèle suppose aussi que seules les entités homologues comprennent le protocole utilisé, et qu'elles seules peuvent le décoder. Aucune des autres entités sur aucune des autres couches ne peut décoder les protocoles qui ne lui appartiennent pas. La figure 1.1 montre la relation entre couches, entités, protocoles, SAP et PDU.

**Figure 1.1**
*Protocoles, PDU et SAP.*



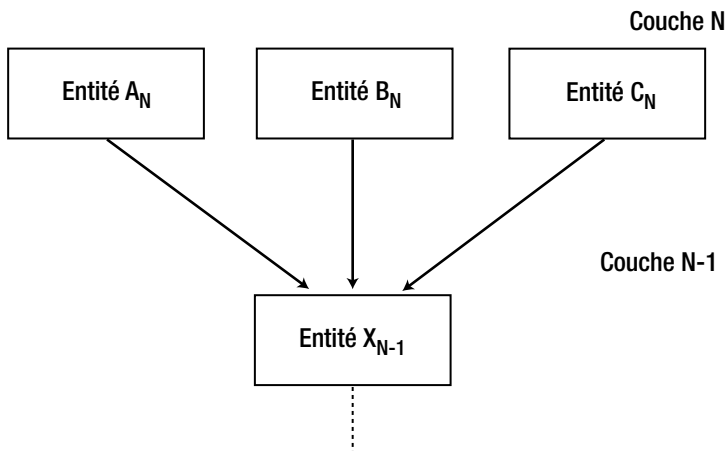

© copyright Éditions Eyrolles

On peut voir sur la figure 1.1 deux entités, A(N) et B(N) de la couche (N), et deux autres X (N-1) et Y(N-1) de la couche (N-1). Les entités A (N) et B (N) échangent une PDU en utilisant le protocole (N). Bien que l'entité A (N) adresse la PDU à l'entité B (N), elle ne peut le faire que *via* le service fourni par les entités X (N-1) et Y (N-1).

L'un des principes essentiels du modèle organisé en couches stipule que l'entité destinataire recevra sans altération la PDU envoyée par l'entité homologue émettrice. Ce principe, tout en facilitant l'indépendance des couches entre elles, rend également possible le fonctionnement des protocoles.

Le modèle organisé en couches permet à plusieurs entités de la couche (N) de coexister au sein d'un même nœud et d'utiliser les services d'une seule et même entité de la couche (N-1). C'est le mécanisme du multiplexage/démultiplexage. Ainsi, quand plusieurs entités de la couche (N) utilisent un SAP de l'entité de la couche (N-1), il s'agit du multiplexage. Le démultiplexage intervient quand une entité de la couche (N-1) distribue les PDU aux entités concernées de la couche (N). La figure 1.2 illustre le cas d'un multiplexage.

**Figure 1.2**
*Multiplexage.*




Le but des modèles de communication organisé en couches est de faciliter le développement du logiciel et du matériel de réseau. L'intégrité des données pendant les transferts étant garantie pour chaque protocole de chaque couche, le concepteur peut se concentrer sur un protocole à la fois, sans se préoccuper du fonctionnement des protocoles des autres couches.

Le modèle organisé en couches, grâce au multiplexage/démultiplexage, permet la conception modulaire du logiciel et du matériel. L'entité dans un tel modèle peut être un module réseau un pilote de périphérique (*driver*) ou même un équipement. Un tel modèle vous permet de charger séparément les modules réseau qui appartiennent à des couches différentes ; d'avoir plusieurs modules réseau fonctionnant à la couche (N) et d'utiliser un seul et même module de la couche (N-1).

Les termes entité et module réseau se recouvrent suffisamment pour être interchangeables. Nous emploierons le terme entité dans le cas général et celui de module pour désigner une entité d'implémentation particulière à un modèle donné.

© copyright Éditions Eyrolles

## Le modèle OSI

Dans la section précédente, nous avons fait part d'un modèle générique de communication organisé en couches qui n'a pas de grande signification pratique. Pour être utile, un modèle doit nous aider à concevoir des systèmes de communications répondant à des besoins. Il ne doit être ni trop sommaire, pour éviter aux concepteurs d'avoir à entasser trop de fonctionnalités dans chaque protocole, ni trop élaboré, pour ne pas faire peser trop de contraintes sur ses performances réelles.

Le modèle de référence de l'ISO fut introduit comme modèle pratique en 1984 par le comité de normalisation de l'ISO (*International Standards Organization*). Ce modèle comprend les sept couches suivantes :

- la couche application – activités specialisées de réseau comme le terminal virtuel, le transfert de fichiers et le courrier électronique ;
- la couche présentation – formatage de données, transcodage de caractère et cryptage ;
- la couche session – établissement de sessions entre un utilisateur et un nœud de réseau tel le login ;
- la couche transport – livraison des données de bout en bout, en mode sécurisé ou non ;
- la couche réseau – routage des PDU à travers des réseaux multiples ; gestion de la congestion intermédiaire ;
- la couche liaison – formatage des données en trames et leur transmission sans erreur à travers un réseau physique ;
- la couche physique – transmission d'éléments binaires ou bits (*binary digits*) sur le support physique de communication.

Aux fonctionnalités présentes dans tout modèle de communication organisé en couches, l'OSI ajoute la méthode d'*encapsulation de paquet* qui permet de conserver l'intégrité nécessaire des PDU échangées entre entités homologues utilisant les services fournis par les entités des couches inférieures.

La PDU de chaque couche, à l'exception de celle de la couche physique, est composée de deux parties : l'*en-tête* et les *données*. L'en-tête contient des informations annexes utilisées uniquement par l'entité ou le module particulier. Les données, quant à elles, sont reçues pour traitement par la couche immédiatement supérieure. Rappelons que le modèle organisé en couches garantit à l'entité destinataire, la réception intacte d'une PDU telle que l'a envoyée l'entité émettrice. L'OSI se conforme à cette règle, faisant en sorte que l'entité émettrice qui a construit la PDU, la passe dans son intégralité (en-tête inclus) sous forme de données, à l'entité immédiatement inférieure. Quand la correspondante de l'entité inférieure à l'autre bout effectue le démultiplexage des données vers l'entité de la couche supérieure, celle-ci reçoit exactement ce qui lui est destiné. Ce procédé est valable pour toutes les couches sauf pour la couche application qui reçoit, en fait, les données finales en provenance du réseau.

Vous remarquerez que l'encapsulation à la couche liaison ajoute à la remorque (*trailer*) un élément d'information de même type que l'en-tête, qui n'est utile qu'aux entités de cette couche. Cet élément n'ayant pas d'importance dans notre sujet sur l'encapsulation, nous ne le détaillerons pas plus.

On ne discerne pas encore clairement comment le multiplexage/démultiplexage est réalisé à travers le processus d'encapsulation. Celui-ci a pour rôle principal d'assurer l'intégrité des

© copyright Éditions Eyrolles

PDU lors de leur acheminement entre entités homologues. Il ne peut pourvoir aux besoins du multiplexage/démultiplexage. Cette dernière fonction, pour être remplie, nécessite donc un élément supplémentaire

Pour envoyer des données à l'entité de la couche inférieure, l'entité de la couche supérieure peut se contenter de connaître le SAP du service de la couche inférieure, à savoir l'interface à ce service. Plusieurs entités situées sur la même couche peuvent partager le service d'une même entité de la couche inférieure. Les SAP sont donc nécessaires à la fonction de multiplexage. Quand l'entité de la couche inférieure destinatrice reçoit les PDU, il lui faut associer chacune d'elle à l'entité destinataire de la couche supérieure. Comme le format des PDU, passé en tant que données à l'entité de la couche inférieure, est spécifié par le protocole des entités de la couche supérieure qu'elles sont seules à comprendre, il est impossible pour l'entité inférieure d'inspecter ces données pour en déterminer le destinataire. Le seul moyen d'identifier ce dernier est de marquer les données avec son propre en-tête. Ce marquage se fait par un identificateur que l'entité de la couche inférieure range dans l'en-tête de sa propre PDU. Cet identificateur qui permet précisément le démultiplexage, est souvent appelé *clé de démultiplexage*,.

## Le modèle Internet

Le modèle OSI, malgré sa définition assez exhaustive de la communication à couches, comporte quelques lacunes. Conçu à l'origine pour servir de cadre opératoire aux protocoles fonctionnant sur des réseaux locaux ou LAN (*Local Area Networks*), homogènes, il est peu adapté aux réseaux étendus ou WAN (*Wide Area Networks*). Si une fonction de routage est bel et bien spécifiée au niveau de la couche réseau du modèle OSI, sa description reste sommaire quant au rôle des routeurs, sachant que ces derniers constituent les nœuds permettant de relier des réseaux mixtes de bout en bout. L'autre modèle le plus utilisé est le modèle Internet, *alias* TCP/IP. À la différence du modèle OSI, le modèle Internet fut conçu pour servir de cadre opératoire aux protocoles fonctionnant sur des réseaux hétérogènes LAN et WAN.

Le modèle Internet comprend quatre couches :

- la couche application – assure des activités spécialisées de réseau comme le terminal virtuel, le transfert de fichiers et le courrier électronique ;
- la couche transport – assure la livraison de données de bout en bout, sécurisées ou non ;
- la couche Internet – assure le routage de données à travers des réseaux hétérogènes et un contrôle de flux rudimentaire ;
- la couche d'accès réseau – assure le formatage de données en trames et leur acheminement sans erreur à travers un réseau physique ; c'est là que s'effectue la transmission de bits sur un support physique de communication.

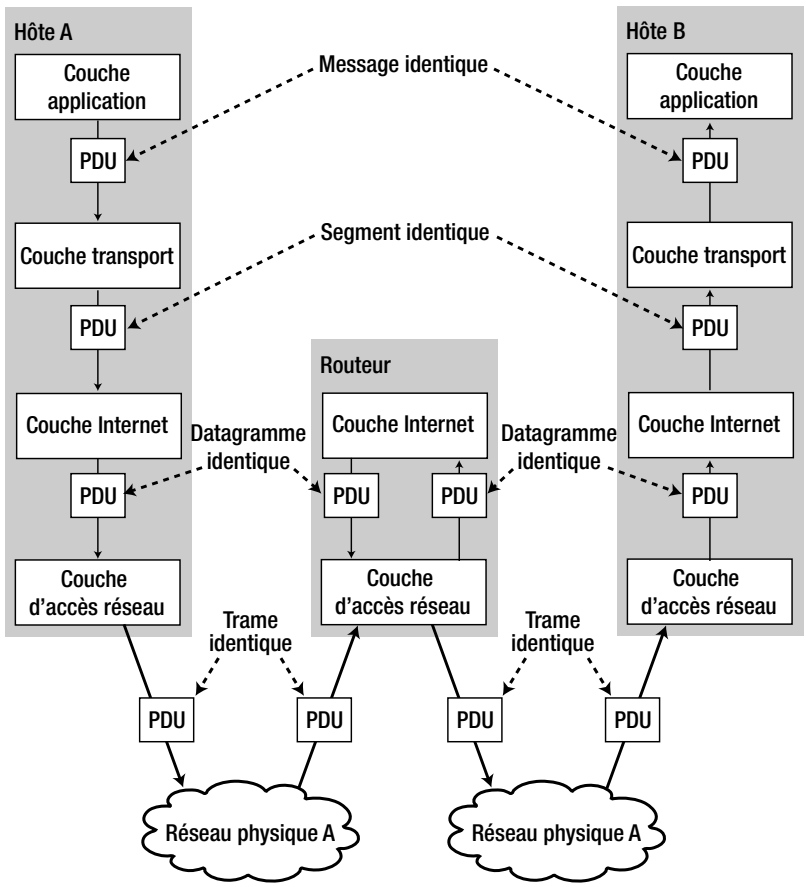
La fonctionnalité de ces couches est équivalente à celle de leurs homologues dans le modèle OSI. Il faut cependant remarquer que la couche d'accès réseau regroupe les fonctionnalités de deux couches (liaison et physique). De même la couche application peut recouvrir les couches session et présentation.

Le modèle Internet se différencie encore de celui de l'OSI pour ce qui est de la communication entre réseaux physiques divers, par l'introduction explicite du concept de *routeur.* Il possède deux types de piles à couches : l'une pour les nœuds terminaux ou « hôtes », dans la terminologie Internet, et l'autre pour les routeurs, autrefois appelés « *gateways* » – cette

© copyright Éditions Eyrolles

dernière dénomination date du début de l'ère Internet et ne manquerait pas de prêter à confusion aujourd'hui. Ces deux types de pile sont illustrés sur la figure 1.3.

**Figure 1.3**
*Deux types de pile dans le modèle Internet.*




Le modèle Internet emploie des noms distincts pour désigner les PDU de la couche Internet et celles de la couche transport : *datagramme* dans le premier cas et *segment* dans le second. Dans l'exemple de la figure 1.3, le datagramme créé par le module IP de l'hôte A transite par le module Internet d'un routeur avant d'être livré au module IP de l'hôte B, car A et B ne sont pas situés sur le même réseau physique. Ce qu'on ne voit pas sur la figure 1.3, c'est la modification que le routeur apporte à certains champs des en-têtes de datagrammes en cours de traitement. Si ceux-ci sont trop grands pour tenir dans la taille maximale de trame ou MTU (*Maximum Transfer Unit*) du réseau par lequel le routeur les expédie, il va s'employer à les découper en morceaux plus petits. En d'autres termes, le module IP de l'hôte B n'est pas sûr de recevoir intacts, les datagrammes créés par le module IP de l'hôte A.

Cependant, cette différence par rapport à l'OSI est moins prononcée qu'elle n'y paraît. Dans la figure 1.3 on voit que le datagramme reste intact en traversant le réseau physique séparant deux nœuds adjacents, que ceux-ci soient des hôtes ou des routeurs. Si le nœud récepteur immédiatement voisin est un routeur, il va modifier certains champs de l'en-tête du datagramme, voire le découper en plusieurs. Les nouveaux datagrammes qui en résultent traversent intacts le réseau intermédiaire suivant. Ce procédé n'a pas d'incidence sur le segment, de

© copyright Éditions Eyrolles

niveau supérieur, créé par le module de la couche transport et acheminé à l'intérieur des datagrammes ; ce segment arrive finalement intact, même après un parcours à travers plusieurs réseaux intermédiaires. Les routeurs n'ont pas de module de couche transport, aussi les segments créés à ce niveau dans l'hôte A ne peuvent-ils être destinés qu'à un destinataire de couche inférieure dans l'hôte B, en l'occurrence, un module Internet. Il en est ainsi à chaque traversée d'un réseau intermédiaire par le datagramme. Dans le modèle Internet, les PDU arrivent donc intactes à destination dans les couches respectives.

Pour tenir compte de ce nouveau schéma d'encapsulation, le modèle Internet distingue deux types de communication selon la couche concernée : le *bout en bout* ou *hôte à hôte* (*host to host*) et le *proche en proche* ou *saut en saut* (*hop by hop*). La communication de bout en bout suppose que les PDU du nœud émetteur sont expédiées vers le nœud récepteur sans se préoccuper du nombre de réseaux physiques intermédiaires à traverser. La communication de proche en proche n'intervient qu'entre deux nœuds situés sur le même réseau physique. Il est clair que la couche transport du modèle Internet assure la communication de bout en bout, tandis que la couche Internet assure la communication de proche en proche.

## Les composants invisibles

Jusqu'à présent nous avons étudié des composants appartenant aux deux modèles ; ils sont suffisamment explicites, dérivant du concept de modèle de communication organisé en couches. Il reste à évoquer les quelques composants invisibles qui jouent un rôle déterminant pendant tout le processus de communication.

Pour illustrer notre propos, nous allons nous en tenir à un seul modèle de communication organisé en couches, le modèle Internet. C'est le modèle que nous utiliserons généralement parce qu'il comporte l'usage des routeurs. Ailleurs, nous préférerons le modèle de l'OSI parce qu'il fournit une description complète des deux couches les plus basses : la couche liaison et la couche physique, alors que le modèle Internet les rassemble en une couche informe, celle de l'accès réseau, insuffisamment précise dans la plupart des cas.

La figure 1.4 montre un réseau très simple qui comprend deux nœuds, par exemple un PC sous Windows et un serveur Unix reliés par un câble Ethernet. Examinons de près ce qui se passe quand un utilisateur établit une session du PC vers le serveur par une connexion Telnet.

Pour se connecter au serveur, l'utilisateur du PC doit entrer la commande suivante :

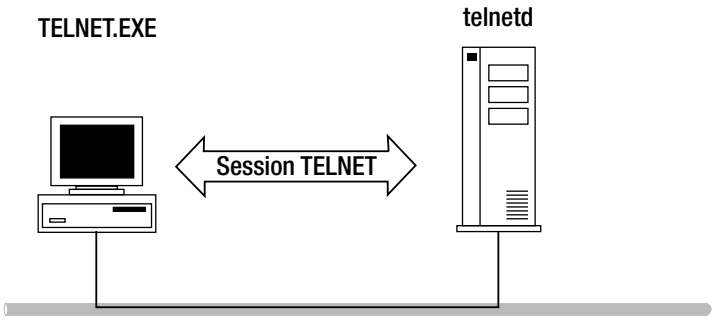
```
C:\>telnet 10.1.0.1
```

Dans ce cas, 10.1.0.1 est l'adresse IP du serveur. En utilisant cette adresse, l'application `telnet.exe` se connecte à l'application `telnetd` résidant sur le serveur Unix.

Les applications `telnet.exe` et `telnetd` sont des entités homologues qui communiquent en utilisant le protocole Telnet. Les deux applications doivent utiliser le service de communication d'un module quelconque de la couche transport. Les spécifications du protocole Telnet imposent l'utilisation de TCP (*Transmission Control Protocol*) qui intervient aussi bien du côté de `telnet.exe` que du côté de `telnetd` pour l'échange des messages Telnet à travers le réseau. Dans ce processus, le protocole Telnet est encapsulé dans des segments TCP, qui sont à leur tour transmis au module IP. Celui-ci encapsule les segments TCP dans des datagrammes et fait appel au pilote de la carte Ethernet pour les envoyer sous forme de trames sur le réseau physique.

© copyright Éditions Eyrolles

**Figure 1.4**
*Session Telnet entre un PC et un serveur Unix.*




Telnet n'est bien entendu pas la seule application à faire partie de la série de protocoles TCP/IP. On peut citer entre autres FTP, HTTP, Finger, etc. Pour effectuer le multiplexage/démultiplexage pour l'ensemble de ces protocoles applicatifs, TCP utilise des ports, qui sont stockés sous la forme de champs de deux octets dans les en-têtes des segments TCP.

Les ports TCP sont essentiellement des numéros de protocole de la couche application permettant au système d'exploitation de l'hôte d'identifier les modules correspondants. Les ports utilisés par TCP se divisent en deux catégories, les ports de destination, qui sont des numéros réservés – « connus de tous » (*well-known*), et les ports source qui sont des numéros adoptés aléatoirement (*random*). Les premiers identifient les modules qui exécutent des fonctions dans le serveur, comme répondre à des requêtes ; les seconds identifient les modules qui exécutent des fonctions de client, telles que le lancement de requêtes. Dans notre exemple de session Telnet, les modules client et serveur sont respectivement `telnet.exe` et `telnetd`. Le système d'exploitation alloue aux ports clients des numéros temporaires , à la différence des ports serveur, qui ont des numéros réservés invariables, quel que soit le nombre de modules actifs sur le serveur. Les ports serveur doivent en outre rester les mêmes d'un hôte à l'autre, en conformité avec TCP/IP, d'où leur qualificatif de « connus de tous ». Or, il est possible de lancer plusieurs sessions vers un même hôte. Les ports TCP ne suffisent donc pas à identifier de manière unique les modules d'application, du moins quand ceux-ci tournent sur un serveur. À titre d'exemple, si vous avez lancé plusieurs sessions vers un même hôte, le module TCP est incapable d'identifier quel module `telnetd` utiliser à partir des seuls ports TCP.

Le seul moyen d'identifier les modules d'une manière non ambiguë est de combiner les deux numéros de ports TCP avec les deux adresses IP. Ce procédé est utilisé par TCP lors du démultiplexage des connexions en retour vers les modules d'application. Quand le module TCP appelle le module IP, il utilise les adresses IP pour identifier les destinations vers lesquelles envoyer les segments. Or, comme vous le savez, TCP n'est pas le seul protocole de la couche transport dans TCP/IP ; le protocole UDP (*User Datagram Protocol*) en fait aussi partie. En principe, le modèle Internet n'est pas limité à ces deux seuls protocoles de transport ; il doit pouvoir mettre à disposition tout autre protocole de transport nécessaire à l'utilisateur. En d'autres termes, le module IP doit pouvoir identifier le protocole de la couche de transport auquel appartient le contenu des datagrammes entrants. Pour cela, le module IP utilise le numéro de protocole, rangé dans un champ d'un octet de l'en-tête du datagramme. Comme la présence de plusieurs instances du même module de la couche transport sur une seule et même instance du module IP est impossible, le module IP n'a aucune difficulté à identifier le module transport à partir du numéro de protocole.

© copyright Éditions Eyrolles

On pourrait croire que ce problème ne se pose pas au niveau de la couche d'accès réseau, car IP, constituant la pierre angulaire du modèle Internet, est le seul protocole à résider sur la couche Internet. Mais, TCP/IP n'a pas le monopole de l'utilisation de la couche d'accès réseau. Par exemple, il est de nos jours très fréquent d'avoir TCP/IP et SPX/IPX actifs sur la même machine. Le module de la couche d'accès réseau (par exemple, une carte réseau Ethernet et son pilote) doit donc aussi savoir lequel des modules de la couche Internet doit recevoir le contenu des trames entrantes. Encore une fois, une méthode semblable à celle de IP et de TCP est utilisée : un champ dans l'en-tête de la trame contient le numéro du protocole de la couche Internet auquel les données sont destinées.

Par cet exemple simple on s'aperçoit que le fonctionnement du modèle Internet n'est pas réellement complexe en lui-même, à la différence de certains de ses composants. Même si cet ouvrage n'a pour objet que l'étude des routeurs, les protocoles des couches transport et application restent d'une grande importance. En effet, les routeurs Cisco sont d'une maîtrise assez difficile, du fait que leur contrôle s'étend précisément aux modules de la couche transport, et même à ceux de la couche application résidant dans les hôtes. Dans cet ouvrage, vous serez confronté à des situations qui nécessitent de comprendre les autres couches, outre celles d'accès réseau et d'Internet.

## IP, protocole Internet

Dans le modèle Internet, le protocole du même nom (IP, *Internet Protocol*), exécute deux fonctions : le routage de datagrammes et un contrôle de congestion rudimentaire. La première fonction, le routage, est spécifique à IP. Aucun autre protocole, sur aucune autre couche du modèle Internet, ne peut effectuer le routage de paquets à travers des réseaux intermédiaires hétérogènes. Une fonction comparable appelée « pontage » (*bridging*), existe bien dans la couche d'accès réseau du modèle Internet et dans la couche liaison du modèle OSI, mais elle ne peut opérer que si tous les supports physiques interconnectés par le pontage sont presque homogènes (le « presque » sera justifié au chapitre 2). Contrairement au routage, l'autre fonction de IP, (le contrôle de congestion) est présente quasiment dans toutes les couches du modèle Internet. Celui de la couche Internet est qualifié de « rudimentaire » parce qu'il est très peu évolué, comparé à celui de TCP. La fonction principale de IP restant tout de même la livraison des datagrammes de l'expéditeur au destinataire, le cas échéant, à travers un grand nombre de réseaux hétérogènes.

Quand les hôtes sont situés sur le même réseau physique, ils peuvent communiquer en n'utilisant que les services de routage fournis par leurs propres modules IP. Mais, quand ces hôtes sont séparés par plusieurs réseaux intermédiaires, des équipements spéciaux appelés *routeurs* s'avèrent nécessaires.

Les routeurs sont des nœuds de réseau comportant plusieurs interfaces. Ils reçoivent des datagrammes en provenance des réseaux auxquels ils sont directement attachés, et grâce à l'adresse de destination, les dirigent vers les interfaces concernées.

**REMARQUE** Notons que les routeurs ne sont pas les seuls nœuds de réseau autorisés à avoir des interfaces multiples. Les hôtes *multihomed* ou « multidomiciliés », par exemple, le sont aussi. À cette différence près que les hôtes n'exécutent pas de fonction de routage sur les datagrammes.

Un exemple typique de routeur IP est le routeur Cisco capable par ailleurs d'exécuter d'autres fonctions. Les routeurs Cisco sont multiprotocole, dans la mesure où ils peuvent fonctionner sur une grande diversité de protocoles, tels que IPX, APPLETALK, DECNET, CLSN, etc.

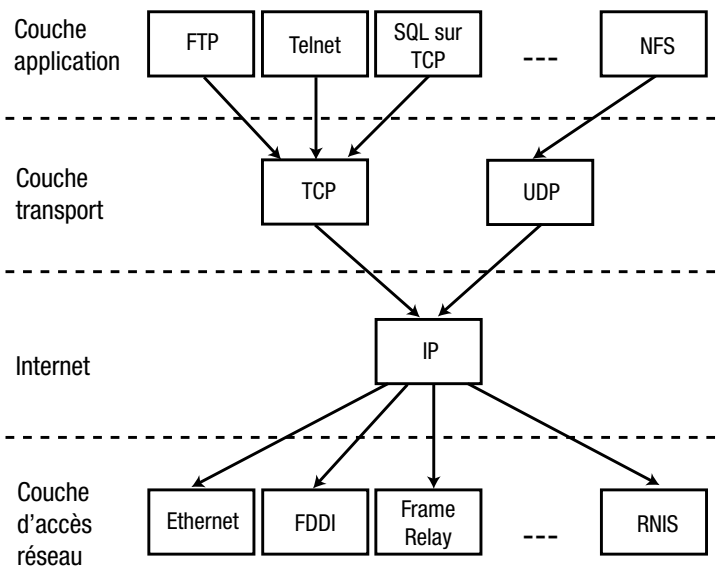
© copyright Éditions Eyrolles

Ces routeurs peuvent aussi jouer le rôle de ponts, c'est-à-dire opérer au niveau de la couche d'accès réseau. L'utilisation des ponts est décrite au chapitre 2.

## La suite de protocoles TCP/IP

La famille de protocoles conçue autour des deux protocoles principaux, TCP et IP, est souvent appelée *suite de protocoles TCP/IP*. La figure 1.5 donne une représentation graphique de la relation entre quelques-uns uns des protocoles de la série TCP/IP.

**Figure 1.5**
*Série de protocoles TCP/IP.*




Comme le montre la figure 1.5, IP occupe une position privilégiée dans la suite TCP/IP étant le seul protocole de la couche Internet à exécuter les fonctions de routage. D'autres protocoles auxiliaires, non représentés dans la figure 1.5, comme ICMP, IGRP, EIGRP, etc. sont présents dans la couche Internet, mais aucun d'entre eux ne fournit les services dont ont besoin les protocoles de la couche transport.

Le service original fourni par IP aux protocoles de la couche transport masque le détail des technologies sous-jacentes fonctionnant au niveau de la couche d'accès réseau, créant ainsi l'illusion que les hôtes sont séparés par un support physique homogène à « un saut ». C'est exactement ce que les protocoles de la couche transport attendent de IP et aucun autre protocole Internet supplémentaire n'est nécessaire pour l'instant.

## Les caractéristiques du service IP

La fonction de routage IP consiste à livrer les datagrammes de l'expéditeur au destinataire. Ce service comporte trois caractéristiques conceptuelles qui vont au-delà du modèle Internet, mais qui sont inhérentes à IP. Ces caractéristiques sont très importantes pour comprendre le comportement du routage IP dans différents scénarios :

- non connecté – ce service fait la livraison de chaque datagramme de manière absolument indépendante par rapport aux autres datagrammes ; ceux-ci peuvent emprunter des chemins différents, et arriver à destination hors séquence, etc ;

© copyright Éditions Eyrolles

- non sécurisé – le service ne donne aucune garantie que tout datagramme arrivera à destination en toute intégrité ; les datagrammes peuvent se perdre en chemin ou subir des avaries lors de leur parcours vers la destination finale ;
- au mieux (*best effort*) – le service « au mieux » signifie que IP fera de son mieux pour que les datagrammes arrivent à destination, sauf s'il est contraint, pour des raisons de baisse de ressources (saturation de la mémoire de file d'attente dans les routeurs, par exemple), de mettre au rebut une partie des datagrammes ; des cas de dysfonctionnement de matériel réseau ou de pilote peuvent avoir la même conséquence.

Comme tout protocole, IP définit le format de ses datagrammes. De ce que nous avons appris plus haut, nous savons que les datagrammes en cours de traitement dans les routeurs intermédiaires sont recréés. Ces nouveaux datagrammes reçoivent la partie données intacte, sauf en cas de fragmentation. Leurs en-têtes conservent la plupart des champs d'en-têtes des datagrammes originaux, sans altération . Tous ces changements, fragmentation comprise, sont le travail des routeurs. Nous allons à présent examiner le processus de transmission d'un datagramme de la source à la destination.

## *Un bref aperçu sur l'opération de routage IP*

Le modèle Internet, malheureusement, ne prend pas en compte le problème de la taille des PDU des différentes couches. Penchons-nous de plus près sur le processus d'encapsulation et essayons d'imaginer ce qui se passerait si l'implémentation suivait aveuglément le modèle Internet. (En toute équité, il est important de reconnaître que cette lacune n'est pas unique au modèle Internet ; l'OSI, ainsi que tout autre modèle « pur » organisé en couches n'en est pas exempt, non plus.)

TCP opérant dans la couche transport ne connaît pas la taille de trame maximale (MTU) du réseau sur lequel réside l'hôte, car cette information relève de la couche d'accès réseau. Il en est de même pour IP.

Si TCP ne faisait aucune hypothèse sur la MTU du réseau il construirait le plus grand segment possible de façon à minimiser la surcharge induite par la multiplication des en-têtes. Il passerait ensuite ce segment à IP, qui aurait à l'encapsuler dans son propre datagramme. Quand IP passerait ce datagramme au pilote de l'interface réseau, il y aurait de fortes chances que la transmission échoue, simplement parce que la MTU du réseau physique sous-jacent peut être inférieure à la taille du datagramme reçu.

Par conséquent, bien que cela puisse paraître comme une violation des principes de l'organisation en couches, les protocoles de la couche transport et IP doivent au moins avoir une certaine idée de la MTU des réseaux directement connectés. Car les segments créés à la couche transport doivent rentrer dans les datagrammes qui, à leur tour, doivent tenir dans les trames physiques. Théoriquement, la fonction de fragmentation dans IP, peut être utilisée pour découper les segments trop grands en morceaux, de façon à ce que chacun de ces derniers tienne dans une trame physique. Cependant, cette utilisation de la fragmentation est impraticable pour deux raisons. Tout d'abord, la fragmentation comme moyen de pallier la différence des MTU des réseaux intermédiaires n'est, de fait, implémentée que dans les routeurs : bien que les hôtes soient capables d'accomplir la fragmentation, ils n'en ont pas besoin. Ensuite, comme nous le verrons sous peu, la fragmentation mène à une détérioration des performances, et son utilisation abusive doit être évitée autant que possible.

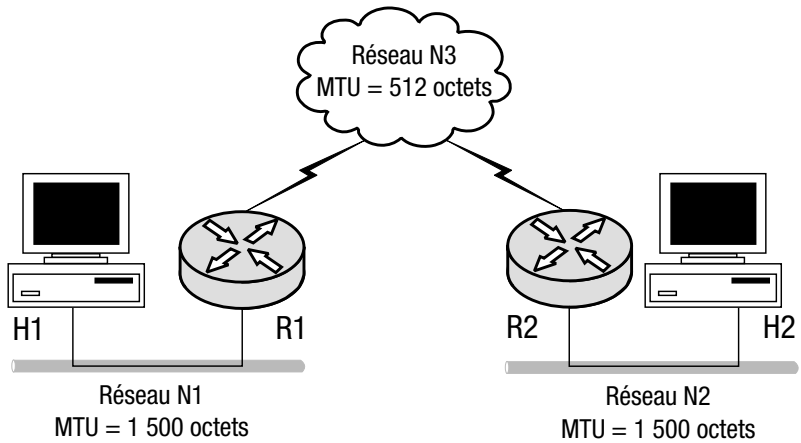
© copyright Éditions Eyrolles

Supposons que les protocoles de la couche transport et IP connaissent la MTU des réseaux directement attachés. Supposons aussi qu'ils utilisent cette connaissance pour créer leurs PDU respectives, de façon à ce que l'encapsulation dans une trame physique n'échoue pas. Que se passe-t-il, si un réseau intermédiaire entre la source et la destination, possède une MTU inférieure à celle des réseaux directement attachés ?

Ce cas est représenté sur la figure 1.6

**Figure 1.6**

*Datagrammes fragmentés quand ils traversent des réseaux dont la MTU est inférieure à celle du réseau de leur provenance.*




L'hôte H1 envoie un datagramme à l'hôte H2. Le premier sait que la MTU du réseau physique sur lequel il réside est de 1500 octets. Cependant, il ne sait rien du réseau N3 et de sa MTU. Par conséquent, le module TCP construit un segment dont la taille est seulement compatible avec le réseau N1. Quand l'interface réseau de l'hôte envoie la trame correspondante sur le support physique, la taille de la trame est de 1500 octets. Cependant, l'hôte H1 sait que la destination n'est pas sur le même réseau physique et qu'il doit recourir au routeur R1 pour expédier les données vers la destination finale. L'hôte H1 envoie la trame physique au routeur R1 qui, lors de sa réception, en extrait le datagramme et se rend compte que sa taille ne permet pas de l'envoyer vers le réseau N3 sans fragmentation. Le routeur R1 découpe donc le datagramme en quatre morceaux appelés *fragments* (nous saurons incessamment pourquoi il est découpé en quatre), et chacun d'eux est encapsulé séparément dans une trame physique du réseau N3 pour être envoyé au routeur R2. Ce dernier réalise que tous les quatre morceaux font partie d'un même datagramme plus grand. La MTU du réseau N2 est assez grande pour contenir le datagramme d'origine, et le routeur R2 se trouve donc devant le choix difficile soit de rassembler lui-même les morceaux, soit de laisser ce travail au destinataire. Selon les spécifications de IP, le soin de réassembler les fragments incombe à l'hôte destinataire. Le routeur R2 envoie donc tous les quatre fragments séparément sur le réseau N3 à destination de l'hôte H2 qui, lui, va se charger de les réassembler en datagramme d'origine. Ce processus se déroule dans le module IP de l'hôte H2. Le module TCP de celui-ci ignore qu'une fragmentation a eu lieu parce qu'il reçoit exactement ce que son expéditeur homologue a créé dans l'hôte H1.

Pendant le processus de fragmentation, le datagramme d'origine envoyé par l'hôte H1 a subi des modifications notables. Non seulement certains champs de l'en-tête ont été changés, mais le datagramme lui-même a dû être fragmenté pour se conformer à la MTU du réseau intermédiaire. Néanmoins, la couche transport de l'hôte H2, qui est TCP dans ce cas, n'a remarqué

© copyright Éditions Eyrolles

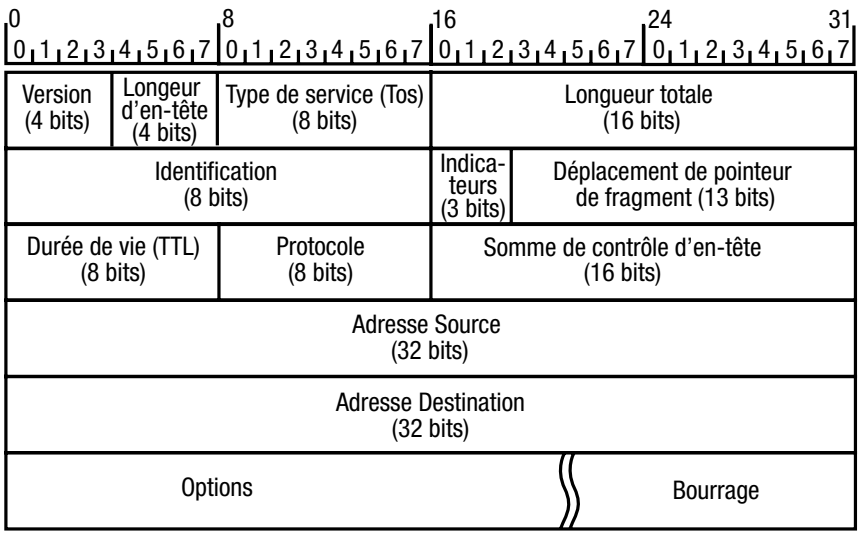
aucun changement dans le segment envoyé par son homologue de l'hôte H1. Tout ce processus s'est déroulé dans les modules IP des nœuds de réseau impliqués dans la transmission, donc de manière totalement transparente pour la couche transport.

Examinons maintenant le contenu du datagramme et voyons le changement qu'il a subi pendant le processus de transmission.

## Les datagrammes IP

Un datagramme IP comprend un en-tête et une charge utile (*payload*). Les données constituent la charge utile que le module IP reçoit des protocoles de niveau supérieur, tels que TCP ou UDP, pendant le processus d'encapsulation. L'en-tête, c'est l'information auxiliaire créée par le module IP que lui seul utilise pour le routage du datagramme vers sa destination finale. Nous avons vu plus haut, que IP est un service en mode non connecté. Ainsi, l'en-tête de chaque datagramme est conçu pour contenir tous les renseignements utiles aux décisions de routage indépendant.

**Figure 1.7**
*Format du datagramme IP*




### Version

Le champ version prend 4 bits et contient le numéro de version IP qui est actuellement de 4.

### Longueur d'en-tête *(IHL)*

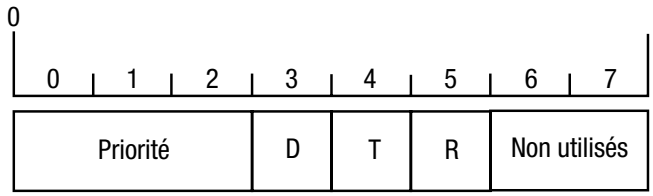
Le champ de longueur d'en-tête (*IP Header Length*) est la taille d'en-tête du datagramme exprimée en nombre entier de mots de 32 bits. Comme nous le verrons plus loin, le champ Options (selon qu'il est présent ou non), rend variable la taille d'en-tête du datagramme. Quand aucune option n'est présente, l'en-tête occupe 20 octets, soit une longueur d'en-tête de 5 mots.

### Type de service *(TOS)*

Le champ type de service (*Type Of Service*) spécifie la manière dont le datagramme doit être géré. Il comporte plusieurs sous-champs, chacun définissant la manière souhaitable de traiter le datagramme (voir figure 1.8).

© copyright Éditions Eyrolles

**Figure 1.8**
*Sous-champs de type de service.*




Le champ de priorité est le premier sous-champ du champ TOS (type de service). Il précise le degré d'importance du datagramme : plus sa valeur est élevée, plus le datagramme doit être traité en priorité (voir tableau 1.1). Ce champ comporte 3 bits, pour spécifier huit niveaux de priorité différents.

**Tableau 1.1. Valeurs de préséance.**

| Décimal | Binaire | Description |
|---|---|---|
| 7 | 111 | Contrôle réseau |
| 6 | 110 | Contrôle inter-réseau |
| 5 | 101 | CRTIC/ECP |
| 4 | 100 | Flash Override |
| 3 | 011 | Flash |
| 2 | 010 | Immediate |
| 1 | 010 | Priority |
| 0 | 000 | Routine (Normal) |

Les trois autres champs – dénommés D (*Delay* ou délai), T (pour *Throughput* ou débit) et R (pour *Reliability* ou fiabilité) dans la figure 1.8 – sont binaires et spécifient le critère d'évaluation de coût optimal du chemin par lequel envoyer le datagramme. Comme le montre la figure 1.8, ces champs n'ont qu'un bit, qui peut prendre la valeur de un (vrai) ou zéro (faux). Si le bit D est à un, le datagramme est envoyé par le chemin de délai minimal. Si le bit T est à un, le datagramme est envoyé par le chemin de plus haut débit. Enfin, si le bit R est à un, le datagramme est envoyé par le chemin le plus fiable. Lorsqu'un champ a pour valeur zéro, le critère correspondant est ignoré.

Conceptuellement, le type de service est une idée fertile. Si le routeur possède plus d'un chemin vers une destination, il est en mesure de choisir celui qui remplit la condition requise. Par exemple, si vous utilisez Telnet pour communiquer avec un hôte distant, vous vous attendez à une réponse rapide de l'hôte. Vous aurez donc à positionner le bit D à un pour tous les datagrammes de toutes les sessions Telnet, de façon à forcer la connexion pour qu'elle prenne le chemin de délai minimal. Au contraire, si vous voulez télécharger un fichier volumineux *via* FTP, ce n'est pas le temps de réponse qui vous importe, mais la rapidité de téléchargement des fichiers. Dans ce cas, vous aurez à positionner le bit T à un pour obtenir un meilleur débit pour la connexion FTP.

Malheureusement, cette fonctionnalité n'est en général pas assurée par les routeurs.

© copyright Éditions Eyrolles

### Longueur totale

Le champ de longueur totale indique la taille du datagramme en nombre d'octets. Ce champ comporte 16 bits, ce qui permet en théorie d'avoir des datagrammes ayant jusqu'à 65 535 octets de longueur. Mais comme nous le savons, des datagrammes aussi grands seront probablement rejetés par le pilote d'interface réseau, incapable de les encapsuler dans des trames au niveau du réseau physique. La taille effective des datagrammes ne doit pas dépasser la MTU des réseaux physiques sur lesquels ils sont envoyés. Néanmoins, tous les hôtes et les routeurs sont obligés d'accepter des datagrammes dont la taille est de 576 octets ou moins.

### Identification

Le champ identification, à l'instar des deux champs suivants, contribue au processus de fragmentation et de réassemblage des datagrammes. Avant de les décrire, il est important de noter que les fragments de datagramme sont aussi des datagrammes. Leur structure est la même que celle des datagrammes IP ordinaires.

Le champ identification occupe 16 bits et sert à repérer les fragments appartenant à un même datagramme. Chaque fois que l'entité source crée un datagramme, elle attribue un numéro unique à son champ identification. Quand un routeur juge bon de fragmenter un datagramme, il recopie la plupart des champs de l'en-tête de ce dernier dans celui des fragments apparentés. Le champ identification est également reporté dans chaque fragment, procurant ainsi à l'hôte destinataire le moyen de reconnaître les fragments faisant partie du datagramme d'origine.

### Indicateurs

Le champ indicateurs (*flags*) occupe 3 bits et sert uniquement à la fragmentation. Chaque bit est interprété indépendamment, comme suit :

- le bit 0 est réservé ;
- le bit 1, aussi appelé DF (*Don't Fragment*), signifie « ne pas fragmenter » ; positionné à 0, il indique l'autorisation de fragmenter.
- le bit 2, aussi appelé MF (*More Fragments*), signifie « encore des fragments » ; positionné à 1 il indique que d'autres fragments suivent ; postionné à 0, il indique le dernier fragment ; bien entendu, ce bit reste positionné à 0 si le datagramme n'est pas fragmenté.

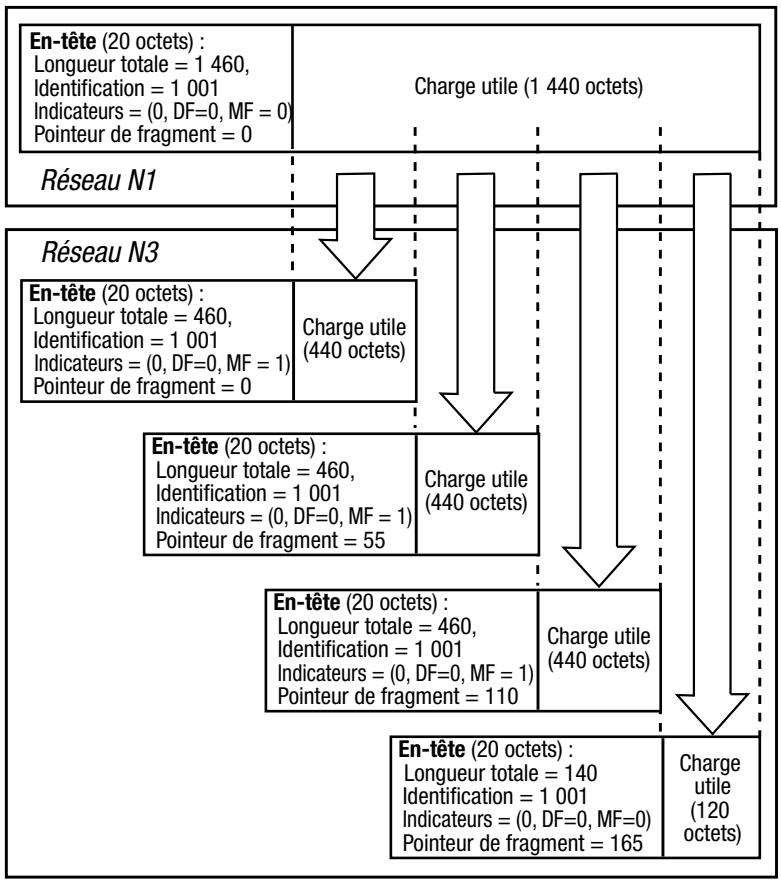
### Pointeur de fragment

Le champ pointeur de fragment (*Fragment Offset*) occupe 13 bits et sert de compteur (par groupe de huit octets) pour calculer le déplacement de la charge utile du fragment courant dans le datagramme d'origine. La figure 1.9 montre comment le datagramme d'origine est fragmenté avant d'être envoyé sur le réseau N3 ; on peut y voir aussi l'en-tête et la charge utile du datagramme ainsi que ceux des fragments ; seuls les champs d'en-tête concernés par le processus de fragmentation y sont représentés.

Pour mieux comprendre une opération de fragmentation, et quels sont les champs modifiés lors de son déroulement, reportons-nous à la figure 1.6. Supposons que le routeur R1 reçoive des datagrammes d'une taille de 1460 octets en provenance de l'hôte H1. Le routeur R1 aura à fragmenter ces datagrammes pour se conformer à la MTU du réseau N3, qui est de 512 octets. Supposons aussi que la couche d'accès réseau prélève 32 octets des trames pour son propre usage, ne laissant ainsi que 480 octets pour IP.

© copyright Éditions Eyrolles

**Figure 1.9**
*Opération de fragmentation du routeur R1 illustré sur la figure 1.6*




Bien entendu, la fragmentation nuit au rendement dans une transmission. Si les fragments ont à traverser un réseau intermédiaire dont la MTU est plus grande que celle des fragments, ceux-ci n'en utiliseront qu'une partie. Par exemple, dans la figure 1.6, le routeur R2 aura à envoyer les petits fragments qu'il reçoit du routeur R1 sur le réseau N2 dont la MTU est au moins trois fois supérieure à celle des fragments. Très souvent, le dernier fragment se trouve être bien plus petit que ce que le réseau physique peut admettre. Dans notre exemple, la MTU du réseau N3 est de 512 octets, ce qui autorise une taille maximale de 480 octets pour un datagramme. Cependant, le dernier fragment ne contient que 120 octets de données. Pendant la fragmentation, l'en-tête d'origine est remplacé par plusieurs autres, ce qui augmente le coût. Dans notre exemple, cela revient à créer quatre nouveaux en-têtes à la place de celui d'origine[1].

### Durée de vie (TTL)

Le champ durée de vie (*Time To Live*) représente la durée de validité du datagramme, avant que l'un des routeurs le rende obsolète. Le terme durée de vie est un peu inapproprié car il ne met pas uniquement en jeu une unité de temps, comme la seconde.

1. De plus, le fait de compter en multiples de 8 octets pour calculer le déplacement implique un mauvais remplissage de la charge utile qui, toujours selon la figure 1.6, a une taille maximale permise de 460 octets, alors que les trois premiers fragments n'en utilisent que 440.

© copyright Éditions Eyrolles

La durée de vie est calculée de la manière suivante :

1. Quand un routeur reçoit un datagramme, il décrémente le champ durée de vie de un.
2. Si la valeur du champ tombe à zéro, le routeur met le datagramme au rebut ; sinon, il l'envoie soit au routeur du saut suivant (*next hop*) soit au destinataire final.
3. Si le routeur est obligé de stocker le datagramme plus d'une seconde, il décrémente le champ de un à chaque seconde.

Le but principal du champ durée de vie est d'éviter la congestion dans un réseau en cas de boucle. Imaginons un datagramme envoyé le long d'un chemin, qui en fin de compte le ramène à son point de départ. Si le champ durée de vie n'existait pas, le datagramme circulerait en permanence.

### Protocole

Le champ protocole occupe un octet, et peut donc servir à identifier 255 protocoles différents utilisant le service IP et destinataires de la charge utile du datagramme. Le tableau 1.2 donne quelques valeurs de ce champ avec les protocoles correspondants. Il est à noter que ces protocoles n'appartiennent pas tous à la couche transport. Des protocoles auxiliaires comme ICMP (*Internet Control Message Protocol*) et EIGRP (*Enhanced Interior Gateway Routing Protocol*) peuvent recourir aussi au service IP.

**Tableau 1.2. Quelques protocoles et leurs valeurs.**

| Valeur de protocole en hexadécimal | Valeur de protocole en décimal | Nom du protocole |
|---|---|---|
| 6 | 6 | Transmission Control Protocol (TCP) |
| 11 | 17 | User Datagram Protocol (UDP) |
| 1 | 1 | Internet Control Protocol (ICMP) |
| 9 | 9 | Cisco Interior gateway Routing Protocol (IGRP) |
| 58 | 88 | Cisco extended IGRP (EIGRP) |
| 59 | 89 | Open Shortest Path First (OSPF) |

### Somme de contrôle

Le champ somme de contrôle (*header checksum*) contient uniquement la somme de contrôle de l'en-tête. Toute modification en chemin de la charge utile n'est pas détectée par IP. Il appartient au protocole de la couche supérieure de procéder à sa propre vérification d'intégrité des données, et de demander une retransmission en cas d'avarie.

### Adresses destination et source

Les adresses destination et source désignent respectivement l'adresse IP de l'hôte émetteur et celle de l'hôte censé recevoir le datagramme (voir section suivante).

### Options et Bourrage

Les champs options peuvent servir éventuellement à une mise au point logicielle ou débogage. Ces champs n'ayant que peu d'utilité en pratique, ils sont rarement utilisés.

Le champ bourrage sert à aligner les options sur une limite de mot (32 bits).

© copyright Éditions Eyrolles

## Les adresses IP

Presque tous les utilisateurs d'ordinateurs ont entendu parler et même utilisé des adresses IP. Mais qu'est-ce qu'une adresse IP ? Est-ce une manière d'identifier un hôte dans un réseau conforme à TCP/IP ? Que se passe-t-il si un hôte a plusieurs cartes d'interface réseau ? Dans ce cas, devrait-il avoir plusieurs adresses IP, une pour chaque interface ? Serait-ce donc une manière d'identifier plutôt l'interface réseau d'un hôte résidant dans un réseau conforme à TCP/IP ?

Nous admettrons qu'une adresse IP appartient au protocole Internet, qui fait partie de la couche du même nom. Partant de cette hypothèse, en quoi une adresse IP serait-elle concernée par l'interface réseau, sachant que cette dernière appartient à la couche d'accès réseau ?

On ne peut justifier pleinement pourquoi il fut décidé que chaque hôte IP devait posséder une adresse IP pour chacune de ses interfaces réseau. Tout ce qu'on peut constater, c'est qu'un hôte ayant une adresse IP spécifique pour chacune de ses interfaces réseau prend autant d'identités.

Les adresses IP servent **aussi** à identifier les réseaux sur lesquels résident les hôtes. L'adresse IP, d'une taille fixe de 32 bits, est divisée en deux parties de taille variable. Elles permettent l'attribution d'une identité unique au réseau (*network ID*) et à l'hôte (*host ID*). Pour simplifier, la première désigne le support physique auquel est rattaché l'hôte, et la deuxième, l'hôte lui-même.

Le format utilisé pour les adresses IP, pour qu'elles soient d'une lecture facile, s'appelle notation décimale pointée. Dans cette notation, les 32 bits de l'adresse IP sont condensés en quatre chiffres décimaux séparés par des points.

```
Adresse IP en binaire : 11010000100000010000000111000011
Notation décimale pointée : 208.129.1.195
```

### *La conception de l'adresse IP et son évolution*

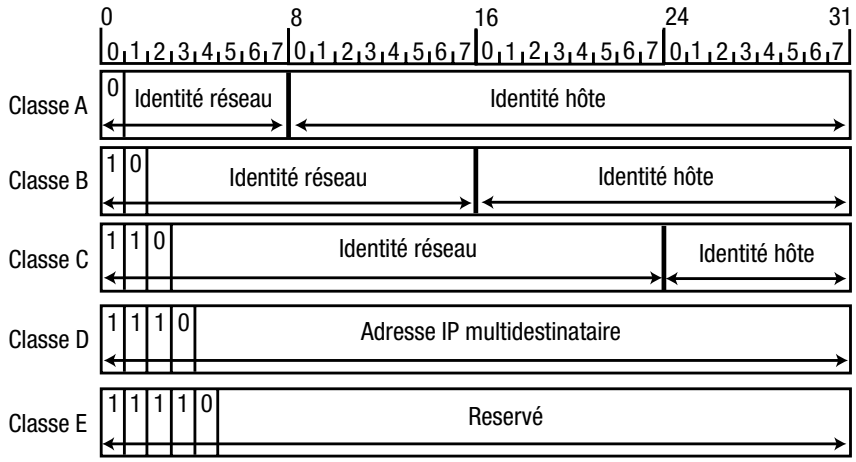
Au début de sa conception, le protocole TCP/IP supposait que la plupart des communications IP seraient monodestinataire (*unicast*). Cependant, ses concepteurs avaient prévu que certains hôtes auraient besoin d'envoyer des messages à plusieurs destinataires d'un coup (*multicast*). Il fut donc décidé de diviser l'espace d'adressage en trois catégories inégalement réparties, une catégorie d'adresses monodestinataire, une catégorie d'adresses multidestinataire (*multicast*) et enfin une catégorie d'adresses réservées. La part la plus importante de cette partition était dévolue à la catégorie monodestinataire, elle-même subdivisée en trois classes disjointes (une adresse IP ne pouvait appartenir qu'à une seule classe à la fois). L'appartenance à une classe ou à une autre déterminait la partie de l'adresse IP à interpréter comme étant l'adresse réseau. Il en va différemment des adresses multicast qui représentent un ensemble d'hôtes et ne présentent pas cette subdivision.

Chacun des groupes, à savoir d'une part les trois classes d'adresses unicast, et d'autre part les adresses multicast et réservées, ont été respectivement appelées *classes de réseau* A, B, C, D et E. Pour éviter toute confusion, on parlera simplement de classes d'adresses (sans mentionner le terme de réseau) pour les adresses individuelles, de classe A, B ou C.

L'arrangement des bits du premier octet détermine la classe à laquelle appartient une adresse IP, comme illustré sur la figure 1.10.

© copyright Éditions Eyrolles

**Figure 1.10**
*Classes de réseau IP*




Il existe cependant certaines dérogations à ce schéma, sous la forme d'adresses fonctionnelles, qui sont détaillées dans la RFC 1700, *Internet Assigned Numbers*).

Par convention (cf. RFC 1700), une adresse réseau dont tous les bits sont à zéro désigne de manière réflexive le réseau sur lequel se trouve l'ordinateur. Cette adresse particulière de classe A peut exceptionnellement être assignée à un hôte particulier dans son réseau particulier lorsqu'il n'y a pas de risque d'ambiguïté Une telle adresse réseau ne peut apparaître que comme adresse source.

L'adresse IP 0.0.0.0 (tous les bits sont à zéro), désigne réflexivement l'hôte « qui parle ». Cette adresse ne doit apparaître que dans un contexte non ambigu et comme adresse source uniquement.

Ces deux formes d'adresse sont utilisées par des hôtes qui ne connaissent pas encore l'identité de leur réseau ni, *a fortiori*, leur propre adresse.

L'adresse IP avec tous les bits à 1 (255.255.255.255 en décimal) est appelée adresse de diffusion locale (*local broadcast*). Elle est envoyée à tous les hôtes d'un même support physique, et ne peut apparaître que comme destination.

L'adresse 127.0.0.0 de classe A sert au rebouclage (loopback). Quand une machine reçoit une adresse IP de ce type (par exemple, 127.0.0.1), son module Internet en déduit que cette adresse lui est destinée, et qu'il ne doit pas l'émettre vers l'extérieur par le pilote réseau. Il le renvoie donc au module TCP. Ces adresses sont souvent utilisées pour vérifier le bon fonctionnement en local du logiciel réseau.

Il existe deux autres adresses d'hôtes réservées. La première, avec tous les bits à 0, désigne le réseau lui-même (par exemple, en notation décimale pointée, 10.0.0.0 représente le réseau de classe A dans son intégralité, sans mention d'un hôte particulier). La deuxième, avec tous les bits à 1, désigne tous les hôtes d'un réseau donné. Une adresse IP *unicast* (de classe A, B ou C) où tous les bits de la partie hôte sont positionnés à 1, s'appelle adresse de diffusion dirigée (*directed broadcast*).

Les trois classes de réseau unicast A, B et C, étaient censées s'adapter à des réseaux physiques de différentes envergures, comprenant un nombre plus ou moins important d'hôtes. En tenant

© copyright Éditions Eyrolles

compte des bits affectés à l'identité réseau et des adresses réservées, la disponibilité par classe est la suivante :

- Pour la classe A on dispose de $2^7 - 2$, soit 126 réseaux, et chaque réseau peut contenir jusqu'à $2^{24} - 2$, soit 16 777 214 hôtes ; la classe A définit une fourchette de réseaux d'adresses IP allant de 1.0.0.0 à 126.0.0.0.
- Pour la classe B on dispose de $2^{14} - 1$, soit 16 383 réseaux, et chaque réseau peut contenir jusqu'à $2^{16} - 2$, soit 65 534 hôtes ; la classe B définit une fourchette de réseaux d'adresses IP allant de 128.0.0.0 à 191.255.0.0.
- Pour la classe C on dispose de $2^{21} - 1$, soit 2 097 151 réseaux, et chaque reseau peut contenir jusqu'à $2^8 - 2$, soit 254 hôtes ; la classe C définit une fourchette de réseaux d'adresse IP allant de 192.0.0.0 à 223.255.255.0.

Jusqu'à présent nous avons évoqué le tout premier schéma officiel de l'adressage IP. Depuis, il a subi bien des changements qui sont dans une certaine mesure incompatibles entre eux, mais qui concernent heureusement davantage les routeurs que les hôtes. Les anciens hôtes peuvent donc continuer à communiquer sans difficulté avec les hôtes plus récents, à condition que les routeurs intermédiaires implémentent toutes les versions de IP nécessaires. Les paragraphes suivants exposent succinctement les changements majeurs qui sont intervenus dans le schéma d'adressage IP et leur raison d'être.

L'invention de la pile de protocoles TCP/IP marqua le début de l'Internet, dont la popularité conduisit de plus en plus de sociétés à choisir cette norme de communication pour leurs besoins commerciaux. Très vite, le nombre de réseaux TCP/IP dépassa celui de tous les autres basés sur d'autres protocoles. Quand la plupart des sociétés manifestèrent leur volonté d'être connectées à l'Internet, celui-ci devint le réseau mondial le plus important. Bien que cette popularité ait prouvé la robustesse de conception et l'interopérabilité entre un nombre croissant de réseaux, de machines, d'équipements et de logiciels d'origine diverse, la flexibilité de l'adressage IP d'origine laissait à désirer.

Il apparut très tôt au cours de l'évolution de l'Internet que le schéma initial d'adressage IP ne conviendrait plus, du fait que peu de réseaux prenaient en charge un nombre d'hôtes dépassant le millier. En d'autres termes, l'espace d'adressage des classes A et B était en bonne partie gaspillé. Et la classe C, de son côté, ne disposait pas d'un nombre suffisant d'hôtes. Pour pallier ces contraintes, on choisit d'étendre les réseaux existants.

La conception initiale d'adressage IP fut rapidement élargie par l'ajout du découpage en sous-réseaux ou *subnetting*. Ceci aboutit à diviser les classes de réseaux de la version initiale d'adressage IP en sous-groupes. Ainsi, la partie hôte elle-même fut décomposée en une partie sous-réseau et une partie hôte. La relation d'origine entre le segment de réseau physique et l'adresse réseau fut également revue de façon à ce que ce soit désormais l'adresse de sous-réseau qui soit liée à l'appartenance à un réseau physique. L'adresse de réseau, quant à elle, identifia le niveau hiérarchique auquel plusieurs sous-réseaux dépendant d'une même juridiction pouvaient être rattachés.

Contrairement aux classes de réseaux, le découpage en sous-réseaux n'impose pas de frontière prédéfinie entre la partie sous-réseau et la partie hôte de l'adresse. En outre, le découpage en sous-réseaux comme tel permet d'attribuer des bits discontinus aux identités de sous-réseau et d'hôte. La démarcation se fait au moyen d'un composant additionnel appelé *masque de sous-réseau* (*subnetmask*). Celui-ci contient un arrangement de mots de 32 bits appliqué à une adresse IP pour différencier les bits de l'identité de sous-réseau de ceux de l'identité

© copyright Éditions Eyrolles

d'hôte. Suivant que le bit du masque est positionné à un ou à zéro, le bit correspondant de l'adresse IP fait partie de l'identité de sous-réseau ou de celle de l'hôte. Les masques de sous-réseaux sont souvent transcrits en notation décimale pointée (par exemple, 255.255.255.0).

Un masque de sous-réseau ne fait pas partie de l'adresse IP ; il s'agit d'une règle appliquée aux adresses IP de tous les hôtes d'un support physique pour déterminer l'identité de sous-réseau de celui-ci.

Le découpage en sous-réseaux introduisit deux adresses fonctionnelles supplémentaires qui sont décrites dans la RFC 1700. La première est l'adresse de diffusion (dirigée) de sous-réseau, ou *subnet directed broadcast* ; elle est composée des adresses de réseau et de sous-réseau, et d'une série de bits positionnés à un pour la partie hôte ; elle permet de joindre tous les hôtes du sous-réseau concerné. La deuxième adresse fonctionnelle comprend l'identité de réseau, avec tous les bits à un dans les parties sous-réseau et hôte ; cette adresse appelée diffusion dirigée ou *directed broadcast* permet de joindre tous les sous-réseaux du réseau concerné. Ces deux adresses ne peuvent apparaître que comme adresses de destination.

Bien que la RFC 1700, décrivant la norme d'affectation des adresses Internet, ne le dise pas expressément, une adresse composée des adresses réseau et de sous-réseau, et, pour la partie hôte, de bits positionnés à zéro, serait bien pratique, pour désigner de manière concise un sous-réseau particulier au sein d'un réseau divisé en sous-réseaux. Ce serait le cas, par exemple, pour un réseau 10.0.0.0 segmenté avec un masque de sous-réseau de 255.255.0.0. Au lieu de mentionner séparément le réseau et le sous-réseau sous la forme d'une identité réseau 10 et d'une identité sous-réseau 5, on pourrait combiner les deux en une seule notation sous la forme 10.5.0.0. Celle-ci, appelée *adresse de sous-réseau* sera adoptée tout au long de cet ouvrage.

Au début, le découpage en sous-réseaux ne touchait pas la partie réseau proprement dite, à savoir celle qui déterminait la classe du réseau. En d'autres termes, l'interprétation des bits du masque de sous-réseau correspondant à la partie réseau n'était pas définie. Ce type de découpage en sous-réseaux est qualifié de *classful*. Plus tard, la raréfaction des nouvelles adresses aidant, la décision fut prise de revoir le concept de classe. C'est ainsi qu'apparut le concept de réseau sans classe ou *classless*, qui mit fin aux classes de réseaux unicast. Toute implémentation de TCP/IP en mode hors classe doit, pour s'y conformer, utiliser un masque de sous-réseau pour délimiter les parties réseau des parties hôte. L'appellation de *super-réseau* est parfois utilisée quand un préfixe réseau regroupe plusieurs réseaux d'une classe, en souvenir de l'adressage par classe. Par exemple, l'adresse réseau 193.0.0.0 avec un masque de sous-réseau 255.0.0.0 est un super-réseau, et n'est valide que dans la version sans classe de l'adressage IP.

Comme indiqué plus haut, ces nouvelles spécifications concernaient peu les hôtes eux-mêmes. Ce sont les routeurs qui ont dû implémenter de nouveaux algorithmes de routage sans classe pour être conformes à ces spécifications. La différence entre le routage avec classes et sans classe est décrite au chapitre 3.

## *Le découpage en sous-réseaux ou* subnetting

L'un des concepts les plus importants de l'adressage IP fut le découpage en sous-réseaux, bien qu'il apparût au début plus comme un raccommodage pour pallier les défauts de conception initiale. En témoignent certains défauts, qui font penser à l'histoire du processeur Intel X86. Celui-ci, avec sa capacité en mémoire très restreinte, mena la vie dure aux ingénieurs de Intel

© copyright Éditions Eyrolles

qui cherchèrent à contourner ses limites. De même, les concepteurs de IP furent contraints de pallier les limites d'adressage en introduisant les masques de sous-réseaux. Quand les limites furent de nouveau presque atteintes, on dut trouver d'autres moyens d'optimiser ce découpage. Le masque de sous-réseau de longueur variable ou VLSM (*Variable Length Subnet Mask*), les protocoles de routage avec ou sans classes et les problèmes d'interopérabilité qui en découlent, font tous partie de cette évolution. Encore une fois, la plupart de ces innovations concernent les routeurs et non les hôtes.

Pour configurer un routeur de façon à ce qu'ils adoptent les différentes stratégies de découpage de sous-réseaux, nous devons aborder un certain nombre de concepts additionnels relevant de ce domaine.

Les masques de sous-réseaux peuvent être continus ou discontinus. Dans le premier cas, les masques commencent à 1 et le premier 0 ne peut être suivi que par d'autres 0. La figure 1.11 en donne un exemple. En utilisant la notation décimale pointée, l'adresse IP et le masque de sous-réseau sont transcrits sous la forme 67.240.1.3 et 255.255.240.0, respectivement.

Si un 0 est suivi d'un ou de plusieurs 1, le masque est discontinu. La figure 1.12 illustre un cas de ce type. Avec la notation décimale pointée, la transcription de l'adresse IP et celle du masque de sous-réseau sont de 67.240.1.3 et 255.252.31.0, respectivement.

**Figure 1.11**
*Masque de sous-réseau contigu* (*réseau de classe*)

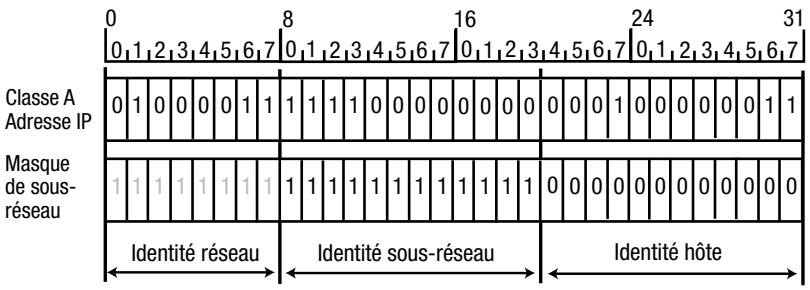
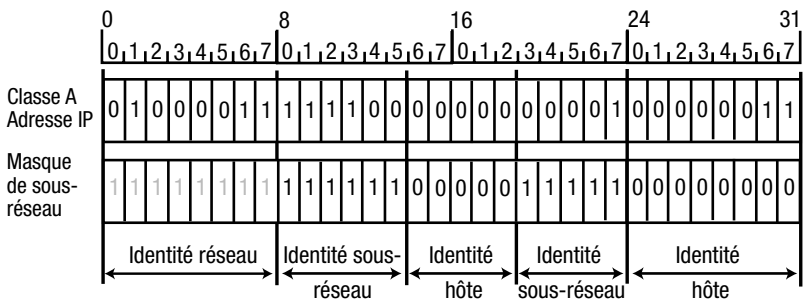



**Figure 1.12**
*Masque de sous-réseau discontinu* (*réseau de classe*)




Les masques discontinus sont très rares (en fait, on n'a pas connaissance d'un seul cas de déploiement de TCP/IP qui en fasse usage). Ils présentent beaucoup d'inconvénients parmi lesquels la difficulté à gérer l'espace d'adressage, le caractère imprévisible des algorithmes de routage du fait que ni l'identité de sous-réseau, ni l'identité d'hôte, ne peuvent être représentés par des champs de bits continus ; et pour les utilisateurs, la difficulté d'extraire les identités de sous-réseaux et d'hôtes. Par exemple, la plage d'adresses IP de la figure 1.11 allant de 67.240.0.1 à 67.240.15.255, facilite la tâche d'assignation des adresses aux hôtes situés sur ce réseau. En revanche, le masque discontinu de la figure 1.12, rend presque impossible la tâche

© copyright Éditions Eyrolles

de définir une plage d'adresses utilisables avec une notation aussi simple. La gestion administrative d'affectation des adresses IP au moyen de masques de sous-réseaux devient trop lourde.

Les déploiements de réseaux TCP/IP utilisent le plus souvent les masques continus. Une caractéristique particulièrement utile de ces derniers est de pouvoir être identifiés par une notation qui ne mentionne que le nombre de bits positionnés à un. C'est une autre notation des masques de sous-réseaux. Dans la figure 1.11, on pourrait par exemple écrire 20 au lieu de la notation classique en décimale pointée 255.255.240.0. Habituellement, on fait précéder ce nombre d'un caractère slache « / », comme dans /20. On peut ainsi rassembler les informations d'adresse et de masque de sous-réseau en une notation compacte, par exemple 67.240.1.3/20.

**REMARQUE** Tout au long de ce livre, nous utiliserons pour spécifier un masque de sous-réseau la nouvelle notation, plutôt que la notation décimale pointée – sauf indication contraire. Pour simplifier, toute référence à un masque de sous-réseau supposera de celui-ci qu'il est continu, car les masques discontinus ne sont d'aucune utilité pratique.

Du fait de l'utilisation exclusive des masques de sous-réseaux continus, les identifiants de sous-réseau contenus dans les adresses IP ont également été appelés *préfixes réseau*. De même, les masques de sous-réseaux ont été considérés comme les indicateurs de la *taille de préfixe réseau* (*network prefix lengths*). Ces nouvelles dénominations sont utilisées de plus en plus souvent pour tout ce qui concerne le routage IP.

**REMARQUE** Les termes de *préfixe réseau* et *longueur de préfixe réseau* seront employés tout au long de cet ouvrage quand nous évoquerons le routage IP. Dans les autres situations, on utilisera les termes *adresse de sous-réseau* et *masque de sous-réseau.*

Il existe une relation directe entre la longueur d'un masque de sous-réseau et le nombre d'hôtes qu'il peut mettre à disposition. Ainsi, le but principal du découpage en sous-réseaux est de diviser l'adresse réseau attribuée en sous-réseaux plus petits, suivant le nombre d'hôtes que l'on souhaite avoir dans chaque segment physique.

Le calcul du nombre d'hôtes possibles pour une longueur donnée de masque de sous-réseau, est assez simple. Le nombre de bits de la partie hôte (nombre de bits positionnés à zéro) s'obtient en soustrayant de 32 le nombre de bits du masque (positionnés à un). Le nombre d'hôtes est le total de toutes les combinaisons possibles des uns et des zéros parmi les bits qui appartiennent à l'identité d'hôte, soustraction faite de l'adresse de sous-réseau et de celle de diffusion. On a ainsi :

$$N = 2^{(32 - L)} - 2$$

N est le nombre d'hôtes ; L est la longueur du masque de sous-réseau.

Voici quelques exemples de calcul.

### Le masque de sous-réseau /30

En appliquant la formule pour le masque de sous-réseau /30, on obtient :

$$N = 2^{(32 - 30)} - 2 = 2^{2} - 2 = 2$$

© copyright Éditions Eyrolles

### Le masque de sous-réseau /24

En appliquant la formule pour le masque de sous-réseau /24, on obtient :

$$N = 2^{(32-24)} - 2 = 2^8 - 2 = 256 - 2 = 254$$

Ce masque de sous-réseau est celui par défaut utilisé pour les réseaux de classe C. Il n'est donc pas surprenant que le nombre d'hôtes qu'il génère soit égal à celui des hôtes défini pour cette classe.

### Le masque de sous-réseau /31

En appliquant la formule pour le masque de sous-réseau /31, on obtient :

$$N = 2^{(32-31)} - 2 = 2^1 - 2 = 2 - 2 = 0$$

On constate que l'utilisation de ce masque ne génère aucune adresse IP utile. Les seules adresses hôtes générées sont une adresse de sous-réseau (lorsque l'unique bit de la partie hôte est positionné à 0) et une adresse de diffusion (lorsque l'unique bit hôte est positionné à 1).

### Le masque de sous-réseau /32

En appliquant la formule pour le masque de sous-réseau /32, on obtient :

$$N = 2^{(32-32)} - 2 = 1 - 2 = -1$$

Apparemment, le masque de sous-réseau /32 est aussi interdit. Mais on l'utilise exceptionnellement pour désigner une adresse d'hôte qui n'appartient à aucun sous-réseau ; nous reviendrons sur ce cas dans les chapitres consacrés aux protocoles de routage.

En pratique, on a plus souvent besoin de déduire la longueur du masque de sous-réseau à partir du nombre d'hôtes que l'inverse ; ce qui rend la formule précédente peu utile. Or, la formule adéquate pour calculer la longueur du masque est assez difficile à manier :

$$L = 32 - \log_2(N + 2)$$

On préfère donc recourir à un tableau de valeurs pré-calculées pour connaître la longueur du masque de sous-réseau. Les réseaux physiques comprenant plus de mille hôtes étant rares, un tableau tel que le 1.3 est généralement suffisant.

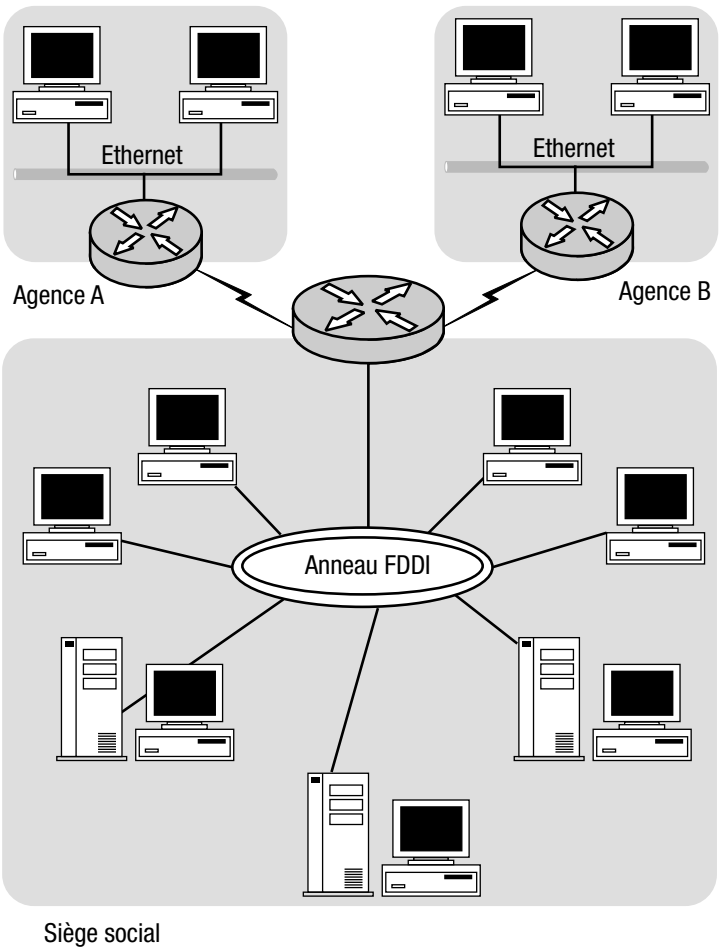
**Tableau 1.3. Relation entre longueur de masque de sous-réseau et nombre d'hôtes.**

| Nombre d'hôtes maximum | Longueur de masque de sous-réseau | Masque de sous-réseau |
|---|---|---|
| 2 | /30 | 255.255.255.252 |
| 6 | /29 | 255.255.255.248 |
| 14 | /28 | 255.255.255.240 |
| 30 | /27 | 255.255.255.224 |
| 62 | /26 | 255.255.255.192 |
| 126 | /25 | 255.255.255.128 |
| 254 | /24 | 255.255.255.0 |
| 510 | /23 | 255.255.254.0 |
| 1022 | /22 | 255.255.252.0 |
| 2046 | /21 | 255.255.248.0 |

© copyright Éditions Eyrolles

Calculons les masques de sous-réseau pour l'exemple illustré figure 1.13.

**Figure 1.13**
*Allocation d'adresse IP en utilisant les masques de sous- réseau*




Supposons que les agences A et B aient toutes deux un besoin maximum de 100 hôtes, en incluant les connexions des routeurs. Pour le réseau FDDI du siège social, on suppose que le besoin en nombre d'hôtes est inférieur à 1000. On a deux réseaux physiques reliant le routeur du siège à ceux des agences ; et pour chacun d'eux le nombre d'hôtes est de deux (une adresse d'hôte par routeur). En consultant le tableau 1.3, on détermine les longueurs de masque de sous-réseau suivantes :

- l'anneau FDDI devrait avoir un masque de longueur /22 ;
- les deux segments Ethernet des agences A et B devraient avoir un masque de longueur /25 ;
- les connexions entre les routeurs devraient utiliser un masque de longueur /30.

Dans la pratique courante, le calcul des masques de sous-réseaux ne suffit pas pour éviter les risques de chevauchement ou de gaspillage de l'espace d'adressage. Cet aspect sort du cadre de notre sujet actuel, mais nous l'aborderons au chapitre 4 quand il sera question de diviser l'espace d'adressage selon la méthode du masque de sous-réseau de taille variable ou VLSM (*Variable Length Subnet Mask*).

© copyright Éditions Eyrolles

## ICMP, protocole des messages de contrôle

Le protocole ICMP (*Internet Control Message Protocol*) fait partie intégrante de IP. Bien que résidant à la même couche que ce dernier, il n'en est pas moins un protocole à part qui utilise les services de IP, tout en ne fournissant pas de services aux protocoles de la couche transport.

La fonction principale de ICMP consiste à envoyer des messages d'erreurs qui surviennent lors de transmission de datagrammes. Les messages ICMP sont couramment générés par les routeurs tout au long du parcours vers la destination finale. Cependant, quelques messages ICMP peuvent être envoyés par les hôtes. Il existe aussi des messages ICMP de contrôle. Aussi, les paquets ICMP sont-ils souvent appelés messages d'erreur ou de contrôle.

### *Les messages de contrôle ICMP*

Les PDU de ICMP sont encapsulées dans des datagrammes IP comme celles de n'importe quel autre protocole qui utilise les services de IP. Comme ICMP réside à la couche Internet, les transmissions des messages de contrôle de ce dernier ne sont pas garanties. Ils peuvent se perdre en route ou subir des avaries.

Si ICMP est utilisé pour signaler un cas d'erreur, le message correspondant est envoyé à l'adresse source du datagramme qui a provoqué l'erreur. Le destinataire du datagramme, quant à lui, n'est pas notifié. Le module IP de la machine source doit, à la réception d'un message d'erreur ICMP, prendre les mesures nécessaires pour avertir les protocoles de la couche supérieure du cas d'erreur. Le nœud de réseau dont le module ICMP a généré le message d'erreur ne cherche pas à corriger le problème.

Chaque message ICMP possède son propre format, avec une partie commune qui comprend trois champs situés au début du message :

- un champ Type, d'un octet de longueur, qui identifie le message ;
- un champ Code, d'un octet, qui fournit d'autres précisions sur le message ;
- un champ de deux octets pour contenir une somme de contrôle de 16 bits, servant à vérifier l'intégrité du message ICMP.

En outre, les messages ICMP qui signalent des cas d'erreurs incluent toujours l'en-tête et les 64 premiers bits utiles du datagramme qui a provoqué l'erreur.

Le tableau 1.4 donne des indications sur les valeurs que peut prendre le champ type.

Nous n'examinerons que les types de message ICMP qui seront évoqués dans les chapitres suivants. Ceux qui voudraient connaître les détails de tous les messages ICMP, peuvent se reporter à la spécification de ICMP (cf. RFC 792) et à la dernière version du document « Internet Assigned Number » (cf. RFC 1700).

#### Les messages écho et réponse écho

Les messages écho et réponse écho sont des messages de contrôle. Ils ne sont utilisés que pour vérifier si un hôte est accessible *via* le réseau. Un hôte recevant un message écho doit répondre par un message réponse écho. Si le message ICMP écho contient des données optionnelles le message réponse écho doit le renvoyer sous la même forme. Le champ code contient toujours un zéro.

Une méthode très employée de diagnostic est le **ping** qui utilise les messages ICMP écho. Contrairement à d'autres applications réseau qui nécessitent que deux hôtes en communication

© copyright Éditions Eyrolles

**Tableau 1.4. Types de message ICMP.**

| Champ type | Description du champ type |
|---|---|
| 0 | Réponse écho |
| 1 | Non utilisé |
| 2 | Non utilisé |
| 3 | Destination inaccessible |
| 4 | Extinction de source |
| 5 | Redirection |
| 6 | Adresse hôte alternative |
| 7 | Non utilisé |
| 8 | Echo |
| 9 | Publication routeur |
| 10 | Selection routeur |
| 11 | Temps dépassé |
| 12 | Problème paramètre |
| 13 | Horodatage |
| 14 | Réponse horodatage |
| 15 | Demande d'information |
| 16 | Réponse d'information |
| 17 | Demande de masque d'adresse |
| 18 | Réponse de masque d'adresse |
| 19 | Réservé (sécurité) |
| 20-29 | Réservé (pour test de robustesse) |
| 30 | Traçage route |
| 31 | Erreur conversion datagramme |
| 32 | Redirection hôte mobile |
| 33 | IP v6 Où êtes-vous |
| 34 | IP v6 Je suis là |
| 35 | Demande inscription de mobile |
| 36 | Réponse inscription de mobile |
| 37-255 | Réservé |

possèdent des modules réseau responsables de la transmission de données, le **ping** s'appuie sur la capacité de réponse en interne de IP, et peut toujours vérifier l'accessibilité des hôtes IP, indépendamment de leur logiciel opératoire.

### Le message « destination inaccessible »

Le message « destination inaccessible » est envoyé en retour à un hôte source dont le datagramme n'a pu être livré au destinataire.

© copyright Éditions Eyrolles

Le champ code précise la raison pour laquelle le datagramme ne peut être livré. Le tableau 1.5 contient les différentes valeurs du champ code avec une description suffisamment explicite pour la plupart d'entre elles.

**Tableau 1.5. Valeur du champ code pour le message d'erreur ICMP destination inaccessible.**

| Code | Description |
|---|---|
| 0 | Réseau inaccessible |
| 1 | Hôte inaccessible |
| 2 | Protocole inaccessible |
| 3 | Port inaccessible |
| 4 | Fragmentation nécessaire et le bit DF (*Don't Fragment*) est à 1 |
| 5 | Echec route source |
| 6 | Réseau destinataire inconnu |
| 7 | Hôte destinataire inconnu |
| 8 | Hôte source isolé |
| 9 | Communication avec réseau destinataire adminisitrativement prohibée |
| 10 | Communication avec hôte destinataire administrativement prohibée |
| 11 | Réseau destinataire inaccessible pour type de service |
| 12 | Hôte destinataire inaccessible pour Type de service |

Selon la valeur du champ code, le message ICMP « destination inaccessible » peut aussi bien être envoyé par un routeur intermédiaire que par le destinataire final. Par exemple, le code 0 « réseau inaccessible », n'est envoyé que par un routeur qui ne trouve pas le chemin vers la destination. Cependant, le code 3, « port inaccessible », peut être envoyé par l'hôte destinataire lui-même, si le port du protocole de la couche supérieure auquel sont destinées les données n'est pas disponible.

### Le message extinction de source *(source quench)*

Le message « extinction de source » est utilisé quand IP doit pratiquer le contrôle de congestion. Un routeur intermédiaire ou l'hôte destinataire envoie habituellement ce message chaque fois qu'il doit mettre un datagramme au rebut. À la réception de ce message, le taux d'émission des datagrammes de la source est réduit et va en diminuant tant que celle-ci continue de recevoir ce type de message. Comme il n'existe aucun message ICMP pour indiquer que l'auteur des messages « extinction de source » s'est allégé, la source commence à augmenter le taux dès qu'elle ne reçoit plus de message « extinction », et ce jusqu'à ce que la source atteigne le taux maximum ou reçoive de nouveaux messages « extinction de source ».

### Le message « redirection » *(redirect)*

Quand il existe plusieurs routeurs disponibles, la source du datagramme reçoit le message de redirection ICMP du routeur qui connaît le meilleur chemin vers la destination. Le message contient l'adresse IP du meilleur routeur et les 64 premiers bits du datagramme. Tout en envoyant le message redirection, le routeur achemine quand même le datagramme.

© copyright Éditions Eyrolles

Bien que les messages de redirection soient une bonne idée, ils ont une grave lacune. Si un routeur intermédiaire reçoit un datagramme d'un autre routeur au lieu de la source, et s'il a connaissance d'un meilleur chemin *via* un autre routeur, il envoie le message ICMP redirection à la source plutôt qu'au routeur en « méprise ». La raison en est que ICMP fonctionne à la couche Internet, et qu'il ne peut pas utiliser l'adresse MAC du routeur dont le datagramme est issu, sachant que cette adresse a été dépouillée par le pilote réseau qui fonctionne à la couche d'accès réseau (le processus inverse de l'encapsulation). Par ailleurs, le datagramme ne contient que les adresses source et destination, et ne donne donc aucun moyen de savoir quel routeur intermédiaire l'a envoyé. Les messages ICMP redirection sont par conséquent de peu d'utilité, sauf s'il s'agit du premier routeur de la chaîne vers la destination, qui les envoie à la source même du datagramme.

### Le message « temps dépassé » *(time exceeded)*

Le message ICMP « temps dépassé » est envoyé par un routeur intermédiaire s'il constate que le champ durée de vie (TTL) du datagramme reçu a atteint zéro ; auquel cas il le met au rebut et avertit la source par un message temps dépassé.

## Technologies de la couche d'accès réseau et routage IP

Nous savons déjà que IP dépend de la couche d'accès réseau pour acheminer ses datagrammes à destination. Mais il nous reste à étudier comment IP communique ses informations, telle que l'adresse physique du destinataire ou celle du routeur suivant (*next hop*), à l'interface réseau et à son pilote.

Avant d'apprendre comment IP entre en interaction avec les technologies de la couche d'accès réseau, voyons d'abord leur typologie et leur nature.

Les technologies de la couche d'accès réseau sont mieux connues sous le nom de technologies de la couche liaison. L'adresse physique LAN souvent gravée en ROM dans l'interface réseau est appelée adresse MAC (*Media Access Control*) qui forme la sous-couche contrôle d'accès au support physique au sein de la couche liaison. Le modèle Internet n'ayant aucun terme spécifique pour les adresses physiques, nous les appellerons adresses MAC. De plus, les PDU de la couche d'accès réseau seront qualifiées de *trames*, puisqu'il s'agit d'une appellation de fait (par exemple, une trame Ethernet). Le réseau physique, quant à lui, sera dénommé *segment*, pour les mêmes raisons.

Les technologies de la couche d'accès réseau peuvent être classées en trois catégories :

- les réseaux à diffusion – permettent de s'adresser à tous les nœuds d'un segment, ce qui revient à envoyer une trame destinée à tous (par exemple, Ethernet, Token Ring, FDDI, etc.) ;
- les réseaux point à point – supposent qu'il n'existe que deux équipements interconnectés (par exemple, les réseaux de lignes louées, commutées, etc.) ;
- les réseaux d'accès multiples non diffusés ou NBMA (*Non-Broadcast Multiple Acces Networks*) – permettent d'interconnecter plusieurs équipements, mais sans possibilité de diffusion de paquets à tous (par exemple, X 25, Frame Relay).

Du point de vue de la couche d'accès réseau, les segments connectés par des routeurs sont, en réalité, déconnectés. C'est la fonction de la couche Internet qui permet d'envoyer des données d'un segment à un autre. Les diffusions qui se produisent à la couche d'accès réseau ne sont donc pas propagées au-delà des limites du segment.

© copyright Éditions Eyrolles

## *Adressage inter-couches et routage IP*

Comment IP procède-t-il quand il doit envoyer un datagramme au routeur de saut suivant en utilisant les technologies de la couche d'accès réseau ? Le module IP reçoit l'adresse IP du destinataire en provenance des couches supérieures, mais pas l'adresse MAC du routeur de saut suivant. Comment l'interface réseau et son pilote connaissent-ils l'adresse MAC du nœud destinataire, s'ils peuvent se passer des services du routeur de saut suivant ? S'ils doivent au contraire recourir à ce dernier, comment font-ils pour connaître l'adresse MAC de son interface ?

Comme nous l'avons appris dans les sections précédentes de ce chapitre, TCP et UDP utilisent l'adresse IP de l'hôte source pour identifier la connexion sans ambiguïté. Ni l'en-tête de TCP, ni celui de UDP ne contiennent la moindre information sur les adresses IP des hôtes source et destination. L'en-tête IP est retiré de la charge utile avant que TCP ou UDP ne le reçoive. Comment peuvent-ils donc connaître l'adresse IP de l'hôte source ?

Pour répondre à ces questions, commençons par la couche la plus haute, celle de l'application, et essayons de comprendre tout le déroulement du processus.

Comme nous le savons, un module de la couche application, comme `telnet.exe`, utilise une interface de programmation d'application pour établir une communication avec le module TCP de la machine locale. Ce module passe ensuite l'adresse IP de destination à la procédure concernée qui, elle, lance une connexion vers l'hôte destinataire. C'est ainsi que TCP ou UDP connaît l'adresse IP de destination. Il la passe ensuite à son tour au module IP comme paramètre en utilisant le point d'accès (SAP) du protocole Internet. Jusqu'à ce point, l'opération paraît élémentaire.

Et c'est au niveau de IP que ce processus devient complexe. D'abord, le module IP doit décider s'il doit envoyer le datagramme directement au destinataire ou s'il doit passer par un routeur. Ensuite, le module IP doit d'une certaine manière déterminer comment dire au pilote réseau l'adresse MAC à inclure dans sa trame pour atteindre soit l'hôte destinataire soit le routeur de saut suivant. Néanmoins, le pilote réseau situé à la couche d'accès réseau, ne sachant rien sur l'adressage IP, l'adresse IP ne lui est d'aucune utilité.

Ces problèmes sont résolus en utilisant deux concepts qui sont au cœur du fonctionnement de IP : la résolution d'adresse IP et le routage IP.

### Le routage IP

Un hôte IP utilise le routage IP pour procéder à l'envoi d'un datagramme vers sa destination. Le routage IP utilise son propre algorithme pour déterminer si la communication avec le destinataire peut se faire directement ou nécessite de passer par des routeurs intermédiaires. Si ces derniers sont nécessaires, l'algorithme de routage IP facilite le choix du meilleur chemin vers la destination. La simplicité du routage IP n'est qu'apparente. La complexité provient du parcours qui peut traverser de multiples réseaux interconnectés à de nombreux routeurs. Ceux-ci doivent connaître les réseaux auxquels ils ne sont pas directement connectés et doivent déterminer le meilleur chemin vers toutes les destinations possibles. Les réseaux concernés doivent aussi être capables de gérer les changements, les défaillances, les modifications, les extensions et ainsi de suite ; les routeurs doivent tenir compte de cette instabilité et s'y adapter pour pouvoir recalculer les chemins en conséquence. Le degré de complexité augmente fortement quand des segments individuels sont interconnectés.

© copyright Éditions Eyrolles

L'algorithme de routage IP est fondé sur un principe simple qui restera valable quelles que soient les futures améliorations de la fonction de routage. Ce principe est le suivant :

*Le routage des datagrammes est fondé sur la partie réseau des adresses de destination, et non pas sur les adresses individuelles des hôtes*.

On pourra trouver de plus amples détails sur le routage IP, au chapitre 3.

### La résolution d'adresse IP

La résolution d'adresse est une technique qui consiste à apparier les adresses IP à celles qui leur correspondent à la couche d'accès réseau, telles que les adresses MAC. Il faut noter cependant que l'adresse de la couche d'accès réseau a un sens large et nous en verrons quelques exemples typiques dans la section « Solutions de configuration » de ce chapitre. Les adresses IP ne doivent être appariées à celles leur correspondant à la couche d'accès réseau que dans le cas des hôtes qui sont situés sur des réseaux directement attachés, car s'ils ne l'étaient pas, la couche d'accès réseau les considère comme inaccessibles.

Les appariements entre adresses MAC et IP sont connus de IP soit par apprentissage soit par configuration manuelle.

La procédure normale pour que IP apprenne que telle adresse IP correspond à telle adresse MAC, dans un réseau de diffusion directement attaché (par exemple, un LAN), est l'écoute. Supposons qu'un hôte veuille envoyer des données à un autre situé sur le même réseau physique, dont l'adresse MAC est inconnue du premier. Celui-ci suppose que son destinataire est toujours en mesure de recevoir une trame envoyée à l'adresse de diffusion. Il envoie donc une trame en précisant qu'il a besoin de l'adresse MAC de l'hôte ayant une certaine adresse IP. Tous les hôtes sur ce réseau physique reçoivent la trame, mais seul l'hôte qui reconnaît son adresse IP répond.

L'implémentation de cette méthode se trouve concrétisée dans un protocole spécial appelé résolution d'adresse ou ARP (*Address Resolution Protocol*). Seul IP utilise ARP en tant que protocole auxiliaire qui, comme ICMP, réside à la couche Internet, mais sans fournir aucun service aux protocoles de la couche transport. En revanche, contrairement à ICMP, les PDU de ARP, au lieu d'être d'encapsulées en datagrammes IP, le sont directement en trames LAN.

Le fonctionnement de ARP est simple. Il tient à jour une table de correspondance entre adresses MAC et IP. Lors du démarrage d'un hôte, sa table ARP est vide. Quand il a besoin d'envoyer un datagramme à un autre sur le même réseau, le module IP de l'hôte fait appel au module ARP pour traduire l'adresse IP destinataire en adresse MAC. Comme la table est vide au démarrage, ARP doit passer par le réseau. Ne sachant rien du destinataire à part son adresse IP, il envoie une requête ARP dans une trame de diffusion contenant cette adresse. Comme il s'agit d'une adresse de diffusion, tous les hôtes de ce réseau reçoivent la trame, mais seul celui qui reconnaît l'adresse IP comme sienne envoie une réponse directement à l'adresse MAC du demandeur. Quand le module ARP de l'hôte reçoit la réponse, il l'inscrit dans sa table sous forme de paire d'adresses IP/MAC et notifie IP qu'une correspondance a été trouvée. Si une adresse MAC n'est pas utilisée pendant un certain temps, elle est effacée de la table pour éviter qu'elle ne devienne périmée (par exemple, l'adresse MAC d'un hôte peut changer si son interface réseau est remplacée, suite à une panne).

La configuration manuelle de ARP devient nécessaire quand la technologie du réseau ne permet pas d'envoyer une adresse de diffusion (par exemple, le RNIS).

© copyright Éditions Eyrolles

### *Le filtrage de paquets*

La technologie du filtrage de paquets permet de mettre au rebut des PDU qui ne répondent pas à certains critères. Dans le cas de IP, les datagrammes peuvent subir le même traitement.

L'implémentation du filtrage de paquets dans le système d'exploitation réseau de Cisco, IOS (*Internetwork Operating System*), se fait au moyen des listes d'accès. Elles constituent des expressions logiques qui contiennent des conditions appliquées aux datagrammes avant qu'une action soit prise à leur encontre.

Les listes d'accès sont décrites au chapitre 6.

## Outils pratiques

Nous utiliserons plusieurs outils très pratiques tout au long de cet ouvrage. Une explication succincte de chacun d'eux peut être utile.

La commande **debug** du système IOS de Cisco est l'outil par excellence pour savoir exactement ce qui se passe à l'intérieur du routeur. Mais dans un environnement opérationnel, particulièrement si le routeur est chargé, il n'est pas conseillé d'activer la commande **debug** pour éviter de détériorer sa performance. Il se peut même qu'il ne soit plus capable d'exécuter les lignes de commande dans de telles conditions, et que le redémarrage à froid soit nécessaire, ce qui est rédhibitoire.

L'analyseur LAN de Microsoft (Microsoft Network Monitor) est un outil élémentaire mais ses fonctionnalités suffisent dans la plupart des cas pour le dépannage et le test. Il est moins onéreux que d'autres analyseurs tel que le Network Associates Sniffer dont l'usage est préférable. Dans cet ouvrage nous ne ferons néanmoins qu'un usage limité de l'analyseur réseau.

D'autres outils sont disponibles sur le site web de Tsunami Computing : http:/www.hugewave.com/blackbook. Visitez-le à l'occasion. Son contenu va s'enrichir avec le temps.

# *Solutions de configuration*

Nous supposerons que la plupart des lecteurs ont la connaissance requise sur le système IOS de Cisco et ses lignes de commandes pour ne pas avoir à détailler chaque étape de la saisie dans les exemples.

Nous utiliserons aussi la commande complète au lieu de son abréviation, mais il est recommandé d'utiliser plutôt celle-ci lors des configurations réelles pour gagner du temps.

Les exemples de cette section n'ont pas pour vocation de vous enseigner la configuration des interfaces du routeur Cisco. Leur but est de faire la démonstration des caractéristiques majeures de IP vues dans la section précédente de ce chapitre, et aussi d'asseoir la base qui aidera à assimiler le contenu des chapitres ultérieurs de cet ouvrage.

Lors de la transcription de la syntaxe des différentes commandes, nous nous en tiendrons aux conventions les plus proches possibles de celles qui sont dans la documentation de Cisco. Elles sont les suivantes :

© copyright Éditions Eyrolles

- **le style gras** est employé pour transcrire les commandes et les mots clefs qui doivent être saisis comme indiqué ;
- *<les éléments en italique entre chevrons simples>* sont employés pour transcrire les paramètres des commandes ; ceux-ci peuvent être des chaînes de caractères ou des chiffres ;
- [les mots entre crochets] sont employés pour transcrire les éléments optionnels ;
- Les mots en gras en accolades séparés par des barres verticales, tels que **{choix1|choix2|choix3}**, sont employés pour indiquer les mots clefs alternatifs, trois choix possibles dans cet exemple.

Comme vous pouvez le constater, il y très peu de différence entre les conventions de Cisco et celles de cet ouvrage, sauf dans l'emploi des chevrons simples autour des mots en italique pour indiquer les paramètres des commandes. Nous croyons ainsi mettre en valeur le rôle fonctionnel de ces éléments.

## Configuration de IP sur LAN avec ARP et Proxy ARP

La configuration d'une interface LAN sur un routeur est probablement une des tâches les plus faciles. Examinons la mise en œuvre des interfaces Ethernet et Token Ring pour le routage IP.

Il faut d'abord assigner une adresse IP et un masque de sous-réseau à l'interface, puis il reste à activer celle-ci.

Supposons que la paire adresse/masque de sous-réseau qu'on veuille assigner soit 10.1.0.1/24. Le listing 1.1 montre les étapes à suivre pour assigner une adresse IP à une interface Ethernet.

**Listing 1.1. Assignation d'une adresse IP à l'interface Ethernet 0 et son activation.**

```
Router (config)#interface ethernet 0
Router (config-if)#ip address 10.1.0.1 255.255.255.0
Router (config-if)#no shutdown
```

L'interface Token Ring nécessite une étape supplémentaire : l'assignation du débit de l'anneau qui apparaît sur le listing 1.2.

**Listing 1.2. Assignation d'une adresse IP à l'interface Token Ring 0 et son activation.**

```
Router (config)#interface tokenring 0
Router (config-if)#ip address 10.1.0.1 255.255.255.0
Router (config-if)#no shutdown
```

S'il n'y a aucun problème matériel, l'interface devrait être prête à l'emploi. Pour utiliser le routeur, les hôtes du segment Ethernet ou de l'anneau Token Ring doivent avoir un pointeur vers l'adresse IP qu'on vient d'assigner.

La route la plus couramment implantée dans les hôtes est celle de la passerelle par défaut. Celle-ci est utilisée uniquement quand l'hôte ou le routeur ne peut trouver une entrée correspondant à l'adresse réseau dans sa table de routage. Il est à noter que la route pointant sur la passerelle par défaut (*default gateway*) est celle qui correspond à l'adresse réseau 0.0.0.0, qui est aussi l'adresse de *ce réseau*, comme nous l'avons vu dans la section précédente de ce chapitre. On peut cependant trouver une explication logique à l'utilisation d'une telle adresse.

© copyright Éditions Eyrolles

D'abord, cette adresse ne doit pas apparaître comme destination, ce qui implique qu'aucun hôte n'essayera de lui envoyer un datagramme. Ensuite, c'est une adresse unicast admissible. On peut donc l'utiliser dans un sens spécifique, en la faisant pointer sur la passerelle par défaut, chaque fois qu'une recherche d'adresse réseau dans la table de routage s'avère infructueuse.

Bien entendu, une adresse de sous-réseau ne peut exister sans le masque de sous-réseau. Par conséquent, si le découpage en sous-réseaux est utilisé, la table de routage doit contenir les masques de sous-réseaux. Par définition, le masque de sous-réseau utilisé pour la route de la passerelle par défaut, a tous ses bits positionnés à 0, ce qui donne 0.0.0.0, réduisant ainsi à néant l'adresse de réseau réelle. En principe, n'importe quelle adresse peut être utilisée comme route vers la passerelle par défaut, tant que tous les bits du masque de sous-réseau associé sont positionnés à 0.

Considérons maintenant le cas illustré sur la figure 1.14.

Les interfaces Ethernet sur le routeur R sont configurées comme le montre le listing 1.3.

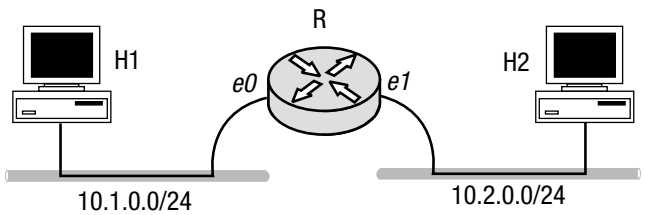
**Listing 1.3. Configuration du routeur R.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Ethernet1
 ip address 10.1.0.1 255.255.255.0
```

Les deux stations de travail utilisent un mauvais masque de sous-réseau /8, celui utilisé par défaut dans un réseau de classe A. En supposant que l'hôte H1 soit une station Windows NT, la configuration de TCP/IP dans la boîte de dialogue des propriétés réseau (*Network Properties*) doit ressembler à celle de la figure 1.15.

**Figure 1.14**

*Stations de travail configurées avec un masque « faux » de sous-réseau : /8.*




La question se pose de savoir si les deux hôtes H1 et H2 seront capables de communiquer *via* IP. En principe, la réponse est non. L'hôte H1 voit l'hôte H2 comme appartenant au même réseau que lui, ce qui va l'amener à lancer une requête ARP pour résoudre l'adresse IP de l'hôte H2 en adresse MAC correspondante. Mais nous voyons que l'hôte H2 est situé sur un autre segment, ce qui va faire échouer la requête ARP lancée par l'hôte H1, par débordement de temporisation. Le module IP de ce dernier va donc retourner une condition d'erreur indiquant que la communication est impossible.

Dans la pratique, on constate avec surprise que cette configuration fonctionne parfaitement. Par exemple, si l'on essaie d'envoyer un ping de l'hôte H1 vers H2, on obtient la sortie du listing 1.4.

© copyright Éditions Eyrolles

**Figure 1.15**

*Configuration de Proxy ARP implicite dans Windows NT 4.0.*

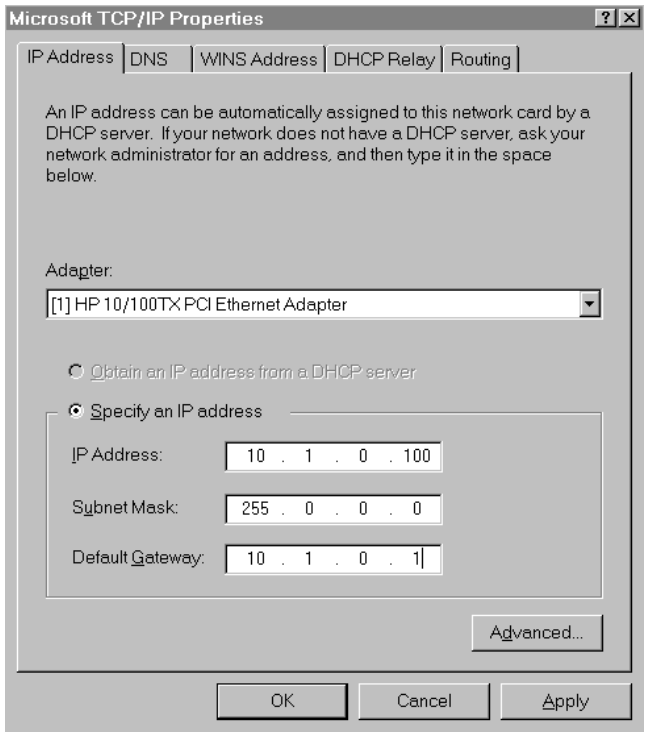
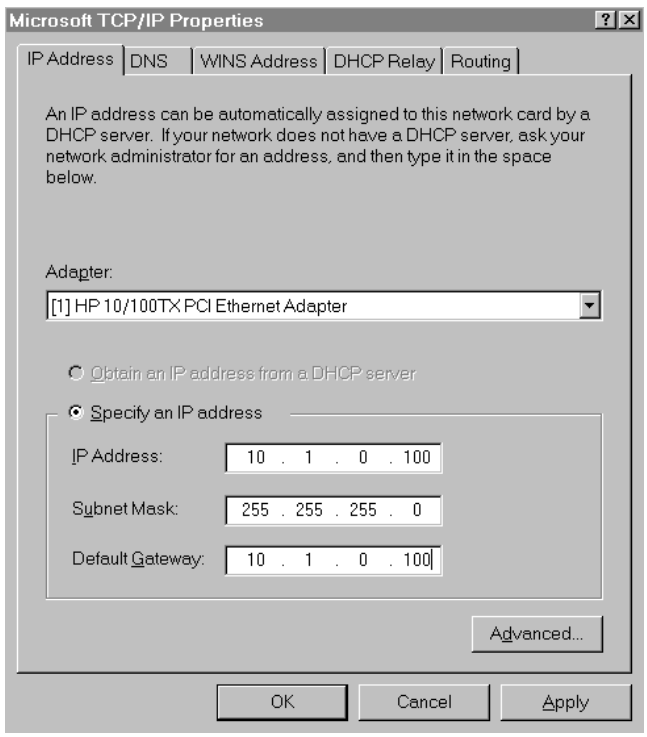



**Figure 1.16**

*Configuration de Proxy ARP explicite dans Windows NT 4.0.*




© copyright Éditions Eyrolles

**Listing 1.4. Résultat du ping qui indique l'accessibilité de l'hôte H2 par l'hôte H1.**

```
C:\>ping 10.2.0.100

Pinging 10.2.0.100 with 32 bytes of data:

Reply from 10.2.0.100: bytes=32 time=10ms TTL=253
Reply from 10.2.0.100: bytes=32 time=10ms TTL=253
Reply from 10.2.0.100: bytes=32 time=10ms TTL=253
Reply from 10.2.0.100: bytes=32 time=10ms TTL=253
```

Le fait que l'hôte H1 puisse recevoir la réponse au ping qu'il a lancé envers l'hôte H2, prouve que ce dernier a bien reçu la requête ARP venant de H1 pour son adresse IP et lui a transmis son adresse MAC en retour. Si nous demandons une sortie de la table ARP de l'hôte H1, nous y verrons le contenu du listing 1.5.

**Listing 1.5. Table ARP de l'hôte H1.**

```
C:\>arp -a

Interface: 10.1.0.100 on Interface 2
  Internet Address      Physical Address      Type
  10.1.0.1              00-e0-b0-64-50-63     dynamic
  10.2.0.100            00-e0-b0-64-50-63     dynamic
```

Cette table contient une entrée pour l'adresse IP de l'hôte H2, ce qui *a priori* n'a pas de sens. L'hôte H2 étant situé sur un réseau physique différent de celui de H1, ne pouvait pas répondre à la requête ARP de ce dernier. Si nous analysons de plus près la table ARP, nous pouvons y voir une autre entrée qui est celle du routeur R. Les adresses MAC pour ces deux entrées sont les mêmes. Nous pouvons donc en conclure que le routeur R a répondu à la requête ARP, à la place de l'hôte H2. Ce trait curieux de IP est appelé Proxy ARP. Il en est fait mention dans la documentation de Cisco, bien qu'il ait été quelque peu omis dans la plupart des documentations relatives à l'implémentation de IP sur les hôtes.

L'idée sous-jacente au Proxy ARP est simple. Si un routeur reçoit une requête ARP pour une adresse IP, différente de celle du réseau de provenance, il peut choisir de répondre à la requête avec sa propre adresse MAC, s'il connaît le chemin vers cette destination. Cependant, sans répondre systématiquement à toutes les requêtes, le routeur ne les prendra en compte que si elles proviennent du réseau dont l'adresse IP est la même que celle de son interface de réception. Par exemple, si l'adresse IP sur la figure 1.15 était 10.1.1.100 au lieu de 10.1.0.100, le routeur aurait ignoré la requête ARP, et l'hôte H1 n'aurait pas pu communiquer avec l'hôte H2.

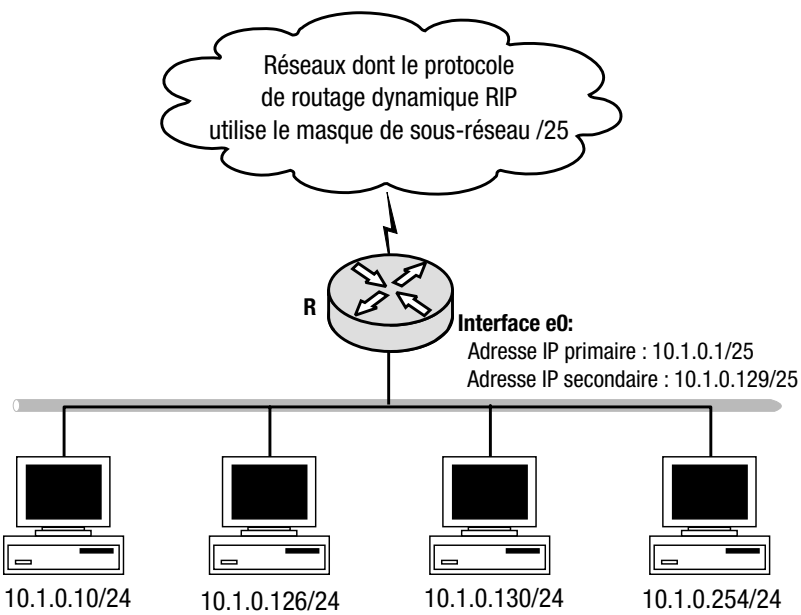
La seule différence entre une configuration qui utilise le Proxy ARP et celle qui ne l'utilise pas, réside dans le masque de sous-réseau configuré dans les hôtes. Celui-ci doit être plus court (le masque étant contigu, on peut le qualifier de « plus court ») que le masque « réel » avec lequel le routeur est configuré.

Appelons ce cas, configuration *implicite* de Proxy ARP. Il est aussi possible de le mettre en œuvre explicitement, comme on le voit sur la figure 1.16. Il s'agit de la même configuration de TCP/IP dans la boîte de dialogue *Network Properties* que celle qu'on a vue à la figure 1.15, à ceci près que l'adresse IP du routeur par défaut est, dans ce cas, identique à celle de l'interface réseau. En effectuant ce changement, on arrive à faire communiquer un hôte avec un autre, indépendamment du réseau sur lequel réside l'adresse IP de ce dernier, à condition que le routeur intermédiaire possède la fonction Proxy ARP.

© copyright Éditions Eyrolles

Le Proxy ARP permet de changer l'adresse IP du routeur utilisé comme passerelle par défaut sans reconfigurer les stations de travail. En outre, s'il y a plus d'un routeur sur un segment, tous ces routeurs peuvent être utilisés en tant que passerelle par défaut par les hôtes configurés avec le Proxy ARP. En principe, le routeur le moins chargé répond à la requête ARP en premier, facilitant ainsi un partage de charge de trafic entre les routeurs. Mais en pratique, ce scénario n'est pas toujours possible, car si les réseaux comportent plusieurs segments connectés par des ponts, le routeur le moins chargé peut se trouver derrière un pont et être retardé par ce dernier dans sa réponse à un point tel que les routeurs les plus chargés réussissent à répondre plus vite. Les réseaux mettant en œuvre le Proxy ARP peuvent fonctionner de concert avec les protocoles de routage dynamique de réseaux à classe (*classful*), tels que RIP ou IGRP. Ceux-ci, ne tolèrent pas plus d'un masque de sous-réseau par adresse réseau à classe. Cependant, si on doit utiliser un masque de sous-réseau plus court dans un segment, que celui utilisé avec le protocole de routage de réseau à classes, le Proxy ARP nous en donne les moyens. Nous étudierons ces protocoles plus en détail dans le chapitre 4. L'utilisation du Proxy ARP peut parfois sembler justifiée, bien que non souhaitable. Prenons le cas illustré sur la figure 1.17. Le masque de sous-réseau utilisé par RIP, un protocole de routage dynamique de réseau de classe, est /25. Pour un fonctionnement correct de RIP, tous les routeurs doivent avoir leur interfaces où ce protocole est actif, configurées avec le masque /25. Prenons un site n'ayant qu'un seul segment physique, avec 200 hôtes y résidant. Le masque /25 ne permet pas d'en loger autant. Supposons que vous décidiez alors d'utiliser un masque /24 dans tous les hôtes, tout en gardant celui de /25 pour l'interface Ethernet 0 du routeur, et de mettre en œuvre le Proxy ARP pour donner aux hôtes l'accès aux autres réseaux. Supposons maintenant que l'interface Ethernet, étiquetée e0 utilise l'adresse IP « primaire ». Du point de vue du routeur, les hôtes dont la plage d'adresses IP va de 10.1.0.130 à 10.1.0.254 sont illégalement connectés au segment. Toute requête ARP en provenance de ces hôtes sera de ce fait ignorée par le routeur. Pour y remédier, la solution consiste à assigner une deuxième adresse appelée « secondaire » (*secondary*) à l'interface e0.

**Figure 1.17**
*Utilisation de Proxy ARP de concert avec les protocoles de routage de réseau à classe et les adresses IP secondaires.*




© copyright Éditions Eyrolles

La configuration de l'interface Ethernet 0 est indiquée sur la figure 1.6.

**Listing 1.6. Configuration de l'interface Ethernet 0 du routeur R.**

```
interface Ethernet 0
 ip address 10.1.0.129 255.255.255.128 secondary
 ip address 10.1.0.1 255.255.255.128
```

La solution proposée est tout à fait incohérente avec le schéma d'adressage IP. Les hôtes « croient » qu'ils sont situés sur le réseau 10.1.0.0/24, tandis que le routeur « pense » avoir deux réseaux logiques attachés à sa seule interface. Les autres routeurs qui participent au protocole RIP perçoivent le réseau du site comme divisé en deux.

La mise en œuvre du Proxy ARP implique aussi un trafic excessif de messages. Dans un environnement routé, les hôtes ne lancent qu'une requête ARP pour avoir l'adresse MAC de la passerelle par défaut et l'inscrire en une seule entrée dans leur table. Celle-ci ne sera pas périmée, tant que les hôtes l'utiliseront pour l'accès vers l'extérieur. Par contre, si le Proxy ARP est mis en œuvre, les hôtes auront à lancer des requêtes ARP même pour les adresses IP auxquelles ils auraient pu accéder *via* le routeur. Ceci entraîne de multiples entrées dans la table ARP où toutes les adresses IP sont appariées à la même adresse MAC du routeur. Chacune de ces entrées a été créée par une requête ARP superflue.

Étant donné ces inconvénients, comment désactiver le Proxy ARP dans un routeur ? En mode configuration d'interface LAN, la commande **ip Proxy-arp** active le Proxy ARP. Celle-ci est active par défaut mais n'est pas affichée à la configuration du routeur (un changement interviendra peut-être sur ce point dans les versions futures de l'IOS de Cisco). Pour désactiver cette commande, il suffit d'entrer **no ip Proxy-arp**. Le listing 1.7 montre comment la configuration de l'interface a été modifiée.

**Listing 1.7. Nouvelle configuration de l'interface Ethernet 0 du routeur R avec le Proxy ARP désactivée.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0
 no ip proxy-arp
```

Si l'interface Ethernet 0 de la figure 1.14 était configurée de cette manière, l'hôte H1 ne pourrait pas communiquer avec l'hôte H2.

En général, il n'est pas nécessaire de désactiver le Proxy ARP sur les routeurs Cisco, si les hôtes n'en dépendent pas. Il serait même utile en cas de dépannage, dans la situation où certains hôtes sont mal configurés avec un masque de sous-réseau par défaut alors qu'ils devraient avoir leur masque spécifique.

## Configuration d'une interface série

L'interface série dans les routeurs Cisco est utilisée dans diverses configurations. La façon dont le routeur interprète les données transmises ou reçues *via* cette interface dépend du type d'encapsulation défini dans celle-ci. Si vous entrez la commande **encapsulation ?**, vous obtenez la sortie du listing 1.8.

© copyright Éditions Eyrolles

**Listing 1.8 Types d'encapsulation disponibles sur une interface série.**

```
R(config-if)#encapsulation ?
  atm-dxi         ATM-DXI encapsulation
  bstun           Block Serial tunneling (BSTUN)
  frame-relay     Frame Relay networks
  hdlc            Serial HDLC synchronous
  lapb            LAPB (X.25 Level 2)
  ppp             Point-to-Point protocol
  sdlc            SDLC
  sdlc-primary    SDLC (primary)
  sdlc-secondary  SDLC (secondary)
  smds            Switched Megabit Data Service (SMDS)
  stun            Serial tunneling (STUN)
  x25             X.25
```

Prenons les deux cas d'encapsulation les plus répandus que sont HDLC et Frame Relay. PPP est aussi très courant, mais comme il est plus souvent utilisé dans les lignes commutées (*dial-up*), comme le RNIS, nous en parlerons dans la section « Configuration de IP sur RNIS » plus loin dans ce chapitre.

### L'encapsulation HDLC

Le HDLC (*High Level Data Link Control*) est un protocole d'usage général, développé par l'ISO, qui se situe à la couche liaison du modèle OSI ou à la couche d'accès réseau du modèle Internet. Étant un protocole générique, le HDLC ne fournit aucune spécification détaillée, prête à être implémentée, mais sert plutôt de fondation à d'autres protocoles tels que le LLC (*Logical Link Control*), le LAPB (*Link Access Procedure Control Balanced*) et le LAPD (*Link Access Procedure Control for D Channel*) de RNIS. Pour les connexions des interfaces série, Cisco Systems a complété les spécifications de HDLC par une version propriétaire et l'a référencée dans sa documentation par le même nom.

La configuration d'une interface série avec l'encapsulation HDLC est vraiment simple. Une interface série est configurée par défaut avec HDLC, ce qui ne nécessite aucune action particulière. Il s'agit d'une connexion en point à point, qui ne possède aucune procédure automatique d'appariement entre les adresses IP et celles de la couche d'accès réseau. L'implémentation de HDLC dans la version de Cisco fournit heureusement toute la fonctionnalité requise pour faire cet appariement. Il est automatique et activé par défaut nous épargnant ainsi toute action manuelle.

Configurer une interface série avec l'encapsulation HDLC revient simplement à assigner une adresse IP et démarrer l'interface comme indiqué sur le listing 1.9.

**Listing 1.9. Configuration d'une interface série avec l'encapsulation HDLC et assignation d'une adresse IP.**

```
R(config)#interface serial 1
R(config-if)#ip address 10.1.0.1 255.255.255.0
R(config-if)#no shutdown
```

© copyright Éditions Eyrolles

## Configuration de IP sur Frame Relay en mapping statique et ARP inverse

Les interfaces série des routeurs Cisco sont plus couramment utilisées de nos jours pour se connecter aux réseaux Frame Relay.

Frame Relay est une technologie de commutation de paquets qui fonctionne à la couche d'accès réseau. Il définit deux types de nœuds : l'Équipement Terminal de Traitement de données ETTD ou DTE (*Data Terminal Equipment*) et l'Équipement de Terminaison de Circuit ETCD ou DCE (*Data communication Equipment*). Les ETTD sont des équipements d'utilisateurs tels que les routeurs, les ponts et les hôtes ; les ETCD, quant à eux, constituent les appareils de réseaux, tels que les commutateurs Frame Relay.

La vitesse de transmission est optimisée dans Frame Relay. Les commutateurs, dans ce but, ne détectent que les erreurs sans tenter de retransmettre les trames qui en contiennent. Comme dans IP, cette fonction est laissée aux protocoles des couches supérieures.

Frame Relay permet de multiplexer plusieurs connexions logiques appelées circuits virtuels sur la même liaison physique. À présent, Frame Relay définit les circuits virtuels permanents ou CVP (*PVC, Permanent Virtual Circuits*), qui sont configurés de manière statique dans le réseau. Il est aussi question d'adjoindre aux spécifications de Frame Relay, les circuits virtuels commutés ou SVC (*Switched Virtual Circuits*). L'identificateur de circuit de données ou DLCI (*Data Link Circuit Identifier*) sert à identifier les CVP. Du point de vue de IP, il est à noter que le DLCI est simplement une adresse de la couche d'accès réseau.

Frame Relay définit également l'interface de gestion locale ou LMI (*Local Management Interface*), qui permet l'échange d'informations de gestion entre l'ETTD et l'ETCD comme, par exemple, l'état de certains CVP. La LMI utilise un CVP prédéfini de façon à ce que l'échange entre l'ETTD et l'ETCD puisse se dérouler même dans le cas où aucun PVC n'a été configuré. Il y a plusieurs versions de LMI. Le système IOS de Cisco en supporte trois qui sont : Cisco, ANSI annexe D et ITU-T Q.933 annexe A. La LMI par défaut, Cisco, utilise le CVP 1023.

Comment l'appariement (*mapping*) des adresses IP se fait-il aux DLCI correspondants ? Frame Relay est un réseau de type NBMA (*Non-Broadcast Multiple Access*), réseau de non-diffusion, et le ARP normal ne pourra donc pas fonctionner. Il en existe une version adaptée qu'est le ARP inverse ou InARP, disponible dans le système IOS de Cisco. Sa fonction est très similaire à celle du ARP normal, mais au lieu d'apparier comme celui-ci des adresses IP aux adresses MAC, le InARP associe les premières aux DLCI. Cette opération peut aussi se faire en configuration manuelle.

Il y deux façons de configurer une interface série en Frame Relay. La première, c'est de faire toute la configuration sous l'interface elle-même. L'autre consiste à définir uniquement le type d'encapsulation et la LMI sous l'interface et d'utiliser des sous-interfaces (*sub-interfaces*). Celles-ci sont des interfaces logiques qui héritent de la plupart des paramètres tel que le type d'encapsulation de l'interface parentale. Du point de vue de IP, il n'y a pas de différence entre les deux.

Les sous-interfaces peuvent être point à point ou multipoint. Dans le premier cas elles sont configurées avec un seul CVP et dans le second avec plusieurs. Nous aurons une idée plus précise de ce que sont les sous-interfaces dans les chapitres suivants, quand nous évoquerons les protocoles de routage dynamique. Pour l'instant, contentons-nous de savoir comment les configurer.

© copyright Éditions Eyrolles

La configuration d'une interface série en Frame Relay sans sous-interfaces s'effectue selon les étapes suivantes :

1. Configurer l'encapsulation du Frame Relay par la commande **encapsulation Relay**.
2. Choisir le cas échéant l'option **type** de LMI par la commande **Relay lmi-type** <*type de LMI* >.
3. Assigner l'adresse IP avec la commande **ip address** <*adresse IP*> <*masque de sous-réseau*>.
4. Activer l'interface avec la commande **no shutdown**.

Examinons le cas illustré à la figure 1.18.

Il faut noter que les types de LMI et les DLCI sont différents de chaque côté de la connexion au réseau Frame Relay, parce que leur signification est locale. Celle-ci concerne l'ETTD (le routeur, dans notre cas) et l'ETCD (les commutateurs Frame Relay, non représentés). En suivant les instructions de configuration des routeurs qu'on vient de décrire, on obtient les listings 1.10 et 1.11.

**Listing 1.10. Configuration du routeur R1.**

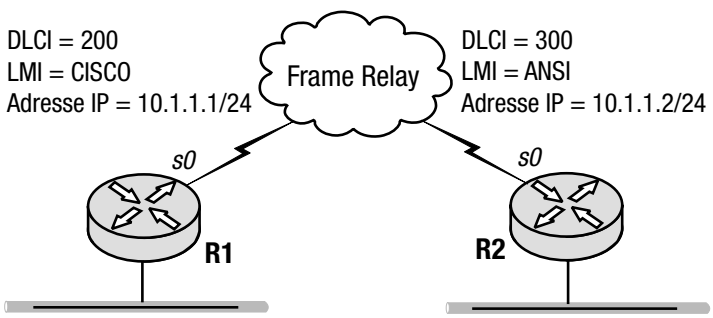
```
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 encapsulation frame-relay
```

**Listing 1.11. Configuration du routeur R2.**

```
interface Serial0
 ip address 10.1.1.2 255.255.255.0
 encapsulation frame-relay
 frame-relay lmi-type ansi
```

**Figure 1.18**

*Deux routeurs connectés sur un réseau Frame Relay.*




La configuration du routeur R2 diffère de celle du routeur R1 par la commande **Relay lmi-type ansi**. Comme dit précédemment, la LMI par défaut est Cisco.

La configuration d'une interface série en Frame Relay avec des sous-interfaces s'effectue selon les étapes suivantes :

1. Configurer l'encapsulation de Frame Relay sur l'interface série adéquate par la commande **encapsulation Relay**.
2. Choisir le cas échéant l'option type de LMI pour toute l'interface, par la commande **Relay lmi-type** <*type de LMI*>.

© copyright Éditions Eyrolles

3. Définir une ou plusieurs sous-interfaces point à point et/ou multipoint, par la commande **interface serial** <*numéro d'interface.numéro de sous-interface*> [**multipoint**|**point-to-point**] (le numéro d'interface correspond à l'interface physique séparé par un point du numéro de sous-interface logique qui a été choisi).
4. Assigner une adresse IP à chaque sous-interface par la commande **ip address** <*adresse IP*><*masque de sous-réseau*>.
5. Pour les sous-interfaces point à point, affecter un DLCI par la commande **Relay interface-dlci** <*DLCI*> *;* pour les sous-interfaces multipoint, si c'est le InARP qui est choisi, utiliser la même commande que dans le cas précédent ; sinon, pour le *mapping* statique, la commande se fait par **Frame-relay map ip** <*adresse IP distante*> <*DLCI*>, aussi bien en multipoint qu'en point à point.
6. L'option [**broadcast]** autorise la diffusion de mises à jour d'informations vers l'adresse indiquée, notamment dans le cas du protocole de routage dynamique OSPF.

La figure 1.19 montre un exemple de réseau Frame Relay qui utilise des sous-interfaces. Les listings 1.12 à 1.14 contiennent les configurations des trois routeurs.

**Listing 1.12. Configuration du routeur R1.**

```
interface Serial0
 no ip address
 encapsulation frame-relay

interface Serial0.1 multipoint
 ip address 10.1.0.1 255.255.255.0
 frame-relay map ip 10.1.0.2 200 broadcast

interface Serial0.2 multipoint
 ip address 10.2.0.1 255.255.255.0
 frame-relay interface-dlci 100
```

**Listing 1.13. Configuration du routeur R2.**

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.1.0.2 255.255.255.0
 frame-relay interface-dlci 300
```

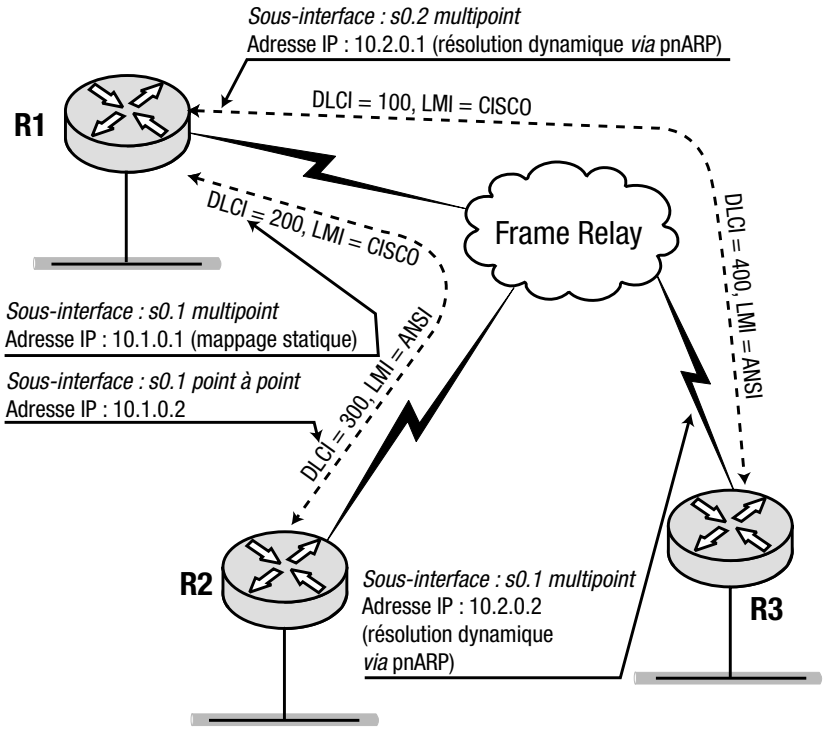
**Listing 1.14. Configuration du routeur R3.**

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 multipoint
 ip address 10.2.0.2 255.255.255.0
 frame-relay interface-dlci 400
```

© copyright Éditions Eyrolles

**Figure 1.19**

*Routeurs connectés via Frame Relay configurés avec des sous-interfaces.*




## Configuration de IP sur RNIS

Le RNIS (Réseau numérique à intégration de services) ou ISDN (*Integrated Service Digital Network*) est un service de téléphonie numérique (encore un exemple de la diversité des technologies sous-jacentes aux réseaux physiques). Sans étudier le fonctionnement de RNIS en profondeur, ce qui n'est pas du ressort de cet ouvrage, nous nous limiterons à ce qui est nécessaire à notre propos, les principes de base régissant cette technologie.

La connexion élémentaire en téléphonie numérique s'appelle un canal dont le débit est de 64 Kbit/s, exactement celui requis pour passer une communication vocale en numérique. Ceci consiste à échantillonner la voix humaine dont la fréquence utile se situe entre 30 et 3100 Hz. En arrondissant la limite haute au millier supérieur on obtient 4 KHz. Pour une qualité d'audition suffisante, l'échantillonnage doit s'effectuer au double de cette fréquence haute, ce qui donne 8000 échantillons par seconde, chacun d'entre eux étant codé sur 8 bits, le débit nécessaire sera de 8 X 8000 = 64000 bits par seconde (64 Kbit/s), débit d'un canal de communication vocale. La technique de numérisation mise en œuvre s'appelle modulation par impulsion codée (MIC) ou PCM (*Pulse Code Modulation*).

Le RNIS possède deux types de service qui sont : l'accès de base ou BRI (*Basic Rate Interface*) et l'accès primaire ou PRI (*Primary Rate Interface*). BRI dispose de deux canaux de 64 Kbit/s, appelés canaux B qui peuvent être utilisés en voix ou données, et d'un canal de 16 Kbit/s, appelé canal D qui sert uniquement à la signalisation.

Il existe deux versions de PRI, la version d'Amérique du nord et du Japon qui fournit 23 canaux B et un canal D à 64 Kbit/s et la version du reste du monde, qui comporte 30 canaux B et un canal D à 64 Kbit/s.

© copyright Éditions Eyrolles

Dans cet ouvrage, il ne sera question que de BRI, bien que les principes de fonctionnement puissent être étendus au PRI.

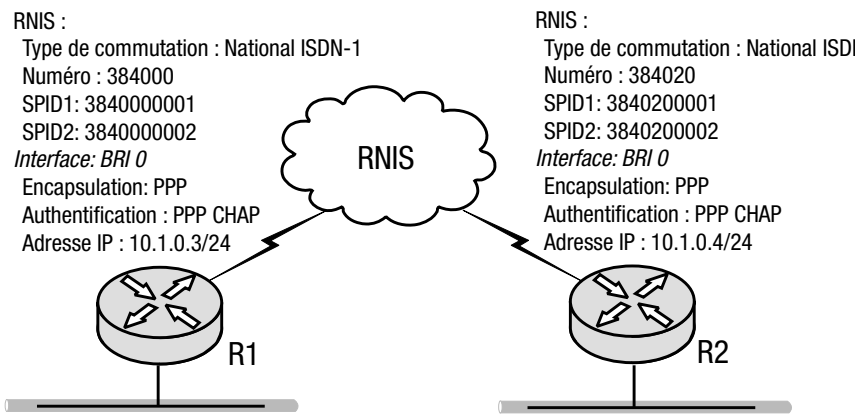
Dans le RNIS, l'adresse de la couche d'accès réseau est le numéro de téléphone utilisé pour l'appel. Le couplage entre l'adresse IP et le numéro RNIS se fait manuellement.

Voyons les particularités de RNIS. Étant une ligne téléphonique, le coût d'une communication RNIS dépend de sa durée. Éviter une trop grande durée de communication nous incite à prendre certaines mesures. Le système IOS de Cisco permet à cet effet de définir le *trafic privilégié* qui seul peut activer une connexion. Si aucun trafic privilégié n'est défini au préalable, la connexion RNIS ne pourra se faire, même si des données sont en attente d'être envoyées. Une fois la connexion RNIS établie, le trafic privilégié ou non peut être transmis. Dès que le trafic privilégié cesse, la connexion est coupée après une temporisation dont la durée est configurable. À chaque reprise de la transmission du trafic privilégié, cette temporisation est réinitialisée. Une autre mesure optionnelle concerne l'authentification de l'appelant. Comme la connexion RNIS passe souvent par un opérateur du réseau public, un individu non autorisé peut essayer de se connecter *via* celui-ci, à un site privé. Pour éviter une telle intrusion, plusieurs types d'authentification sont disponibles. Dans notre exemple à la fin du chapitre, c'est celui du *chap* qui est utilisé.

Le chapitre 7 donne quelques compléments d'information sur la configuration de RNIS. Pour l'heure, nous nous limiterons à l'exemple illustré sur la figure 1.20 et les listings correspondants 1.14 et 1.15.

**Figure 1.20**

*Deux routeurs connectés via le réseau RNIS.*




**Listing 1.14. Configuration du routeur R1.**

```
username g4 password 0 tsunami
isdn switch-type basic-ni1

interface BRI0
 ip address 10.1.0.3 255.255.255.0
 encapsulation ppp
 isdn spid1 3840000001
 isdn spid2 3840000002
 dialer map ip 10.1.0.4 name g4 broadcast 384020
 dialer-group 1
 ppp authentication chap

dialer-list 1 protocol ip permit
```

© copyright Éditions Eyrolles

**Listing 1.15. Configuration du routeur R2.**

```
username g3 password 0 tsunami
isdn switch-type basic-ni1

interface BRI0
 ip address 10.1.0.4 255.255.255.0
 encapsulation ppp
 isdn spid1 3840200001
 isdn spid2 3840200002
 dialer map ip 10.1.0.3 name g3 broadcast 348000
 dialer-group 1
 ppp authentication chap

dialer-list 1 protocol ip permit
```

Dans les listings de configuration des routeurs R1 et R2, la commande **isdn switch-type** <*type de commutateur*> définit le type de commutateur RNIS utilisé par l'opérateur. La commande **dialer map** est très comparable en termes de fonctionnalité à celle de **Relay map** que nous avions utilisée dans la configuration du Frame Relay. Ces commandes permettent d'associer une adresse IP à une adresse de la couche d'accès réseau qui, dans le cas du RNIS, est le numéro d'appel. Les commandes **encapsulation ppp**, **ppp authentication** <*type d'authentification*> et **username** <*nom du routeur distant*> **password** <*type de chiffrage*> <*mot de passe*> définissent le type d'encapsulation, l'authentification par chap, et enfin, la vérification de l'appel entrant. La commande **dialer-list** <*numéro de groupe*> **protocol <** *nom du protocole*> **permit** définit le protocole IP comme trafic privilégié. Pour avoir un contrôle plus fin sur ce trafic, il est conseillé d'associer à cette commande, une liste d'accès au lieu du **permit** simple. Enfin, la paire de commandes **isdn spid** <*numéro de profil du service*> concerne chacun des canaux B de l'interface BRI. Certains commutateurs RNIS peuvent ne pas en avoir besoin.

© copyright Éditions Eyrolles

# 2

# Le pontage avec les routeurs Cisco

**Solutions de configuration présentées dans ce chapitre**



Les ponts sont des appareils qui opèrent à la couche d'accès réseau du modèle Internet. Alors que celui-ci ne fournit que des spécifications sommaires, le modèle OSI est extrêmement précis. À la couche d'accès réseau du modèle Internet correspondent deux couches du modèle OSI, la couche liaison et la couche physique. La couche liaison est décomposée en deux sous-couches que sont la couche LLC (*Logical Link Control*) et la couche MAC (*Media Access Control*) de contrôle d'accès au support physique.

© copyright Éditions Eyrolles

Le détail de ces spécifications sort du cadre de cet ouvrage, mais pour la bonne compréhension du pontage, il faut connaître le mécanisme d'adressage, fonctionnalité de la sous-couche MAC.

Nous utiliserons le terme de couche liaison chaque fois qu'il sera fait allusion à la couche accès réseau du modèle Internet. Nous appellerons segments LAN, ceux non reliés par des ponts, et LAN pontés, les autres.

## Adresses MAC

Dans le datagramme Internet, la trame MAC ou la PDU créée par l'entité de la couche liaison, contient l'information d'adressage sur la source ou la destination, appelée adresse MAC.

Nous n'examinerons la structure des adresses MAC que des technologies LAN Ethernet et Token Ring qui sont les plus utilisées avec FDDI, en sachant que cet adressage est similaire pour d'autres normes.

Les adresses MAC sont typiquement utilisées pour joindre les hôtes situés sur les segments réseau. Elles sont gravées en mémoire morte ou ROM (*Read Only Memory*) sur la carte d'interface réseau ou NIC (*Network Interface Card*). Les adresses MAC d'Ethernet et de Token Ring ont une longueur de 48 bits, parmi lesquels deux ont une signification particulière précisée plus loin.

Une différence notable entre les adresses MAC d'Ethernet et celles de Token Ring est l'ordre des bits au sein des octets individuels. L'exemple suivant de la même adresse MAC le montre :

```
Ethernet : 00000000.11001100.10101111
Token Ring : 00000000.00110011.11110101
```

Si l'adresse MAC a tous ses bits positionnés à 1, il s'agit d'une adresse de *diffusion* (*broadcast*) permettant de joindre tous les hôtes d'un segment. Les deux bits mentionnés plus haut font partie des deux derniers du premier octet dans le cas d'Ethernet et des deux premiers du même octet pour Token Ring. Voir figure 2.1.

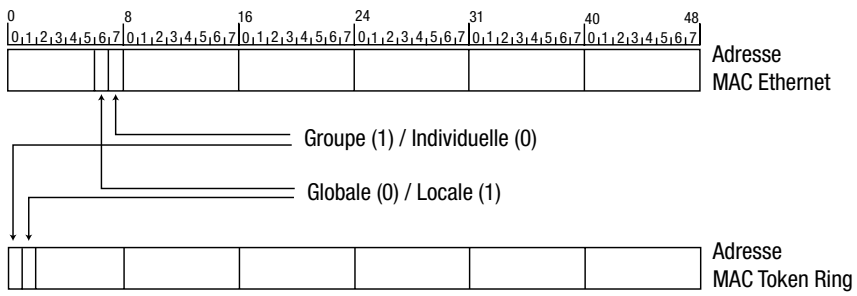



**Figure 2.1**
*Bits particuliers dans les adresses MAC d'Ethernet et de Token Ring.*

Ces deux bits sont appelés *Groupe/Individuelle* (G/I) et *Globale/Locale* (G/L), respectivement. Si le bit G/I est à zéro, l'adresse MAC désigne une adresse individuelle ; s'il est à un, on est dans le cas d'une adresse de groupe logique d'hôtes appelée aussi adresse multidestinataire (*multicast*). L'adresse MAC multidestinataire ne doit pas apparaître comme adresse source dans l'en-tête des trames, sauf exception abordée plus loin dans la section « configuration du pontage à routage par la source » (*source route bridging*) de ce chapitre.

© copyright Éditions Eyrolles

Le bit G/L précise si l'adresse MAC a été assignée par l'IEEE (*Institute of Electronic and Electrical Engineers*), un comité des normes américaines, qui distribue les adresses MAC. Les fabricants de cartes d'interface peuvent acquérir un bloc d'adresses de cet organisme. Dans ce cas, le bit G/L est toujours à zéro, indiquant ainsi que les trois octets suivants ne peuvent être assignés que par le fabricant. En revanche, les adresses avec le bit G/L à un, sont d'utilisation libre.

## Le pontage transparent

Le pontage transparent est comme son nom l'indique, totalement transparent aux hôtes reliés à un LAN ponté. La spécification formelle de cette technique, rédigée dans le document de l'IEEE (cf. la norme 802.1D), stipule les règles suivantes :

- Un pont possède plusieurs ports qui peuvent être en état *acheminement* ou *bloqué ;* dans le premier cas, il peut envoyer ou recevoir des trames ; dans le second, il en est incapable ; la mise en état bloqué d'un port permet d'éviter la duplication des trames due à la présence de deux chemins parallèles actifs simultanément.
- Pour minimiser la quantité de trafic sur un LAN ponté, les adresses MAC circulant dans chaque segment sont mémorisées par le pont dans une base de données de filtrage pour chacun de ses ports ; tout envoi d'une trame déclenche la consultation de cette base et ne devient effectif à travers le port concerné que si le pont y trouve l'adresse de destination correspondante.
- Une trame dont l'adresse MAC de destination est absente de la base est envoyée à travers les ports (*flooding*) en état acheminement, sauf celui par lequel il a l'a reçue.
- Même processus si la trame à envoyer contient une adresse *multidestinataire* (*multicast*) ou de *diffusion générale* (*broadcast*).
- Les ponts échangent des informations de topologie pour mettre leurs ports dans un état bloqué ou acheminement ; ce protocole de pontage se déroule suivant un algorithme dit d'*arbre de recouvrement* (*spanning tree*) décrit dans la section suivante.

### L'algorithme d'arbre de recouvrement

Le problème qui se pose est de trouver un ensemble de chemins sans boucle reliant tous les segments du réseau ; la résolution de ce problème – complexe du fait du caractère changeant de la situation (rajout ou panne d'un segment…) – fait appel à la théorie des graphes qui dénomme arbre de recouvrement un tel ensemble. L'algorithme qui à un réseau donné associe un arbre de recouvrement s'appelle algorithme d'arbre de recouvrement. Un tel algorithme permet une reconfiguration dynamique des différents composants d'un LAN ponté, tels qu'un hôte, un segment ou un pont, qui peuvent être ajoutés ou retirés. Il permet aussi de gérer les changements intervenus à la suite de la défaillance de l'un de ces composants, pour maintenir en permanence un état de connectivité globale sans boucles en tenant compte des chemins parallèles. Chaque fois qu'un événement survient, l'algorithme d'arbre de recouvrement recalcule un nouvel arbre.

L'algorithme d'arbre de recouvrement se déroule ainsi : dans tout LAN ponté, un seul pont est élu « racine » (*root) ;* tous les autres déterminent lequel parmi leurs ports permet d'accéder par le plus court chemin à ce pont racine, qui devient ainsi le *port racine* (*root port*) mis en état acheminement ; les ponts rattachés à chaque segment doivent en désigner un ayant le plus court chemin au pont racine ; ce *pont désigné* (*designated bridge*), seul habilité à envoyer des paquets vers le pont racine et à en recevoir pour ce segment, choisit un seul de ses ports en tant

© copyright Éditions Eyrolles

que *port désigné* (*designated port*) pour les opérations de transfert de paquets ; le pont racine est le pont désigné pour tous les segments auxquels il est rattaché ; enfin, tous les ports de tous les ponts, qui ne sont ni port racine, ni port désigné, sont mis en état bloqué.

L'exemple suivant décrit le déroulement de l'algorithme de l'arbre de recouvrement sur un réseau dont le plan est illustré à la figure 2.2.

En supposant que B1 soit un candidat pour le pont racine, le calcul de l'arbre de recouvrement mène à la topologie de la figure 2.3. Les ports en état acheminement sont reliés par des traits épais, et les ports racine sont étiquetés avec le sigle RP (*root port*) entouré d'un cercle.

**Figure 2.2**
*Plan physique d'un LAN ponté.*

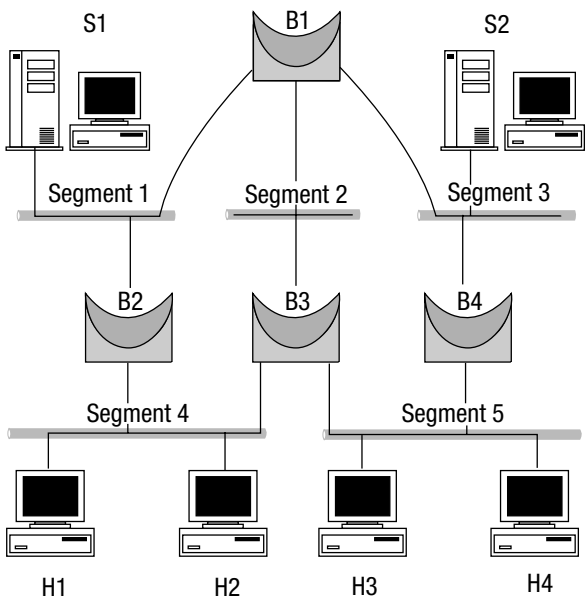
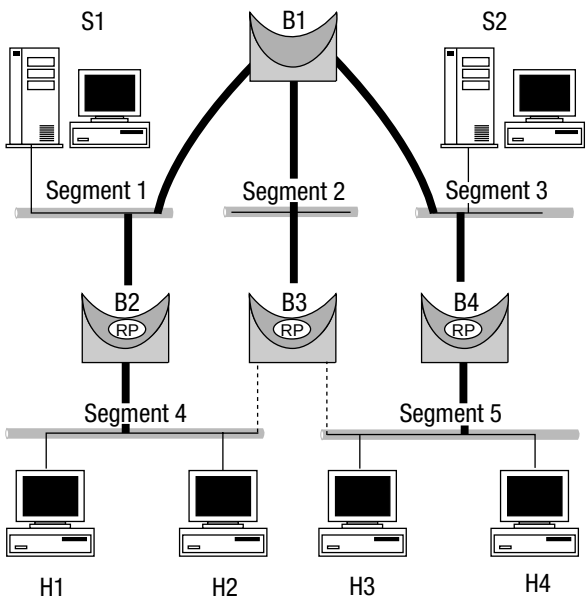



**Figure 2.3**
*Topologie d'arbre de recouvrement d'un LAN ponté.*




© copyright Éditions Eyrolles

Suite au déroulement de l'algorithme d'arbre de recouvrement, B2 et B4 sont devenus les ponts désignés pour les segments 4 et 5, respectivement. Le pont B3 met donc en état bloqué, ses deux ports. On y voit aussi que le pont racine B1 est aussi le pont désigné pour tous les segments auxquels il est rattaché.

Ce qui saute aux yeux après l'exécution de l'algorithme de l'arbre de recouvrement est le maintien de la connectivité générale permettant à tout hôte de communiquer avec un autre dans n'importe quel segment ainsi que la création d'une topologie arborescente évitant les paquets dupliqués ou pis encore en multiples exemplaires.

Cependant la topologie d'arbre de recouvrement peut ne pas générer un chemin optimal pour les paquets. Par exemple, si les hôtes H1 et H3 doivent communiquer, leurs données doivent transiter par les ponts B2, B1 et B4 au lieu d'aller directement par le pont B3.

Que se passe-t-il si l'un des ponts est défaillant ? Par exemple, si le pont B4 est en panne l'algorithme d'arbre de recouvrement recalcule la topologie telle qu'elle est montrée à la figure 2.4. Le pont B3 devient le pont désigné pour le segment 5, sauvegardant ainsi la connectivité générale sans boucles.

L'algorithme d'arbre de recouvrement fonctionne même en cas de défaillance du pont racine ; la topologie résultante est illustrée à la figure 2.5.

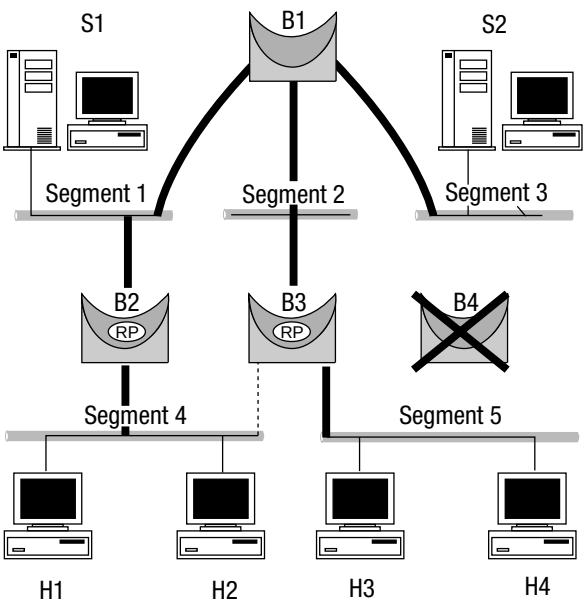
Le pont B2 devient le pont racine, et les ponts B3 et B4 s'y adaptent en conséquence. Le pont B3 devient le pont désigné pour les segments 2 et 5, et le pont B4 devient le pont désigné uniquement pour le segment 3, maintenant la connectivité requise.

Le but n'était pas de couvrir tous les scénarios possibles dans ces exemples, mais de mettre en évidence les caractéristiques principales de l'algorithme d'arbre de recouvrement : maintien de la connectivité générale en utilisant les chemins parallèles et calcul d'une nouvelle topologie arborescente après chaque événement déstabilisant.

Les détails de fonctionnement de l'algorithme de l'arbre ne sont pas du ressort de cet ouvrage. Nous étudierons cependant les paramètres les plus importants de cet algorithme dans la section suivante de ce chapitre « Solutions de configuration ».

**Figure 2.4**

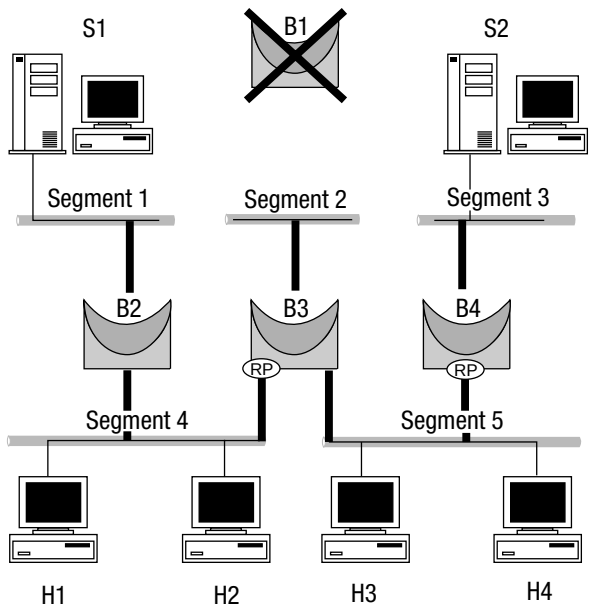
*Topologie d'arbre de recouvrement recalculée suite à défaillance du pont B4.*




© copyright Éditions Eyrolles

**Figure 2.5**

*Topologie d'arbre de recouvrement recalculée suite à défaillance du pont racine B1.*




## Pontage avec routage par la source (SRB)

IBM avait conçu le *pontage à routage par la source* ou SRB (*Source Route Bridging*) spécialement pour les réseaux Token Ring ou ceux capables de l'émuler. Le SRB confie à l'hôte source la mission d'insérer dans l'en-tête toutes les données nécessaires au routage, ce qui le différencie du pontage transparent.

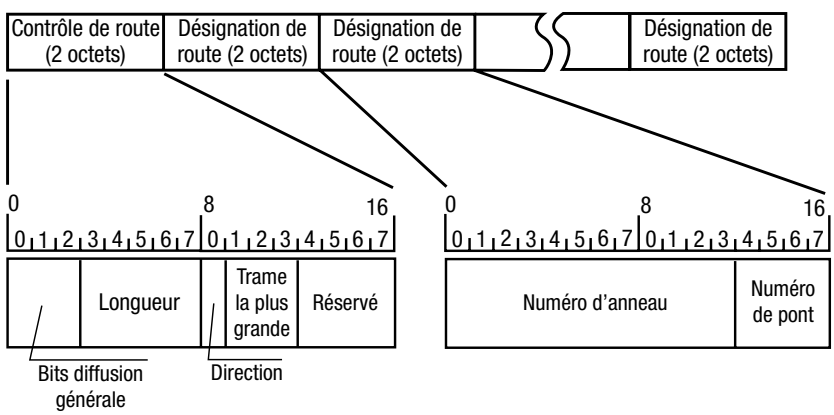
Un LAN ponté à routage par la source doit remplir les conditions suivantes :

- À chaque anneau Token Ring doit être assigné un numéro unique.
- À chaque pont connecté à un anneau doit être assigné un numéro unique.
- Chaque pont à routage par la source « classique » ne peut être connecté à plus de deux anneaux.

L'en-tête des trames Token Ring contient le champ *information de routage* ou RIF (*Routing Information Field*) dont le format est illustré sur la figure 2.6.

**Figure 2.6**

*Champ RIF d'une trame Token Ring.*




© copyright Éditions Eyrolles

La présence du champ RIF est signalée en positionnant le bit G/I à un. La trame contenant ce champ n'est pas nécessairement propre à Token Ring ; il peut s'agir d'une trame ordinaire de diffusion générale contenant une PDU liée à une fonction de la couche supérieure, telle une requête ARP.

Les bits *diffusion générale* (*broadcast*) font partie du sous-champ *contrôle de route* du RIF ; ils indiquent s'il s'agit d'une trame de diffusion générale multiroute (*all routes broadcast*), monoroute (*single route broadcast*) ou *route spécifique*. Les différentes valeurs de ce champ se trouvent dans le tableau 2.1.

**Table 2.1. Signification des bits de diffusion générale.**

| Bits diffusion générale | Description |
|---|---|
| 000 | route spécifique (non diffusion) |
| 100 | diffusion générale multiroute |
| 110 | diffusion générale monoroute |

Le champ *longueur* indique la longueur totale du RIF en octets. Seuls les nombres pairs de 2 à 30 sont admis. Selon la spécification de IBM, le nombre maximal de ponts traversés ne peut être supérieur à sept.

Le champ *direction* indique le sens de lecture des *signalisations de route* (route designators). Si le bit correspondant est positionné à un, les ponts doivent interpréter ces informations en partant de la dernière *signalisation* vers la première.

Le champ *trame la plus grande* (*largest frame*) indique la MTU supportée par tous les ponts traversés le long du parcours défini dans le RIF. Le tableau 2.2 donne pour chaque valeur de trame la plus grande, la MTU correspondante et la taille maximale du datagramme IP.

**Tableau 2.2. Relation entre valeur de trame la plus grande, la MTU et taille maximale de datagramme IP.**

| Trame la plus grande | MTU | Taille maximale de datagramme IP |
|---|---|---|
| 000 | 552 | 508 |
| 001 | 1064 | 1020 |
| 010 | 2088 | 2044 |
| 011 | 4136 | 4092 |
| 100 | 8232 | 8188 |

Bien entendu, le nœud source doit connaître la route vers la destination, ce qu'il découvre en lançant des trames d'exploration de diffusion générale multiroute (*all routes broadcast*) ou monoroute (*single route broadcast*).

Le nœud source envoie une trame d'exploration multiroute diffusée systématiquement sur toutes les interfaces des ponts traversés (*flooding*). Cette trame positionne le premier bit de l'adresse source, en l'occurrence dénommé RII (*Routing Information Indicator*), à un pour indiquer la présence du champ RIF dont seul le sous-champ contrôle de route est renseigné. Le sous-champ diffusion générale contient la valeur 100 pour indiquer qu'il s'agit d'une diffusion

© copyright Éditions Eyrolles

multiroute. Au fur à mesure que la trame d'exploration traverse le réseau, chaque pont adjoint au champ RIF un enregistrement qui comprend les numéros de l'anneau de provenance et le sien propre. Le pont n'envoie pas la trame d'exploration à l'anneau suivant si elle contient déjà son numéro dans le RIF. S'il existe plusieurs chemins vers le destinataire, celui-ci va recevoir plusieurs copies de la trame d'exploration, chacune renfermant une route particulière. Il répond normalement à toutes celles qui sont reçues. Et la source choisit la route de la première réponse, suite à l'envoi de sa trame d'exploration d'origine.

Une autre façon de découvrir une route vers la destination, c'est l'envoi de trames d'exploration monoroute (*single route explorers*) appelées aussi trames d'exploration de recouvrement (*spanning explorers*). Le mode opératoire de cette dernière méthode diffère sur quelques détails de l'exploration multiroute (*all routes explorers*). Quand le nœud crée un champ RIF, il remplit le sous-champ diffusion générale avec la valeur 110 qui correspond à la diffusion générale monoroute. Les ponts traversés y adjoignent les désignations de route comme dans le cas des trames d'exploration multiroute, mais au lieu de les envoyer à travers tous les chemins possibles (*flooding*), ils ne les envoient que par les ports qui ont été configurés manuellement ou automatiquement pour l'envoi en diffusion générale monoroute. Le destinataire, quant à lui, répond selon la même procédure.

Comme on peut le constater, le pontage SRB nécessite l'intervention des hôtes. Le pilote de la carte d'interface réseau Token Ring fournit toutes les fonctionnalités nécessaires aux protocoles des couches supérieures qui choisissent de l'utiliser, ainsi que le type de trame à envoyer.

La RFC 1042 est le document de spécifications pour le fonctionnement de IP et de ARP dans un environnement ponté à routage par la source. Les points qu'on peut en citer sont les suivants :

- Les protocoles IP et ARP requièrent l'utilisation de la fonctionnalité de pontage à routage par la source.
- Quand le protocole ARP doit traduire une adresse IP en adresse MAC, il doit recourir en premier à l'adresse de *diffusion générale toutes stations* (*all stations broadcast*), c'est-à-dire une adresse destinée à la diffusion générale ne contenant pas de champ RIF (RII = 0) ; s'il n'obtient pas de réponse dans un délai prédéterminé, il doit envoyer une trame d'exploration en diffusion générale, soit multiroute, soit monoroute.
- Les diffusions générales de IP doivent utiliser uniquement la trame d'exploration monoroute.

# Solutions de configuration

## Configuration du pontage transparent

Le routage et le pontage sont deux fonctions comparables qui consistent à l'envoi de paquets. Mais leur règles d'exécution ne sont pas les mêmes, quand elles ne sont pas contradictoires, ce qui rend leur fonctionnement simultané pratiquement impossible au sein d'un même appareil. Au fil des ans, les ingénieurs de chez Cisco trouvèrent des solutions innovantes pour que ces deux fonctions ne soient pas mutuellement exclusives. Ces solutions se traduisirent par trois versions majeures, chacune d'elles étant une amélioration de la précédente.

À l'origine, le pontage transparent et le routage IP (ou d'autres protocoles) ne pouvaient fonctionner simultanément au sein d'un routeur Cisco, même si ces deux fonctions étaient

© copyright Éditions Eyrolles

incluses. On devait désactiver l'une pour pouvoir utiliser l'autre. Dans la version de l'IOS 11.0, on vit apparaître le routage et le pontage simultanés ou CRB (*Concurrent Routing and Bridging*). Le CRB permet le routage et le pontage en même temps, mais sur des interfaces différentes, sans aucune communication entre elles. C'était comme s'il y avait deux appareils (un pont et un routeur) en un. En fin de compte, c'est dans la version 11.2 de l'IOS que l'on aboutit à une meilleure solution, le routage et le pontage intégrés ou IRB (*Integrated Routing and Bridging*). Celle-ci permet de passer le trafic IP ou celui d'autres protocoles entre les processus de routage et de pontage.

Heureusement, le CRB et le IRB n'ajoutent que quelques nouvelles commandes à celles qui étaient disponibles auparavant. La section suivante décrit les versions que sont le pontage seul, le CRB et le IRB.

### Utilisation d'un pontage monogroupe sur un routeur

La configuration du pontage la plus élémentaire nécessite de désactiver le routage IP. Par conséquent, toutes les configurations sauf celles se rapportant à CRB et IRB, ont en commun la commande **no ip routing**.

Les étapes à suivre pour configurer un routeur Cisco en pontage transparent pour IP, sont les suivantes :

1. Désactiver le routage IP par la commande **no ip routing** en mode de configuration globale ;
2. Créer un numéro de groupe de pontage en mode de configuration globale par la commande **bridge** <*numéro de groupe*> **protocol** <*protocole*> *;* chaque numéro de groupe définit un pont logique séparé au sein du même routeur ; le paramètre protocole comporte trois choix possibles : **ieee**, **dec** ou **ibm** ; le premier est conforme à la norme IEEE 802.1D, le second est celui de Digital Equipment Corporation, le premier à implanter l'arbre de recouvrement ; le troisième, enfin, est la version d'IBM, utilisée dans un environnement de pontage à routage par la source.
3. Assigner les interfaces créées au numéro de groupe de pontage par la commande **bridge-group** <*numéro de groupe*>.

**AVERTISSEMENT** La commande **ip routing** n'apparaît pas à la configuration du routeur, ce qui peut entraîner un oubli concernant sa désactivation par **no ip routing**. Sans cette dernière commande aucun pontage n'est possible, à moins d'utiliser le CRB ou le IRB. Malheureusement, le routeur ne rendra compte d'aucune erreur et n'affichera aucun message d'avertissement pour signaler cet oubli. Quand le trafic IP censé être ponté, atteindra le routeur, celui-ci essayera de le router sans y parvenir. Là encore, aucune erreur ne sera signalée pour indiquer cette mauvaise configuration.

La figure 2.7 montre un routeur utilisé pour le pontage du trafic IP entre deux segments. La configuration de ce routeur se trouve sur le listing 2.1.

**Listing 2.1. Configuration du routeur R1.**

```
no ip routing

bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1
```

© copyright Éditions Eyrolles

```
interface Ethernet1
 bridge-group 1
```

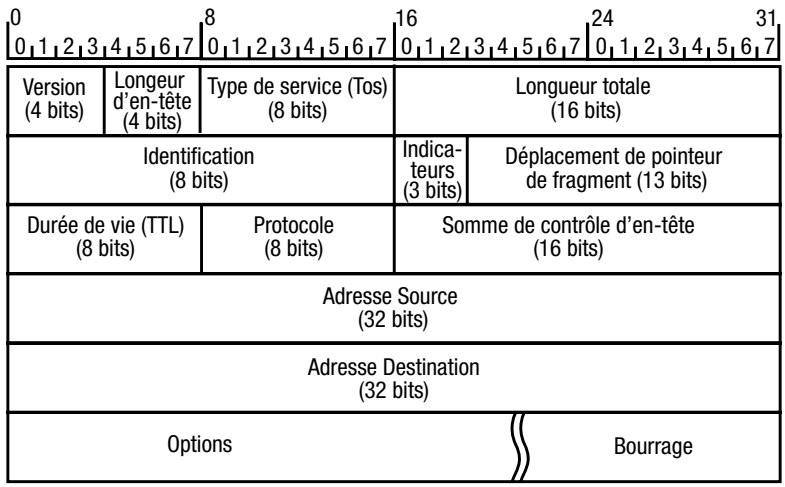



**Figure 2.7**
*Utilisation d'un routeur en pontage entre deux segments.*

Bien que séparés par un routeur, les hôtes H1 et H2 doivent être configurés avec la même adresse de sous-réseau.

Comme nous l'avons évoqué dans la section précédente, les ponts maintiennent une base de données de filtrage d'adresses. Un routeur permet d'afficher cette base au moyen de la commande **show bridge** <*numéro de groupe*>. Si on omet ce dernier paramètre, le routeur affiche les bases pour tous les groupes qui ont été définis. Les résultats de cette commande sur le routeur R1 de la figure 2.7 se trouvent sur le listing 2.2.

**Listing 2.2. Résultats de la commande `show bridge` sur le routeur R1.**

```
R1#show bridge

Total of 300 station blocks, 295 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action   Interface  Age   RX count  TX count
0260.8c4c.1132  forward   Ethernet1   3          41        32
0060.5cc4.f4c5  forward   Ethernet0   0         475         0
0060.b01a.9e1c  forward   Ethernet0   3         500       202
```

Le champ **Age** indique le nombre de minutes écoulé depuis la dernière émission ou réception par l'adresse MAC correspondante. Les champs **RX count** et **TX count** affichent le nombre de trames émises ou reçues par cette même adresse.

**ASTUCE** Bien que le routage IP soit désactivé, les interfaces individuelles peuvent quand même être configurées avec une adresse IP par la commande **ip address** <*adresse IP*> < *masque de sous-réseau*>, permettant ainsi d'accéder au routeur *via* telnet ou d'autres télécommandes. Cette adresse doit appartenir au même sous-réseau que celui des hôtes connectés à l'interface du routeur.

© copyright Éditions Eyrolles

### Utilisation du pontage en multigroupe

Il est possible de configurer un pontage en multigroupe dans un même routeur Cisco. Comme nous l'avons déjà évoqué, les groupes de pontage constituent autant de ponts logiques séparés dans un même routeur.

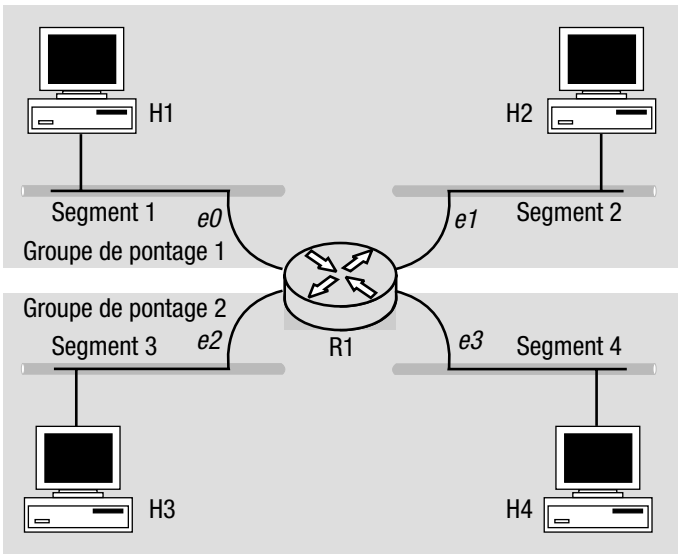
Les étapes à suivre pour configurer un routeur en pontage multigroupe sont les suivantes :

1. Désactiver le routage IP par la commande **no ip routing** en mode de configuration globale ;
2. Créer un pontage multigroupe en mode de configuration globale par la commande **bridge** <*numéro de groupe*> **protocol** <*protocole*>, autant de fois qu'il est nécessaire de créer des groupes différents ;
3. Assigner les interfaces aux groupes de pontage appropriés par la commande **bridge-group** <*numéro de groupe*>.

Prenons le cas qui est illustré sur la figure 2.8. Le routeur R1est configuré selon le listing 2.3.

**Figure 2.8**

*Un même routeur configuré avec un pontage multigroupe.*




```
no ip routing

bridge 1 protocol ieee
bridge 2 protocol ieee

interface Ethernet0
 bridge-group 1

interface Ethernet1
 bridge-group 1

interface Ethernet2
 bridge-group 2

interface Ethernet3
 bridge-group 2
```

© copyright Éditions Eyrolles

Bien que les quatre segments soient connectés au même routeur, aucune communication n'est possible entre la branche comprenant les segments 1/2 et celle comprenant 3/4. Même si les hôtes des quatre segments appartiennent au même sous-réseau, ils ne peuvent communiquer qu'à l'intérieur d'une même branche. Par exemple, les hôtes H1 et H2 peuvent communiquer entre eux, mais ni l'un, ni l'autre ne pourront communiquer avec H3 ou H4. Ces derniers, bien évidemment, peuvent communiquer entre eux.

La commande **show bridge** utilisée sur le routeur R1 sans paramètre a pour effet d'afficher les deux bases de données de filtrage d'adresses montrées sur le listing 2.4.

**Listing 2.4. Les bases de données de filtrage du routeur R1.**

```
R1#show bridge

Total of 300 station blocks, 295 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action   Interface  Age  RX count  TX count
0260.8c4c.1132  forward  Ethernet1   3         41        32
0060.5cc4.f4c5  forward  Ethernet0   0        475         0
0060.b01a.9e1c  forward  Ethernet0   3        500       202

Bridge Group 2:

0060.97fb.566a  forward  Ethernet3   0          5         4
0260.8ca3.28cd  forward  Ethernet2   0         41        10
```

## Configuration du pontage transparent sur support physique mixte

Il est possible de configurer le pontage pour des supports non LAN ou pour des LAN mixtes comme Ethernet et Token Ring.

Il est important de pouvoir utiliser le pontage sur des supports physiques non LAN pour deux raisons. La première, c'est d'avoir à fournir une connexion à la couche liaison pour des protocoles non routables tel que le NetBEUI ou le LAT. La deuxième, c'est de pouvoir disposer d'une ligne de secours pour un support de connexion principal ponté de plus haut débit, comme Ethernet.

Contrairement au pontage sur des supports non LAN, celui qui implique des LAN mixtes comme Ethernet et Token Ring semble peu justifié. Cette dernière technique est très proche de celle du *pontage traducteur* entre ces deux types de LAN, que nous verrons à la fin de ce chapitre.

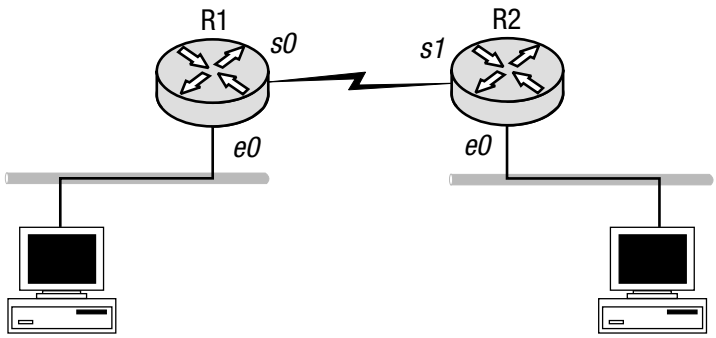
### Pontage sur HDLC

La méthode la plus facile pour configurer un pontage sur un support non LAN, c'est de l'effectuer sur une connexion HDLC. Cela revient à utiliser la même méthode que celle déjà vue lors des configurations monogroupe ou multigroupe sur un support LAN.

© copyright Éditions Eyrolles

La figure 2.9 montre un exemple de deux routeurs en pontage de segments Ethernet sur une ligne série. Les listings 2.5 et 2.6 contiennent les commandes correspondantes pour ces routeurs.

**Figure 2.9**
*Pontage de deux segments Ethernet avec deux routeurs connectés par une ligne série et configurée en encapsulation HDLC.*




**Listing 2.5. Configuration du routeur R1.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial0
 bridge-group 1
```

**Listing 2.6. Configuration du routeur R2.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial1
 bridge-group 1
```

La commande **show-bridge** dans chaque routeur donne la sortie typique des listings 2.7 et 2.8. Mais au lieu de pointer sur l'interface LAN, les adresses MAC pointent sur la ligne série.

**Listing 2.7. Base de données de filtrage d'adresses de R1.**

```
R1#show bridge

Total of 300 station blocks, 296 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  Serial0      0         15         4
```

© copyright Éditions Eyrolles

```
0060.b01a.9e1c  forward  Ethernet0    0         10        10
00e0.b064.30a9  forward  Ethernet0    1         30         0
0260.8ca3.28cd  forward  Serial0      0         42         4
```

**Listing 2.8. Base de données de filtrage d'adresses de R2.**

**Solution apparentée**
Encapsulation, HDLC, p. 42.

```
R2#show bridge

Total of 300 station blocks, 296 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  Ethernet0    0         19         6
0060.b01a.9e1c  forward  Serial1      0         12        11
00e0.b064.30a9  forward  Serial1      1         30         0
0260.8ca3.28cd  forward  Ethernet0    0         42         4
```

### Pontage sur Frame Relay

Par rapport à HDLC, les autres types d'interface série nécessitent un peu plus d'efforts de la part de l'administrateur réseau. C'est le cas de Frame Relay.

Les étapes à suivre pour configurer une interface en encapsulation Frame Relay, sont les suivantes :

1. Désactiver le routage IP par la commande **no ip routing** en mode de configuration globale.
2. Créer un groupe de pontage ou plus par la commande **bridge** <*numéro de groupe*> **protocol** <*protocole*> en mode de configuration globale.
3. Configurer une interface série pour le Frame Relay par la commande **encapsulation frame-relay**, et le cas échéant, par la commande **frame-relay lmi-type** <*type de LMI*>. À vous de choisir suivant vos besoins, s'il faut créer ou non des sous-interfaces qui seront point à point ou multipoint.
4. Assigner par la commande **bridge-group**, un numéro de groupe de pontage (défini à l'étape 2), à toutes les interfaces concernées ou aux sous-interfaces si elles existent.
5. Si vous utilisez des sous-interfaces multipoint ou si vous n'utilisez pas de sous-interfaces, par la commande **frame-relay map bridge** <*DLCI*> [**broadcast**], vous pouvez affecter la fonction de pontage au DLCI correspondant ; (pour les sous-interfaces point à point, cela se fait automatiquement).

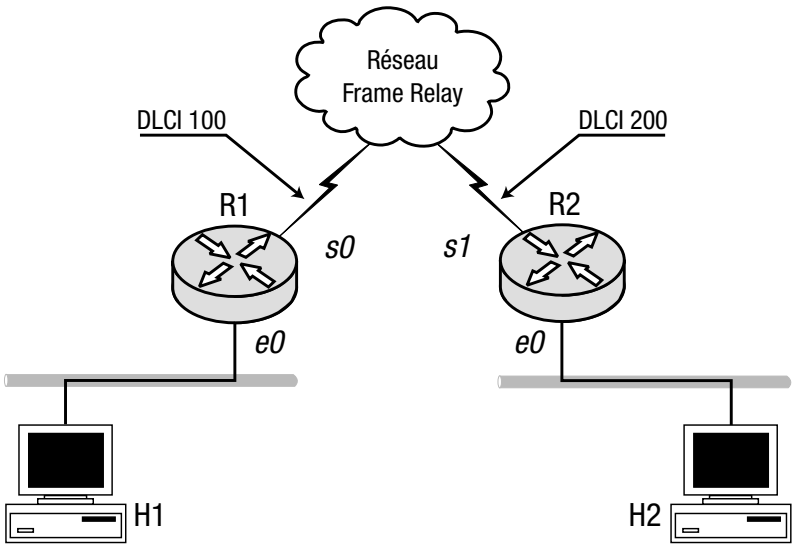
**REMARQUE** Le mot clef **broadcast**, bien qu'optionnel, doit figurer dans la commande pour permettre aux routeurs d'échanger des paquets spéciaux appelés BPDU (*Bridge Protocol Data Units*).

La figure 2.10 montre comment deux routeurs peuvent connecter, un segment Ethernet chacun, à un réseau Frame Relay *via* un CVP.

Les listings 2.9 et 2.10 montrent la configuration des routeurs pour le pontage en utilisant une sous-interface multipoint.

© copyright Éditions Eyrolles

**Figure 2.10**
*Pontage de deux segments Ethernet par deux routeurs connectés à un réseau Frame Relay.*




**Listing 2.9. Configuration du routeur R1.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 multipoint
 frame-relay map bridge 100 broadcast
 bridge-group 1
```

**Listing 2.10. Configuration du routeur R2.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial1
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial1.1 multipoint
 frame-relay map bridge 200 broadcast
 bridge-group 1
```

© copyright Éditions Eyrolles

On peut aussi définir alternativement une sous-interface point à point, auquel cas on utilise les commandes des listings 2.11 et 2.12.

**Listing 2.11. Configuration du routeur R1.**

```
no ip routing
bridge 1 protocol ieee

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 frame-relay interface-dlci 100
 bridge-group 1
```

**Listing 2.12. Configuration du routeur R2.**

```
no ip routing
bridge 1 protocol ieee

interface Serial1
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial1.1 point-to-point
 frame-relay interface-dlci 200
 bridge-group 1
```

Les sorties des listings 2.13 et 2.14 sont les résultats de la commande **show bridge** sur les routeurs R1 et R2, respectivement. Comme nous pouvons le constater, certaines lignes pointent sur des sous-interfaces.

**Listing 2.13. Base de données de filtrage d'adresses du routeur R1.**

```
R1#show bridge

Total of 300 station blocks, 296 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  Serial0.1    2         11         4
0060.97fb.566a  forward  Ethernet0    0         73         4
0060.b01a.9e1c  forward  Ethernet0    0         39         9
0260.8ca3.28cd  forward  Serial0.1    0         36        10
```

**Listing 2.14. Base de données de filtrage d'adresses du routeur R2.**

```
R2#show bridge

Total of 300 station blocks, 296 free
```

© copyright Éditions Eyrolles

**Solution apparentée**
Configuration de IP sur Frame Relay, p. 44.

```
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  Ethernet0    0         12         4
0060.97fb.566a  forward  Serial1.1    0         73         4
0060.b01a.9e1c  forward  Serial1.1    0         34         9
0260.8ca3.28cd  forward  Ethernet0    0         36        10
```

### Pontage sur RNIS

La configuration du pontage sur une interface d'accès de base (BRI) du RNIS est assez comparable à celle d'une sous-interface multipoint en encapsulation Frame Relay. Le travail consiste à en substituer la partie spécifique à RNIS par les étapes suivantes :

1. Désactiver le routage IP par la commande **no ip routing** en mode de configuration globale.
2. Créer un groupe de pontage par la commande **bridge** <*numéro de groupe*> **protocol** <*protocole*>.
3. Affecter le type de commutateur RNIS par la commande **isdn switch-type** <*type de commutateur*> en mode de configuration globale.
4. Définir le trafic privilégié par la commande **dialer-list** <*numéro de groupe d'appel*> **protocol** <*protocole*> {**permit**|**deny**} ou **list** <*numéro de liste de code type Ethernet*>, en mode de configuration globale ; le paramètre protocole est **bridge** dans notre cas.
5. Définir le type d'encapsulation, PPP ou HDLC, sur l'interface BRI ; si l'authentification est nécessaire, les utilisateurs doivent être définis par la commande **username** <*nom utilisateur*> **passsword** <*mot de passe*> en mode de configuration globale avec le type d'authentification défini par la commande **ppp authentification** {**pap**|**chap**} en mode de configuration interface ; suivant la version du système IOS de Cisco installée dans les routeurs, des lignes supplémentaires de configuration peuvent être nécessaires pour l'authentification ; se reporter à la documentation du constructeur.
6. Assigner un numéro de groupe à l'interface BRI par la commande **dialer-group** <*numéro de groupe d'appel*> qui est défini dans la commande **dialer-list**.
7. Le cas échéant, utiliser la commande **isdn spid** <*numéro de profil de service*> pour l'affecter à chacun des canaux B.
8. Assigner à l'interface BRI, le numéro de groupe de pontage défini à l'étape 2, par la commande **bridge-group** <*numéro de groupe*>.
9. Enfin, définir le mapping entre la fonction de pontage et le numéro d'appel du routeur distant par la commande **dialer map** <*protocol*> **name** <*nom du routeur distant*> [**broadcast**] <*numéro d'appel*> ; le paramètre protocole est **bridge** dans notre cas ; pour plus de détails, se reporter à la documentation de Cisco.

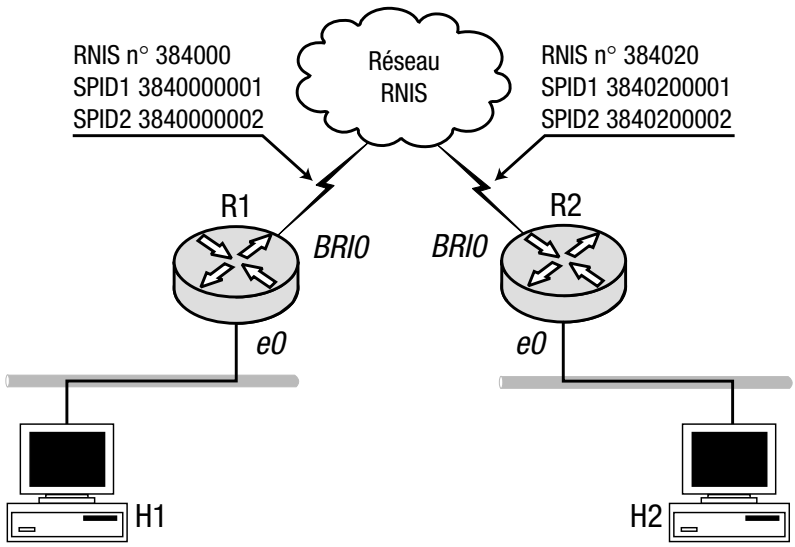
La figure 2.11 montre un exemple de deux routeurs Cisco en pontage de trafic IP entre deux segments Ethernet, à travers un réseau RNIS. Le type de commutateur, dans ce cas, est un *National ISDN-1*.

Les listings 2.15 et 2.16 contiennent les commandes de configuration pour les routeurs R1 et R2.

© copyright Éditions Eyrolles

**Figure 2.11**

*Pontage de segments Ethernet avec deux routeurs reliés* via *un réseau RNIS.*




**Listing 2.15. Configuration du routeur R1.**

```
username R2 password cisco
no ip routing
isdn switch-type basic-ni1
dialer-list 1 protocol bridge permit
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface BRI0
 encapsulation ppp
 isdn spid1 3840000001
 isdn spid2 3840000002
 dialer map bridge name R2 broadcast 384020
 dialer-group 1
 ppp authentication chap
 bridge-group 1
```

**Listing 2.16. Configuration du routeur R2.**

```
username R1 password cisco
no ip routing
isdn switch-type basic-ni1
dialer-list 1 protocol bridge permit
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface BRI0
 encapsulation ppp
 isdn spid1 3840200001
 isdn spid2 3840200002
```

© copyright Éditions Eyrolles

```
dialer map bridge name R1 broadcast 384000
dialer-group 1
ppp authentication chap
bridge-group 1
```

La commande **show bridge** sur les routeurs R1 et R2 donne les résultats imprimés sur les listings 2.17 et 2.18, où on voit les lignes pointer sur l'interface BRI du RNIS.

**Listing 2.17. Base de données de filtrage d'adresses du routeur R1.**

```
R1#show bridge

Total of 300 station blocks, 296 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  BRI0         1          5         4
0060.97fb.566a  forward  Ethernet0    0          6         0
0060.b01a.9e1c  forward  Ethernet0    0          5         5
0260.8ca3.28cd  forward  Ethernet0    0          2         0
```

**Listing 2.18. Base de données de filtrage d'adresses du routeur R2.**

**Solution apparentée**
Configuration de IP sur RNIS, p. 47.

```
R2#show bridge

Total of 300 station blocks, 297 free
Codes: P - permanent, S - self

Bridge Group 1:

    Address      Action  Interface   Age  RX count  TX count
0260.8c4c.1132  forward  Ethernet0    0          5         4
0060.b01a.9e1c  forward  BRI0         0          5         5
0260.8ca3.28cd  forward  BRI0         3          1         0
```

## Configuration du routage et du pontage en simultané

Pour configurer un routeur de façon à ce qu'il exécute le routage et le pontage simultanément, il faut recourir au CRB. Cette fonction est activée par la commande **bridge crb** en mode de configuration globale. La commande **no ip routing** n'est plus nécessaire avec le CRB. Le routage IP étant actif, en même temps que le CRB, certaines interfaces pourront fonctionner en IP, tandis que d'autres seront en pontage. Les recommandations pour la configuration du CRB sont les mêmes que pour le pontage transparent, mais sans la commande **no ip routing**.

La figure 2.12 montre un routeur configuré pour le pontage IP entre les segments 1 et 2, et le routage IP entre les segments 3 et 4.

La configuration du routeur R1 est contenue dans le listing 2.19.

**Listing 2.19. Configuration du routeur R1.**

```
bridge crb
bridge 1 protocol ieee

interface Ethernet0
```

© copyright Éditions Eyrolles

```
 bridge-group 1

interface Ethernet1
 bridge-group 1

interface Ethernet2
 ip address 10.0.1.1 255.255.255.0

interface Ethernet3
 ip address 10.0.2.1 255.255.255.0
```

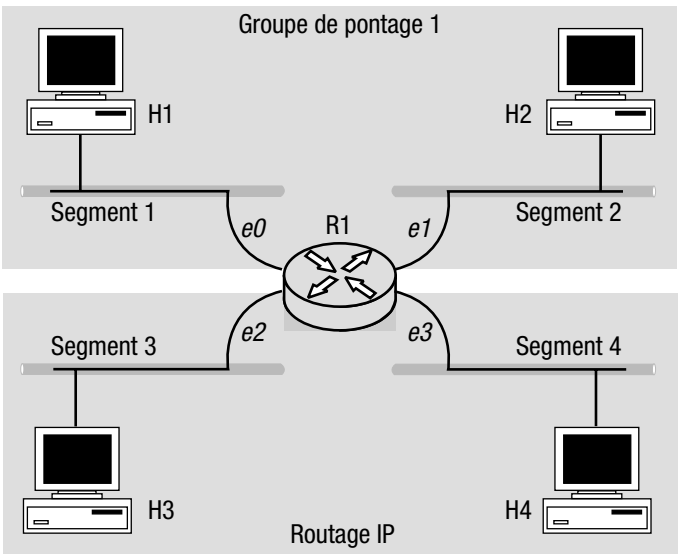



**Figure 2.12**
*Le CRB permet au routeur de fonctionner en routage IP et en pontage simultanément.*

**ASTUCE** Quand vous utilisez la commande **bridge crb**, tous les protocoles, y compris IP, sont en pontage sur les interfaces qui sont définies avec cette fonction. Pour réactiver le routage sur ces interfaces, il faut utiliser la commande **bridge** <*numéro de groupe*> **route ip**. Vous pourrez ainsi basculer toutes les interfaces de ce groupe, du pontage en routage IP.

### Configuration du routage et du pontage intégrés

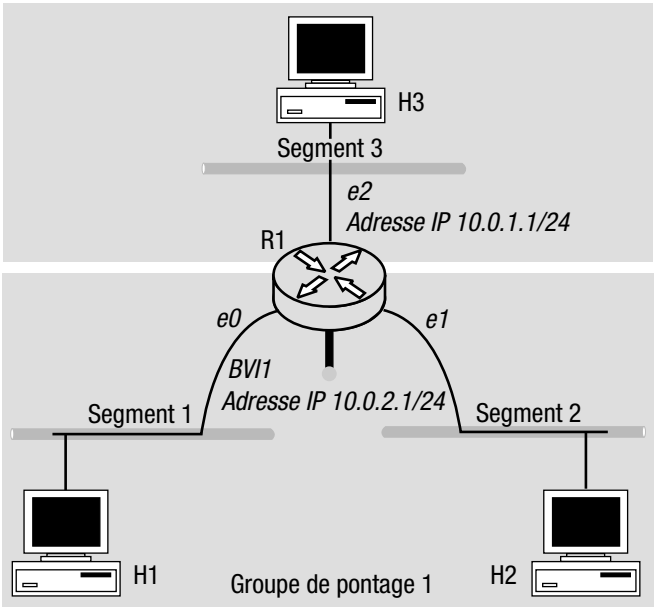
Pour que les interfaces en routage et celles en pontage fonctionnent simultanément, et que le trafic IP puisse passer entre ces deux types d'interface, on introduit un concept qui définit le groupe de pontage tout entier comme une *interface virtuelle de pontage* (*virtual bridge interface*).

La commande **bridge irb** en mode de configuration globale active cette fonction. Une fois celle-ci activée, la commande **interface BVI** <*numéro de groupe*> permet de définir une interface virtuelle de pontage comprenant toutes les interfaces du numéro du groupe de pontage concerné. L'adresse IP de l'interface virtuelle doit provenir du sous-réseau auquel appartiennent les hôtes connectés aux segments desservis par les interfaces, membres du groupe concerné. Les hôtes pourront utiliser cette adresse IP comme celle de la passerelle par défaut, pour accéder au reste du réseau.

La figure 2.13 illustre le cas du routeur R1 configuré en IRB.

© copyright Éditions Eyrolles

**Figure 2.13**
*Routeur R1 configuré en IRB avec le routage de trafic IP entre les interfaces du groupe de pontage (e0 et e1) et l'interface de routage IP (e2).*




**Listing 2.20. Configuration du routeur R1.**

```
interface Ethernet0
 bridge-group 1

interface Ethernet1
 bridge-group 1

interface Ethernet2
 ip address 10.0.1.1 255.255.255.0

interface BVI1
 ip address 10.0.2.1 255.255.255.0

bridge irb
bridge 1 protocol ieee
 bridge 1 route ip
```

**REMARQUE** La commande **bridge** <*numéro de groupe*> **route ip** est celle qui permet de passer le trafic IP entre l'interface virtuelle de pontage définie par la commande **interface BVI** et l'interface de routage IP.

### Réglage des paramètres de l'arbre de recouvrement

Les ponts déroulent l'algorithme d'arbre de recouvrement en utilisant un protocole de pontage sans nom. Celui-ci peut être conforme soit à la norme IEEE 802.1D, soit à celle de DEC.

Ces deux versions utilisent des paquets spéciaux appelés BPDU (*Bridge Protocol Data Unit*) pour communiquer les informations de topologie qui permettent de rétablir l'arbre de recouvrement. Celles-ci sont envoyées à une adresse MAC multicast, uniquement réservée aux ponts. Sa valeur, en hexadécimal, selon la notation de Cisco, est 0180.C200.0000.

© copyright Éditions Eyrolles

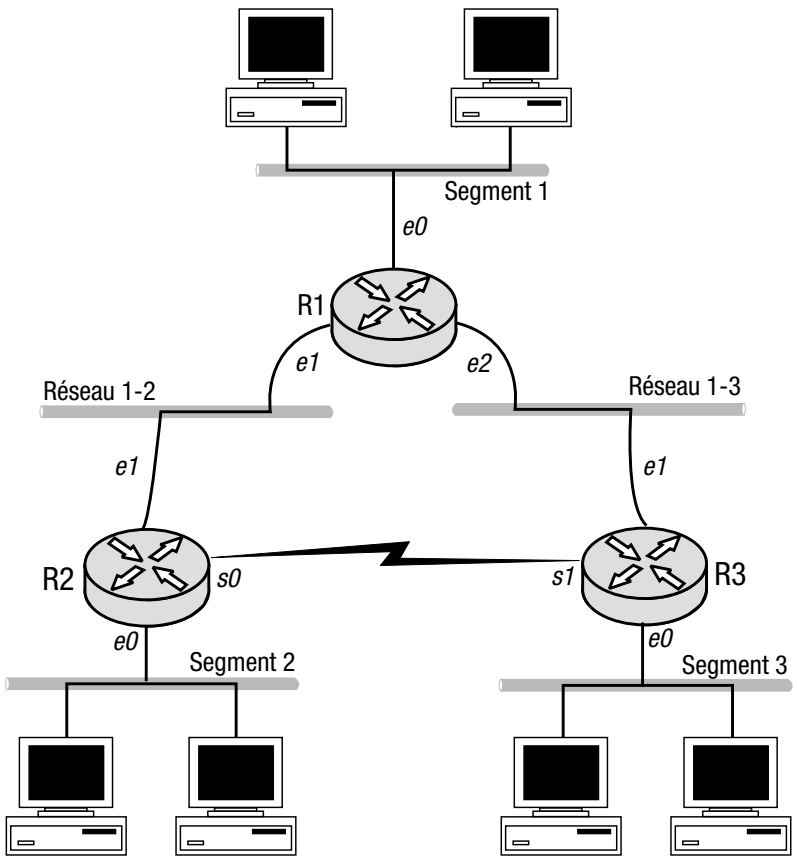
Les ponts eux-mêmes ont certains paramètres qui influent sur l'algorithme d'arbre de recouvrement, quand il recalcule la topologie. Parmi ces paramètres, les plus importants sont l'*identité du pont* (*bridge ID*) et le *coût du chemin* (*path cost*) associé à ses différents ports. Le pont possédant l'identité la plus petite devient le pont racine. Le coût du chemin détermine l'état d'un port, bloqué ou en acheminement.

L'identité du pont se compose de deux parties, son adresse MAC et sa priorité. Cette dernière configurée par l'administrateur est plus significative que la première. La comparaison d'identité des ponts se fait d'abord par leur priorité (le pont dont la priorité est la plus petite se voit affecter l'identité la plus petite). Si les priorités sont égales, ce qui arrive quand elles sont laissées à leur valeur par défaut, la comparaison se fait entre les adresses MAC des ponts. Celui ayant la plus petite adresse possède alors l'identité la plus petite.

Prenons le cas de la figure 2.14 comme exemple pour étudier comment l'algorithme d'arbre de recouvrement recalcule sa topologie avec les paramètres par défaut, et comment on peut l'améliorer en réglant ces derniers.

**Figure 2.14**

*Topologie avec trois Cisco routeurs configurés pour le pontage IP.*




Les configurations des trois routeurs figurent aux listings 2.21 à 2.23.

© copyright Éditions Eyrolles

**Listing 2.21. Configuration du routeur R1.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Ethernet1
 bridge-group 1

interface Ethernet2
 bridge-group 1
```

**Listing 2.22. Configuration du routeur R2.**

```
Router R2

no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial0
 bridge-group 1

interface Ethernet1
 bridge-group 1
```

**Listing 2.23. Configuration du routeur R3.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface Serial1
 bridge-group 1

interface Ethernet1
 bridge-group 1
```

Jusqu'à présent, nous n'avons pas encore évoqué la topologie qui résulte du déroulement de l'algorithme d'arbre de recouvrement. Car on n'avait considéré que des plans de réseau comportant au plus deux ponts reliés par une seule ligne, ce qui implique la mise en état acheminement de tous leurs ports. Nous abordons maintenant le cas de trois ponts (en fait, des routeurs configurés en ponts), chacun connecté aux deux autres. Un port d'au moins l'un de ces ponts doit être mis en état bloqué.

L'examen de la topologie d'arbre de recouvrement telle qu'elle est visible d'un pont donné, se fait par la commande **show spanning-tree** <*numéro de groupe*>. Vous pouvez utiliser la même commande sans renseigner le paramètre pour avoir les topologies de tous les groupes qui ont été définis.

Les résultats de cette commande sur les 3 routeurs, avec les sorties correspondantes, se trouvent sur les listings 2.24 à 2.26. Vu le grand nombre de lignes imprimées, les éléments principaux ont été mis en italique.

© copyright Éditions Eyrolles

**Listing 2.24. Sortie de la commande `show spanning-tree` sur le routeur R1.**

```
R1#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
protocol
 Bridge Identifier has priority 32768, address 0010.1111.1111
 Configured hello time 2, max age 20, forward delay 15
 Current root has priority 32768, address 0000.0000.1000
 Root port is 7 (Ethernet1), cost of root path is 100
 Topology change flag not set, detected flag not set
 Times:  hold 1, topology change 30, notification 30
         hello 2, max age 20, forward delay 15, aging 300
 Timers: hello 0, topology change 0, notification 0

Port 6 (Ethernet0) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0010.1111.1111
 Designated port is 6, path cost 100
 Timers: message age 0, forward delay 0, hold 0

Port 7 (Ethernet1) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.1000
 Designated port is 2, path cost 0
 Timers: message age 0, forward delay 0, hold 0

Port 8 (Ethernet2) of bridge group 1 is blocking
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.2000
 Designated port is 2, path cost 100
 Timers: message age 2, forward delay 0, hold 0
```

**Listing 2.25. Sortie de la commande `show spanning-tree` sur le routeur R2.**

```
R2#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
protocol
 Bridge Identifier has priority 32768, address 0000.0000.1000
 Configured hello time 2, max age 20, forward delay 15
 We are the root of the spanning tree
 Topology change flag not set, detected flag not set
 Times:  hold 1, topology change 30, notification 30
         hello 2, max age 20, forward delay 15, aging 300
 Timers: hello 1, topology change 0, notification 0

Port 2 (Ethernet0) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
```

© copyright Éditions Eyrolles

```
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.1000
 Designated port is 2, path cost 0
 Timers: message age 0, forward delay 0, hold 0

Port 4 (Serial0) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.1000
 Designated port is 4, path cost 0
 Timers: message age 0, forward delay 0, hold 0

Port 3 (Ethernet1) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.1000
 Designated port is 3, path cost 0
 Timers: message age 0, forward delay 0, hold 0
```

**Listing 2.26. Sortie de la commande `show spanning-tree` sur le routeur R3.**

```
R3#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
protocol
 Bridge Identifier has priority 32768, address 0000.0000.2000
 Configured hello time 2, max age 20, forward delay 15
 Current root has priority 32768, address 0000.0000.1000
 Root port is 5 (Serial1), cost of root path is 100
 Topology change flag not set, detected flag not set
 Times:  hold 1, topology change 30, notification 30
         hello 2, max age 20, forward delay 15, aging 300
 Timers: hello 0, topology change 0, notification 0

Port 2 (Ethernet0) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.2000
 Designated port is 2, path cost 100
 Timers: message age 0, forward delay 0, hold 0

Port 5 (Serial1) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.1000
 Designated port is 4, path cost 0
 Timers: message age 1, forward delay 0, hold 0

Port 3 (Ethernet1) of bridge group 1 is forwarding
 Port path cost 100, Port priority 128
 Designated root has priority 32768, address 0000.0000.1000
 Designated bridge has priority 32768, address 0000.0000.2000
 Designated port is 3, path cost 100
 Timers: message age 0, forward delay 0, hold 0
```

© copyright Éditions Eyrolles

Supposons que le routeur R1 soit le plus performant des trois, et qu'on veuille faire transiter par lui le maximum de trafic, c'est-à-dire qu'il devienne, en somme, le routeur racine. Rappelons qu'un tel routeur est le pont désigné pour tous les segments auxquels il est connecté, et est de ce fait en mesure d'écouler le maximum de trafic.

Or, on constate à la ligne 4 du listing pour le routeur R2 que c'est ce dernier qui est racine. Pourquoi ? Parce que les valeurs par défaut des priorités dans ces routeurs (configurés en ponts) n'ont pas été modifiées. Comme ces priorités sont restées égales, pour choisir le pont racine ayant l'identité la plus petite, c'est l'adresse MAC qui a servi de facteur discriminant. Parmi les trois ponts R1, R2 et R3 dont les adresses sont respectivement 0010.1111.1111, 0000.0000.1000 et 0000.0000.2000, R2 qui possède la plus petite adresse devient pont racine.

En outre, on ne peut garder les interfaces série des routeurs R1 et R2 en état acheminement, tandis que l'interface Ethernet 2 du routeur R1 est en état bloqué. Il faut noter que le coût du chemin est de 100, pour toutes les interfaces, qu'elles soient série ou Ethernet. Ainsi, parce que l'interface série du routeur R3 donne le chemin le plus court, en termes de coût, vers le pont racine (routeur R2), elle est mise en état d'acheminement.

Le moyen le plus facile de changer cette topologie consiste à attribuer une valeur plus faible à la priorité du routeur R1, qui est à sa valeur maximale par défaut, c'est-à-dire, 32768. En la réduisant, par exemple à 1000, ce routeur devient le pont racine.

Pour changer la priorité du pont, on utilisera la commande **bridge** <*numéro de groupe*> **priority** <*priorité*> qui affecte le groupe renseigné en paramètre.

Une fois ce changement effectué, sur les listings 2.27 à 2.29 on voit le résultat de la topologie recalculée. Seules les lignes qui présentent un intérêt ici y sont reproduites.

**Listing 2.27. Sortie de la commande `show spanning-tree` sur le routeur R1, après changement de priorité.**

```
R1#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
protocol
 Bridge Identifier has priority 1000, address 0010.1111.1111
 ...
 We are the root of the spanning tree
 ...

Port 6 (Ethernet0) of bridge group 1 is forwarding
 ...

Port 7 (Ethernet1) of bridge group 1 is forwarding
 ...

Port 8 (Ethernet2) of bridge group 1 is forwarding
 ...
```

**Listing 2.28. Sortie de la commande `show spanning-tree` sur le routeur R2, après changement de priorité.**

```
R2#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
```

© copyright Éditions Eyrolles

```
protocol
 Bridge Identifier has priority 32768, address 0000.0000.1000
 ...
 Current root has priority 1000, address 0010.1111.1111
 Root port is 2 (Ethernet0), cost of root path is 100
 ...

Port 2 (Ethernet0) of bridge group 1 is forwarding
 ...

Port 4 (Serial0) of bridge group 1 is forwarding
 ...

Port 3 (Ethernet1) of bridge group 1 is forwarding
 ...
```

**Listing 2.29. Sortie de la commande `show spanning-tree` sur le routeur R3, après changement de priorité.**

```
R3#show spanning-tree 1

Bridge Group 1 is executing the IEEE compatible Spanning Tree
protocol
 Bridge Identifier has priority 32768, address 0000.0000.2000
 ...
 Current root has priority 1000, address 0010.1111.1111
 Root port is 2 (Ethernet0), cost of root path is 100
 ...

Port 2 (Ethernet0) of bridge group 1 is forwarding
 ...

Port 5 (Serial1) of bridge group 1 is blocking
 ...

Port 3 (Ethernet1) of bridge group 1 is forwarding
 ...
```

La topologie d'arbre de recouvrement est à présent satisfaisante. R1 est le pont racine, et seuls les ports Ethernet sont en état acheminement. Néanmoins, on pourrait avoir besoin de renseigner la variable de coût du chemin en lui donnant la bonne valeur pour tous les ports impliqués dans l'algorithme d'arbre de recouvrement. En principe, la valeur du coût du chemin dépend du débit de la connexion. Si, par exemple, une ligne série a un débit qui est le tiers de celui d'Ethernet, on peut donner donner une valeur de 300 au coût du chemin de son interface. La commande utilisée à cet effet est **bridge-group** <*numéro de groupe*> **path-cost** <*coût du chemin*>.

**ASTUCE** Il est possible de changer l'adresse MAC inscrite sur la carte d'interface réseau d'un routeur Cisco. Bien que déconseillée, la commande pour effectuer ce changement est **mac-address** <*nouvelle adresse MAC*>, en mode de configuration interface.

**REMARQUE** Les adresses MAC qui apparaissent sur les listings ont toutes été assignées par la commande **mac-address**, pour aider à comprendre comment l'algorithme d'arbre de recouvrement recalcule sa topologie. De plus, la valeur par défaut du coût du chemin a été changée pour l'interface série, avec la commande **bridge-group** 1 **path-cost** 100 pour la forcer à l'état acheminement, afin d'illustrer nos propos. Bien que la valeur par défaut d'une interface série soit plus grande que celle d'une interface Ethernet, elle peut ne pas refléter son débit réel, et par conséquent nécessiter un réajustement.

© copyright Éditions Eyrolles

## Configuration du pontage à routage par la source (SRB)

Le *pontage à routage par la source* ou SRB (*Source Route Bridging*) et le pontage transparent servent le même but, l'acheminement des PDU d'un segment à un autre. Mais le premier est basé sur des principes complètement différents que nous allons évoquer dans cette section.

### Utilisation du pontage « classique » à routage par la source

Le pont classique SRB ne peut avoir plus de deux ports connectés en même temps, condition si restrictive en pratique, que les concepteurs de Cisco ont dû la contourner en introduisant le concept d'*anneau virtuel*. Du point de vue du pontage SRB, ce dernier apparaît comme un anneau normal dont le numéro est référencé dans le champ désignation de route des trames acheminées. Cependant, il n'a qu'une définition logique, d'où son nom. Le routeur lui-même émule en fait plusieurs ponts SRB, chacun ne possédant que deux ports connectés à l'anneau virtuel. Cette particularité se trouve répercutée dans les trames SRB, dans lesquelles le routeur insère deux désignations de route lors de leur passage.

Pour configurer un routeur Cisco en pontage de trafic IP à routage par la source, on doit suivre les étapes suivantes :

1. Désactiver le routage IP par **no ip routing** en mode de configuration globale.
2. Définir un anneau virtuel par la commande **source-bridge ring-group** *<numéro d'anneau virtuel>* en mode de configuration globale.
3. Assigner à chaque interface Token Ring un numéro d'anneau physique et un numéro de pont logique qui connecte le premier à l'anneau virtuel, par la commande **source-bridge** *<numéro d'anneau physique> <numéro de pont logique> <numéro d'anneau virtuel>* en mode de configuration globale.
4. Utiliser la commande **multiring ip** pour activer sur l'interface, le pontage à routage par la source des trames MAC contenant les datagrammes IP.

Le listing 2.30 donne un exemple de configuration qui active le SRB du trafic IP dans un routeur avec trois interfaces Token Ring.

**Listing 2.30. Configuration du SRB classique.**

```
no ip routing
source-bridge ring-group 1000

interface TokenRing0
 ring-speed 16
 source-bridge 110 1 1000
 multiring ip

interface TokenRing1
 ring-speed 16
 source-bridge 120 2 1000
 multiring ip

interface TokenRing2
 ring-speed 16
 source-bridge 130 3 1000
 multiring ip
```

© copyright Éditions Eyrolles

### Configuration du pontage distant à routage par la source

Comme le pontage transparent, le pontage à routage par la source peut aussi être utilisé sur des réseaux non LAN et non Token Ring. Cette variante est appelée pontage distant à routage par la source ou RSRB (*Remote Source Route Bridging*).

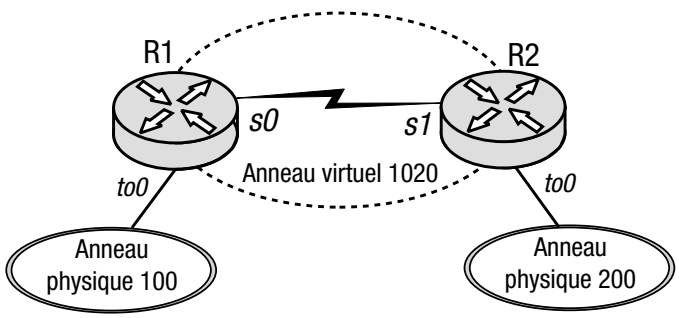
Contrairement au cas du pontage transparent, dans la configuration du RSRB on ne peut rattacher le service de pontage directement à l'interface non Token Ring, car celle-ci, en général, ne possède pas la fonctionnalité de pontage à routage par la source. On a donc recours à une extension de l'anneau virtuel aux interfaces non Token Ring, de façon à ce que tous les routeurs qui y participent apparaissent comme faisant partie de ce même anneau. Il s'agit d'une émulation de Token Ring.

Cette extension se fait en rattachant à l'anneau virtuel tous les partenaires distants, par la commande **source-bridge remote-peer** avec un certain nombre de paramètres, dont le type de transport (TCP, Frame Relay ou autres) et l'identification du partenaire distant qui, dans le cas de TCP, est l'adresse IP.

Prenons l'exemple de la figure 2.15 où l'on voit deux routeurs configurés en RSRB qui acheminent le trafic IP entre deux anneaux Token Ring reliés par une connexion distante en ligne série.

**Figure 2.15**

*Connexion en ligne série de deux anneaux Token Ring par deux routeurs configurés en RSRB.*




Les listings 2.31 et 2.32 montrent la configuration de ces routeurs.

**Listing 2.31. Configuration du routeur R1.**

```
no ip routing
source-bridge ring-group 1020
source-bridge remote-peer 1020 tcp 10.1.1.2 local-ack
source-bridge remote-peer 1020 tcp 10.1.1.1

interface Serial0
 ip address 10.1.1.1 255.255.255.0

interface TokenRing0
 ring-speed 16
 multiring ip
 source-bridge 100 11 1020
```

**Listing 2.32. Configuration du routeur R2.**

```
no ip routing
source-bridge ring-group 1020
```

© copyright Éditions Eyrolles

```
source-bridge remote-peer 1020 tcp 10.1.1.1 local-ack
source-bridge remote-peer 1020 tcp 10.1.1.2

interface Serial1
 ip address 10.1.1.2 255.255.255.0

interface TokenRing0
 ring-speed 16
 multiring ip
 source-bridge 200 12 1020
```

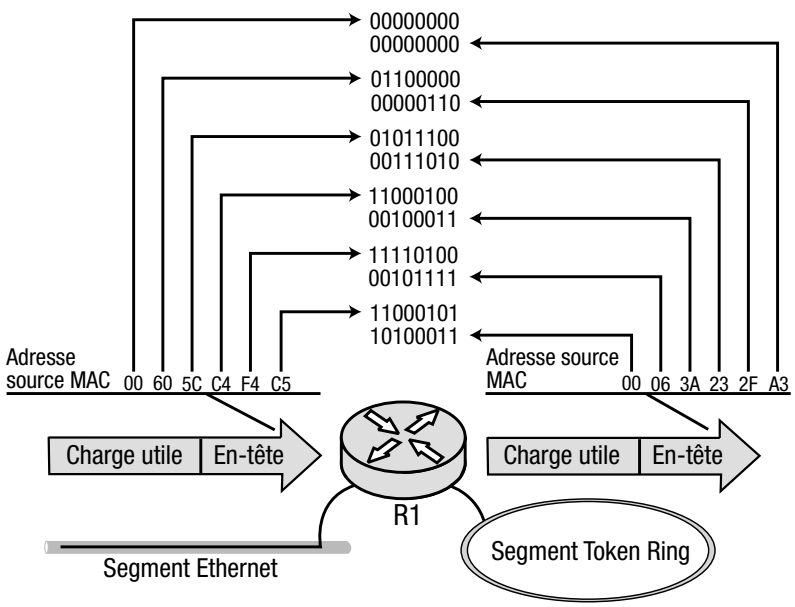
### Configuration du pontage traducteur à routage par la source et du pontage transparent

Le pontage transparent n'est pas limité au seul environnement Ethernet, il peut fonctionner aussi dans un cadre exclusivement Token Ring ou mixte (Ethernet/Token Ring). On peut comprendre aisément le fonctionnement du pontage transparent dans le cadre du Token Ring ; l'environnement mixte, lui, soulève quelques questions.

Le pontage à routage par la source n'a été conçu que pour les environnements Token Ring. Certains ponts perfectionnés peuvent essayer de « traduire » les informations du SRB en celles du pontage transparent et *vice versa*, permettant ainsi de fusionner les LAN de ces deux technologies. Cette méthode est appelée *pontage traducteur à routage par la source* (*source route translational bridging*). Comme nous l'avons remarqué plus haut, les situations qui justifient la mise en œuvre d'un tel environnement mixte, sont non seulement rares, mais elles induisent aussi des problèmes complexes.

Tout d'abord, nous avons appris dans la section précédente que les adresses MAC d'Ethernet et de Token Ring ont un ordre différent des bits dans chacun des octets, ce qui nécessite une traduction si un pont doit relier des LAN comportant ces deux technologies. Un exemple de cette traduction se trouve à la figure 2.16.

**Figure 2.16**
*Traduction de l'ordre des bits dans les octets d'adresse MAC par un routeur en pontage entre un LAN Ethernet et un autre en Token Ring.*




© copyright Éditions Eyrolles

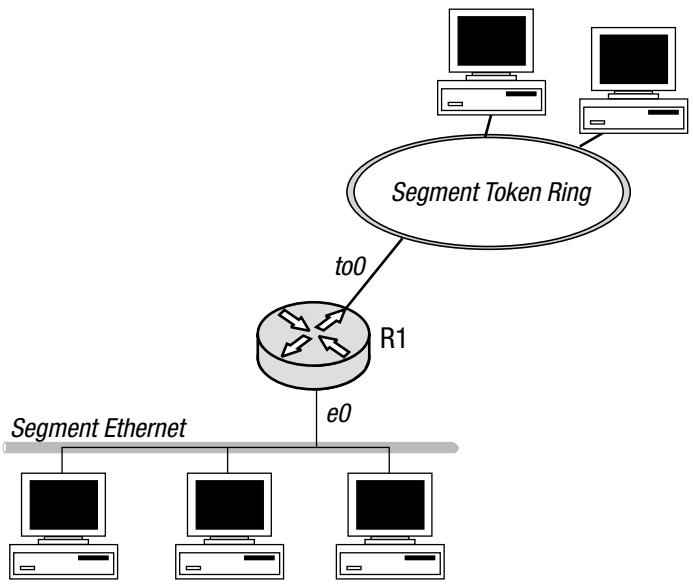
La traduction en elle-même ne serait pas un problème majeur, si le protocole ARP ne sauvegardait pas une copie de l'adresse MAC source dans sa propre PDU. Au niveau de la couche liaison, la PDU de l'ARP n'est rien d'autre qu'une charge utile à encapsuler dans une trame MAC. Le pont, en tant qu'appareil fonctionnant à la couche liaison, n'a donc aucun moyen d'examiner la PDU et de traduire ainsi la copie de l'adresse MAC qui s'y trouve. Le module ARP destinataire va utiliser cette adresse MAC source non traduite dans sa réponse, l'envoyant ainsi dans la nature. Le seul moyen d'éviter ce problème est de configurer toutes les entrées ARP manuellement, ce qui serait très fastidieux.

Un autre problème provient de la différence de MTU entre Ethernet et Token Ring. Le premier a une MTU de 1500 octets, tandis que celle du second peut compter jusqu'à 4096 octets. Si un routeur est utilisé pour acheminer le trafic IP entre un segment Ethernet et un anneau Token Ring, il peut fragmenter les datagrammes en provenance du Token Ring de façon à ce qu'ils puissent tenir dans une MTU Ethernet, mais les ponts n'ayant pas cette capacité, vont tout simplement mettre au rebut sans aviser, tout datagramme trop grand pour être envoyé dans la MTU du réseau destinataire dont la taille n'est pas suffisante. Il est à noter qu'une communication peut toujours commencer entre deux stations reliées par un pont, dont l'une se trouve sur Ethernet et l'autre sur Token Ring car au début, les protocoles n'envoient pas leur PDU à la taille maximale. C'est seulement en cours de session que la taille des PDU augmente progressivement pour atteindre la MTU du réseau physique sous-jacent. Dès que la taille des PDU envoyées par la station sur le Token Ring dépasse la MTU du réseau Ethernet, le pont intermédiaire va les mettre au rebut, provoquant ainsi une coupure de session. Il est donc nécessaire de réduire la MTU sur tous les nœuds du Token Ring susceptibles également de communiquer avec des nœuds sur un segment Ethernet.

Parce que ces deux types de pontage de trafic IP en environnement mixte sont très rares, on ne montre qu'un exemple de configuration pour chacun, avec un seul routeur reliant un segment Ethernet et un anneau Token Ring. Le schéma auquel pourraient s'appliquer ces types de pontage se trouve illustré à la figure 2.17.

**Figure 2.17**

*Routeur en pontage de trafic IP entre un segment Ethernet et un anneau Token Ring.*




© copyright Éditions Eyrolles

Le listing 2.33 montre une configuration de routeur reliant un segment Ethernet et un anneau Token Ring, en pontage transparent de trafic IP.

Le listing 2.34 montre une configuration de routeur reliant un segment Ethernet et un anneau Token Ring, en pontage traducteur.

**Listing 2.33. Configuration de pontage transparent dans un environnement de supports physiques mixtes.**

```
no ip routing
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface TokenRing0
 ring-speed 16
 bridge-group 1
```

**Listing 2.34. Configuration de pontage traducteur dans un environnement de supports physiques mixtes.**

```
no ip routing
source-bridge ring-group 1000
source-bridge transparent 1000 200 15 1
bridge 1 protocol ieee

interface Ethernet0
 bridge-group 1

interface TokenRing0
 ring-speed 16
 source-bridge 100 1 1000
 multiring ip
```

© copyright Éditions Eyrolles

# 3

# Routage statique

### Solutions de configuration présentées dans ce chapitre



Dans les chapitres précédents, nous avons étudié les principes des protocoles à couches et les détails du modèle Internet, tels que l'adressage IP, l'acheminement de datagrammes, la fragmentation, etc. Nous y avons appris que les routeurs IP sont les appareils qui servent à acheminer les datagrammes à leur destination, à travers des réseaux divers. Il est temps de savoir **comment** ces appareils prennent leur décision dans la fonction de routage.

Les routeurs acheminent les datagrammes entre les segments qui leur sont directement rattachés, et leurs décisions de routage sont basées sur l'adresse réseau et non sur celle de l'hôte. D'une certaine manière, les routeurs doivent déterminer quelles adresses correspondent à quelles interfaces. Bien évidemment, il serait commode d'avoir tous les segments connectés à un seul routeur, sachant que ce dernier sait exactement quelles sont les adresses réseau de ses interfaces. Mais si plusieurs routeurs sont utilisés pour assurer le trafic entre différents

© copyright Éditions Eyrolles

segments d'un réseau, et si le trafic doit passer par plus d'un saut, les décisions de routage deviennent difficiles.

Pour faire face à ces difficultés, les routeurs doivent :

- mémoriser les adresses réseau disponibles sur les segments qui ne leur sont pas directement rattachés ;
- appliquer certaines règles pour résoudre des cas d'ambiguïté éventuels pouvant survenir quand la même adresse IP correspond à plusieurs adresses réseau ; rappelons que l'adresse réseau peut être celle d'un réseau direct, d'un sous-réseau, d'un hôte ou même celle d'un super-réseau dans le modèle d'adressage sans classe.

Concernant le premier point, les routeurs utilisent une base de données appelée *table de routage*, qu'ils sont censés tenir à jour. Cette table contient des entrées ou *routes* dont chacune comprend une seule adresse réseau avec un masque de sous-réseau (ou, dans le vocabulaire récent, un « préfixe réseau » et une « longueur de préfixe réseau »), ainsi qu'une liste de références pour atteindre cette adresse réseau. Cette liste spécifie généralement l'adresse IP du routeur de saut suivant qui devrait être accessible directement par l'une des interfaces locales. Cependant, si l'adresse réseau est celle d'un segment directement rattaché, le champ concernant le routeur de saut suivant est laissé vide. La table de routage peut aussi contenir des informations auxiliaires telles que des métriques et des signalisations de route.

Les routeurs utilisent deux procédés pour remplir la table de routage, la configuration de routes statiques et l'acquisition de routes dynamiques. Dans le premier cas, c'est l'administrateur réseau qui doit manuellement configurer les routes dans la table ; dans le deuxième cas, des protocoles auxiliaires appelés *protocoles de routage dynamique* assurent la découverte automatique des routes en échangeant des paquets spéciaux qui servent aux mises à jour de la table.

Ce chapitre est consacré au routage statique et à la façon de le configurer sur les routeurs Cisco. Le routage dynamique sera traité exhaustivement dans les deux chapitres suivants.

Avant de voir les tâches de configuration statique, examinons d'abord comment les routeurs choisissent les routes adéquates à partir de leurs tables. Ce rôle est dévolu à l'algorithme de routage implanté sur le routeur.

## Algorithme de routage

La première version de l'algorithme de routage concernait les réseaux à classe et fut constamment améliorée pour être enfin consignée dans la norme d'adressage réseau sans classe. Les détails de cette norme se trouvent décrits dans la RFC 1812 traitant des spécifications des routeurs pour IPv4. Notre exposé sur l'algorithme de routage sera basé sur la version sans classe et relèvera les différences avec la version originale.

L'algorithme de routage utilise deux arguments, l'adresse IP de destination et la table de routage. Le résultat est une route unique que le routeur utilise pour acheminer les datagrammes.

La version simplifiée de l'algorithme de routage sans classe comporte deux étapes :

1. **La correspondance élémentaire (*basic match*)**, qui consiste à ne conserver que les préfixes réseau qui correspondent à l'adresse de destination et à écarter tous les autres ; par exemple, si l'adresse de destination est 10.234.24.194, le résultat de l'algorithme peut donner les préfixes réseau suivants : 8.0.0.0/5, 10.0.0.0/8 et 10.234.0.0/16 ; les autres, 9.0.0.0/8, 10.200.0.0/16 et 146.123.45.0/24, s'ils existent dans la table, sont écartés.

© copyright Éditions Eyrolles

2. **La correspondance la plus longue (*longest match*)** appliquée au résultat précédent permet de choisir le préfixe réseau le plus long comme résultat final ; si la correspondance élémentaire n'avait produit aucun résultat, l'algorithme aboutirait à la mise au rebut du datagramme ; dans le cas contraire, comme dans notre exemple, la correspondance la plus longue choisirait le préfixe 10.234.0.0/16.

Les deux étapes précédentes peuvent être combinées en une seule fonction, c'est-à-dire la recherche directe de la correspondance la plus longue. Mais le document officiel spécifie les deux étapes qui nous permettront de relever plus facilement les différences entre la version à classe et celle sans classe :

1. La correspondance élémentaire, dans la version à classe, extrait en premier l'adresse réseau à classe de l'adresse IP de destination ; si le routeur est directement connecté à un ou plusieurs des sous-réseaux relevant de cette adresse réseau, l'algorithme va choisir en premier les préfixes correspondants, y compris celui de l'adresse réseau (égal à l'adresse elle-même) pouvant figurer dans la table de routage (parmi ces préfixes, seuls seront retenus ceux qui correspondent à l'adresse de destination) ; si le routeur n'est connecté à aucun des sous-réseaux, l'algorithme va conserver tous les préfixes qui correspondent à l'adresse IP de destination.

**REMARQUE** Autrement dit, si le routeur est connecté à un ou plusieurs sous-réseaux de l'adresse réseau à classe à laquelle appartient l'adresse IP de destination, il n'acheminera jamais ce datagramme en utilisant les routes de super-réseau de l'adresse de destination.

2. La correspondance la plus longue pour la version à classe est la même que pour l'algorithme de routage sans classe.

L'étape de la correspondance élémentaire peut paraître un peu complexe et nous allons la clarifier par un exemple pratique. Supposons qu'un routeur contienne les routes pour les préfixes : 10.1.1.0/24, 10.1.2.0/24, 10.2.1.0/24, 10.2.2.0/24, 172.16.0.0/16 et 0.0.0.0/0. Supposons aussi que ce routeur soit directement connecté aux sous-réseaux correspondant aux deux premiers préfixes. S'il reçoit un datagramme dont l'adresse IP de destination est 10.3.1.10, l'algorithme de routage à classe extrait en premier l'adresse réseau qui est 10.0.0.0/8 ; ensuite, il ne sélectionne que les préfixes qui sont des sous-réseaux de 10.0.0.0/8 dans sa table : 10.1.1.0/24, 10.1.2.0/24, 10.2.1.0/24 et 10.2.2.0/24. Aucun de ces préfixes ne correspond à l'adresse de destination, ce qui oblige le routeur à mettre au rebut le datagramme. Notons que l'algorithme de routage, dans sa version à classe, n'a pas pris pas en compte la route par défaut, 0.0.0.0/0, contrairement à celui de la version sans classe.

## *Partage de charge*

Encore une fois, les routes consignées dans les tables de routage consistent en un préfixe réseau associé à une liste de références pour atteindre la destination correspondant à ce préfixe. Dans le cas où plusieurs routes existent pour une même destination, le routeur doit choisir l'une d'entre elles voire les utiliser concurremment.

Lorsqu'un routeur utilise plus d'une route vers une même destination, il effectue un *partage de charge* ou *équilibrage de charge* (*load splitting* ou *load balancing*). Dans un cas, le paquet emprunte des routes à coût égal, où le flux des paquets est partagé de manière égale entre des

© copyright Éditions Eyrolles

routes de même débit ; dans l'autre, le trafic s'écoule à travers des routes à coût inégal proportionnellement à leur débit.

**REMARQUE** On peut aussi considérer qu'une table de routage qui contient plusieurs routes vers la même destination peut les utiliser pour le partage de charge. Vu ainsi, l'algorithme de routage génère en sortie plusieurs routes au lieu d'une seule. Ces deux descriptions sont équivalentes, quoique celle qui figure dans le texte principal semble plus logique.

Les routeurs Cisco permettent en outre d'employer deux stratégies de partage de charge, *par destination* ou *par paquet*. Dans le premier cas, le routeur choisit aléatoirement une route et l'utilisera toujours pour acheminer tous les datagrammes vers cette destination. Dans le second cas, le routeur achemine les datagrammes à tour de rôle (*round robin*) à travers toutes les routes disponibles pour le préfixe réseau correspondant.

La fonction de partage de charge peut être désactivée, si elle s'avère inutile.

# Solutions de configuration

Les solutions qui sont proposées dans les sections suivantes supposent que toutes les interfaces des routeurs ont été préalablement bien configurées et en même temps affectées d'une adresse IP adéquate.

## Utilisation d'interfaces connectées pour le routage de base

Si un seul routeur est utilisé pour relier les différents segments d'un réseau, aucune configuration supplémentaire n'est nécessaire pour le routage inter-segments, car le routeur lui-même ajoute automatiquement dans la table de routage les adresses IP qui ont été configurées sur toutes ses interfaces actives. L'état « actif » se vérifie par la commande **show interfaces** <*interface*> <*numéro d'interface*>. La sortie d'une telle commande se trouve sur le listing 3.1.

**Listing 3.1. Résultat de la commande `show interfaces` ethernet 0 qui donne les états physique et logique de l'interface.**

```
R1#show interfaces ethernet 0
Ethernet0 is up, line protocol is up
```

Les messages « *Ethernet is up* » et « *line protocol is up* » signifient que l'interface est physiquement et logiquement en état actif. En supposant que l'interface ait été assignée l'adresse IP 10.1.1.1, en utilisant la commande **show ip route** comme sur le listing 3.2, nous pouvons vérifier que la table de routage contient bien cette adresse (en italique).

**Listing 3.2. Table de routage du routeur R1.**

```
R1#show ip route
...
10.0.0.0/24 is subnetted, 3 subnets
C       10.2.2.0 is directly connected, TokenRing0
C       10.1.1.0 is directly connected, Ethernet0
C       10.0.255.0 is directly connected, Serial1
```

© copyright Éditions Eyrolles

Dans certains cas, une interface peut être physiquement active (*up*) et logiquement inactive (*down*). Si cela se produit, l'adresse IP correspondante est effacée de la table de routage. Par exemple, si un routeur utilise une interface Ethernet reliée à un concentrateur (*hub*), tant que l'échange des messages *Keep alive* n'est pas interrompu pour une raison quelconque (défaillance du câble, erreur de configuration ou taux d'erreur excessif), cette interface reste logiquement active. Par contre, une mauvaise communication peut entraîner une désactivation logique de l'interface, et la commande **show interfaces** ethernet 0 du routeur R1 donne ce qui suit :

```
R1#show interfaces ethernet 0
Ethernet0 is up, line protocol is down
```

Même si l'interface n'est que logiquement inactive, l'adresse réseau ou sous-réseau à laquelle appartient son adresse IP n'apparaîtra pas dans la table de routage, comme le montre la sortie de la commande **show ip route** du listing 3.3.

**Listing 3.3. Table de routage du routeur R1.**

```
R1#show ip route
...
10.0.0.0/24 is subnetted, 2 subnets
C       10.2.2.0 is directly connected, TokenRing0
C       10.0.255.0 is directly connected, Serial1
```

Les hôtes de segments interconnectés par un même routeur doivent avoir une route qui pointe sur l'adresse IP correspondant à l'interface de ce dernier, qui devient ainsi la *default gateway.*

## Configuration du routage de base

Pour configurer le routage statique sur un routeur, suivre les étapes suivantes :

1. Repérer les réseaux ayant besoin d'être accessibles *via* le routeur.
2. Créer les entrées dans la table de routage pour chaque réseau distant par la commande **ip route** <*adresse réseau distant*> <*masque de sous-réseau*> <*adresse IP du routeur de saut suivant*>.

Quelques embûches sont à éviter lors de la configuration du routage statique. On doit s'assurer tout d'abord qu'il est tenu compte aussi bien du trafic entrant que sortant. Cette précaution est cruciale pour un routeur qui relie des réseaux dont aucun ne lui est directement attaché. La figure 3.1 illustre ce cas.

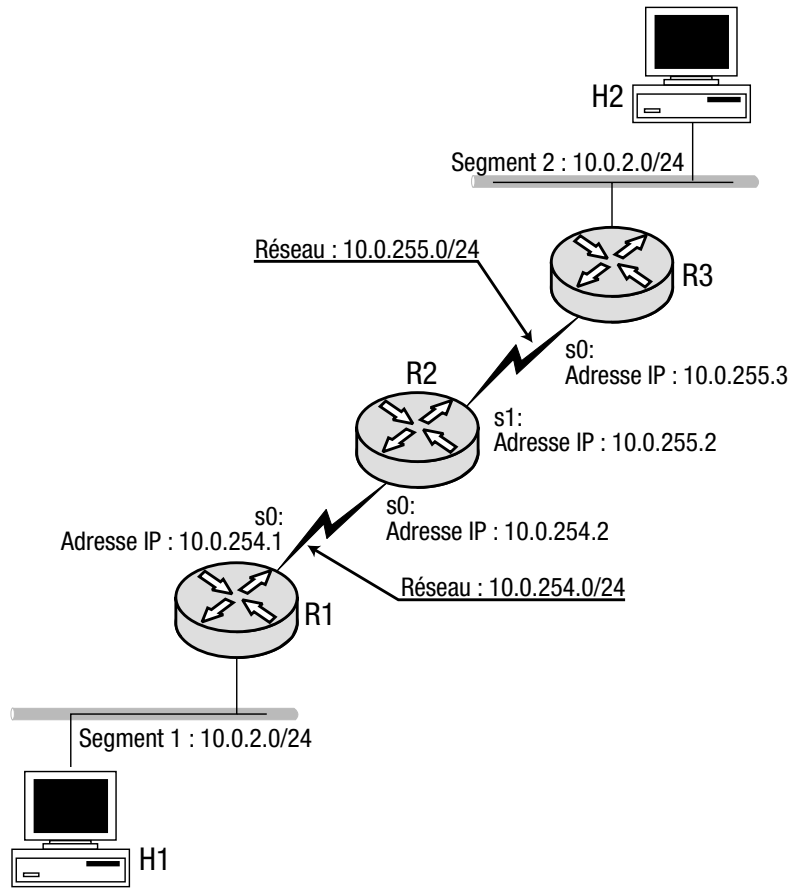
Nous supposerons que les routeurs R1 et R3 sont bien configurés, et que l'hôte H1 du segment 1 doit communiquer avec l'hôte H2 du segment 2. Nous pourrions n'ajouter qu'une seule route statique par la commande qui suit :

```
R2(config)#ip route 10.0.2.0 255.255.255.0 10.0.255.3
```

Cette route est empruntée chaque fois que le routeur R2 reçoit un paquet de l'hôte H1 destiné à l'hôte H2. Il consulte alors sa table de routage qui donne l'adresse du routeur de saut suivant R3, auquel il envoie le paquet. Une fois que ce paquet a atteint sa destination, l'hôte H2 répond avec son propre paquet qui est envoyé au routeur R3 pour l'acheminement à travers le réseau intermédiaire.

© copyright Éditions Eyrolles

**Figure 3.1**
*Routeur utilisé pour acheminer le trafic entre deux réseaux distants.*




Si le routeur R3 est bien configuré, il envoie le paquet de l'hôte H2 au routeur R2. Mais celui-ci ne connaît pas la route vers le segment 1, ce qui l'oblige à mettre le paquet au rebut. Il est facile de corriger cette situation en ajoutant une autre route dans la table de R2 par la commande suivante :

```
R2(config)#ip route 10.0.1.0 255.255.255.0 10.0.254.1
```

Les configurations en routage statique qui ne concernent qu'un routeur sont rares ; généralement, elles en impliquent deux ou plus. La configuration en routage statique d'un seul routeur ne crée une connectivité que dans un sens. Le paquet est livré à destination, mais lorsque le destinataire répond, le routeur à l'autre bout ignore la route vers la source, l'obligeant ainsi à mettre le paquet de réponse au rebut. C'est le cas de la figure 3.2 où le routeur R1 contient l'entrée suivante dans sa table de routage :

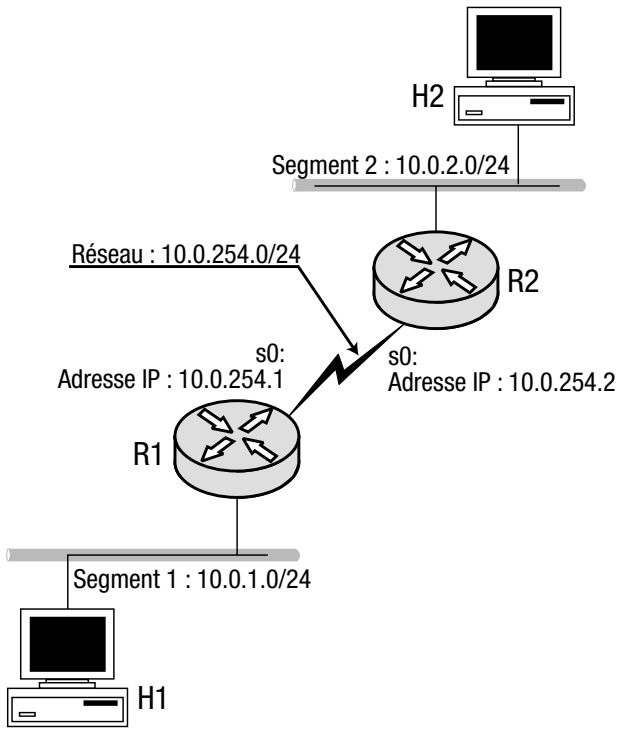
```
ip route 10.0.2.0 255.255.255.0 10.0.254.2
```

Quant au routeur R2, il n'a aucune route statique configurée. La communication initiée par l'hôte H1 vers l'hôte H2 échoue parce que la réponse de ce dernier se trouve mise au rebut par le routeur R2. Pour corriger cet état de fait, il suffit d'ajouter l'entrée suivante dans la table de routage de R2, par la commande :

```
R2(config)#ip route 10.0.1.0 255.255.255.0 10.0.254.1
```

© copyright Éditions Eyrolles

**Figure 3.2**
*Seul le routeur R1 est configuré avec une route statique pour le segment 2.*




Il est à noter que ce problème, bien que simple, peut être difficile à diagnostiquer. Si nous utilisons par exemple la commande **ping** sur le routeur R1 pour vérifier l'accessibilité de l'hôte H2, on y arrive parfaitement. Mais au cas où le **ping** est utilisé sur l'hôte H2 pour vérifier l'accessibilité de l'interface Ethernet du routeur R1, on n'obtient aucune réponse.

À première vue, de tels symptômes pourraient nous laisser penser qu'il s'agit d'un problème de connectivité à sens unique : le trafic circule du routeur R1 vers l'hôte H2, mais pas dans le sens inverse. Comme cela se produit sur le routeur R1, et que l'hôte H2 ne peut accéder au segment Ethernet connecté derrière ce routeur, c'est celui-ci qui semble être mal configuré. Mais si nous étudions le problème de près, nous nous rendrons compte que ce n'est pas le cas. Le **ping** sur le routeur R1 montre que l'hôte H2 est accessible, ce qui signifie que ce routeur reçoit les réponses au **ping**. Nous pouvons déjà en conclure qu'il ne s'agit pas d'un problème de connectivité à sens unique.

Nous savons également que le routeur R1 utilise son interface la plus proche pour envoyer le **ping**, dans ce cas l'interface série 0 directement connectée par une ligne à l'interface série 0 du routeur R2. Quand l'hôte H2 répond, les paquets sont envoyés à l'adresse IP de l'interface série 0 du routeur R1 qui est l'adresse de provenance, et pour laquelle le routeur R2 a une connexion directe. C'est seulement dans le cas où l'hôte H2 cherche à envoyer un **ping** à l'interface Ethernet du routeur R1, que cette commande ne reçoit pas de réponse, parce que le routeur R2 ne connaît aucune route vers cette interface.

La confusion vient du fait qu'on a tendance à associer une adresse IP au routeur lui-même, alors qu'il faudrait plutôt l'associer à une interface particulière de ce dernier. Les hôtes qui en général n'ont qu'une interface ne connaissent pas ce problème.

© copyright Éditions Eyrolles

Enfin, un autre point important à retenir est la configuration des hôtes. Le routage statique des routeurs ne peut fonctionner convenablement sans une configuration adéquate sur les hôtes qui doivent disposer d'une route pointant vers l'adresse IP de l'interface correspondante du routeur. Une configuration incomplète de l'hôte et une mauvaise configuration de routage statique sur le routeur le plus proche vont souvent de pair. Autrement dit, un administrateur qui configure le routage statique n'effectue souvent que la moitié du travail, oubliant que le site concerné doit communiquer avec d'autres.

## Utilisation de métrique avec les routes statiques

La commande **ip route** <*adresse réseau distant*> <*masque de sous-réseau*> <*adresse IP du routeur de saut suivant*> peut prendre un paramètre optionnel appelé [<*distance*>]. Celui-ci sert à choisir la route statique qui doit figurer dans la table de routage, s'il en existe plusieurs pour un même réseau de destination. Plus la valeur de distance est petite, plus la route qui lui est associée prime sur les autres. Le routeur va donc ajouter dans sa table, la route ayant la valeur de distance la plus petite, mettant les autres en réserve. La valeur par défaut de distance est égale à 1, qui est la plus petite possible pour une route statique. Si la distance est 255, la route associée n'a aucune chance de figurer dans la table de routage.

**REMARQUE** Il faut noter que les valeurs de distance n'entrent en ligne de compte que si les longueurs des préfixes réseau sont égales. Si elles sont différentes, les routes correspondantes sont aussi considérées comme différentes. Par exemple, s'il existe deux routes statiques avec les préfixes réseau 10.1.0.0/16 et 10.1.0.0/24, elles seront toutes les deux ajoutées dans la table de routage, indépendamment de leur valeur de distance.

Pourquoi aurait-on besoin de routes qui ne sont pas ajoutées dans la table de routage ? Pour répondre à cette question, on doit faire appel à l'un des principes qui régit l'ajout d'une route dans la table de routage. Le routeur n'ajoute pas une route dans sa table sans avoir connaissance du routeur de saut suivant. En outre, si cette route vient à disparaître, suite à une panne de connexion, elle est effacée de la table. Normalement, le routeur de saut suivant est accessible par l'un des réseaux directement connectés. Par conséquent, si l'interface correspondant à ce réseau est active, le routeur garde une route pour l'adresse réseau à laquelle appartient l'adresse IP assignée à cette interface. Si celle-ci tombe en panne, le routeur efface immédiatement la route correspondante. Toute autre route pointant sur un routeur de saut suivant accessible par cette interface, est aussi effacée. Entre-temps, elle est immédiatement remplacée par la route statique configurée pour la même destination qui remplit le mieux au critère de la valeur de distance. On peut ainsi avoir des routes de rechange en cas de défaillance de la route principale.

Les routes statiques qui ne deviennent effectives qu'après disparition de toutes les autres ayant une valeur de distance inférieure, s'appellent *routes statiques flottantes* (*floating static routes*).

Les routes statiques flottantes sont souvent utilisées pour procurer une ligne de secours à une connexion principale. La figure 3.3 montre deux routeurs R1 et R2 connectés par une paire de lignes série dont les configurations se trouvent sur les listings 3.4 et 3.5.

**Listing 3.4. Configuration du routeur R1.**

```
interface Serial0
 ip address 195.0.0.1 255.255.255.0
```

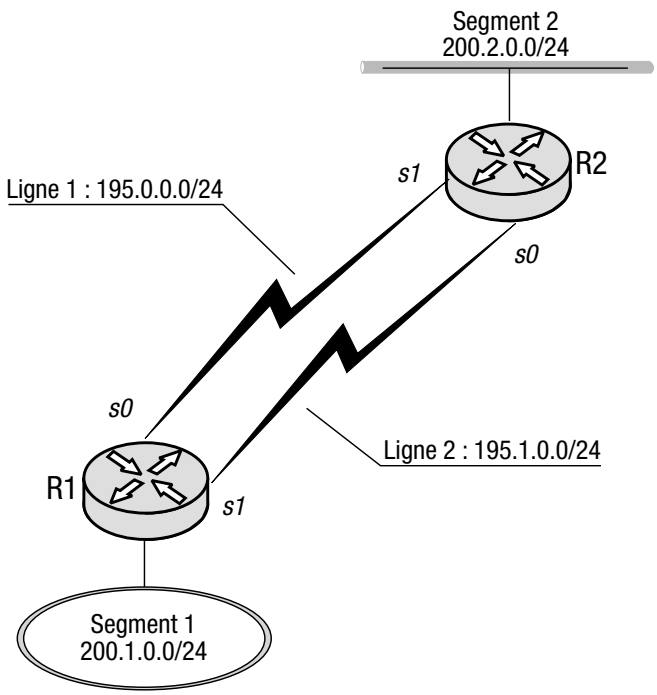
© copyright Éditions Eyrolles

```
interface Serial1
 ip address 195.1.0.1 255.255.255.0

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

ip route 200.2.0.0 255.255.255.0 195.0.0.2
ip route 200.2.0.0 255.255.255.0 195.1.0.2 10
```

**Listing 3.5. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 195.1.0.2 255.255.255.0

interface Serial1
 ip address 195.0.0.2 255.255.255.0

 ip route 200.1.0.0 255.255.255.0 195.0.0.1
p route 200.1.0.0 255.255.255.0 195.1.0.1 10
```

**Figure 3.3**
*Ligne 2 utilisée uniquement en cas de panne de la ligne 1.*




Si nous examinons la table de routage de chacun des routeurs, nous constatons que la route dont la valeur de distance est 10, n'y figure pas. Voir listings 3.6 et 3.7.

© copyright Éditions Eyrolles

**Listing 3.6. Table de routage du routeur R1.**

```
R1#show ip route
...
C    195.1.0.0 is directly connected, Serial1
C    195.0.0.0 is directly connected, Serial0
C    200.1.0.0 is directly connected, TokenRing0
S    200.2.0.0 [1/0] via 195.0.0.2
```

**Listing 3.7. Table de routage du routeur R2.**

```
R2#show ip route
...
C    195.1.0.0/24 is directly connected, Serial0
C    195.0.0.0/24 is directly connected, Serial1
S    200.1.0.0/24 [1/0] via 195.0.0.1
C    200.2.0.0/24 is directly connected, Ethernet0
```

Si nous provoquons une panne de ligne reliant l'interface série 0 du routeur R1 à l'interface série 1 du routeur R2, nous pouvons voir sur le listing 3.8 que la route qui pointait sur l'interface série 0 pointe maintenant sur l'interface série 1 pour le routeur R1, et inversement pour le routeur R2. La ligne de secours a donc été substituée à la ligne principale.

**Listing 3.8. Panne provoquée de la ligne 1 reliant les routeurs R1 et R2 fait disparaître celle-ci de leur table de routage.**

```
R1#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,
changed state to down
%LINK-3-UPDOWN: Interface Serial0, changed state to down
R1#show ip route
...
C    195.1.0.0 is directly connected, Serial1
C    200.1.0.0 is directly connected, TokenRing0
S    200.2.0.0 [10/0] via 195.1.0.2
         R2#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
changed state to down
%LINK-3-UPDOWN: Interface Serial1, changed state to down
R2#show ip route
...
C    195.1.0.0/24 is directly connected, Serial0
S    200.1.0.0/24 [10/0] via 195.1.0.1
C    200.2.0.0/24 is directly connected, Ethernet0
```

Quand la ligne en panne est rétablie, la route qui pointait sur l'interface série de celle-ci, sur les deux routeurs, y pointe de nouveau, comme on peut le voir sur le listing 3.9.

**Listing 3.9. Rétablissement de la ligne 1 avec restauration de la route d'origine qui pointe de nouveau sur l'interface d'avant.**

```
R1#
%LINK-3-UPDOWN: Interface Serial0, changed state to up
```

© copyright Éditions Eyrolles

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,
changed state to up
R1#show ip route
...
C    195.1.0.0 is directly connected, Serial1
C    195.0.0.0 is directly connected, Serial0
C    200.1.0.0 is directly connected, TokenRing0
S    200.2.0.0 [1/0] via 195.0.0.2

R2#
%LINK-3-UPDOWN: Interface Serial1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
changed state to up
R2#show ip route
...
C    195.1.0.0/24 is directly connected, Serial0
C    195.0.0.0/24 is directly connected, Serial1
S    200.1.0.0/24 [1/0] via 195.0.0.1
C    200.2.0.0/24 is directly connected, Ethernet0
```

Il existe une relation entre la valeur de distance et les routes étiquetées « C » (*connected*) dans la table de routage. Quand une adresse IP est assignée à une interface, le routeur ajoute une étiquette de route statique dans sa table de routage. Comme le routeur est directement connecté à cette interface, il attribue la valeur de distance minimale qui est 0, ce qui signifie que c'est la meilleure route vers ce réseau. On ne peut pas la remplacer par une route statique dont la valeur de distance minimale est 1.

## Routage statique avec utilisation d'une interface de sortie au lieu du routeur de saut suivant

Pour configurer une route statique sur une interface, nous devons utiliser la commande **ip route** <*adresse réseau distant*> <*masque de sous-réseau*> <*interface de sortie*>.

Du point de vue du routage, il peut sembler curieux d'utiliser une interface plutôt que le routeur de saut suivant. En pareil cas, comment le routeur configuré avec une route statique pointant sur une interface peut-t-il choisir le routeur de saut suivant ?

La réponse peut se trouver assez aisément dans la table de routage, si on l'examine de près. Les routes connectées qui y figurent, pointent sur des interfaces et non des routeurs de saut suivant. Dans le cas des routes connectées, cela se comprend car le routeur est lui-même le dernier saut vers ces réseaux. Nul besoin d'un autre routeur. Comme nous le savons d'un chapitre précédent, quand le routeur de dernier saut doit acheminer un datagramme à sa destination finale, il essaye de traduire l'adresse IP en celle de la couche accès réseau, qui est l'adresse MAC dans le cas des LAN. S'il y parvient, le routeur envoie directement le datagramme à destination.

Les routes statiques pointant sur des interfaces se comportent comme des routes connectées. Le routeur considère ces réseaux comme directement connectés à ses interfaces, et tout le trafic qui leur est destiné aboutit sur les segments correspondants. Le routeur par conséquent s'attribue le rôle de celui du saut suivant, et même du dernier saut ; il n'a donc recours à aucun autre routeur. Pour traduire l'adresse IP de destination en adresse de la couche accès réseau, c'est le protocole ARP qui est utilisé dans le cas des LAN. Nous qualifierons ces routes statiques spéciales de *pseudo-connectées*.

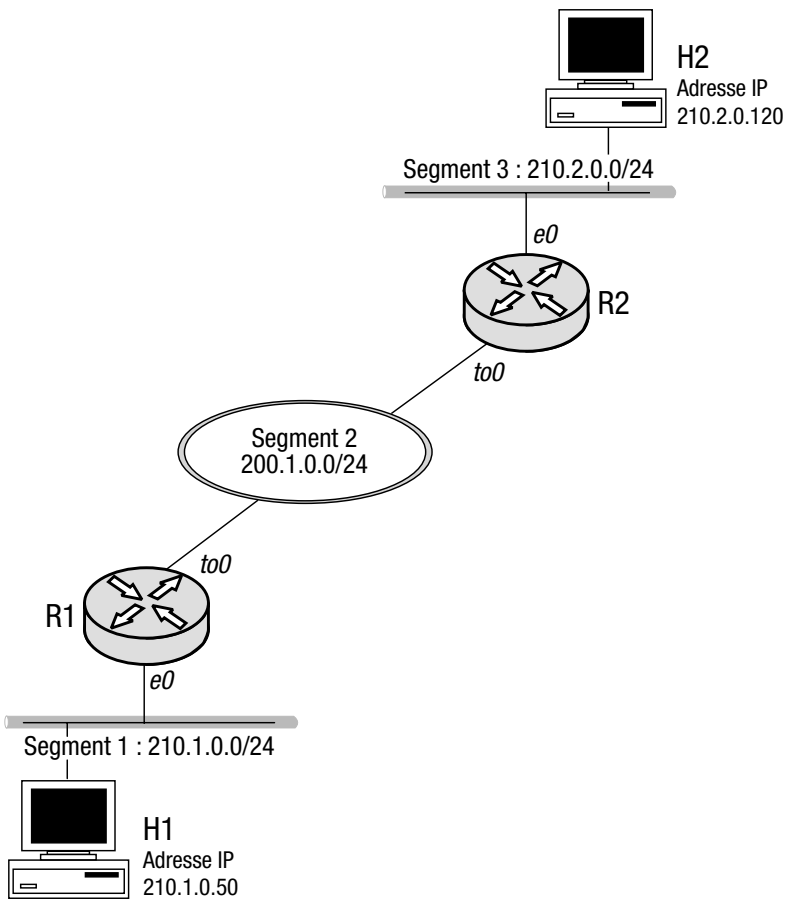
© copyright Éditions Eyrolles

Bien qu'une route pseudo-connectée se comporte de la même façon qu'une route statique normale, le routeur n'a pas d'adresse IP sur ce réseau « connecté ». Bien évidemment, si un hôte réside sur le réseau pseudo-connecté, il ne pourra pas utiliser ce routeur comme passerelle par défaut parce que ce dernier n'y possède pas d'adresse.

Prenons l'exemple de la figure 3.4 où deux routeurs R1 et R2 sont interconnectés par un anneau Token Ring (segment 2) ; ils sont configurés avec des routes statiques pseudo-connectées pour les réseaux situés derrière eux (segment 1 pour R1 et segment 3 pour R2) ; chaque route pointe sur son interface respective du segment 2. Les listings 3.10 et 3.11 montrent leurs configurations ; les listings 3.12 et 3.13, leurs tables de routage.

**Figure 3.4**

*Routeurs configurés avec des routes statiques pseudo-connectées qui pointent sur leur interface respective du Token Ring pour la communication entre les hôtes H1 et H2.*




**Listing 3.10. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 210.1.0.1 255.255.255.0

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

ip route 210.2.0.0 255.255.255.0 TokenRing0
```

© copyright Éditions Eyrolles

**Listing 3.11. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 210.2.0.1 255.255.255.0

interface TokenRing0
 ip address 200.1.0.2 255.255.255.0
 ring-speed 16

ip route 210.1.0.0 255.255.255.0 TokenRing0
```

**Listing 3.12. Table de routage du routeur R1.**

```
R1#show ip route
...
C    200.1.0.0 is directly connected, TokenRing0
S    210.2.0.0 is directly connected, TokenRing0
C    210.1.0.0 is directly connected, Ethernet0
```

**Listing 3.13. Table de routage du routeur R2.**

```
R2#show ip route
...
C    200.1.0.0/24 is directly connected, TokenRing0
C    210.2.0.0/24 is directly connected, Ethernet0
S    210.1.0.0/24 is directly connected, TokenRing0
```

Nous constatons dans les tables de routage que les étiquettes des routes connectées normales sont différentes de celles des routes pseudo-connectées. Les premières sont codées avec la lettre « C » pour « connectée », tandis que les secondes le sont avec la lettre « S » pour « statique ».

Lançons une commande **ping** sur l'hôte H1 pour vérifier l'accessibilité de l'hôte H2. En utilisant la commande **debug arp** sur le routeur R1, nous pouvons voir s'afficher sur le listing 3.14, le déroulement de l'échange entre les routeurs R1 et R2 qui jouent mutuellement le rôle de serveur Proxy ARP.

**Listing 3.14. Déroulement de l'échange du protocole ARP entre les routeurs R1 et R2.**

```
R1#debug arp
ARP packet debugging is on
R1#
IP ARP: creating incomplete entry for IP address: 210.2.0.120
IP ARP: sent req src 200.1.0.1 0007.0d26.0a46,
                dst 210.2.0.120 0000.0000.0000 TokenRing0
IP ARP: rcvd rep src 210.2.0.120 0007.0d26.0c15,
 dst 200.1.0.1 TokenRing0
IP ARP: rcvd req src 200.1.0.2 0007.0d26.0c15,
 dst 210.1.0.50 TokenRing0
IP ARP: creating entry for IP address: 200.1.0.2,
 hw: 0007.0d26.0c15
IP ARP: sent rep src 210.1.0.50 0007.0d26.0a46,
                dst 200.1.0.2 0007.0d26.0c15 TokenRing0
```

© copyright Éditions Eyrolles

Selon le scénario de cet échange, le routeur R1 tente de traduire l'adresse IP de l'hôte H2 en adresse MAC par une requête ARP. Le routeur R2 répond avec l'adresse MAC de son interface Token Ring. Le paquet du **ping** de l'hôte H1 qui est à l'origine de ce dialogue, est remis à l'hôte H2. Celui-ci envoie une réponse pour laquelle le routeur R2 doit traduire l'adresse IP de l'hôte H1 en adresse MAC. C'est maintenant au tour du routeur R1 de répondre à la requête ARP du routeur R2 en lui envoyant l'adresse MAC de son interface Token Ring.

Il existe une certaine similitude entre une route statique pseudo-connectée qu'on vient d'évoquer et une adresse IP secondaire qui peut être assignée à une interface, comme celle vue dans un chapitre précédent. Ces deux moyens permettent de créer une route vers l'adresse réseau correspondante dans la table de routage, qui pointe non pas sur un routeur de saut suivant, mais sur une interface. La seule différence, c'est que l'adresse IP secondaire assignée à l'interface d'un routeur peut être utilisée comme *default gateway* par les hôtes situés sur le segment connecté à cette interface, alors que l'interface d'un routeur pointant sur une route pseudo-connectée ne possède pas d'adresse IP sur ce réseau destinataire.

L'utilisation de routes pseudo-connectées ne peut être qu'un palliatif, si l'on veut éviter un trafic de diffusion générale excessif sur les segments concernés. Elle nécessite une gestion complexe et peut être source de problèmes. Il en est de même pour les adresses IP secondaires.

## Configuration du routage sans classe

La commande **ip classless** en mode de configuration globale permet aux routeurs Cisco d'utiliser la meilleure adresse super-réseau disponible si les conditions suivantes sont remplies :

- Une ou plusieurs interfaces d'un routeur sont configurées avec une adresse IP appartenant aux sous-réseaux de l'adresse réseau dont fait partie l'adresse de destination.
- L'adresse IP de destination appartient à un sous-réseau pour lequel le routeur ne possède aucune route.
- Le routeur ne possède aucune route pour l'adresse réseau dont fait partie l'adresse de destination.
- La table de routage contient une route ou plus vers les super-réseaux auxquels appartient l'adresse IP de destination.

Sans la commande **ip classless**, le routeur met au rebut tout paquet dont l'adresse IP de destination remplit les conditions citées ci-dessus. Si cette commande est activée, le routeur achemine le paquet dont l'adresse IP de destination correspond à une ou plusieurs routes de super-réseau.

Prenons l'exemple de la figure 3.5.

Le routeur ne possède pas de route individuelle pour le réseau 150.1.2.0/24. En revanche, il est configuré avec la route statique du super-réseau 150.0.0.0/8 comme on peut le voir sur le listing 3.15. La configuration du routeur R2 est aussi montrée à titre indicatif sur le listing 3.16.

**Listing 3.15. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 150.1.1.1 255.255.255.0

interface Serial0
 ip address 150.1.254.1 255.255.255.0

ip route 150.0.0.0 255.0.0.0 150.1.254.2
```

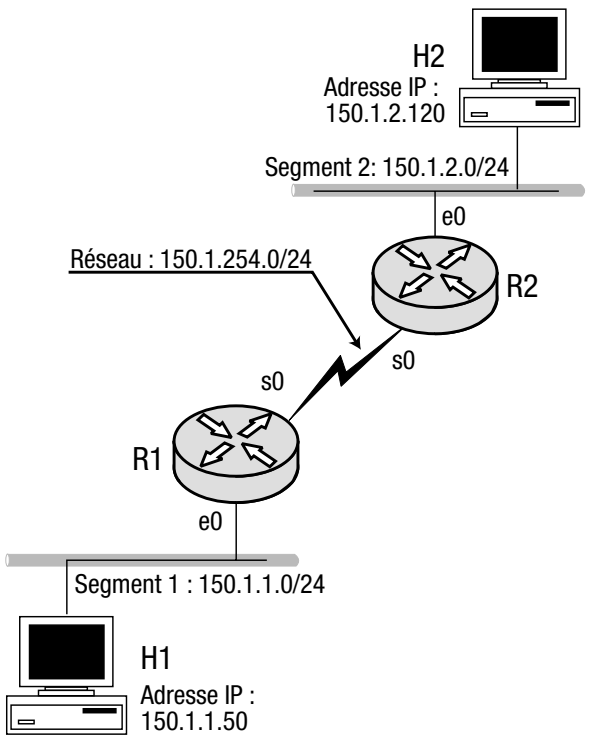
© copyright Éditions Eyrolles

**Listing 3.16. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 150.1.2.1 255.255.255.0

interface Serial1
 ip address 150.1.254.2 255.255.255.0

ip route 150.1.1.0 255.255.255.0 150.1.254.1
```

**Figure 3.5**

*Routeur R1 configuré uniquement avec une route de super-réseau 150.0.0.0/8 pour le segment 2.*




À ce stade, la commande **ip classless** n'est pas activée dans la configuration du routeur R1. Si nous lançons un **ping** de l'hôte H1 vers l'hôte H2, nous voyons sur le listing 3.17 qu'il échoue (destination inaccessible).

**Listing 3.17. En algorithme de routage à classe on ne peut utiliser une route de super-réseau pour l'adresse 150.0.0.0/8.**

```
C:\>ping 150.1.2.120

Pinging 150.1.2.120 with 32 bytes of data:

Reply from 150.1.1.1: Destination host unreachable.
Reply from 150.1.1.1: Destination host unreachable.
Reply from 150.1.1.1: Destination host unreachable.
```

© copyright Éditions Eyrolles

Une fois la commande **ip classless** activée sur le routeur R1, le **ping** peut aboutir, comme le montre le listing 3.18.

**Listing 3.18. Changement d'algorithme de routage de classe à sans classe pour établir une route de super-réseau, par la commande `ip classless`.**

```
C:\>ping 150.1.2.120

Pinging 150.1.2.120 with 32 bytes of data:

Reply from 150.1.2.120: bytes=32 time=40ms TTL=126
Reply from 150.1.2.120: bytes=32 time=30ms TTL=126
Reply from 150.1.2.120: bytes=32 time=30ms TTL=126
Reply from 150.1.2.120: bytes=32 time=30ms TTL=126
```

**REMARQUE** Bien que la documentation de Cisco précise que la commande **ip classless** est désactivée par défaut, les versions ultérieures de l'IOS de Cisco peuvent faire l'inverse.

## Configuration de la route par défaut sur un routeur

La route de super-réseau qui pointe sur la passerelle par défaut est utilisée quand l'adresse de destination ne correspond à aucune autre route. Par définition, cette route est identifiée par le préfixe réseau et sa longueur, sous la forme : 0.0.0.0/0.

Pour configurer une route pointant sur la passerelle par défaut, on doit d'abord utiliser la commande **ip classless** et la faire suivre par celle de **ip route** *<adresse réseau> <masque> <adresse IP du routeur de saut suivant> ;* les deux premiers paramètres doivent être renseignés avec les valeurs 0.0.0.0 et 0.0.0.0, respectivement.

## Configuration de routes individuelles pour des hôtes

Avec la commande **ip route** *<adresse IP de l'hôte> <masque> <adresse IP du routeur de saut suivant>*, en renseignant le masque à 255.255.255.255, on peut configurer une route individuelle pour un hôte. Cette adresse IP n'est pas obligatoirement celle d'un hôte, il peut s'agir aussi d'une adresse IP d'un routeur.

Le routeur configuré sur le listing 3.19 possède une route individuelle d'hôte pour 200.2.0.120.

**Listing 3.19. Configuration du routeur R1.**

```
interface Serial0
 ip address 195.0.0.1 255.255.255.0

interface Serial1
 ip address 195.1.0.1 255.255.255.0

ip route 200.2.0.0 255.255.255.0 195.0.0.2
ip route 200.2.0.120 255.255.255.255 195.1.0.2
```

© copyright Éditions Eyrolles

**Listing 3.20. Table de routage du routeur R1.**

```
R1#show ip route
...
C    195.1.0.0/24 is directly connected, Serial1
C    195.0.0.0/24 is directly connected, Serial0
C    200.1.0.0/24 is directly connected, TokenRing0
     200.2.0.0/24 is variably subnetted, 2 subnets, 2 masks
S    200.2.0.120/32 [1/0] via 195.1.0.2
S    200.2.0.0/24 [1/0] via 195.0.0.2
```

On peut constater sur le listing 3.20 que le routeur utilise une de ses interfaces (ligne série 1) pour une route individuelle dont l'adresse IP est : 200.2.0.120.

Les routes individuelles sont généralement utilisées à des fins de débogage ou quand la route principale vers un réseau particulier est devenue inaccessible.

## Configuration du partage de charge à coût égal en routage statique

Le routeur peut exécuter la fonction de partage de charge sur six routes au maximum vers la même destination (dans les versions futures du système IOS de Cisco, ce nombre étant susceptible de changer). Le routeur considère toutes ces routes comme ayant le même débit, quel que soit leur débit réel, et répartit le trafic de manière égale sur chacune d'elles. Ce type de partage de charge est appelé « partage de charge à coût égal », ce qui a été vu dans la première section de ce chapitre.

**REMARQUE** Rappelons que le partage de charge ne s'exécute que sur le trafic sortant, le routeur n'ayant aucun contrôle sur le trafic entrant.

Pour configurer le partage de charge, nous devons entrer des commandes qui contiennent la même destination, mais pointant sur des interfaces ou des routeurs de saut suivant, différents.

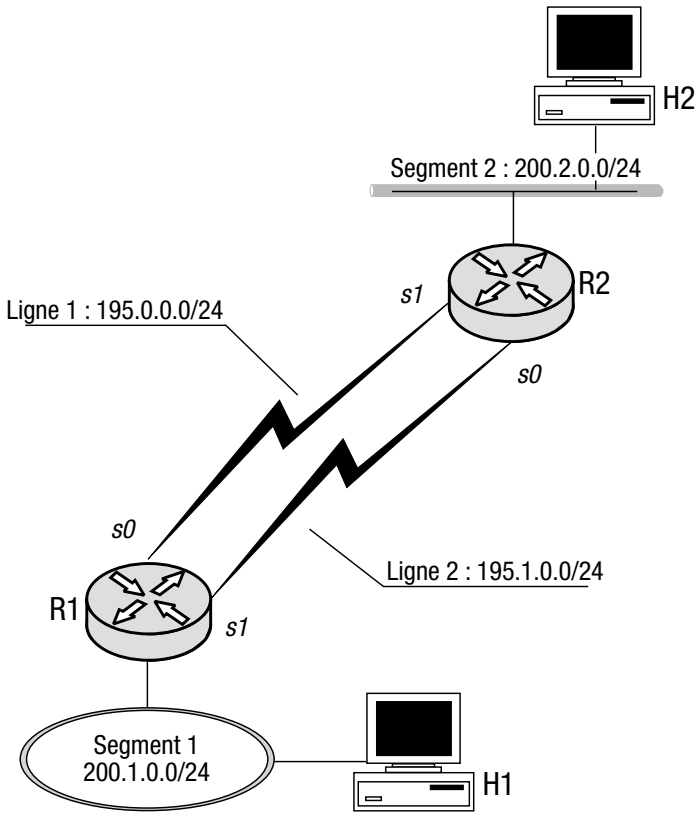
La figure 3.6 illustre le cas de deux routeurs R1 et R2 interconnectés par une paire de lignes série. Les listings 3.21 et 3.22 montrent leur configuration respective. Les tables de routage sont imprimées sur les listings 3.23 et 3.24 où on peut voir une même adresse réseau pointer sur deux routeurs de saut suivant, différents.

**Listing 3.21. Configuration du routeur R1.**

```
interface Serial0
 ip address 195.0.0.1 255.255.255.0

interface Serial1
 ip address 195.1.0.1 255.255.255.0

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

ip route 200.2.0.0 255.255.255.0 195.0.0.2
ip route 200.2.0.0 255.255.255.0 195.1.0.2
```

© copyright Éditions Eyrolles

**Figure 3.6**
*Deux routeurs configurés en partage de charge sur une paire de lignes série.*




**Listing 3.22. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 195.1.0.2 255.255.255.0

interface Serial1
 ip address 195.0.0.2 255.255.255.0

ip route 200.1.0.0 255.255.255.0 195.0.0.1
ip route 200.1.0.0 255.255.255.0 195.1.0.1
```

**Listing 3.23 Table de routage du routeur R1.**

```
R1#show ip route
...
C    195.1.0.0 is directly connected, Serial1
C    195.0.0.0 is directly connected, Serial0
C    200.1.0.0 is directly connected, TokenRing0
S    200.2.0.0 [1/0] via 195.0.0.2
               [1/0] via 195.1.0.2
```

© copyright Éditions Eyrolles

**Listing 3.24. Table de routage du routeur R2.**

```
R2#show ip route
...
C    195.1.0.0/24 is directly connected, Serial0
C    195.0.0.0/24 is directly connected, Serial1
S    200.1.0.0/24 [1/0] via 195.0.0.1
                  [1/0] via 195.1.0.1
C    200.2.0.0/24 is directly connected, Ethernet0
```

La configuration de plusieurs routes statiques vers la même destination, comme dans le cas des routeurs R1 et R2, permet d'exécuter le partage de trafic uniquement sur ces derniers. Si les autres routeurs sont configurés avec une seule route, le trafic n'empruntera que celle-ci. Comme nous l'avons vu dans la première section de ce chapitre, les routeurs Cisco ont deux modes d'exécution du partage de charge : *par destination* et *par paquet*. Le partage de charge par destination s'exécute en commutation rapide (*fast switching*), qui consiste à conserver dans une mémoire cache les adresses de destination, pour éviter de consulter la table de routage chaque fois qu'un paquet pour la même adresse arrive sur une interface. Tant qu'une adresse de destination se trouve dans cette mémoire, le routeur utilise toujours la même interface pour y acheminer les paquets. On peut activer la commutation rapide par la commande **ip route-cache**. Mais celle-ci est normalement active par défaut, bien qu'elle n'apparaisse pas dans la configuration du routeur.

**REMARQUE** Les modules VIP (Versatile Interface Processor) des routeurs haut de gamme tel que le Cisco 7500 et les modules RSM (Route Switch Module) des commutateurs Catalyst, possèdent d'autres commandes de commutation rapide, répertoriées dans la documentation Cisco. S'y reporter pour plus d'informations.

Quant au partage de charge par paquet, il ne peut s'exécuter qu'en désactivant la commutation rapide par la commande **no ip route-cache**. Dans ce cas, le routeur distribue les paquets allant vers une même destination, à tour de rôle, sur toutes les interfaces pointées par cette destination. À moins d'avoir une ligne de faible débit qui peut être saturée ou des pics de trafic importants, il est déconseillé de désactiver la commutation rapide qui demeure la plus performante, sauf quand on doit utiliser la commande **debug ip-packet** pour visualiser l'acheminement des paquets. Si on reste en commutation rapide sous cette commande, on n'aura en sortie que les paquets en provenance ou à destination du routeur.

L'autre utilisation de la commande **debug ip-packet** est la visualisation de l'exécution du partage de charge. Les listings 3.25 et 3.26 montrent un exemple pour les routeurs R1 et R2 de la figure 3.6. On peut y voir le déroulement d'un **ping** lancé de l'hôte H1 à l'adresse 200.1.0.15 vers l'hôte H2 à l'adresse 200.2.0.120. Les routeurs alternent les paquets sur leurs deux interfaces configurées en partage de charge.

**Listing 3.25. Exécution de la commande `debug ip-packet` sur le routeur R1.**

```
IP: s=200.1.0.15 (TokenRing0), d=200.2.0.120 (Serial0),
 g=195.0.0.2, len 82, forward
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0),
 g=200.1.0.15, len 64, forward
IP: s=200.1.0.15 (TokenRing0), d=200.2.0.120 (Serial1),
```

© copyright Éditions Eyrolles

```
 g=195.1.0.2, len 82, forward
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0),
 g=200.1.0.15, len 64, forward
IP: s=200.1.0.15 (TokenRing0), d=200.2.0.120 (Serial0),
 g=195.0.0.2, len 82, forward
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0),
 g=200.1.0.15, len 64, forward
IP: s=200.1.0.15 (TokenRing0), d=200.2.0.120 (Serial1),
 g=195.1.0.2, len 82, forward
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0),
 g=200.1.0.15, len 64, forward
```

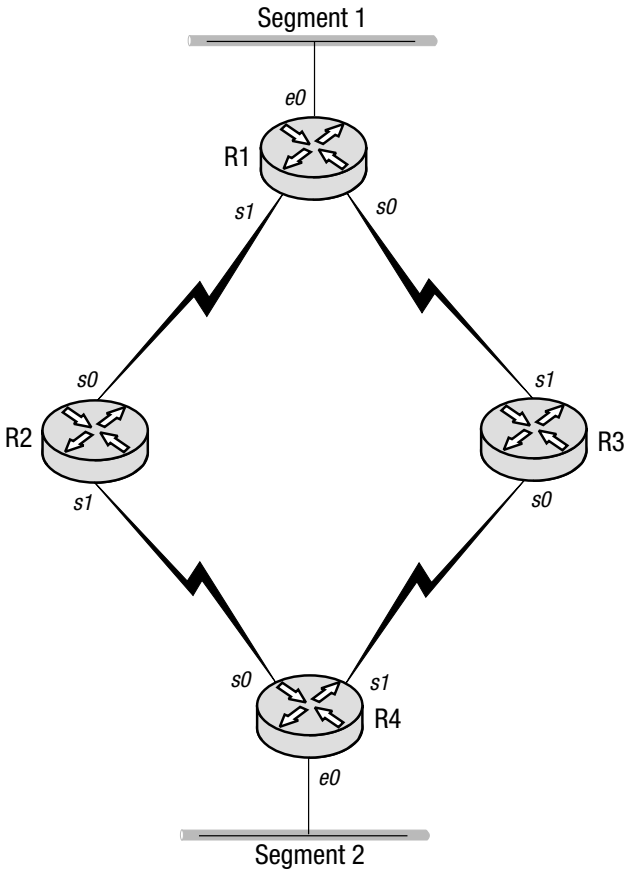
**Listing 3.26. Exécution de la commande `debug ip-packet` sur le routeur R2.**

```
IP: s=200.1.0.15 (Serial1), d=200.2.0.120 (Ethernet0),
 g=200.2.0.120, len 60, forward
IP: s=200.2.0.120 (Ethernet0), d=200.1.0.15 (Serial1),
 g=195.0.0.1, len 60, forward
IP: s=200.1.0.15 (Serial0), d=200.2.0.120 (Ethernet0),
 g=200.2.0.120, len 60, forward
IP: s=200.2.0.120 (Ethernet0), d=200.1.0.15 (Serial0),
 g=195.1.0.1, len 60, forward
IP: s=200.1.0.15 (Serial1), d=200.2.0.120 (Ethernet0),
 g=200.2.0.120, len 60, forward
IP: s=200.2.0.120 (Ethernet0), d=200.1.0.15 (Serial1),
 g=195.0.0.1, len 60, forward
IP: s=200.1.0.15 (Serial0), d=200.2.0.120 (Ethernet0),
 g=200.2.0.120, len 60, forward
IP: s=200.2.0.120 (Ethernet0), d=200.1.0.15 (Serial0),
 g=195.1.0.1, len 60, forward
```

Le partage de charge ne s'applique pas uniquement au cas d'une paire de lignes série. On peut aussi configurer des routes statiques qui pointent sur deux routeurs de saut suivant, différents. Cependant, la complexité de leur configuration augmente avec le nombre de routeurs et de lignes les reliant. Il est préférable d'utiliser les protocoles de routage dynamique capables de mettre à jour automatiquement les tables de routage de manière optimale. Le partage de charge en routage statique est donc limité au cas des routeurs reliés par une paire de lignes série (cf. figure 3.6) ou à celui d'un routeur relié symétriquement à deux autres comme dans la figure 3.7. Les routeurs R1 et R4 de cet exemple, peuvent être configurés facilement en routes statiques de façon à ce que le trafic entre les segments 1 et 2 soit également réparti entre les routes transitant par les routeurs R2 et R3.

© copyright Éditions Eyrolles

**Figure 3.7**
*Configuration à quatre routeurs qui permet le partage de charge du trafic sur des lignes symétriques.*




## Configuration du partage de charge à coût inégal en routage statique

Si curieux que cela puisse paraître, nous pouvons néanmoins configurer en routage statique le partage de charge à coût inégal. Bien qu'assez limité, un tel partage de charge peut revêtir un caractère élégant et original.

**REMARQUE** La solution suivante fut suggérée par Steve Kann à un groupe de discussion sur le web (*comp.dcom.sys.cisco*). Son caractère élégant mis à part, elle a pu ébranler quelque peu l'idée reçue selon laquelle les technologues suivent servilement les instructions des constructeurs qui soutiennent que « le routage statique ne permet que le partage de charge à coût égal ». Voilà une contribution qui mérite d'être soulignée !

Cette solution est basée sur le fait tout simple qu'un routeur ne répartit pas réellement le trafic vers une même destination entre ses interfaces, mais plutôt entre les routes qui pointent sur cette destination. Par exemple, si un routeur possède, pour une même destination, trois routes sur une première interface, et deux autres sur une deuxième, il va fractionner le trafic aux trois cinquièmes sur la première et aux deux cinquièmes sur la deuxième. Bien évidemment, cette manière de répartir le trafic est limitée par le nombre de routes qui peuvent pointer sur une même destination, au maximum six dans la version actuelle du Système IOS de Cisco. Le tableau 3.1 donne les différents fractionnements utiles dans certains cas.

© copyright Éditions Eyrolles

**Tableau 3.1. Fractionnement de trafic selon le nombre de routes par interface.**

| Fractionnements de trafic | Nombre de routes sur interface 1 | Nombre de routes sur interface 2 |
|---|---|---|
| 1/3 et 2/3 | 1 | 2 |
| 1/4 et 3/4 | 1 | 3 |
| 1/5 et 4/5 | 1 | 4 |
| 1/6 et 5/6 | 1 | 5 |
| 2/5 et 3/5 | 2 | 3 |

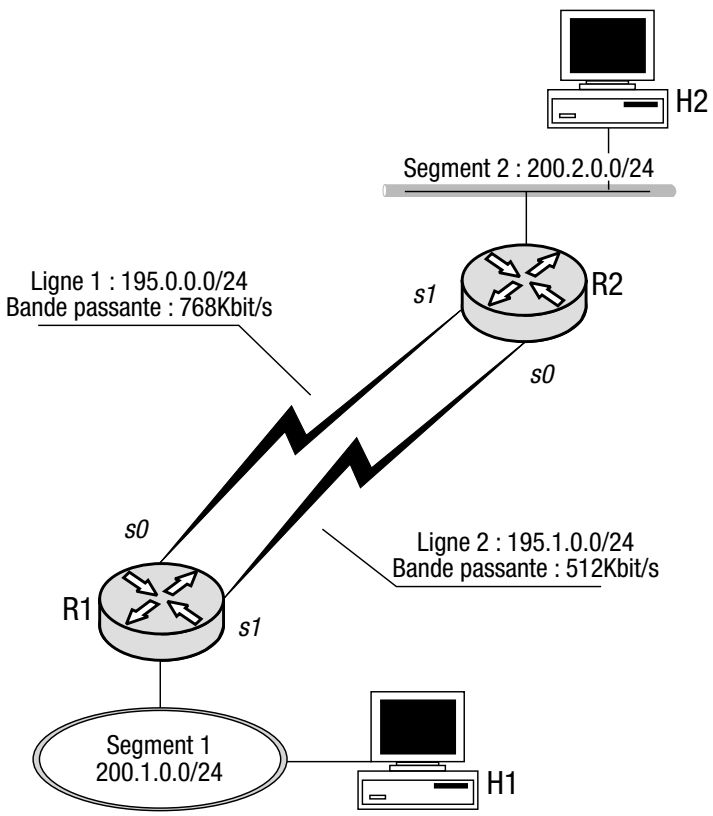
La méthode la plus facile pour implémenter le partage de charge à coût inégal dans un routeur, c'est de définir des adresses secondaires sur les interfaces locales et de configurer plusieurs routes statiques pointant sur l'adresse primaire et les adresses secondaires des interfaces du routeur de saut suivant.

Prenons le cas de la figure 3.8 où les routeurs R1 et R2 sont connectés par deux lignes dont les débits respectifs (768 Kbit/s et 512 Kbit/s) sont dans les proportions de 3/5 et 2/5. D'après le tableau 3.1, le fractionnement de trafic peut s'effectuer en pointant trois routes sur les interfaces de la première ligne et deux autres sur les interfaces de la deuxième ligne.

Les listings 3.27 et 3.28 contiennent un exemple de configuration des routeurs R1 et R2 selon le partage de charge décrit ci-dessus.

**Figure 3.8**

*Le partage de charge à coût inégal sur deux lignes reliant les routeurs R1 et R2.*




© copyright Éditions Eyrolles

**Listing 3.27. Configuration du routeur R1.**

```
interface Serial0
 ip address 195.0.0.1 255.255.255.0
 ip address 195.0.0.2 255.255.255.0 secondary
 ip address 195.0.0.3 255.255.255.0 secondary

interface Serial1
 ip address 195.1.0.1 255.255.255.0
 ip address 195.1.0.2 255.255.255.0 secondary

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

ip route 200.2.0.0 255.255.255.0 195.0.0.4
ip route 200.2.0.0 255.255.255.0 195.0.0.5
ip route 200.2.0.0 255.255.255.0 195.0.0.6
ip route 200.2.0.0 255.255.255.0 195.1.0.3
ip route 200.2.0.0 255.255.255.0 195.1.0.4
```

**Listing 3.28. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 195.1.0.3 255.255.255.0
 ip address 195.1.0.4 255.255.255.0 secondary

interface Serial1
 ip address 195.0.0.4 255.255.255.0
 ip address 195.0.0.5 255.255.255.0 secondary
 ip address 195.0.0.6 255.255.255.0 secondary

ip route 200.1.0.0 255.255.255.0 195.0.0.1
ip route 200.1.0.0 255.255.255.0 195.0.0.2
ip route 200.1.0.0 255.255.255.0 195.0.0.3
ip route 200.1.0.0 255.255.255.0 195.1.0.1
ip route 200.1.0.0 255.255.255.0 195.1.0.2
```

En désactivant la commutation rapide et en utilisant la commande **debug ip-packet** sur le routeur R1, on peut voir sur le listing 3.29 comment R1 exécute le partage de charge en alternant les interfaces de sortie (mises en italique sur le listing avec des lignes tronquées pour une lecture plus aisée).

**Listing 3.29. Sortie de la commande `debug ip-packet` sur le routeur R1.**

```
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0), ...
```

© copyright Éditions Eyrolles

```
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial0), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0), ...
IP: s=200.2.0.120 (Serial1), d=200.1.0.15 (TokenRing0), ...
```

© copyright Éditions Eyrolles

# 4

# Routage dynamique : protocoles à vecteur de distance

**Solutions de configuration présentées dans ce chapitre**



© copyright Éditions Eyrolles

Les protocoles de routage dynamique ont été spécifiquement conçus pour mettre à jour la table de routage d'un hôte ou d'un routeur. Pour ce faire, ils s'échangent des paquets appelés *mises à jour de routage* contenant des informations destinées à remplir la table de routage. Elles n'ont de sens que pour un protocole de routage donné, et ne peuvent être ni reçues, ni *a fortiori* interprétées par un autre protocole.

**REMARQUE** Les protocoles de routage ne doivent pas être confondus avec les protocoles routés (comme IP et IPX parmi d'autres). Un protocole de routage utilise une table de routage pour chaque protocole routé. Par exemple, si dans un routeur IPX et IP fonctionnement simultanément, ils ont chacun leur table de routage.

Les protocoles de routage peuvent être classés en deux catégories : ceux qui sont intra-domaine appelés IGP (*Interior Gateway Protocol*) et ceux qui sont inter-domaines appelés EGP (*Exterior Gateway Protocol*). Ces derniers servent au routage entre réseaux relevant d'autorités administratives différentes telles que des entreprises ou des fournisseurs d'accès à l'Internet – ISP (*Internet Service Provider*).

Les protocoles de routage intra-domaine constituent le sujet de ce chapitre. Les protocoles inter-domaines qui concernent principalement le routage dans l'Internet sont d'une telle complexité, qu'un livre entier pourrait leur être consacré.

Ces deux catégories de protocoles se subdivisent encore selon l'algorithme sur lequel ils sont basés : *vecteur de distance* (*distance vector*) ou *état des liens* (*link state*). Ce chapitre traite des IGP du premier groupe, tandis que le prochain est consacré à celui du second groupe, OSPF.

## Algorithme à vecteur de distance

Bellman, un chercheur à l'université de Princeton aux États-Unis, fut à l'origine de l'équation de routage en 1957, qui contribua plus tard en 1962, à la conception par deux autres chercheurs de la même université, Ford et Fulkerson, du prototype de l'algorithme à vecteur de distance dont sont issus ceux utilisés de nos jours. Ils portent le nom collectif de leurs auteurs : Bellman-Ford ou Ford-Fulkerson.

Avant d'étudier le fonctionnement de l'algorithme à vecteur de distance, voyons de plus près sa terminologie et ses principes dans les paragraphes suivants.

Les routeurs qui utilisent le même protocole de routage pour échanger des mises à jour ne peuvent être séparés par plus d'un réseau physique. Autrement dit, les paquets de mise à jour ne peuvent être routés, précisément parce que le routage dépend d'eux. Les routeurs engagés dans ces mises à jour sont appelés *voisins* (*neighbors*).

L'échange de paquets de mise à jour s'effectue sous un format qui correspond à la PDU du protocole de routage concerné. Les mises à jour de routage contiennent normalement les préfixes réseau et les métriques associées. Ces préfixes sont dits être *annoncés* par le routeur quand il les diffuse à ses voisins.

© copyright Éditions Eyrolles

Les métriques indiquent généralement à quelle distance le routeur expéditeur se place lui-même par rapport au préfixe réseau qu'il annonce. Le routeur en réception insère une route dans sa table, selon le critère de la meilleure métrique associée à ce préfixe, ignorant toutes les autres mises à jour reçues le concernant.

Les métriques dépendent en fait du protocole de routage et varient quant à la façon dont elles sont définies et calculées. Néanmoins, tous les protocoles de routage à vecteur de distance appliquent les règles qui sont les suivantes :

- Chaque interface d'un routeur desservie par un protocole de routage est affectée d'un coût qui est spécifique à ce protocole.
- Quand un routeur reçoit une mise à jour pour un préfixe réseau par annonces diffusées, il recalcule la métrique associée à ce dernier, en tenant compte du coût de l'interface par laquelle il va envoyer à son tour le message de mise à jour pour ce préfixe. Dans la plupart des cas, les interfaces de réception et d'envoi sont les mêmes.
- La nouvelle métrique calculée par le routeur doit être supérieure à celle qui a été reçue auparavant, car il doit y inclure le coût de l'interface de sortie.

De ce qui est exposé ci-dessus on peut déduire qu'un routeur, à l'initialisation, n'annonce que les préfixes dont il a une connaissance immédiate, c'est-à-dire ceux des réseaux auxquels il est directement connecté. Au fur et à mesure qu'il reçoit les préfixes annoncés par ses voisins, il en recalcule les métriques pour les diffuser à son tour. Peu à peu, tous les routeurs de tous les segments finiront par apprendre tous les préfixes réseau dans leur totalité.

**REMARQUE** Il faut noter que les routeurs qui diffusent les préfixes réseau par le protocole à vecteur de distance, n'ont aucune idée des réseaux intermédiaires qu'il faut traverser pour atteindre ces préfixes. L'algorithme à vecteur de distance est parfois qualifié, pour cette raison, de « routage par ouï-dire ».

Dans un réseau stable, la table de routage de chaque routeur devrait rester la même, une fois qu'elle a été remplie. Mais c'est rarement le cas, car des changements interviennent dans un réseau même stable, ajouts, mises hors service, pannes, etc. Les protocoles de routage sont chargés de les gérer selon deux procédés :

- Si un nouveau segment est ajouté à un réseau, le routeur connecté à ce segment commence par annoncer le préfixe de ce dernier à ses voisins, et par propagation successive, tous les routeurs du réseau apprendront l'existence de ce nouveau segment. En même temps, le routeur envoie des mises à jour concernant le reste du réseau à ce segment dont il recevra en retour celles émises par les routeurs qui s'y trouvent.
- Les pannes de composants réseau ou leur mise hors service sont prises en compte par l'échange régulier de mises à jour entre les routeurs, par exemple toutes les 30 secondes. En outre, pour chaque route insérée dans sa table, le routeur démarre un temporisateur qui est réinitialisé à chaque mise à jour successive de cette route. Si le temporisateur arrive à échéance sans qu'il y ait eu de nouvelle mise à jour, la route concernée est effacée de la table.

**REMARQUE** L'usage des mises à jour et des temporisateurs peut varier d'un protocole à l'autre, et peut ne pas correspondre à ce qui est décrit précédemment. Les protocoles de première génération tels que RIP et IGRP s'échangent régulièrement des mises à jour sur tous les préfixes pour lesquels ils ont une route. Ceux qui sont plus perfectionnés, tel que le EIGRP, au lieu des mises à jour périodiques, s'échangent plutôt des messages de « salut » (*hello*) pour se tenir informés de l'état des routeurs voisins. Ils n'ont pas non plus un temporisateur pour chaque route, mais consignent les messages *hello* ; au bout d'un certain nombre de messages manqués, le routeur voisin correspondant est déclaré hors service, et les routes qui en dépendent sont recalculées.

© copyright Éditions Eyrolles

## Algorithme à vecteur de distance amélioré : règle de clivage d'horizon, temporisateur de maintien et mises à jour déclenchées

Il prend un certain temps pour que les informations sur les changements se propagent sur le réseau tout entier. Pendant ce laps de temps, certains segments devenus inaccessibles et dont les routes de rechange ont déjà été transmises, peuvent ne pas avoir atteint les routeurs les plus éloignés. La mise en cohérence des tables de routage de tous les routeurs amène l'état dit de *convergence*.

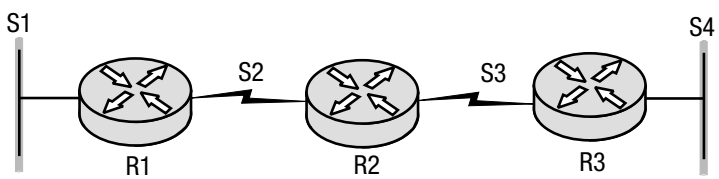
La durée qu'il faut pour parvenir à cet état s'appelle *temps de convergence*. Cette durée dépend du protocole de routage dynamique utilisé. L'implémentation particulière de l'algorithme d'un protocole de routage n'empêche pas qu'il soit sujet au phénomène du *comptage à l'infini*.

Pour comprendre le comptage à l'infini, prenons le cas de la figure 4.1.

Supposons que les trois routeurs R1, R2 et R3 utilisent le même protocole de routage et que leurs tables aient convergé, chaque routeur étant informé des réseaux accessibles par les deux autres. Nous savons que les routeurs annoncent les préfixes réseau *via* toutes leurs interfaces. Par exemple, le routeur R2 diffuse le préfixe du segment S1 à travers ses deux interfaces. Le routeur R3 n'ayant pas une meilleure route pour ce segment, prendra en compte la mise à jour diffusée par le routeur R2, tandis que le routeur R1, directement connecté au segment concerné, va l'ignorer. Supposons maintenant que le segment S2 soit rompu ; le routeur R2 démarre un temporisateur pour le segment S1 qui, arrivée à échéance, va lui indiquer que ce segment n'est plus accessible via le routeur R1. Dans l'intervalle, le routeur R2 continue à diffuser la mise à jour du segment S1, de nouveau prise en compte par le routeur R3 qui ignore que l'annonce concernant ce segment est périmée. Quand le temporisateur pour le segment S1 arrive enfin à échéance, et que le routeur R2 considère une route alternative pour ce segment, il reçoit une mise à jour pour celui-ci du routeur R3. Le routeur R2 recalcule sa propre métrique pour le segment S1 et l'envoie à son tour au routeur R3. Ce dernier constatant que la métrique pour le segment S1 a empiré, calcule une nouvelle métrique pour la retourner au routeur R2. Ce va-et-vient va aggraver la métrique associée au segment, à chaque itération, sans toutefois le faire disparaître.

**Figure 4.1**

*Le problème du comptage à l'infini dans une topologie de réseau simple.*




La méthode la plus facile pour résoudre ce problème est de considérer la route vers un réseau comme impraticable, dès que la métrique qui lui est associée devient supérieure à une certaine valeur, qu'on appelle *infini*. D'où l'expression, « comptage à l'infini ». Une fois cette valeur atteinte, le préfixe réseau concerné n'est plus accessible.

Le temps d'atteindre la valeur infinie, les routeurs croient encore que la route vers le segment en panne est toujours disponible. Les routeurs se transmettent des messages erronés jusqu'à ce que le champ durée de vie (TTL) des paquets tombe à zéro provoquant ainsi une boucle de routage. Pendant ce comptage à l'infini, les lignes reliant les routeurs dont les paquets tournent en rond, peuvent être saturées à un tel point que le trafic utile est retardé ou même mis au rebut.

© copyright Éditions Eyrolles

Quelques techniques ont été développées pour éliminer ou minimiser le problème du comptage à l'infini.

## Clivage d'horizon

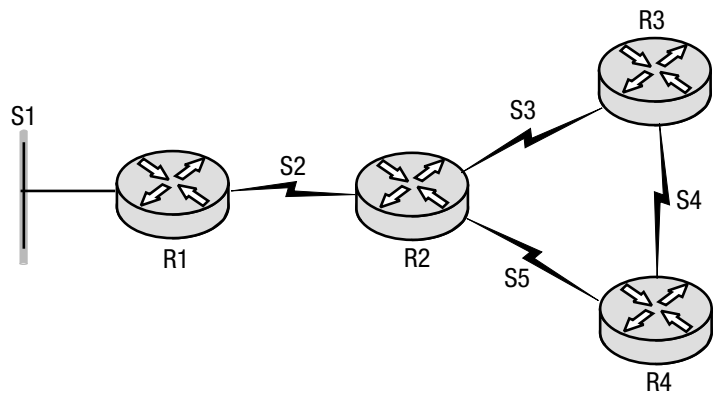
La première technique appelée *clivage d'horizon* (*split horizon*) définit une règle qui interdit à tout routeur d'annoncer un préfixe réseau *via* l'interface par laquelle il a appris l'existence de celui-ci. Dans le cas cité à la figure 4.1, les routeurs ne seraient pas confrontés au problème du comptage à l'infini, s'ils appliquaient cette règle.

Une deuxième technique, plus hardie, connue sous le nom de *clivage d'horizon à l'antidote* (*split horizon with poison reverse*), consiste à ordonner au routeur de retourner le préfixe par l'interface au travers de laquelle il l'a appris, en lui associant une métrique de valeur infinie.

Les règles énoncées précédemment, tout en améliorant le temps de convergence, peuvent n'avoir aucun effet dans certains cas, comme celui qui est illustré à la figure 4.2.

**Figure 4.2**

*Comptage à l'infini dans une topologie de réseau où la règle du clivage d'horizon à l'antidote ne résout pas le problème.*




Supposons que tous les routeurs appliquent la règle combinée du clivage d'horizon et d'antidote. Le segment S2 tombe en panne, et le temporisateur de route pour le segment S1 *via* le routeur R1, démarré par le routeur R2 arrive à échéance. Avant que cela se produise, le routeur R2 continue à annoncer le préfixe du segment S1 comme disponible aux routeurs R3 et R4. Ces derniers s'annoncent aussi mutuellement le même préfixe. Après que le routeur R2 a cessé d'annoncer le préfixe du segment S1, les temporisateurs de route pour ce préfixe, dans les routeurs R3 et R4, cette fois, arrivent à échéance à leur tour. Ce qui les amène à annoncer le préfixe du segment S1 dont ils s'échangeaient la route jusqu'à présent, vers le routeur R2. Ces trois routeurs se trouvent ainsi dans une boucle temporaire.

## Temporisateur de maintien

Une autre technique plus simple que le clivage d'horizon est celle du *temporisateur de maintien* (*holddown timer*). Toute route, en plus de son temporisateur normal, dispose d'un temporisateur additionnel qui démarre dès que le premier expire. Et la valeur de la métrique associée à la route affectée est mise à l'infini pour indiquer son caractère inaccessible. Tant que le deuxième temporisateur n'est pas arrivé à échéance, cette route ne peut être ni mise à jour ni effacée de la table.

Le but du temporisateur de maintien est d'éviter les boucles de routage pour empêcher toute saturation des lignes.

© copyright Éditions Eyrolles

### *Mises à jour déclenchées*

Il existe encore une dernière technique dite des *mises à jour déclenchées* (*triggered updates*) qui demande à ce que tout changement intervenu dans le réseau soit immédiatement signalé par le routeur qui en a connaissance sans attendre l'annonce périodique. Les autres routeurs sont ainsi informés rapidement de la situation, sans qu'ils aient à annoncer des préfixes de segments qui ne sont plus opérationnels.

## Distance administrative

Un routeur peut exécuter en parallèle plusieurs protocoles de routage IP. Quand c'est le cas, un conflit peut survenir quant à la route à inscrire dans la table de routage, pour une même destination, exprimée sous la forme d'un préfixe réseau. Pour résoudre ce genre de conflit, une valeur numérique spéciale appelée *distance administrative* est affectée à chaque protocole. Cette distance indique le degré de crédibilité accordé par le routeur au protocole correspondant. Plus sa valeur de distance est faible, plus le protocole est crédible. Si par exemple, un protocole de distance 100 préconise une route pour la destination 10.1.0.0/16 et qu'une autre route a déjà été installée dans la table par un autre protocole de distance 70, la route préconisée n'est pas prise en compte. Par contre, si le protocole qui préconise la nouvelle route avait une distance de 50, qui est inférieure à celle du protocole détenteur de la route existante, la nouvelle supplanterait l'ancienne.

Comme il est improbable que deux protocoles de routage essaient d'installer une route pour la même destination simultanément, la valeur de distance du protocole détenteur de la route est mémorisée chaque fois pour permettre d'en faire la comparaison avec celle d'un protocole candidat éventuel au remplacement de cette route. Celui-ci ne prendra effet que si la valeur de distance du candidat est inférieure à celle du détenteur. La distance administrative est, par conséquent, associée non seulement à un protocole mais aussi aux routes installées dans la table de routage.

Dans le système IOS de Cisco, la distance administrative est un entier positif qui va de 0 à 255.

En plus des protocoles de routage, les deux autres origines de l'information de routage sont les routes statiques et les routes implicites des interfaces connectées. Toutes ces sources ont une valeur par défaut pour leur distance administrative qu'illustre la table 4.1 pour les routeurs Cisco.

**Table 4.1. Valeurs par défaut de la distance administrative.**

| Origine de la route | Distance |
|---|---|
| Interface connectée | 0 |
| Route statique | 1 |
| IGRP amélioré (route agrégée) | 5 |
| BGP externe | 20 |
| IGRP amélioré (interne) | 90 |
| IGRP amélioré (externe) | 170 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| EGP | 140 |
| BGP interne | 200 |
| Inconnu | 255 |

© copyright Éditions Eyrolles

La valeur par défaut de la distance administrative est modifiable pour toutes les origines de l'information de routage sauf celle de l'interface connectée (toujours à 0) et celle de la route agrégée du protocole IGRP améliorée (toujours à 5).

Nous allons résumer ci-après, tous les éléments qui concourent au fonctionnement d'un routeur :

- Le routeur n'utilise les métriques que dans le cadre d'un protocole de routage particulier. S'il reçoit plusieurs messages de mise à jour pour le même préfixe, la route à la meilleure métrique prévaudra sur les autres pour être inscrite dans la table de routage.
- Si l'information de routage provient de plusieurs origines (par exemple, les protocoles de routage dynamique, les routes statiques et les routes d'interfaces connectées) pour la même destination, la route installée dans la table sera celle dont l'origine possède la plus petite valeur de distance administrative au détriment des autres.
- Le routeur utilise l'algorithme de recherche par la correspondance la plus longue pour trouver la meilleure route vers une destination, dans sa table de routage.
- Un protocole à vecteur de distance n'annonce pas une route qu'il n'a pas installée dans la table de routage.

**REMARQUE** Si un protocole de routage reçoit une mise à jour mais ne peut installer la route correspondante dans la table de routage, parce qu'elle y existe déjà, détenue par un autre protocole à la valeur de distance plus petite, il s'abstiendra de diffuser cette route.

## Protocoles de routage à classe et sans classe

Les protocoles de routage dynamique, qu'ils soient à vecteur de distance ou à état des liens, peuvent être à classe ou sans classe.

Les protocoles de routage dynamique à classe appliquent strictement les règles et les restrictions dévolues au schéma d'adressage IP à classe. Bien que tous les protocoles à classe supportent le masque de sous-réseau, ils exigent que celui-ci soit le même pour toutes les adresses IP appartenant à une adresse donnée de réseau à classe. Par exemple, si une adresse réseau 10.0.0.0 est utilisée avec un masque de sous-réseau 255.255.255.0 (ou /24), tous les autres sous-réseaux doivent avoir ce même masque. Cependant, si une autre adresse réseau 11.0.0.0 est utilisé, il peut avoir un autre masque de sous-réseau, par exemple 255.255.240.0 (ou /20).

Un trait important des protocoles de routage à classe, c'est leur incapacité à transmettre le masque de sous-réseau dans les messages de mises à jour. Comme ils supposent que le masque de sous-réseau sera le même pour toutes les adresses IP appartenant à une adresse réseau donnée, ce masque est récupéré de l'interface par laquelle ils reçoivent et transmettent les messages de mise à jour.

Les protocoles de routage sans classe envoient toujours le masque de sous-réseau avec le préfixe réseau annoncé dans leurs messages de mise à jour. Ces protocoles peuvent appliquer certaines restrictions des réseaux à classe, mais cette fonction peut être désactivée. Tous les protocoles de routage sans classe supportent le schéma d'adressage IP correspondant.

© copyright Éditions Eyrolles

# Solutions de configuration

Configurer les protocoles à vecteur de distance sur les routeurs Cisco est une opération assez facile et répétitive. Les étapes de base sont toujours les suivantes, quel que soit le protocole choisi :

1. Définir le protocole de routage par la commande **router** <*protocole de routage*> en mode de configuration globale. Certains protocoles nécessitent que soit mentionné le numéro du *système autonome* dont la signification sera précisée dans la section consacrée au protocole IGRP. Cette commande nous met en mode de configuration routeur.
2. Spécifier les adresses réseau qui doivent être annoncées par le protocole de routage, au moyen de la commande **network** <*adresse IP du réseau*>. On peut mentionner n'importe quelle adresse réseau par cette commande, mais le routeur n'annoncera que les réseaux auxquels il est directement connecté. Autrement dit, pour qu'une adresse réseau correspondant au paramètre de la commande **network** soit annoncée par le routeur, il doit avoir au moins une interface active avec une adresse IP appartenant à ce réseau.

**REMARQUE** La commande **network** permet de spécifier uniquement une adresse réseau à classe, par exemple 10.0.0.0, 150.0.0.0, etc. Par contre, on ne peut pas préciser le sous-réseau, même s'il est configuré sur l'interface du routeur.

L'étape 2 demande quelques précisions. À la première commande **network**, toutes les interfaces dont les adresses IP appartiennent à l'adresse réseau renseigné en paramètre, sont enregistrées dans le processus de routage RIP du routeur. Celui-ci peut maintenant traiter les mises à jour reçues et aussi en envoyer à travers ces interfaces. Pour chaque nouvelle commande **network**, le routeur répète l'opération décrite ci-dessus. Les interfaces dont les adresses IP n'appartiennent à aucune adresse réseau donnée en argument à la commande **network**, seront exclues du processus de routage RIP. Le routeur n'enverra jamais de mises à jour ni ne tiendra compte de celles reçues *via* ces interfaces.

## Configuration des protocoles de routage à classe

Les routeurs Cisco disposent de deux protocoles de routage à classe, RIP (*Routing Information Protocol*) et IGRP (*Interior Gateway Routing Protocol*), basés sur l'algorithme à vecteur de distance. Leur fonctionnement est comparable, malgré les différences importantes qui sont les suivantes :

- Le protocole RIP est une norme de fait disponible chez tout constructeur d'hôte ou de routeur, tandis que le protocole IGRP dont Cisco est propriétaire, ne peut s'exécuter que sur les routeurs dotés du système IOS de Cisco.
- La métrique de RIP est représentée par un entier positif allant de 1 à 16 maximum (appelé « infini »). Le calcul de la métrique d'une route est basé sur le cumul du nombre de segments à traverser. La métrique de IGRP est plus complexe et différents facteurs interviennent dans son calcul tels que le débit, le délai, etc. En outre, le diamètre du réseau, c'est-à-dire le nombre de segments à traverser, y est aussi bien plus important.

© copyright Éditions Eyrolles

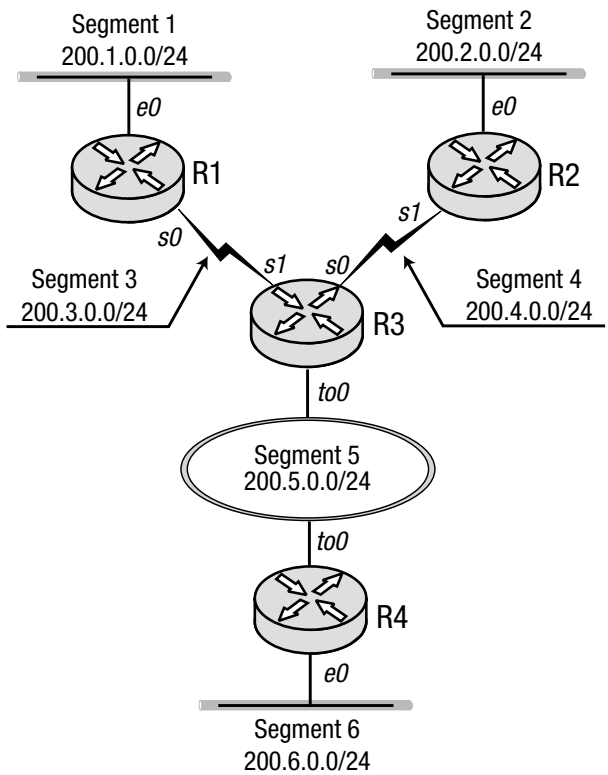
- Il ne peut y avoir qu'un seul processus de routage par routeur pour RIP, tandis que IGRP peut executer plusieurs processus de routage dans un même routeur, chacun servant un domaine différent.
- RIP peut annoncer des adresses individuelles telle que 10.1.0.1/32, ce qui n'est pas le cas de IGRP.

### Configuration de RIP

La configuration de base de RIP doit suivre les étapes définies au début de cette section « Solutions de configuration ». La commande utilisée à l'étape 1 est **router** avec <**rip**> en paramètre. Si on prend l'exemple de la figure 4.3, les listings 4.1 à 4.4 montrent les configurations des quatre routeurs concernés.

**Figure 4.3**

*Quatre routeurs configurés en protocole RIP pour connecter six segments du réseau.*




**Listing 4.1. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial0
 ip address 200.3.0.2 255.255.255.0

router rip
 network 200.1.0.0
 network 200.3.0.0
```

© copyright Éditions Eyrolles

**Listing 4.2. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial1
 ip address 200.4.0.2 255.255.255.0

router rip
 network 200.2.0.0
 network 200.4.0.0
```

**Listing 4.3. Configuration du routeur R3.**

```
interface Serial0
 ip address 200.4.0.1 255.255.255.0

interface Serial1
 ip address 200.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 network 200.3.0.0
 network 200.4.0.0
 network 200.5.0.0
```

**Listing 4.4. Configuration du routeur R4.**

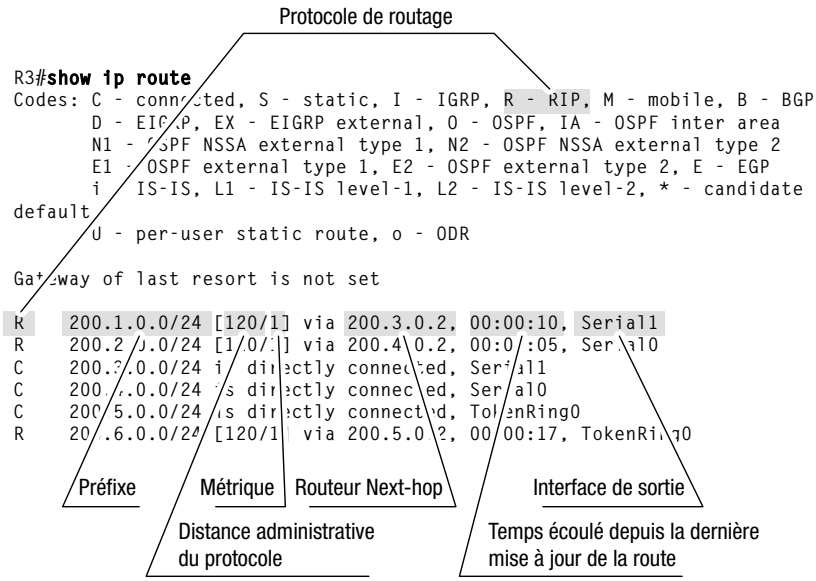
```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 network 200.5.0.0
 network 200.6.0.0
```

Prenons le cas du routeur R3, et examinons sa table de routage. Comme dans le cas du routage statique, nous pouvons l'afficher par la commande **show ip route** dont la sortie se trouve sur la figure 4.4.

Certains éléments de la table de routage ont été ignorés jusqu'à présent, car ils n'intervenaient pas dans les cas de routage statique et d'interfaces connectées. Maintenant que nous abordons le sujet du routage dynamique, nous devons étudier de plus près la table de routage.

© copyright Éditions Eyrolles

**Figure 4.4**
*Table de routage du routeur R3.*




Les points clefs à retenir de la sortie de la commande **show ip route**, sont les suivants :

- Les six premières lignes affichées sous la rubrique *codes* se rapportent aux abréviations employées dans la suite du listing ; par exemple, R pour RIP, I pour IGRP, etc. Comme ces lignes sont toujours les mêmes, nous n'aurons pas à les détailler chaque fois.
- La ligne suivante indique si un routeur par défaut est configuré. La « passerelle de dernier recours » (*gateway of last resort*) est le terme utilisé par Cisco pour désigner ce routeur.
- Les lignes restantes affichent le contenu de la table de routage proprement dite, avec chaque entrée composée d'un certain nombre de champs dont le premier est une lettre qui précise l'origine de l'information qui a permis d'établir la route. Par exempe, « R » pour RIP, ce qui signifie que la route à été apprise via ce protocole ; de même, « S » pour statique.
- Le champ suivant de l'entrée comporte le préfixe réseau et sa longueur. Le format d'affichage dépend de la version du système IOS de Cisco. Les plus récentes font précéder la longueur du préfixe par le caractère « / ». Les versions plus anciennes affichent le préfixe suivi du masque de sous réseau en notation décimale pointée (par exemple, adresse réseau 150.0.0.0 et masque de sous-réseau 255.255.0.0). Il est possible de modifier ce type d'affichage pour la durée de la session par la commande **terminal ip netmask-format bit-count** ; ou de façon permanente par la commande **ip netmask-format bit-count**, en mode de configuration d'interface ligne.
- Les deux champs suivants sont affichés entre parenthèses carrées, separés par le caractère « / » ; le premier concerne la distance administrative du protocole à l'origine de l'information de routage (dans le cas de RIP, la valeur est 120) ; le deuxième se rapporte à la métrique de la route (c'est-à-dire le nombre de sauts pour RIP).
- Le champ suivant indique le routeur de saut suivant.
- L'avant-dernier champ affiche le temps écoulé depuis la dernière mise à jour de cette route.
- Le dernier champ identifie l'interface de sortie.

© copyright Éditions Eyrolles

La sortie de la commande **show ip route** que nous venons de passer en revue, bien qu'incomplète, suffit à notre propos. Nous aurons à y revenir au cours de notre progression dans cet ouvrage.

Le calcul de la métrique de routes dans RIP est très simple. Il considère que chaque segment a un coût de 1, et le cumul du nombre de segments à traverser pour atteindre une destination est le coût total attribué à la route pointant vers celle-ci. Dans l'exemple de la figure 4.3, toutes les routes apprises par RIP ont un coût de 1, ce qui s'explique parfaitement : aucune d'entre elles n'est à plus d'un saut du routeur R3. Mais si nous affichons la table de routage du routeur R4, les métriques varient légèrement plus.

**Listing 4.5. Table de routage du routeur R4.**

```
R4#show ip route
...
R   200.1.0.0 [120/2] via 200.5.0.1,00:00:21, TokenRing0
R   200.2.0.0 [120/2] via 200.5.0.1,00:00:21, TokenRing0
R   200.3.0.0 [120/1] via 200.5.0.1,00:00:21, TokenRing0
R   200.4.0.0 [120/1] via 200.5.0.1,00:00:22, TokenRing0
C   200.5.0.0 is directly connected,TokenRing0
C   200.6.0.0 is directly connected,Ethernet0
```

Bien évidemment, les métriques dans notre exemple ne sont pas très diversifiées, du fait que le réseau a un diamètre plutôt réduit. En pratique, elles peuvent prendre des valeurs bien plus grandes.

Nous allons maintenant utiliser la commande **debug ip rip** sur le routeur R3 pour analyser la transaction de RIP lors de ses mises à jour, telle qu'elle est affichée sur le listing 4.6.

**Listing 4.6. Sortie de la commande `debug ip rip` entrée sur le routeur R3.**

```
R3#debug ip rip
RIP protocol debugging is on
R3#
RIP: received v1 update from 200.3.0.2 on Serial1
     200.1.0.0 in 1 hops
RIP: received v1 update from 200.4.0.2 on Serial0
     200.2.0.0 in 1 hops
RIP: received v1 update from 200.5.0.2 on TokenRing0
     200.6.0.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Serial0
(200.4.0.1)
     network 200.1.0.0, metric 2
     network 200.3.0.0, metric 1
     network 200.5.0.0, metric 1
     network 200.6.0.0, metric 2
RIP: sending v1 update to 255.255.255.255 via Serial1
(200.3.0.1)
     network 200.2.0.0, metric 2
     network 200.4.0.0, metric 1
     network 200.5.0.0, metric 1
     network 200.6.0.0, metric 2
RIP: sending v1 update to 255.255.255.255 via TokenRing0
```

© copyright Éditions Eyrolles

```
(200.5.0.1)
     network 200.1.0.0, metric 2
     network 200.2.0.0, metric 2
     network 200.3.0.0, metric 1
     network 200.4.0.0, metric 1
```

**AVERTISSEMENT** Il faut éviter d'utiliser la commande **debug ip rip** ou toutes celles qui demandent un traitement intensif sur des routeurs d'un réseau opérationnel. C'est particulièrement critique quand ils reçoivent bon nombre de mises à jour de routage. Les commandes en question ne peuvent que bloquer l'accès à leur terminal, tout en pénalisant leur performance. Dans ce cas, seul un redémarrage électrique peut les débloquer.

La plupart des commandes **debug** conviennent uniquement dans un environnement d'essai en laboratoire. Leur utilisation dans cet ouvrage sert à l'illustrer ce qui se passe dans un routeur dans certains cas difficiles à expliquer autrement.

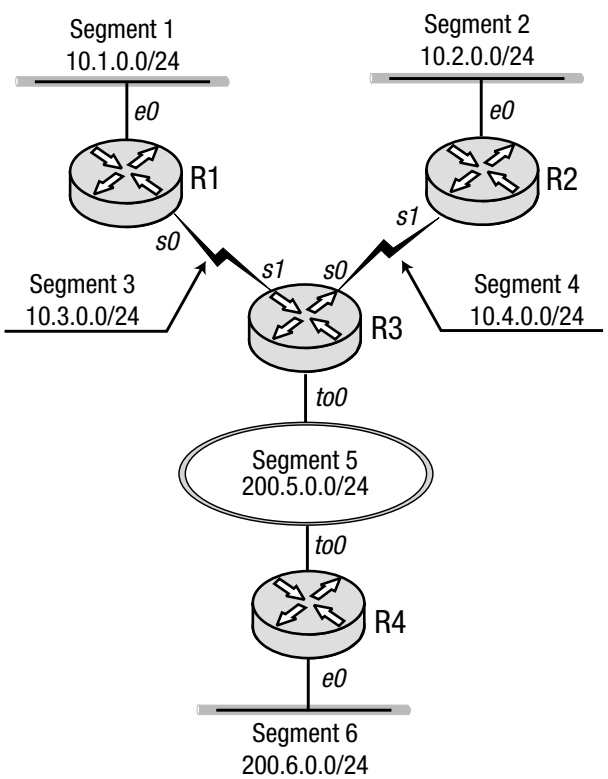
La sortie du listing 4.6 est suffisamment explicite par elle-même. Chaque mise à jour commence par une ligne d'envoi ou de réception RIP (*sending* ou *received*), suivie par les entrées des préfixes réseau ou sous-réseau annoncés. Un message RIP peut contenir jusqu'à 25 préfixes.

Un autre point à noter, c'est l'absence de routes avec la métrique de valeur infinie (16 pour RIP), qui indique que la version de l'IOS de Cisco applique la règle simple du clivage d'horizon, sans l'antidote.

Jusqu'à présent, nous avons utilisé des adresses réseau de classe C avec le masque de sous-réseau par défaut. Il s'agit d'une configuration minimale, et tout s'est déroulé comme prévu, du point de vue du routeur. Remplaçons maintenant certaines des adresses de classe C avec des sous-réseaux appartenant à la classe A 10.0.0.0. La figure 4.5 illustre ce nouveau cas.

**Figure 4.5**

*Segments 1 à 4 configurés en sous-réseaux de 10.0.0.0, segments 5 et 6 inchangés.*




© copyright Éditions Eyrolles

Les listings 4.7 à 4.9 montrent les modifications apportées à la configuration des routeurs concernés par le changement d'adressage.

**Listing 4.7. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router rip
 network 10.0.0.0
```

**Listing 4.8. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router rip
 network 10.0.0.0
```

**Listing 4.9. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 network 10.0.0.0
 network 200.5.0.0
```

Comme prévu, les routeurs R1 et R2 n'ont qu'une seule déclaration par la commande **network**, correspondant à l'ensemble du réseau de classe A (10.0.0.0). Quant au routeur R3, il possède deux déclarations par la même commande, la première pour le réseau 10.0.0.0 et la deuxième pour le réseau 200.5.0.0.

Le listing 4.10 montre la table de routage du routeur R1.

**Listing 4.10. Table de routage du routeur R1.**

```
R1#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/2] via 10.3.0.1, 00:00:23, Serial0
C       10.3.0.0 is directly connected, Serial0
C       10.1.0.0 is directly connected, Ethernet0
```

© copyright Éditions Eyrolles

```
R       10.4.0.0 [120/1] via 10.3.0.1, 00:00:23, Serial0
R    200.5.0.0/24 [120/1] via 10.3.0.1, 00:00:23, Serial0
R    200.6.0.0/24 [120/2] via 10.3.0.1, 00:00:23, Serial0
```

La sortie de la commande **show ip route** montre que le réseau 10.0.0.0 est découpé en sous-réseaux et que le routeur R1 en connaît quatre (première ligne), dont deux directement connectés (lignes commençant par « C »), et deux autres appris *via* RIP (lignes commençant par « R »). Les deux dernières lignes affichent les réseaux, 200.5.0.0 et 200.6.0.0, appris par le routeur, également *via* RIP.

Le listing 4.11 montre la table de routage du routeur R4.

**Listing 4.11. Table de routage du routeur R4.**

```
R4#show ip route
...
R    10.0.0.0 [120/1] via 200.5.0.1, 00:00:08, TokenRing0
C    200.5.0.0 is directly connected, TokenRing0
C    200.6.0.0 is directly connected, Ethernet0
```

Nous constatons que le routeur R4 a un problème. Non seulement il ne sait pas que le réseau 10.0.0.0 est découpé en sous-réseaux, mais il pense aussi que ce réseau tout entier n'est distant que d'un saut. Pourquoi ?

La réponse est à chercher dans le routeur R3 qui est le seul qui sépare le routeur R4 du réseau 10.0.0.0. Pour comprendre ce qui se passe dans le routeur R3, examinons d'abord sa table de routage (cf. listing 4.12) ; et affichons les messages de mise à jour qu'il génère en entrant la commande **debug ip rip** (cf. listing 4.13).

**Listing 4.12. Table de routage du routeur R3.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:04, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:02, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:04, TokenRing0
```

**Listing 4.13. Sortie de la commande `debug ip rip`.**

```
R3#debug ip rip
RIP protocol debugging is on
R3#
RIP: sending v1 update to 255.255.255.255 via Serial0 (10.4.0.1)
     subnet  10.3.0.0, metric 1
     subnet  10.1.0.0, metric 2
     network 200.5.0.0, metric 1
     network 200.6.0.0, metric 2
```

© copyright Éditions Eyrolles

```
RIP: sending v1 update to 255.255.255.255 via Serial1 (10.3.0.1)
     subnet  10.2.0.0, metric 2
     subnet  10.4.0.0, metric 1
     network 200.5.0.0, metric 1
     network 200.6.0.0, metric 2
RIP: sending v1 update to 255.255.255.255 via TokenRing0 (200.5.0.1)
     network 10.0.0.0, metric 1
```

Le routeur R3 sait visiblement que le réseau 10.0.0.0 est découpé en quatre sous-réseaux. Mais dans la transaction affichée sur le listing 4.13, on constate qu'il n'envoie qu'une seule mise à jour pour tout ce réseau, avec une métrique à 1, à travers l'interface de Token Ring. Par contre, quand il s'agit des interfaces ligne S0 et S1, les messages de mise à jour envoyés sont complets.

Pour expliquer ce comportement, nous devons nous rappeler que RIP, un protocole de routage à classe, ne transmet pas un préfixe réseau en même temps que le masque de sous-réseau associé, lors de l'envoi de ses mises à jour. En réception de celles-ci, il ne peut par conséquent que « deviner » le masque de sous-réseau d'une destination particulière. Pour ce faire, il applique le masque de sous-réseau de l'interface à travers laquelle il reçoit les mises à jour pour les routes à destination des sous-réseaux appartenant au même réseau que cette interface. Au cas où certaines adresses IP n'en feraient pas partie, elles sont considérées soit comme des adresses réseau à classe, soit comme des adresses d'hôtes quand les bits correspondants sont positionnés. De même, le routeur n'annonce des sous-réseaux *via* une interface que si ceux-ci appartiennent au même réseau que l'adresse IP de cette dernière. Sinon, les sous-réseaux sont remplacés par les adresses réseau auxquelles ils appartiennent, avant d'être diffusés *via* l'interface, avec une métrique de 1. Ce procédé s'appelle auto-agrégation (*auto-summarization*).

Le protocole RIP n'est pas le seul à utiliser l'auto-agrégation ; tout protocole à classe y a recours chaque fois que les mises à jour de routage ont à traverser les frontières réseau correspondantes. Cela vient du fait que les protocoles de routage à classe, tel RIP, ne transmettent pas le masque de sous-réseau avec le préfixe réseau. L'auto-agrégation produit l'effet de bord suivant :

*Les protocoles de routage dynamique à classe ne peuvent fonctionner correctement que si l'espace d'adressage IP demeure continu pour chaque adresse réseau à classe.*

Autrement dit, deux groupes de sous-réseaux appartenant à un réseau A ne peuvent être séparés par une adresse IP appartenant à un autre réseau que celui-ci. Ce qui signifie que dans l'exemple de la figure 4.6, où l'on voit deux groupes de sous-réseaux appartenant au réseau 10.0.0.0 séparés par une adresse IP du réseau 200.1.0.0/24, les hôtes situés sur les segments de part et d'autre, ne pourront communiquer. La solution à ce problème consiste à relier parallèlement par une autre paire de routeurs, ces deux segments avec une adresse de sous-réseau 10.255.1.0, appartenant au réseau 10.0.0.0, comme dans la figure 4.7.

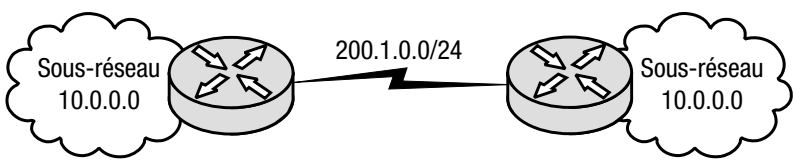



**Figure 4.6**

*Groupes de sous-réseaux appartenant au réseau 10.0.0.0, séparés par une ligne d'adresse réseau 200.1.0.0.*

© copyright Éditions Eyrolles

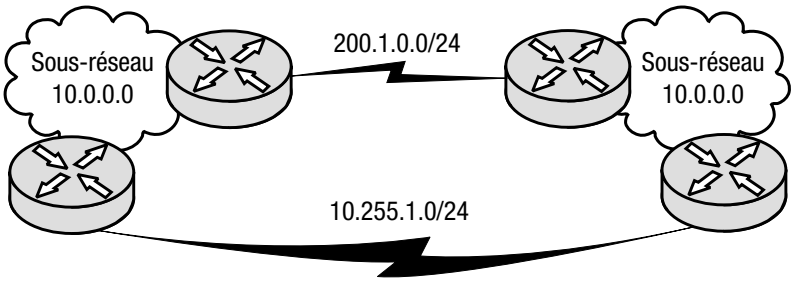



**Figure 4.7**
*Une ligne parallèle d'adresse de sous-réseau appartenant au réseau 10.0.0.0 résout le problème d'interconnexion.*

### Utilisation d'adresses d'hôte individuelles avec RIP

Comme indiqué précédemment, le protocole RIP permet de transmettre dans ses mises à jour de routage, des adresses individuelles d'hôte. La même règle de continuité de l'espace d'adressage s'applique aussi à ce cas comme à celui des sous-réseaux, pour que l'interconnexion se fasse correctement. Une fois cette condition remplie, la configuration s'effectue selon les étapes suivantes :

1. Définir une interface de rebouclage (*loopback*) par la commande **interface loopback** <*numéro*>, en mode de configuration globale.
2. Assigner une adresse IP à l'interface de rebouclage par la commande **ip address** <*adresse IP*> 255.255.255.255.

Si l'adresse IP assignée à l'interface de rebouclage appartient à l'un des réseaux définis par la commande **network**, en mode de configuration globale, RIP annonce cette adresse en tant qu'adresse individuelle d'hôte.

Procédons à cette assignation sur les routeurs R1, R2 et R3, avec les adresses IP 10.0.0.1/32, 10.0.0.2/32 et 10.0.0.3/32, respectivement. Nous voyons apparaître ces adresses individuelles d'hôte dans leur table de routage comme celle du routeur R3 qui est affichée sur le listing 4.14.

**Listing 4.14. Table de routage du routeur R3.**

```
R3#show ip route
...
     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
R       10.0.0.2/32 [120/1] via 10.4.0.2, 00:00:17, Serial0
R       10.2.0.0/24 [120/1] via 10.4.0.2, 00:00:17, Serial0
C       10.0.0.3/32 is directly connected, Loopback0
C       10.3.0.0/24 is directly connected, Serial1
R       10.0.0.1/32 [120/1] via 10.3.0.2, 00:00:11, Serial1
R       10.1.0.0/24 [120/1] via 10.3.0.2, 00:00:11, Serial1
C       10.4.0.0/24 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:01, TokenRing0
```

Si nous examinons le contenu des messages de mise à jour de RIP, par la commande **debug ip rip**, nous y voyons effectivement les routes individuelles des hôtes concernés, comme sur le listing 4.15 (mises en italique).

© copyright Éditions Eyrolles

**Listing 4.15. Sortie de la commande `debug ip rip` où on voit les adresses individuelles d'hôte annoncées dans les mises à jour du routeur R3.**

```
R3#debug ip rip
...
RIP: sending v1 update to 255.255.255.255 via Serial0
(10.4.0.1)
     host    10.0.0.3, metric 1
     subnet  10.3.0.0, metric 1
     host    10.0.0.1, metric 2
     subnet  10.1.0.0, metric 2
     network 200.5.0.0, metric 1
     network 200.6.0.0, metric 2
...
```

### Configuration de RIP pour l'annonce de la route par défaut

Pour qu'un routeur annonce la route par défaut, c'est-à-dire 0.0.0.0/0, la commande **default-information originate** doit être utilisée en mode de configuration routeur (**router rip)**.

Par exemple, supposons que le routeur R4 sur la figure 4.5 soit aussi connecté à l'Internet. Si nous voulons en autoriser l'accès à tous les hôtes de tous les segments, il devient utile de contraindre le routeur R4 à diffuser la route par défaut. Le listing 4.16 montre la configuration modifiée du routeur R4.

Nous voyons apparaître, dans la table de routage du routeur R3 affiché sur le listing 4.17, une entrée correspondant à la route par défaut portant la mention « R » pour indiquer qu'elle a été apprise *via* RIP, suivie d'un astérisque pour préciser qu'elle est *candidate* pour la route par défaut.

**Listing 4.16. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 network 200.5.0.0
 network 200.6.0.0
 default-information originate
```

Le listing 4.17 affiche la table de routage du routeur R3.

**Listing 4.17. Table de routage du routeur R3.**

```
R3#show ip route
10.0.0.0/24 is subnetted, 4 subnets
...
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:12, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:27, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:09, TokenRing0
R*   0.0.0.0/0 [120/1] via 200.5.0.2, 00:00:09, TokenRing0
```

© copyright Éditions Eyrolles

**REMARQUE** Bien que conseillée, la commande **ip classless** n'est pas indispensable pour la configuration décrite ci-dessus. Sans cette commande, le routeur suppose que s'il est connecté à un sous-réseau d'une certaine adresse réseau à classe, il saura aussi communiquer avec ce réseau à classe tout entier. Ceci est valable dans le cas des protocoles de routage dynamique à classe, car leur fonctionnement nécessite que l'espace d'adressage soit continu.

### Configuration de RIP avec des adresses IP secondaires

Le protocole RIP annonce les adresses IP secondaires si elles sont configurées sur l'une des interfaces du routeur, et si les réseaux auxquels appartiennent ces adresses, sont rattachés au processus de routage RIP, par la commande **network**, en mode de configuration routeur.

Quand le routeur doit générer et envoyer des mises à jour de routage à travers une interface configurée avec des adresses IP secondaires, il applique les règles suivantes :

- Il envoie de multiples copies de la même mise à jour à travers une interface configurée avec des adresses IP secondaires. Le nombre de copies équivaut à celui des adresses réseaux à classe configurées sur l'interface, que ces réseaux soient découpés en sous-réseaux ou non. Chaque copie est envoyée comme si elle provenait de l'adresse réseau correspondante.
- Il prélève la première adresse IP parmi celles configurées sur l'interface pour chaque réseau. Les adresses IP sont traitées dans l'ordre où elles apparaissent dans la configuration. L'adresse ainsi prélevée est utilisée comme source du datagramme contenant la copie correspondante de la mise à jour de routage.
- Aucune adresse réseau ou sous-réseau, qu'elle soit secondaire ou primaire, ne se trouve dans les copies de mise à jour de routage. Pour une mise à jour donnée, toutes les copies sont identiques.

Ces règles peuvent paraître quelque peu compliquées. Pour mieux les comprendre, prenons l'exemple de la figure 4.5, en modifiant la configuration des routeurs R3 et R4, pour y ajouter des adresses IP secondaires (cf. listings 4.18 et 4.19).

**Listing 4.18. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 172.16.1.20 255.255.255.0 secondary
 ip address 172.16.1.15 255.255.255.0 secondary
 ip address 210.1.0.1 255.255.255.0 secondary
 ip address 172.16.10.20 255.255.255.0 secondary
 ip address 172.16.10.15 255.255.255.0 secondary
 ip address 172.16.10.10 255.255.255.0 secondary
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 network 10.0.0.0
 network 200.5.0.0
 network 172.16.0.0
 network 210.1.0.0
```

© copyright Éditions Eyrolles

**Listing 4.19. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 172.16.1.120 255.255.255.0 secondary
 ip address 172.16.1.115 255.255.255.0 secondary
 ip address 210.1.0.2 255.255.255.0 secondary
 ip address 172.16.10.120 255.255.255.0 secondary
 ip address 172.16.10.115 255.255.255.0 secondary
 ip address 172.16.10.110 255.255.255.0 secondary
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 network 200.5.0.0
 network 200.6.0.0
 network 172.16.0.0
 network 210.1.0.0
```

Les interfaces Token Ring sur les deux routeurs ont maintenant un certain nombre d'adresses IP secondaires, et celles-ci, comme les adresses IP primaires, appartiennent à trois réseaux dont les adresses sont : 172.16.0.0/16, 200.5.0.0/24 et 210.1.0.0/24. Par conséquent, trois copies de la même mise à jour de routage doivent être envoyées par l'interface Token Ring de chaque routeur.

Vérifions que c'est bien le cas, par la commande **debug ip rip** qui affiche le contenu des messages de mise à jour de routage envoyés par RIP dont on trouve un extrait sur le listing 4.20, pour le routeur R4.

**Listing 4.20. Sortie de la commande `debug ip rip` sur le routeur R4.**

```
R4#debug ip rip
RIP protocol debugging is on
R4#
RIP: sending update to 255.255.255.255 via Ethernet0
(200.6.0.1)
     network 10.0.0.0, metric 2
     network 172.16.0.0, metric 1
     network 200.5.0.0, metric 1
     network 210.1.0.0, metric 1
RIP: sending update to 255.255.255.255 via TokenRing0
(200.5.0.2)
     network 200.6.0.0, metric 1
RIP: sending update to 255.255.255.255 via TokenRing0
(172.16.1.120)
     network 200.6.0.0, metric 1
RIP: sending update to 255.255.255.255 via TokenRing0
(210.1.0.2)
     network 200.6.0.0, metric 1
```

© copyright Éditions Eyrolles

```
RIP: received update from 200.5.0.1 on TokenRing0
     10.0.0.0 in 1 hops
RIP: received update from 172.16.1.20 on TokenRing0
     10.0.0.0 in 1 hops
RIP: received update from 210.1.0.1 on TokenRing0
     10.0.0.0 in 1 hops
```

La sortie de la commande **debug ip rip** du listing 4.20 confirme que trois copies identiques de la mise à jour de routage ont été envoyées à travers l'interface Token Ring par le routeur R4 (*idem* pour R3), démontrant ainsi la mise en application des règles énoncées plus haut. Aucune de ces copies ne contient une adresse de réseau ou de sous-réseau configurée sur l'interface Token Ring. Les adresses IP utilisées comme sources de ces mises à jour pour le réseau 176.16.0.0 sont : 176.16.1.20 pour le routeur R3 et 176.16.1.120 pour le routeur R4. Ces adresses IP apparaissent effectivement en premier, configurées sur l'interface de chacun des routeurs concernés.

**AVERTISSEMENT** Bien que le réseau 176.16.0.0 soit découpé en deux sous-réseaux, seule l'adresse IP appartenant au premier (176.16.1.0), a été utilisée comme source de la copie de mise à jour. Les hôtes résidant sur d'autres sous-réseaux qui sont à l'écoute de RIP, recevront des mises à jour avec une adresse IP source incorrecte, et devront donc les ignorer ; car ils ne sauront pas communiquer avec le routeur d'annonce des préfixes réseau pour lesquels celui-ci devrait servir de routeur de saut suivant pour en établir une route. (La raison pour laquelle les hôtes sont susceptibles de recevoir les mises à jour de RIP, vient du fait qu'elles sont transmises à l'adresse de diffusion générale 255.255.255.255).

Les tables de routage des routeurs R3 et R4 se trouvent modifiées par la présence dans leur configuration d'adresses IP secondaires (cf. listings 4.21 et 4.22)

**Listing 4.21. Table de routage du routeur R3.**

```
R3#show ip route
...
   10.0.0.0/24 is subnetted, 4 subnets
R     10.2.0.0 [120/1] via 10.4.0.2, 00:00:15, Serial0
C     10.3.0.0 is directly connected, Serial1
R     10.1.0.0 [120/1] via 10.3.0.2, 00:00:06, Serial1
C     10.4.0.0 is directly connected, Serial0
   172.16.0.0/24 is subnetted, 2 subnets
C     172.16.10.0 is directly connected, TokenRing0
C     172.16.1.0 is directly connected, TokenRing0
C  200.5.0.0/24 is directly connected, TokenRing0
R  200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:06, TokenRing0
                [120/1] via 172.16.1.120, 00:00:06, TokenRing0
                [120/1] via 210.1.0.2, 00:00:06, TokenRing0
C  210.1.0.0/24 is directly connected, TokenRing0
```

**Listing 4.22. Table de routage du routeur R4.**

```
R4#show ip route
...
R    10.0.0.0 [120/1] via 200.5.0.1, 00:00:13, TokenRing0
              [120/1] via 172.16.1.20, 00:00:13, TokenRing0
```

© copyright Éditions Eyrolles

```
                [120/1] via 210.1.0.1, 00:00:13, TokenRing0
      172.16.0.0 255.255.255.0 is subnetted, 2 subnets
C        172.16.10.0 is directly connected, TokenRing0
C        172.16.1.0 is directly connected, TokenRing0
C     200.5.0.0 is directly connected, TokenRing0
C     200.6.0.0 is directly connected, Ethernet0
C     210.1.0.0 is directly connected, TokenRing0
```

Les lignes qui indiquent que le routeur pratique le partage de charge sur chemins multiples ont été mises en italique sur les listings 4.21 et 4.22, pour mieux les faire ressortir. Remarquons cependant que toutes ces routes pointent vers la même interface Token Ring qui, à elle seule, supporte tout le trafic destiné au routeur de saut suivant. On peut en conclure que ce partage de charge est factice, du fait que les routeurs reçoivent plusieurs mises à jour de routage annonçant les mêmes préfixes réseau (10.0.0.0 et 200.6.0.0) avec la même métrique. Nous verrons sous peu comment l'implémentation de RIP dans l'IOS de Cisco affecte le partage de charge. Contentons-nous simplement de constater pour l'instant que les conditions requises étaient remplies, même si le résultat n'a fait que semer la confusion dans l'esprit de l'administrateur réseau.

**AVERTISSEMENT** Si des adresses IP secondaires sont utilisées avec les protocoles de routage dynamique, tous les routeurs connectés au même segment, doivent avoir les mêmes adresses réseau et sous-réseau, configurées sur leurs interfaces respectives. Tout manquement à cette règle risque de mener à des boucles de routage.

En outre, l'ordre dans lequel apparaissent les sous-réseaux d'un même réseau dans la configuration des interfaces est lui aussi primordial. Seule la première adresse IP pour chaque adresse réseau à classe est utilisée comme adresse source dans les datagrammes transportant les copies de mises à jour de routage censées provenir de ce réseau. Par conséquent, si cet ordre change d'un routeur à l'autre, les mises à jour générées par ces routeurs risquent de traverser des sous-réseaux différents.

Vous voilà maintenant dissuadé d'utiliser des adresses IP secondaires dans un réseau opérationnel, sachant que les routeurs ainsi configurés avec le protocole de routage dynamique peuvent avoir un comportement quelque peu imprévisible. Les règles qu'ils appliquent pour les mises à jour routage en sont d'autant compliquées, et leur configuration en devient inutilement lourde et fastidieuse. Les seuls cas qui peuvent justifier l'utilisation d'adresses IP secondaires sont soit les migrations, soit le besoin urgent d'allouer des adresses IP supplémentaires.

### Inhibition par RIP de l'envoi des mises à jour de routage sur une interface

Pour empêcher RIP de diffuser les mises à jour à travers une interface, la commande **passive-interface** <*numéro d'interface*> doit être utilisée, en passant en mode de configuration routeur (commande **routeur rip**). Cette commande va inhiber les mises à jour de RIP à travers les interfaces qui lui sont données en paramètre, même si les adresses IP de celles-ci appartiennent aux adresses réseau définies par la commande **network**. Mais RIP continuera à traiter les mises à jour entrantes sur ces interfaces.

Entrons la commande **passive-interface** sur l'interface Token Ring du routeur R4 pour en observer les effets. Le listing 4.23 en montre la configuration. Les autres routeurs sont configurés selon le schéma de la figure 4.5, sans les interfaces de rebouclage, ni les adresses IP secondaires.

© copyright Éditions Eyrolles

**Listing 4.23. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 passive-interface TokenRing0
 network 200.5.0.0
 network 200.6.0.0
```

Comme on le voit sur le listing 4.24, la table de routage du routeur R4 n'a pas changé, ce qui était prévu, sachant la fonctionnalité de la commande **passive-interface**.

**Listing 4.24. Table de routage du routeur R4.**

```
R4#show ip route
...
R    10.0.0.0 [120/1] via 200.5.0.1, 00:00:09, TokenRing0
C    200.5.0.0 is directly connected, TokenRing0
C    200.6.0.0 is directly connected, Ethernet0
```

Par contre, le listing 4.25 de la table de routage du routeur R3 montre qu'il n'est plus en mesure de voir l'adresse réseau du segment Ethernet situé derrière le routeur R4.

**Listing 4.25. Table de routage du routeurR3.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:02, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:03, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
```

Nous pouvons tracer les mises à jour envoyées par le routeur R4 par la commande **debug ip rip** (cf. listing 4.26).

**Listing 4.26. Sortie de la commande `debug ip rip` sur le routeur R4.**

```
R4#debug ip rip
RIP protocol debugging is on
R4#
RIP: received update from 200.5.0.1 on TokenRing0
     10.0.0.0 in 1 hops
RIP: sending update to 255.255.255.255 via Ethernet0
(200.6.0.1)
     network 10.0.0.0, metric 2
     network 200.5.0.0, metric 1
```

© copyright Éditions Eyrolles

```
RIP: received update from 200.5.0.1 on TokenRing0
     10.0.0.0 in 1 hops
RIP: sending update to 255.255.255.255 via Ethernet0
(200.6.0.1)
     network 10.0.0.0, metric 2
     network 200.5.0.0, metric 1
```

Comme on pouvait s'y attendre, le routeur R4 ne diffuse plus de mises à jour *via* l'interface Token Ring. Ce qui montre que la commande **passive-interface** a pris effet sur cette interface. On peut aussi l'utiliser sur des hôtes si on veut qu'ils soient uniquement à l'écoute des messages de mise à jour RIP, sans en émettre, pour éviter de charger le segment réseau avec un trafic de diffusion inutile. Par exemple, dans le cas des stations de travail sous Unix, qui exécutent le *daemon* `routed` (la tâche de fond RIP, version BSD), on peut mettre leur interface réseau en mode passif, sauf indication contraire.

**REMARQUE** Même si l'hôte qui exécute la tâche RIP ne possède qu'une seule interface, il peut néanmoins générer des messages de mise à jour d'antidote, chargeant ainsi inutilement le segment auquel il est connecté, avec du trafic RIP.

Le trafic inutile est surtout dû au caractère de diffusion générale de RIP. Tout trafic destiné à l'adresse de diffusion générale, quel que soit le protocole auquel il est destiné, est reçu par tous les hôtes résidant sur un segment, provoquant ainsi un traitement superflu sur ces machines, et pénalisant leur performance. Il est donc important de réduire au maximum le trafic de diffusion générale.

Le mode passif peut s'avérer nécessaire quand un segment ne comprend qu'un seul routeur et des hôtes non habilités à recevoir ses mises à jour RIP. Il serait alors judicieux de mettre cette interface du routeur en écoute passive.

### Envoi de mises à jour monodestinataire par RIP

Si nous examinons la sortie de la commande **debug ip rip**, nous pouvons constater que toutes les mises à jour sont destinées à l'adresse de diffusion générale, 255.255.255.255. Dans certains cas, cités ci-après, il est possible d'envoyer les mises à jour à une adresse monodestinataire (*unicast*). Le système IOS de Cisco nous en donne les moyens par la commande **neighbor** <*adresse IP*> en mode de configuration routeur (**router rip**). Les mises à jour sont ainsi acheminées à l'adresse IP donnée en argument.

**REMARQUE** La commande **neighbor** n'a d'utilité que si l'interface de l'adresse IP concernée est en mode passif. Sinon, le routeur continue à envoyer aussi les mises à jour à l'adresse de diffusion générale sur cette interface.

La commande **passive-interface** tokenring 0 sur le routeur R4 a inhibé l'envoi de ses mises à jour sur cette interface, comme nous l'avons vu dans la section précédente. De ce fait, le routeur R3 n'avait plus connaissance du réseau Ethernet 200.6.0.0/24. Si nous introduisons maintenant l'adresse IP de ce routeur par la commande **neighbor** sur le routeur R4, les mises à jour envoyées par ce dernier vont lui permettre à nouveau d'avoir connaissance du réseau en question. Le listing 4.27 montre cette nouvelle configuration du routeur R4. Et dans le listing 4.28, la table de routage du routeur R3 fait de nouveau apparaître le réseau 200.6.0.0.

© copyright Éditions Eyrolles

**Listing 4.27. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 passive-interface TokenRing0
 neighbor 200.5.0.1
 network 200.5.0.0
 network 200.6.0.0
```

**Listing 4.28. Table de routage du routeur R3.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:12, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:05, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:11, TokenRing0
```

Sur le listing 4.29 on voit par la commande **debug ip rip** que les mises à jour sont de nouveau envoyées par le routeur R4 sur l'interface Token Ring, mais cette fois-ci, uniquement à l'adresse monodestinataire du routeur R3 (ligne en italique).

**Listing 4.29. Sortie de la commande `debug ip rip`.**

```
R4#debug ip rip
...
RIP: sending update to 255.255.255.255 via Ethernet0 (200.6.0.1)
     network 10.0.0.0, metric 2
     network 200.5.0.0, metric 1
RIP: sending update to 200.5.0.1 via TokenRing0 (200.5.0.2)
     network 200.6.0.0, metric 1
```

### Discrimination en entrée des mises à jour de routage

Le système IOS de Cisco procure un moyen de sélection ou même d'exclusion complète des mises à jour de routage en provenance de certaines sources. Cela consiste à assigner une valeur de distance administrative pour les sources de ces mises à jour.

Pour assigner une nouvelle valeur de distance administrative pour les sources de mises à jour entrantes, la commande **distance** <*valeur de distance*> [<*adresse IP source/masque générique*>] doit être utilisée, en mode de configuration routeur (**router rip**). La nouvelle distance administrative s'appliquera à toutes les routes pointant vers les préfixes réseau annoncés par la source dont l'adresse IP correspond au deuxième paramètre de la commande. Si ce paramètre (optionnel) est omis, la nouvelle distance s'appliquera à toutes les mises à jour reçues,

© copyright Éditions Eyrolles

quelle que soit leur origine. Si on assigne à cette distance une valeur de 255, les routes diffusées par la source correspondante, ne seront jamais inscrites dans la table de routage, mais tout simplement ignorées.

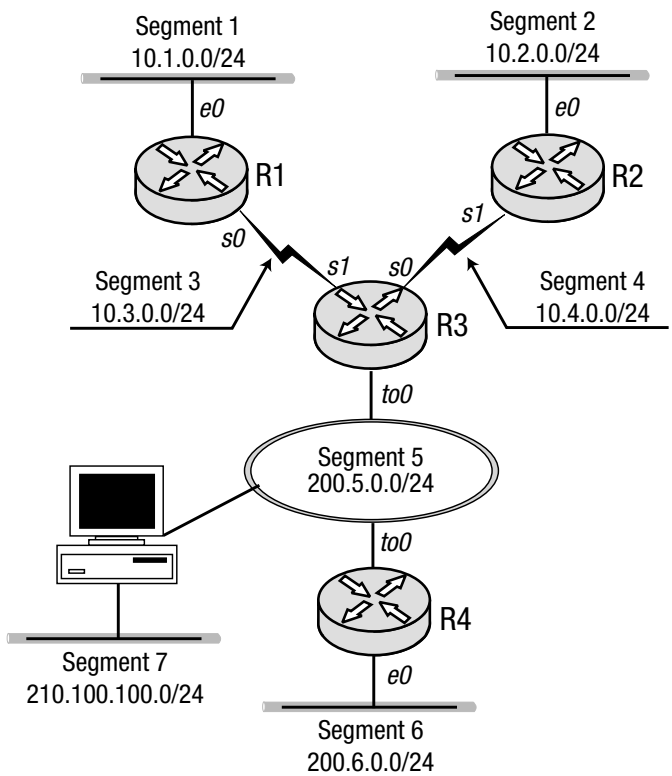
**REMARQUE** Dans le vocabulaire de Cisco, le « masque » générique (*wild card*) appliqué à l'adresse IP ne doit pas être pris dans son acception habituelle, mais dans le sens spécifique à Cisco, comme pour les listes d'accès. À savoir, tout bit positionné à 1 dans le masque indique que le bit correspondant dans l'adresse IP, doit être ignoré. En fait, l'application du masque à une adresse IP, dans ce cas, a l'effet inverse de celui du masque habituel appliqué aux adresses IP pour en extraire l'adresse réseau ou sous-réseau.

**ASTUCE** Plusieurs commandes **distance** peuvent être utilisées pour tenir compte du niveau de crédibilité accordé à chaque source des mises à jour de routage.

Supposons que nous ayons ajouté une source supplémentaire pour l'origine des mises à jour RIP, au schéma du réseau de la figure 4.5 qui est reproduit modifié, en 4.8. Celle-ci concerne l'installation sur le segment 7 d'un hôte multidomicilié (*multihomed*) qui exécute aussi la tâche RIP.

Supposons que le segment en question ne relève pas de l'infrastructure du réseau opérationnel et que nous voulions ignorer les mises à jour qui l'ont pour origine. Si nous étendons cette restriction à tout le segment 5, nous pouvons ignorer les mises à jour de toute source étrangère qui résiderait sur ce segment. Pour mieux illustrer notre exemple, nous allons aussi changer la valeur de distance des mises à jour émises réciproquement par les routeurs R3 et R4 en la faisant passer de 120 (valeur par défaut) à 200.

**Figure 4.8**
*Mises à jour de routage en provenance d'une source étrangère au réseau opérationnel.*




© copyright Éditions Eyrolles

Les listings 4.30 et 4.31 montrent l'état actuel des tables de routage des routeurs R3 et R4.

**Listing 4.30. Table de routage du routeur R3.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:18, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:12, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:24, TokenRing0
R    210.100.100.0/24 [120/2] via 200.5.0.15, 00:00:24, TokenRing0
```

**Listing 4.31. Table de routage du routeur R4.**

```
R4#show ip route
...
R    10.0.0.0 [120/1] via 200.5.0.1, 00:00:23, TokenRing0
C    200.5.0.0 is directly connected, TokenRing0
C    200.6.0.0 is directly connected, Ethernet0
R    210.100.100.0 [120/2] via 200.5.0.15, 00:00:24, TokenRing0
```

**REMARQUE** Le routeur R4, bien que configuré avec les commandes **passive-interface** tokenring 0 et **neighbor** 200.5.0.1, accepte toujours les mises à jour en provenance de l'hôte H1. Ces commandes ne peuvent donc être utilisées pour contraindre le routeur à ignorer les sources de mises à jour étrangères.

Comme les routeurs R3 et R4 sont les seules sources du segment 5 auxquelles nous faisons confiance, nous pouvons associer aux adresses IP sur leur interface Token Ring respectives le masque générique 0.0.0.0, en tant qu'argument à la commande **distance**. Pour bien indiquer que toute autre source pour la mise à jour RIP ne nous inspire pas confiance, nous devons associer à l'adresse réseau 200.5.0.0 du segment 5, le masque générique 0.0.0.255, avant de le passer en argument à la commande **distance** suivante.

Les listings 4.32 et 4.33 montrent les configurations des routeurs R3 et R4 après modification.

**Listing 4.32. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 network 10.0.0.0
 network 200.5.0.0
 distance 200 200.5.0.2 0.0.0.0
 distance 255 200.5.0.0 0.0.0.255
```

© copyright Éditions Eyrolles

**Listing 4.33. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 passive-interface TokenRing0
 network 200.5.0.0
 network 200.6.0.0
 neighbor 200.5.0.1
 distance 200 200.5.0.1 0.0.0.0
 distance 255 200.5.0.0 0.0.0.255
```

En examinant les tables de routage des deux routeurs, nous nous apercevons que les mises à jour envoyées par l'hôte H2 ont disparu. En même temps, la distance administrative des routes apprises réciproquement par les routeurs R3 et R4 est passée à la valeur 200 au lieu de 120 précédemment. Cette valeur est mise en italique dans les listings 4.34 et 4.35.

**Listing 4.34. Table de routage du routeur R3.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 4 subnets
R       10.2.0.0 [120/1] via 10.4.0.2, 00:00:15, Serial0
C       10.3.0.0 is directly connected, Serial1
R       10.1.0.0 [120/1] via 10.3.0.2, 00:00:18, Serial1
C       10.4.0.0 is directly connected, Serial0
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [200/1] via 200.5.0.2, 00:00:16, TokenRing0
```

**Listing 4.35. Table de routage du routeur R4.**

```
R4#show ip route
...
R    10.0.0.0 [200/1] via 200.5.0.1, 00:00:18, TokenRing0
C    200.5.0.0 is directly connected, TokenRing0
C    200.6.0.0 is directly connected, Ethernet0
```

**AVERTISSEMENT** L'ordre dans lequel doivent être introduites les commandes **distance** est primordial. Seule la première commande dont l'argument *<adresse IP source/masque générique>* correspond à l'adresse IP de la source de l'envoi des mises à jour, est prise en compte. Une fois cette condition remplie, les commandes suivantes n'ont plus d'effet. Dans notre exemple, si nous avions interverti les commandes, le routeur aurait ignoré toutes les mises à jour en provenance du segment 5, quelle que soit leur source, car la commande **distance 255 200.5.0.0 0.0.0.255** correspond à toutes les adresses de ce segment dont la valeur de distance administrative vaut 255, ce qui signifie qu'il faut ignorer les mises à jour les ayant pour origine.

**REMARQUE** La commande **distance** est indépendante des protocoles utilisés, et peut donc agir sur le fonctionnement de chacun d'eux.

© copyright Éditions Eyrolles

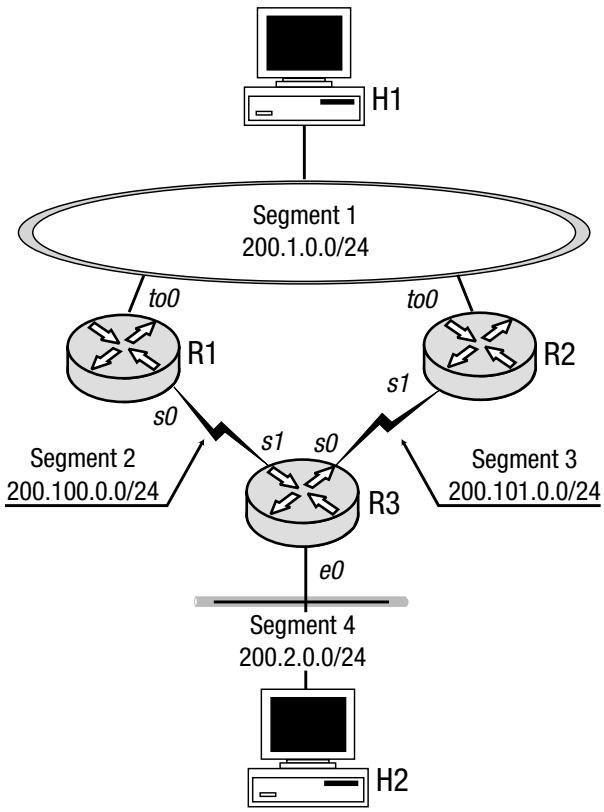
### Configuration de RIP avec le partage de charge à coût égal

L'implémentation de RIP dans l'IOS de Cisco peut répartir le trafic sur plusieurs chemins vers la même destination exprimée sous forme de préfixe réseau, si ces différents chemins ont la même métrique.

**REMARQUE** Le routeur qui pratique le partage de charge n'a cependant aucun contrôle sur le trafic entrant qui peut lui être acheminé sur une seule de ses interfaces.

Prenons le cas de la figure 4.9 où le routeur R3 possède deux chemins vers le segment 1 du Token Ring. Ceux-ci sont considérés comme identiques par RIP quel que soit leur débit propre, et il leur attribue la même métrique 1. Le listing 4.36 en apporte une confirmation.

**Figure 4.9**
*Routeur R3 avec chemins redondants vers le segment Token Ring.*




**Listing 4.36. Table de routage du routeur R3.**

```
R3#show ip route
...
C    200.100.0.0/24 is directly connected, Serial1
C    200.101.0.0/24 is directly connected, Serial0
R    200.1.0.0/24 [120/1] via 200.100.0.2, 00:00:05, Serial1
                  [120/1] via 200.101.0.2, 00:00:21, Serial0
C    200.2.0.0/24 is directly connected, Ethernet0
```

© copyright Éditions Eyrolles

Les lignes en italique de la sortie de la commande **show ip route** sur le routeur R3 devraient nous sembler familières. En effet, elles sont les mêmes que celles du chapitre 3 où nous avons configuré le partage de charge en routage statique sur un routeur.

Dans le listing 4.37 on peut voir la sortie de la même commande sur le routeur R1 qui pratique également le partage de charge en répartissant le trafic destiné au réseau de la ligne série reliant les routeurs R2 et R3. De toute évidence, ce genre de partage de charge a peu d'utilité, sachant que le débit de l'un des chemins (le segment Token Ring) est bien plus important que celui de l'autre chemin (la ligne série). En l'occurrence, le partage de charge va saturer la ligne série, alors que le Token Ring sera sous-utilisé. Heureusement, il est très improbable (sauf acte de malveillance) que la ligne série reçoive un trafic important, vu qu'il s'agit d'un réseau qui n'a que deux hôtes qui sont tous les deux des routeurs.

La table de routage du routeur R1 se trouve sur le listing 4.37.

**Listing 4.37. Table de routage du routeur R1.**

```
R1#show ip route
...
C  200.1.0.0/24 is directly connected, TokenRing0
R  200.2.0.0/24 [120/1] via 200.100.0.1, 00:00:07, Serial0
C  200.100.0.0/24 is directly connected, Serial0
R  200.101.0.0/24 [120/1] via 200.100.0.1, 00:00:07, Serial0
                  [120/1] via 200.1.0.2, 00:00:05, TokenRing0
```

**ASTUCE** Les routeurs Cisco peuvent utiliser jusqu'à quatre chemins (valeur par défaut) pour le partage de charge. La valeur par défaut peut être modifiée par la commande **maximum-paths** <*nombre*>, où ce paramètre est un entier allant de 1 à 6. Pour empêcher le routeur de pratiquer le partage de charge, il faut renseigner le paramètre à la valeur 1.

### Changement de métrique RIP

Il est parfois utile de contraindre RIP à choisir une route plutôt qu'une autre, surtout si deux liaisons parallèles entre deux routeurs ont des débits asymétriques. Dans ce cas, il serait préférable que RIP choisisse la liaison qui possède le plus grand débit. Nous savons d'un chapitre précédent que RIP ne fait pas de distinction entre les segments auxquels il est relié. Il attribue systématiquement à ces derniers, une métrique de 1, sans tenir compte de leurs caractéristiques. Dans la figure 4.9, le schéma du réseau qui y est représenté, incitera RIP à répartir un trafic égal sur les deux liaisons, ce qui ne nous conviendrait pas. Par exemple, si le débit de la première est trois fois plus important que celui de la deuxième, en partage de charge, la première à elle seule aura un potentiel de trafic supérieur à celui qui est effectivement acheminé.

Le système IOS de Cisco procure un moyen très puissant pour modifier la métrique de 1 que RIP attribue aux interfaces qu'il dessert. Il s'agit de la commande **offset-list**, disponible en mode de configuration routeur, qui ne modifie pas la métrique de l'interface elle-même, mais change sélectivement la métrique des mises à jour reçues ou émises, en appliquant une majoration précise (*offset*). Cette sélectivité de la commande permet de préciser en option, parmi les routes, celles qui auront leurs métriques modifiées pour chaque interface.

Le format de la commande est le suivant :

**offset-list** <*liste d'accès*> {**in**|**out**} <*valeur de majoration*> [<*interface*>]

© copyright Éditions Eyrolles

Le premier paramètre peut prendre comme valeur 0 ou le numéro d'une liste d'accès (qui peut aussi être un nom, dans les dernières versions de l'IOS de Cisco). Si celle-ci existe, la majoration de la métrique ne sera appliquée qu'aux routes correspondantes. Dans le cas contraire ou si le paramètre est à 0, la majoration s'appliquera à toutes les routes. Nous aurons à étudier en détail les listes d'accès au chapitre 6.

Les mots clés alternatifs **in** ou **out** précisent si la commande s'applique en entrée ou en sortie des mises à jour RIP.

Le dernier paramètre qui est optionnel permet d'affecter la commande à une interface spécifique sur laquelle RIP appliquera la majoration de la métrique en conformité avec les autres éléments mentionnés ci-dessus.

Selon la documentation de Cisco, la commande **offset-list**, devient « étendue » (*extended*), quand elle concerne une interface spécifique. Quand la même route est comprise aussi bien dans une commande étendue que dans une commande non étendue, c'est la première qui prime sur la seconde. Autrement dit, c'est la majoration de la métrique précisée dans la première commande qui sera appliquée à la route concernée.

Chaque interface peut posséder une commande **offset-list** séparée, en entrée et en sortie. Il peut exister en outre des commandes non étendues, qui s'appliquent aussi bien en entrée qu'en sortie. Toute commande **offset-list** introduite à la suite supprime celle d'avant, à paramètres et mot clé identiques.

Dans le schéma de réseau de la figure 4.9, supposons que le segment 3 ait un débit trois fois plus réduit que celui du segment 2. La routeur, en l'état actuel, pratique le partage de charge sur ces deux segments. Notre but est de majorer la métrique de RIP pour le segment 3 en changeant sa valeur par défaut qui est 1, pour la faire passer à 3. La commande **offset-list** concernée apparaît sur le listing 4.38 qui montre la configuration modifiée du routeur R3.

**Listing 4.38. Configuration modifiée du routeur R3.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.101.0.1 255.255.255.0

interface Serial1
 ip address 200.100.0.1 255.255.255.0

router rip
 offset-list 0 in 3 Serial0
 network 200.2.0.0
 network 200.100.0.0
 network 200.101.0.0
```

L'indication du partage de charge de l'exemple précédent (cf. listing 4.3) a disparu du listing 4.39 qui montre la table de routage du routeur R3. Nous pouvons en avoir confirmation à la lecture du listing 4.40 où la métrique des mises à jour reçues en entrée de l'interface série du segment 3, par le routeur R3, a bien été majorée à 3.

© copyright Éditions Eyrolles

**Listing 4.39. Table de routage du routeur R3.**

```
R3#show ip route
...
C    200.100.0.0/24 is directly connected, Serial1
C    200.101.0.0/24 is directly connected, Serial0
R    200.1.0.0/24 [120/1] via 200.100.0.2, 00:00:11, Serial1
C    200.2.0.0/24 is directly connected, Ethernet0
```

**Listing 4.40. Sortie de la commande `debug ip rip`.**

```
R3#debug ip rip
RIP protocol debugging is on
R3#
RIP: received v1 update from 200.101.0.2 on Serial0
     200.1.0.0 in 4 hops
RIP: received v1 update from 200.100.0.2 on Serial1
     200.1.0.0 in 1 hops
```

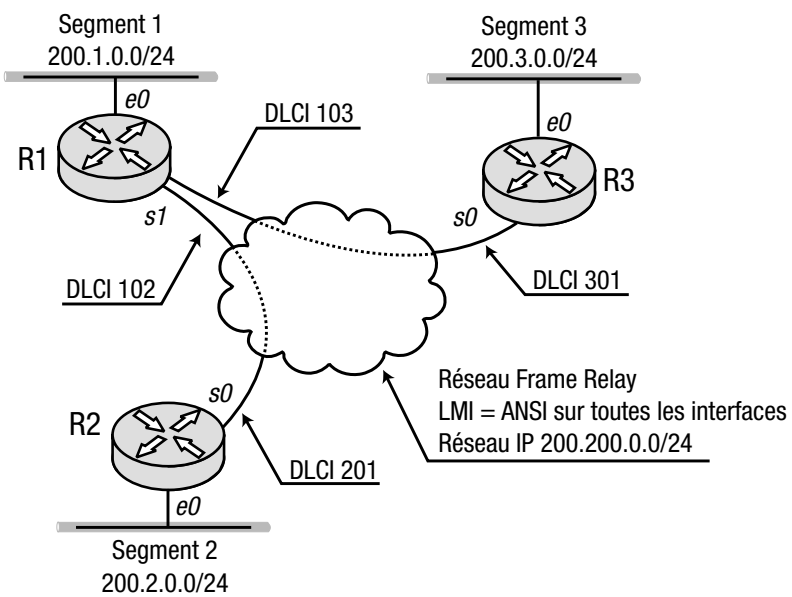
### Configuration de RIP sur un réseau Frame Relay non intégralement maillé

Quel que soit le protocole, la configuration de Frame Relay présente un cas qui diffère des autres supports de transmission, car c'est un réseau de non diffusion à accès multiple ou NBMA (*Non-Broadcast Multiple Access*). La plupart des protocoles de routage ont recours aux adresses de diffusion générale (*broadcast*) ou de diffusion multidestinataire (*multicast*) qui ne sont pas directement utilisables dans Frame Relay, ce qui nécessite des solutions de rechange.

Ces solutions s'avèrent particulièrement indispensables quand il s'agit du protocole OSPF, comme nous le verrons dans le chapitre 5. Bien que les réseaux NBMA soient mieux tolérés par les protocoles à vecteur de distance, ceux-ci peuvent cependant comporter des surprises, surtout quand les réseaux en question ne sont pas intégralement maillés (*non fully meshed*).

Prenons l'exemple de la figure 4.10 qui représente un réseau Frame Relay non intégralement maillé avec trois routeurs. Pour des raisons inconnues, le routeur R1 ne peut être configuré

**Figure 4.10**
*Routeurs reliés par un réseau Frame Relay non intégralement maillé.*




© copyright Éditions Eyrolles

avec des sous-interfaces pour ses deux CVP (circuit virtuel permanent) qui le relient aux deux autres, R2 et R3.

Les trois routeurs sont configurés avec RIP, et d'après ce qu'on en sait, RIP est incapable d'assurer une connectivité globale du réseau. La règle du clivage d'horizon va empêcher le routeur R1 de passer les mises à jour du routeur R2 au routeur R3, et *vice versa*. Car celles-ci nécessitent d'être transmises *via* l'interface qui les a reçues. Observons cependant ce qui se produit en réalité. Les listings 4.41 à 4.43 montrent les configurations des trois routeurs.

**Listing 4.41. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.200.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.2 102 broadcast
 frame-relay map ip 200.200.0.3 103 broadcast
 frame-relay lmi-type ansi

router rip
 network 200.1.0.0
 network 200.200.0.0
```

**Listing 4.42. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.200.0.2 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.1 201 broadcast
 frame-relay lmi-type ansi

router rip
 network 200.2.0.0
 network 200.200.0.0
```

**Listing 4.43. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.200.0.3 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.1 301 broadcast
 frame-relay lmi-type ansi

router rip
 network 200.3.0.0
 network 200.200.0.0
```

© copyright Éditions Eyrolles

Nous constatons que le routeur R1 n'a aucun problème pour voir les segments attachés aux routeurs R2 et R3. Nous doutons néanmoins que le routeur R2 soit capable de voir le segment Ethernet attaché au routeur R3, et réciproquement. Pour en avoir confirmation, examinons la table de routage du routeur R2 (ligne en italique) sur le listing 4.44.

**Listing 4.44. Table de routage du routeur R2.**

```
R2#show ip route
...
C    200.200.0.0/24 is directly connected, Serial0
R    200.1.0.0/24 [120/1] via 200.200.0.1, 00:00:18, Serial0
C    200.2.0.0/24 is directly connected, Ethernet0
R    200.3.0.0/24 [120/2] via 200.200.0.1, 00:00:18, Serial0
```

Visiblement nous avions tort ; le RIP sur le routeur R1 a malgré tout passé outre la règle du clivage d'horizon pour envoyer les mises à jour en provenance du routeur R3 vers le routeur R2, *via* sa même interface. On peut en trouver l'explication en lançant la commande **show ip interface** sur le routeur R1 (cf. listing 4.45). Celle-ci, contrairement à la commande **show interfaces**, affiche les informations spécifiques à IP de l'interface donnée en argument.

Sur le listing 4.45, la ligne en italique montre l'information qui nous concerne particulièrement. On y voit que le clivage d'horizon est tout simplement désactivé, et par conséquent ignoré par le routeur R1. Ce qui n'est pas un hasard, sachant que dans l'IOS de Cisco, sur toutes les interfaces configurées avec Frame Relay, cette désactivation est faite par défaut.

**Listing 4.45. Sortie de la commande `show ip interface` sur le routeur R1.**

```
R1#show ip interface Serial 1
Serial1 is up, line protocol is up
  Internet address is 200.200.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is disabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is enabled
  IP multicast fast switching is disabled
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  Probe proxy name replies are disabled
```

© copyright Éditions Eyrolles

```
Gateway Discovery is disabled
Policy routing is disabled
Network address translation is disabled
```

Une autre surprise nous attend, si nous lançons un ping du routeur R2 vers l'interface Ethernet du routeur R3. Le résultat est affiché sur le listing 4.46.

**Listing 4.46. Échec de la commande `ping 200.3.0.1` sur le routeur R2.**

```
R2#ping 200.3.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.3.0.1, timeout is 2
seconds:
.....
Success rate is 0 percent (0/5)
```

Pour une raison quelconque, la commande **ping** n'aboutit pas. Nous allons savoir pourquoi, en lançant la commande **debug ip packet** sur le routeur R2 (cf. listing 4.47).

**Listing 4.47. Explication de l'échec du ping par la commande `debug ip packet`.**

```
R2#debug ip packet
IP packet debugging is on
R2#ping 200.3.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.3.0.1, timeout is 2 seconds:

IP: s=200.200.0.2 (local), d=200.3.0.1 (Serial0)...
IP: s=200.200.0.2 (local), d=200.3.0.1 (Serial0)...
IP: s=200.200.0.2 (local), d=200.3.0.1 (Serial0)...
IP: s=200.200.0.2 (local), d=200.3.0.1 (Serial0)...
IP: s=200.200.0.2 (local), d=200.3.0.1 (Serial0)...
Success rate is 0 percent (0/5)
```

Les parties en italique mettent en évidence l'adresse IP 200.200.0.2 que le routeur R2 insère dans chaque paquet ping avant de l'envoyer au routeur R3. Mais celui-ci n'a aucune route vers cette adresse. Rappelons-nous la commande **frame-relay map ip** du chapitre 1 qui établit un lien entre une adresse IP distante et le DLCI (numéro de CVP du Frame Relay). Aucun des deux routeurs R2 et R3 n'a cette commande dans sa configuration pour pointer mutuellement sur l'adresse IP de leur interface série respective.

### Configuration de IGRP

Nous allons utiliser la topologie de la figure 4.3 qui a déjà servi pour RIP, car le protocole IGRP lui ressemble sur bien des points en configuration de base. Une fois passés en revue les traits communs, nous aborderons certaines fonctionnalités de IGRP non disponibles dans RIP.

Pour configurer IGRP, les étapes à suivre sont les mêmes que pour RIP, à cette différence près qu'il faut introduire un renseignement supplémentaire par la commande **router igrp** <*numéro de système autonome*>. Ce paramètre désigne le système autonome ou AS (*Autonomous System*) desservi par le processus IGRP qui sera lancé sur le routeur. Dans notre exemple, le AS aura le numéro 10.

© copyright Éditions Eyrolles

Les listings 4.48 à 4.51 montrent les configurations modifiées pour les quatre routeurs de la figure 4.3.

**Listing 4.48. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial0
 ip address 200.3.0.2 255.255.255.0

router igrp 10
 network 200.1.0.0
 network 200.3.0.0
```

**Listing 4.49. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial1
 ip address 200.4.0.2 255.255.255.0

router igrp 10
 network 200.2.0.0
 network 200.4.0.0
```

**Listing 4.50. Configuration du routeur R3.**

```
interface Serial0
 ip address 200.4.0.1 255.255.255.0

interface Serial1
 ip address 200.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.3.0.0
 network 200.4.0.0
 network 200.5.0.0
```

**Listing 4.51. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.5.0.0
 network 200.6.0.0
```

© copyright Éditions Eyrolles

La commande **debug ip igrp transactions** est l'équivalent pour IGRP de la commande de RIP, **debug ip rip**. On peut voir la sortie de cette commande affichée par le routeur R3, sur le listing 4.52.

**Listing 4.52. Sortie de la commande `debug ip igrp transactions`.**

```
R3#debug ip igrp transactions
IGRP protocol debugging is on
R3#
IGRP: received update from 200.4.0.2 on Serial0
      network 200.2.0.0, metric 8576 (neighbor 1100)
IGRP: received update from 200.5.0.2 on TokenRing0
      network 200.6.0.0, metric 1163 (neighbor 1100)
IGRP: received update from 200.3.0.2 on Serial1
      network 200.1.0.0, metric 8576 (neighbor 1100)
IGRP: sending update to 255.255.255.255 via Serial0
(200.4.0.1)
      network 200.1.0.0, metric=8576
      network 200.3.0.0, metric=8476
      network 200.5.0.0, metric=688
      network 200.6.0.0, metric=1163
IGRP: sending update to 255.255.255.255 via Serial1
(200.3.0.1)
      network 200.2.0.0, metric=8576
      network 200.4.0.0, metric=8476
      network 200.5.0.0, metric=688
      network 200.6.0.0, metric=1163
IGRP: sending update to 255.255.255.255 via TokenRing0
(200.5.0.1)
      network 200.1.0.0, metric=8576
      network 200.2.0.0, metric=8576
      network 200.3.0.0, metric=8476
      network 200.4.0.0, metric=8476
```

Les autres commandes disponibles dans IGRP avec les mêmes fonctionnalités que dans RIP, sont : **neighbor**, **distance** et **passive-interface**.

La liste des différences entre RIP et IGRP est donnée ci-dessous :

- La commande **router igrp** *<numéro de système autonome>* permet de lancer plusieurs processus IGRP sur le même routeur, ce qui n'est pas le cas de la commande **router rip** qui ne lance qu'un seul processus RIP. Les différents processus IGRP identifiés par leur numéro d'AS sont hermétiques les uns par rapport aux autres. Ils ne se transmettent pas les informations de routage entre eux. Autrement dit, les routes établies par un processus IGRP ne sont pas diffusées à un autre processus IGRP.
- Contrairement à RIP, IGRP ne peut annoncer aucune adresse individuelle d'hôte.
- Les adresses IP secondaires configurées sur les interfaces ne sont jamais utilisées par IGRP pour l'envoi des mises à jour ; seule l'adresse primaire sert à cet effet. Les adresses réseau et sous-réseau configurées sur une interface ne sont jamais contenues dans une mise à jour. Visiblement, Cisco a bien tiré les enseignements de RIP. S'agissant de gérer les adresses IP secondaires en environnement réel on peut difficilement trouver mieux que ce que fait IGRP.

© copyright Éditions Eyrolles

### IGRP et sa métrique

IGRP est connu sous le sobriquet de « RIP aux hormones », ce qui n'est pas sans raisons. L'une d'elles se rapporte à la métrique de IGRP qui admet des réseaux d'un diamètre bien plus important que celle de RIP.

La métrique de IGRP, contrairement à celle de RIP, est plus complexe et permet de faire la distinction entre chemins aux caractéristiques différentes, alors que pour RIP ils pourraient sembler identiques. Comprendre la métrique de IGRP est important car certaines fonctionnalités tel que le partage de charge à coût inégal sont basées sur cette métrique. La formule que IGRP utilise pour calculer la métrique de chaque route est la suivante :

$$M_{IGRP} = [k1 \times B_{IGRP} + (k2 \times B_{IGRP})/(256 - L) + k3 \times D_{IGRP}] \times k5/(R + k4)$$

Si le coefficient k5 vaut 0, la formule est réduite à l'expression ci-après :

$$M_{IGRP} = [k1 \times B_{IGRP} + (k2 \times B_{IGRP})/(256 - L) + k3 \times D_{IGRP}]$$

$B_{IGRP}$ est le débit IGRP du chemin, calculé selon la formule ci-dessous :

$$B_{IGRP} = 10^7 / B_{MIN}$$

$B_{MIN}$ est le débit logique minimal du chemin exprimé en kilobits par seconde (Kbit/s). Ce paramètre statique est introduit par la commande **bandwidth** en mode de configuration d'interface. Il faut noter cependant que cette valeur devient $B_{MIN}$ pour un chemin spécifique, uniquement si ce débit logique est le minimum parmi ceux de tous les autres segments qui constituent ce chemin. Pour chaque interface, il y a une valeur par défaut, qui normalement correspond à son débit réel. Pour les lignes série, la valeur par défaut est toujours 1544, ce qui peut nécessiter un ajustement par la commande indiquée plus haut.

**REMARQUE** La commande **bandwidth** ne modifie pas le débit réel de l'interface, mais lui associe une valeur logique. Celle-ci est utilisée pour calculer la métrique d'une route par les protocoles tels que IGRP, EIGRP, OSPF, etc. D'autres processus peuvent également utiliser cette valeur logique.

Si la commande **bandwidth** est utilisée, elle doit l'être sur toutes les interfaces de tous les routeurs connectés à un même segment. Par exemple, si deux routeurs sont reliés par une ligne série, la modification de l'interface série de l'un doit aussi être répercutée sur celle de l'autre.

La valeur logique courante du débit d'une interface, ainsi que les autres facteurs qui interviennent dans le calcul d'une métrique dans IGRP peuvent être visualisés par la commande **show interfaces** <*interface*> (cf. listing 4.53).

**Listing 4.53. Valeurs logiques de IGRP pour le calcul de la métrique affichées à la quatrième ligne (en italique).**

```
R3#show interfaces Serial 1
Serial1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 200.100.0.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
...
```

Sur le listing 4.53 les valeurs suivantes sont affichées, avec leurs codes correspondants :

- l'unité maximale de transfert permise au niveau du réseau physique, codée MTU, en nombre d'octets ;
- $B_{IGRP}$, codé `BW`, débit logique modifiable par la commande **bandwidth**, en mode de configuration interface ;

© copyright Éditions Eyrolles

- $D_{IGRP}$, codé `DLY`, délai (exprimé en multiple de 10 microsecondes) du chemin, somme des délais de tous les segments traversés ; modifiable par la commande **delay**, en mode de configuration interface ; transmis comme une variable de 32 bits (valeur à 0xFFFFFFFF en hexadécimal), si une route est inaccessible ;
- R, codé `rely`, fiabilité de l'interface, mesurée dynamiquement par IGRP dont la valeur va de 1 à 255 (haute fiabilité) ;
- L, codée `load`, charge correspondant à l'interface, mesurée dynamiquement par IGRP dont la valeur varie entre 1 et 255 (charge maximale 100 %).
- Le protocole IGRP ne modifie pas intempestivement les paramètres R et L, pour assurer la stabilité de l'opération de routage quand le trafic passe par des pics. Ceux-ci en conséquence peuvent fluctuer beaucoup sans que IGRP en tienne compte, pour éviter que les routes correspondantes soient mises sous temporisation de maintien (*hold down*). Les paramètres L et R d'une interface donnée peuvent être affichés grâce à la commande **show interfaces** (les deux derniers paramètres en italique du listing 4.53).
- Les facteurs de pondération, k1, k2, k3, k4 et k5 peuvent être configurés administrativement ; leurs valeurs par défaut se trouvent dans la table 4.2.

**Table 4.2. Valeur par défaut des facteurs de pondération.**

| Facteur | Valeur par défaut |
|---|---|
| K1 | 1 |
| K2 | 0 |
| K3 | 1 |
| K4 | 0 |
| K5 | 0 |

Si on applique la valeur par défaut pour les facteurs k, la formule précédente se réduit à la suivante :

$$M_{IGRP} = B_{IGRP} + D_{IGRP}$$

Par exemple, calculons la métrique de IGRP, en remplaçant dans la formule, les paramètres avec leurs valeurs correspondantes, telles qu'elles sont annoncées pour l'interface série 1 du listing 4.53. On obtient :

$$M_{IGRP} = 10^7/1544 + 20000/10 = 6476 + 2000 = 8476$$

L'utilisation de la commande **debug ip igrp transactions** permet de vérifier l'exactitude du calcul, ce que confirme le résultat mis en italique sur le listing 4.54.

**Listing 4.54. Sortie de la commande `debug ip igrp` transactions.**

```
R3#debug ip igrp transactions
IGRP protocol debugging is on
R3#
IGRP: sending update to 255.255.255.255 via Ethernet0 (200.2.0.1)
      network 200.100.0.0, metric=8476
      network 200.101.0.0, metric=8476
      network 200.1.0.0, metric=8539
```

© copyright Éditions Eyrolles

### Configuration de IGRP avec le partage de charge à coût égal et inégal

Tout comme RIP, IGRP peut aussi pratiquer par défaut le partage de charge sur des chemins à coût égal.

Par exemple, si nous changions de protocole pour le réseau de la figure 4.9, en passant de RIP à IGRP, nous devrions modifier les configurations des routeurs selon les listings 4.55 à 4.57.

**Listing 4.55. Configuration du routeur R1.**

```
interface Serial0
 ip address 200.100.0.2 255.255.255.0

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.1.0.0
 network 200.100.0.0
```

**Listing 4.56. Configuration du routeur R2.**

```
interface Serial1
 ip address 200.101.0.2 255.255.255.0

interface TokenRing0
 ip address 200.1.0.2 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.1.0.0
 network 200.101.0.0
```

**Listing 4.57. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.101.0.1 255.255.255.0

interface Serial1
 ip address 200.100.0.1 255.255.255.0

router igrp 10
 network 200.2.0.0
 network 200.100.0.0
 network 200.101.0.0
```

Si nous affichions la table de routage du router R3, nous aurions la sortie du listing 4.58.

© copyright Éditions Eyrolles

**Listing 4.58. Configuration du routeur R3.**

```
R3#show ip route
...
C  200.100.0.0/24 is directly connected, Serial1
C  200.101.0.0/24 is directly connected, Serial0
I  200.1.0.0/24 [100/8539] via 200.100.0.2, 00:00:36, Serial1
                [100/8539] via 200.101.0.2, 00:00:15, Serial0
C  200.2.0.0/24 is directly connected, Ethernet0
```

**ASTUCE** La commande **offset-list** est aussi disponible sous IGRP. Mais elle agit sur le paramètre délai des mises à jour entrantes ou sortantes de l'interface donnée en argument. Si nous l'appliquons sous la forme **offset-list 0 in 3 serial 0** sur le routeur R3 du schéma précédent (cf. figure 4.9), après la commande **router igrp 10**, le partage de charge à coût égal sera supprimé, car la métrique des mises à jour reçues sur l'interface série 0 s'en trouvera augmentée.

Supposons maintenant que le débit du segment 2 soit de 1024 Kbit/s et que celui du segment 3 soit de 512 Kbit/s. Nous savons de ce qui a été dit plus haut que la commande **bandwidth** permet d'ajuster la valeur du débit logique que IGRP utilise pour le calcul de sa métrique.

Les listings 4.59 à 4.61 montrent les configurations modifiées des routeurs R1, R2 et R3.

**Listing 4.59. Configuration du routeur R1.**

```
interface Serial0
 ip address 200.100.0.2 255.255.255.0
 bandwidth 1024

interface TokenRing0
 ip address 200.1.0.1 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.1.0.0
 network 200.100.0.0
```

**Listing 4.60. Configuration du routeur R2.**

```
interface Serial1
 ip address 200.101.0.2 255.255.255.0
 bandwidth 512

interface TokenRing0
 ip address 200.1.0.2 255.255.255.0
 ring-speed 16

router igrp 10
 network 200.1.0.0
 network 200.101.0.0
```

**Listing 4.61. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 200.101.0.1 255.255.255.0
 bandwidth 512

interface Serial1
 ip address 200.100.0.1 255.255.255.0
 bandwidth 1024

router igrp 10
 network 200.2.0.0
 network 200.100.0.0
 network 200.101.0.0
```

L'indication du partage de charge à coût égal a disparu du listing 4.62 pour le routeur R3, suite aux modifications faites plus haut.

**Listing 4.62. Table de routage du routeur R3.**

```
R3#show ip route
...
C  200.100.0.0/24 is directly connected, Serial1
C  200.101.0.0/24 is directly connected, Serial0
I  200.1.0.0/24 [100/11828] via 200.100.0.2, 00:00:55,Serial1
C  200.2.0.0/24 is directly connected, Ethernet0
```

Contrairement à RIP, IGRP peut pratiquer le partage de charge à coût inégal, ce qui permet à un routeur de répartir le trafic vers une même destination selon un critère de proportionnalité de la métrique de chacun des différents chemins.

Le partage de charge à coût inégal n'est pas actif par défaut. Pour qu'il le soit, il faut entrer la commande **variance** <*multiplicande*> en mode routeur (**router igrp** <*numéro d'AS*>). La valeur du paramètre va de 1 à 128.

Une fois que cette commande est introduite dans un routeur, les routes qui satisfont les conditions suivantes sont ajoutées à la table de routage :

- Le routeur de saut suivant doit avoir une route candidate pour une même destination, avec une meilleure métrique que celle de la route locale.
- La métrique de la route candidate doit être inférieure ou égale à celle de la route locale multipliée par le paramètre de la commande **variance**.

Essayons de modifier la configuration du routeur R3 de l'exemple précédent (cf. figure 4.9) pour qu'il puisse pratiquer le partage de trafic à coût inégal sur les segments 2 et 3, dirigé vers le segment 1. Calculons d'abord la métrique des chemins que le processus IGRP permet au routeur R3 d'attribuer vers le segment destinataire.

Le débit le plus petit de la route *via* le segment 2 est 1024 Kbit/s. Les délais configurés sont 630 pour le segment 1 et 20000 pour le segment 2. La métrique de cette route se calcule comme suit :

$$M_1 = 10^7/1024 + (630 + 20000)/10 = 9765 + 2063 = 11828$$

La route via le segment 3 ne diffère que par le plus petit débit qui est 512 Kbit/s. Le calcul de la métrique de cette route se fait comme précédemment :

$$M_2 = 10^7/512 + (630 + 20000)/10 = 19531 + 2063 = 21594$$

© copyright Éditions Eyrolles

Le rapport entre la métrique de la meilleure route et celle de la route candidate est inférieur à 2, ce qui permet par la commande **variance 2** de faire du partage de charge à coût inégal, comme le montre la table de routage du routeur R3 sur le listing 4.63.

**Listing 4.63. Table de routage du routeur R3.**

```
R3#show ip route
...
C  200.100.0.0/24 is directly connected, Serial1
C  200.101.0.0/24 is directly connected, Serial0
I  200.1.0.0/24 [100/11828] via 200.100.0.2, 00:00:22,Serial1
                [100/21594] via 200.101.0.2, 00:00:22,Serial0
C  200.2.0.0/24 is directly connected, Ethernet0
```

Les deux numéros en italique confirment les calculs et la présence de deux routes pour le réseau 200.1.0.0/24 prouve que le routeur R3 opère un équilibrage de charge pour ce réseau.

## Configuration des protocoles de routage sans classe

Les sections suivantes donnent les détails de configuration sur deux protocoles de routage sans classe (*classless*) disponibles dans le système IOS de Cisco, à savoir RIP version 2 et EIGRP.

La différence notable entre ces deux protocoles et les deux autres que nous avons traités dans les sections précédentes tient au fait qu'ils peuvent transmettre le masque associé au préfixe réseau ou sous-réseau dans leurs mises à jour de routage, permettant ainsi la mise en application d'un schéma d'adressage de sous-réseaux de taille variable, appelés aussi réseaux VLSM (*Variable Length Subnet Mask*).

Diviser un réseau par la méthode VLSM, alors que son espace d'adressage a déjà été défini, relève souvent d'un exercice jugé difficile. Nous allons donc présenter dans cette première section, un moyen pour aider à y parvenir.

En dépit de leurs différences, les protocoles de routage à classe et sans classe nécessitent des tâches communes de configuration que nous citerons au passage, avant d'aborder celles qui ne le sont pas.

### Division de l'espace d'adressage IP en VLSM

Le sujet sur la division de l'espace d'adressage en VLSM, bien que n'étant pas directement lié à la configuration des routeurs Cisco, est cependant inclus, car il est important pour comprendre le fonctionnement des protocoles sans classe sur ces routeurs.

L'idée dont s'inspire la méthode que nous abordons dans cette section fut formulée pour la première fois dans la RFC 950, intitulée « *Internet Standard Subnetting Procedure* ». Ce document explique le découpage des sous-réseaux à classe et fait quelques suggestions sur la manière de les représenter. Comme nous le savons, les sous-réseaux sont transcrits en notation décimale pointée tout comme le masque associé mis en mémoire dont chaque bit quand il est à 1 désigne la partie identité sous-réseau et quand il est à 0 désigne la partie identité d'hôte. Mais à l'époque où fut rédigée la RFC 950, la décision n'avait pas encore été prise sur l'implémentation des masques de sous-réseaux.

© copyright Éditions Eyrolles

Ainsi le document en question propose cinq méthodes pour le découpage en sous-réseaux, qui sont les suivantes :

- le champ de taille variable ;
- le champ de taille fixe ;
- le champ de taille variable à auto-encodage ;
- le champ de taille fixe à auto-encodage ; et,
- les bits masqués.

Les trois premières méthodes utilisent un nombre fixe de bits pour le découpage en sous-réseaux. La dernière, une fois implémentée, prit le nom de masque de sous-réseau. C'est de la troisième dont on va s'inspirer pour le VLSM.

**REMARQUE** Les méthodes citées ci-dessus ne doivent pas être confondues avec la division d'un espace d'adressage en VLSM. Si elles sont mentionnées, bien qu'elles concernent plus particulièrement le découpage des sous-réseaux en tant que tel, c'est bien parce que l'une d'elles (la troisième) nous servira de modèle pour illustrer notre méthode VLSM.

Cette troisième méthode est elle-même dérivée de l'encodage du réseau à classe où les bits de poids fort déterminent la signification des bits de poids faible suivants, comme dans les réseaux de classe A, B, C, etc.

Prenons le cas d'un réseau de classe B d'adresse 172.16.0.0/16 qui comporte deux types de segments : Ethernet et lignes point à point HDLC. Le premier doit comporter jusqu'à 200 hôtes, et le deuxième en comporte seulement deux. Nous pouvons utiliser le bit de poids le plus fort du troisième octet de chaque adresse pour désigner soit les segments Ethernet, soit les segments de ligne point à point, de la façon suivante :

- Si le bit de poids le plus fort du troisième octet est à 0, le masque par convention sera /24, ce qui engendre 254 hôtes par sous-réseau, nombre suffisant pour ce qui est des segments Ethernet. Une telle division donnera une série d'adresses allant de 172.16.1.0/24 à 172.16.127.0/24.
- Si le bit de poids le plus fort du troisième octet est à 1, le masque par convention sera /30, ce qui engendre 2 hôtes par sous-réseau, exactement ce qui est spécifié pour les lignes point à point. Nous aurons dans ce cas une série d'adresses allant de 172.16.128.4/30 à 172.16.255.252/30.

Mais la méthode qu'on vient d'évoquer n'est pas optimale quant à l'utilisation de l'espace d'adressage alloué. La première série attribue 127 sous-réseaux pour les segments Ethernet, tandis que la seconde attribue 8 190 sous-réseaux pour les lignes point à point. Si le chiffre 127 peut être considéré comme réaliste, 8 190 ne l'est certainement pas, surtout comparé au premier.

Voyons maintenant si au lieu d'utiliser un seul bit de poids le plus fort, en utilisant plutôt un bloc de bits pour encoder la signification des bits suivants, on améliore un peu la méthode.

Pour ce faire, tout le troisième octet sera réservé à l'encodage ; si tous ses bits sont à 0, cela signifiera que le masque est /30 ; toute autre disposition de ces mêmes bits portera le masque à /24. Cette fois-ci, le nombre de sous-réseaux attribués aux segments Ethernet est de 254, et

© copyright Éditions Eyrolles

celui attribué aux lignes point à point est de 63. Sans être parfaite dans tous les cas, cette méthode paraît meilleure que la première. En outre, elle utilise l'adresse de sous-réseau zéro (172.16.0.0/24), ce qui n'était pas le cas pour la première méthode.

L'une ou l'autre de ces méthodes peut convenir plus ou moins à tel ou tel cas, mais elles sont toutes les deux inadaptées par le côté irrégulier et peu systématique de leur mise en œuvre.

Nous allons donc élaborer une méthode VLSM de division d'un espace d'adressage déjà défini, en sous-réseaux dont la taille devra satisfaire au nombre d'hôtes requis sur chaque segment. Cette méthode alloue les adresses de façon très économique, avec comme seule restriction, l'usage de l'identité de sous-réseau zéro, pour éviter une complexité inutile.

Avant de décrire la méthode VLSM elle-même, pour mieux la formaliser, nous allons introduire la terminologie qui suit :

- L'espace d'adressage auquel appartiennent les sous-réseaux alloués est dénommé *A*. Il s'agit simplement de la paire préfixe réseau/longueur de préfixe (ou pour utiliser un terme plus classique, la paire adresse réseau/masque de sous-réseau). Le préfixe réseau est dénommé $L_A$.
- Le symbole *S* est utilisé pour désigner le masque de sous-réseau. La table 4.3 donne les principales valeurs (longueur de préfixe et masque), en tenant compte du nombre d'hôtes requis.
- La longueur du masque de sous-réseau *S* est notée $L_S$. En divisant un espace d'adressage selon la méthode VLSM, nous devons calculer le nombre de sous-réseaux qui utiliseront ce même masque S. Ensuite, nous devons déterminer le nombre de bits nécessaires à l'encodage de chaque sous-réseau qui utilise ce masque. Ce nombre noté $N_S$, peut être obtenu à partir de la table 4.4.

**Table 4.3. Relation entre longueur de préfixe sous-réseau et le nombre d'hôtes qu'elle peut contenir.**

| Nombre d'hôtes | Longueur masque sous-réseau | Masque sous-réseau |
|---|---|---|
| 1 | /32 | 255.255.255.255 |
| Jusqu'à 2 | /30 | 255.255.255.252 |
| Jusqu'à 6 | /29 | 255.255.255.248 |
| Jusqu'à 14 | /28 | 255.255.255.240 |
| Jusqu'à 30 | /27 | 255.255.255.224 |
| Jusqu'à 62 | /26 | 255.255.255.192 |
| Jusqu'à 126 | /25 | 255.255.255.128 |
| Jusqu'à 254 | /24 | 255.255.255.0 |
| Jusqu'à 510 | /23 | 255.255.254.0 |
| Jusqu'à 1022 | /22 | 255.255.252.0 |
| Jusqu'à 2046 | /21 | 255.255.248.0 |

© copyright Éditions Eyrolles

**Table 4.4. Relation entre le nombre de bits et le nombre de sous-réseaux qu'il peut contenir.**

| Nombre maximum de sous-réseaux | Nombre de bits |
|---|---|
| 3 | 2 |
| 7 | 3 |
| 15 | 4 |
| 31 | 5 |
| 63 | 6 |
| 127 | 7 |
| 255 | 8 |
| 511 | 9 |
| 1023 | 10 |

Nous pouvons maintenant définir les étapes à suivre pour implémenter cette méthode :

1. Estimation du nombre de sous-réseaux à prévoir pour le futur.
2. Estimation du nombre maximum d'hôtes sur chaque sous-réseau existant et à prévoir. Avec ce nombre, déterminer le masque de sous-réseau adéquat, en consultant la table 4.3.
3. Pour chaque masque de sous-réseau S, calcul du nombre de sous-réseaux qui l'utiliseront. Au moyen de la table 4.4, trouver le nombre minimum de bits (noté $N_S$) pour pouvoir énumérer tous ces sous-réseaux rattachés au masque S. S'il n'y a qu'un seul sous-réseau pour utiliser un masque donné, $N_S$ vaudra 0.
4. Pour calculer le nombre total de bits, dénommé $T_S$, nécessaire à chaque groupe de sous-réseaux qui utilisent le même masque, on utilise la formule $\mathbf{T_S = N_S + 32 - L_S}$. Si le calcul donne le même $T_S$ pour deux masques différents, le $N_S$ de l'un d'eux doit être augmenté de 1, et ce procédé répété autant de fois qu'il faudra, afin d'arriver à une situation où on n'a plus de $T_S$ de même valeur.
5. L'inégalité $\mathbf{max(T_S) + 1 \leq 32 - L_A}$ permet de vérifier que tous les sous-réseaux résultants peuvent être contenus dans l'espace d'adressage alloué A. Sinon celui-ci doit être étendu par la diminution de $L_A$ (longueur de préfixe réseau) jusqu'à ce que la condition soit remplie.
6. Tous les $T_S$ doivent être triés dans l'ordre décroissant et placés dans une table, comme celle de la figure 4.11. Les cases en grisé sont celles réservées aux hôtes. Dans chaque rangée, une double barre verticale sépare les cases $T_S + 1$ et $T_S$ ; celle-ci est la dernière des cases du groupe de bits du même nom ; $T_S + 1$ est la case située juste avant. Cette barre en quelque sorte marque la frontière entre les bits utilisés pour énumérer les sous-réseaux de même masque S, et les bits inutilisés.
7. Pour chaque S, il faut placer 1 dans la case $T_S + 1$, et 0 dans toutes celles qui sont en dessous. Pour chaque case vide devant $T_S + 1$, mettre un 0 comme le montre la figure 4.12. Les flèches indiquent la direction de rangement de 0 dans les cases en dessous de $T_S + 1$.
8. Le bloc de bits entre la première case et $T_S$ permet d'identifier de manière unique, tous les sous-réseaux de même masque S. Les bits situés entre la case $T_S + 1$ et les cases en grisé identifient les sous-réseaux individuels. Ces bits peuvent être attribués de plusieurs façons

© copyright Éditions Eyrolles

dont celle qui consiste à utiliser une table auxiliaire comme dans la figure 4.13, où chaque rangée est unique.

La combinaison des bits de sous-réseau, les blocs de bits définis à l'étape 7 et les bits de l'espace d'adressage de départ, avec le masque correspondant, génèrent les adresses de sous-réseau pour tous les segments futurs.

**Figure 4.11**
*Table pour calcul des masques de sous-réseaux VLSM.*

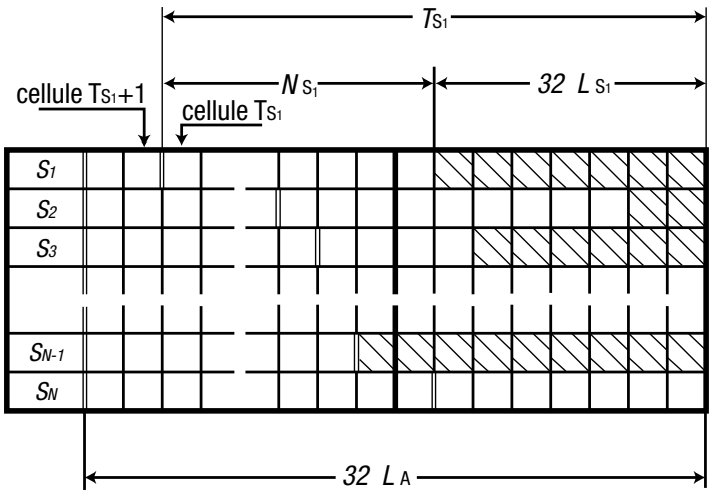



**Figure 4.12**
*Résultats de l'étape 7.*

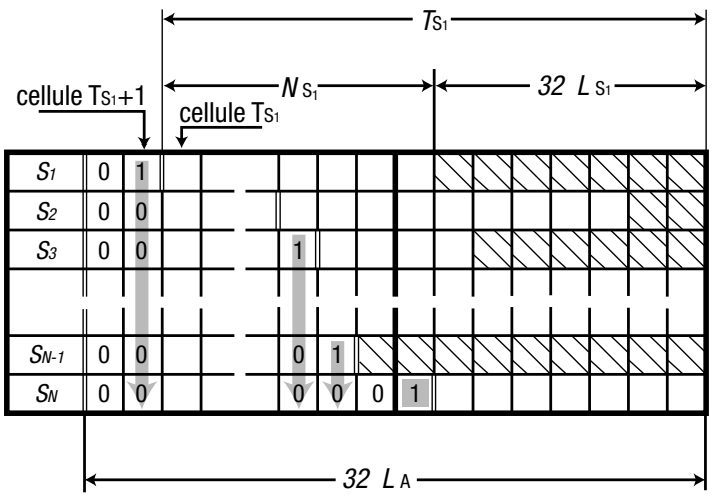



**Figure 4.13**
*Table auxiliaire pour assigner les bits de sous-réseaux.*

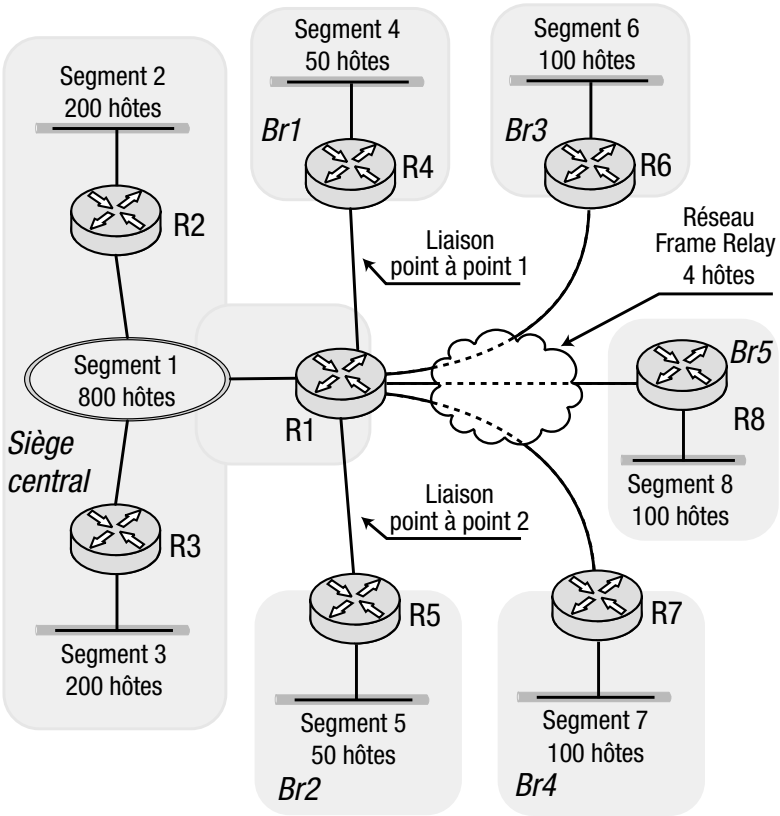
| | | | |
|---|---|---|---|
| Sous-réseau 1 | 0 | 0 | 1 |
| Sous-réseau 2 | 0 | 1 | 0 |
| Sous-réseau 3 | 0 | 1 | 1 |
| Sous-réseau 4 | 1 | 0 | 0 |

**REMARQUE** Les deux dernières rangées de la table de la figure 4.12 montrent deux cas particuliers où dans le premier, le masque $S_{N-1}$ n'est utilisé que par un seul sous-réseau ; le $N_s$ correspondant à ce masque vaut donc zéro. Dans le second cas, le masque $S_N$ vaut /32, et de ce fait, n'ayant pas d'hôtes, la rangée correspondant à ce masque n'a aucune case grise.

© copyright Éditions Eyrolles

Cette procédure peut paraître compliquée, mais elle ne l'est pas tellement. Pour bien comprendre son fonctionnement, prenons l'exemple du réseau de la figure 4.14.

**Figure 4.14**




Une entreprise XYZ possède un siège central relié à cinq branches, tous logés dans des bâtiments séparés avec une infrastructure réseau qui leur est propre. Celui du siège central comprend un anneau FDDI et deux segments Ethernet. Chaque branche comprend un segment Ethernet. La figure 4.14 montre le nombre maximum d'hôtes par segment.

Supposons pour simplifier, que le réseau vient d'être installé et que l'espace d'adressage défini par 200.170.176.0/20 n'a pas encore été implémenté. Nous allons donc mettre en pratique les étapes définies précédemment.

### Première étape

L'extension future du réseau prévoit le plan d'actions suivant :

- Ajouter au siège central trois segments Ethernet, chacun avec 200 hôtes maximum.
- Relier par le Frame Relay au siège central sept branches de plus, chacune avec un segment Ethernet de 100 hôtes.
- Relier par liaisons point à point au siège central quatre branches supplémentaires, chacune avec un segment Ethernet de 50 hôtes maximum.
- En conformité avec la politique de la Direction des systèmes d'information (DSI) de l'entreprise, prévoir une adresse IP individuelle par routeur, pour un total de 60 à ne pas dépasser.

© copyright Éditions Eyrolles

### Deuxième étape

L'infrastructure réseau de l'entreprise comprend sept catégories de segments qui sont répertoriées dans la table 4.5 avec le nombre maximum d'hôtes prévu pour chacun.

**Table 4.5. Types de segments réseau de l'entreprise XYZ dont les noms abrégés sont entre parenthèses.**

| Type de réseau | Nombre maximum d'hôtes |
|---|---|
| FDDI | 800 |
| Ethernet grand (Eg) | 200 |
| Ethernet moyen (Em) | 100 |
| Ethernet petit (Ep) | 50 |
| Frame Relay (Fr) | 10 |
| Lignes point à point (PàP) | 2 |
| Adresses IP individuelles (I) | 1 |

### Troisième étape

Les données de l'étape 2 sont complétées dans la table 4.6, en y incluant les informations sur les masques de sous-réseaux et le nombre de segments par catégorie.

**Table 4.6. Résultats de l'étape 3.**

| Type de réseau | Nombre d'hôtes | Masque de sous-réseau | Nombres de segments | 32 - $L_S$ | $N_s$ |
|---|---|---|---|---|---|
| FDDI | 800 | /22 | 1 | 10 | 0 |
| Eg | 200 | /24 | 5 | 8 | 3 |
| Em | 100 | /25 | 10 | 7 | 4 |
| Ep | 50 | /26 | 6 | 6 | 3 |
| Fr | 10 | /28 | 1 | 4 | 0 |
| PàP | 2 | /30 | 6 | 2 | 4 |
| I | 1 | /32 | 60 | 0 | 6 |

### Quatrième étape

Toutes les valeurs de $T_S$ sont calculées et les résultats se trouvent dans la table 4.7.

Dans cette table on constate aux lignes 2, 3 et 6, 7, que les $T_S$ ont la même valeur (chiffres $T_S$ en gras). Selon notre méthode, pour résoudre ce conflit, nous pouvons augmenter de 1, par exemple le $N_S$ des lignes 2 et 7.

© copyright Éditions Eyrolles

**Table 4.7. Valeurs des $T_S$ incluses, avec deux conflits en lignes 2,3 et 6,7.**

| Type de réseau | Nombre d'hôtes | Masque de sous-réseau | Nombre de segments | $32 - L_S$ | $N_S$ | $T_S$ |
|---|---|---|---|---|---|---|
| FDDI | 800 | /22 | 1 | 10 | 0 | 10 |
| Ethernet | 200 | /24 | 5 | 8 | 3 | **11** |
| Ethernet | 100 | /25 | 10 | 7 | 4 | **11** |
| Ethernet | 50 | /26 | 6 | 6 | 3 | 9 |
| FR | 10 | /28 | 1 | 4 | 0 | 4 |
| Point à Point | 2 | /30 | 6 | 2 | 4 | **6** |
| I | 1 | /32 | 60 | 0 | 6 | **6** |

La table 4.8 affiche les résultats après cette modification (chiffres $N_S$ en gras).

**Table 4.8. Résolution des ambiguïtés.**
**Les nombres en gras montrent les valeurs modifiées.**

| Type de réseau | Nombre d'hôtes | Masque de sous-réseau | Nombre de segments | $32 - L_S$ | $N_S$ | $T_S$ |
|---|---|---|---|---|---|---|
| FDDI | 800 | /22 | 1 | 10 | 0 | 10 |
| Ethernet | 200 | /24 | 5 | 8 | **4** | 12 |
| Ethernet | 100 | /25 | 10 | 7 | 4 | 11 |
| Ethernet | 50 | /26 | 6 | 6 | 3 | 9 |
| FR | 10 | /28 | 1 | 4 | 0 | 4 |
| Point à Point | 2 | /30 | 6 | 2 | 4 | 6 |
| I | 1 | /32 | 60 | 0 | **7** | 7 |

### Cinquième étape

Vérifions maintenant si l'espace d'adressage alloué est suffisant pour contenir tous les sous-réseaux. En appliquant l'inégalité de l'étape 5, le $max(T_S) + 1$ est égal à 13 ; et $32 - L_A$ est égal à 12. La condition n'étant pas remplie, le $L_A$ doit être diminué de 1, ce qui donne le nouvel espace d'adressage de 200.170.160.0/19, avec les deux membres de l'inégalité qui valent maintenant 13.

### Sixième étape

Les résultats de l'étape 4, triés par ordre décroissant de $T_S$, se trouvent sur la figure 4.15.

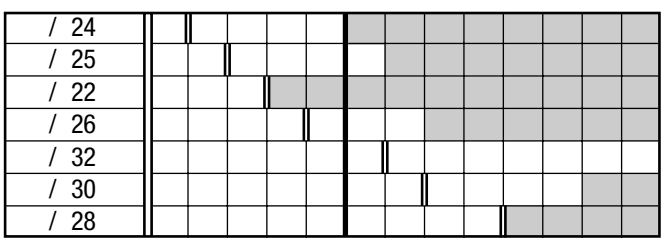



**Figure 4.15**
*Masques de sous-réseau triés par $T_S$ décroissants selon l'étape 6.*

© copyright Éditions Eyrolles

### Septième étape

Entre la première case et $T_S$, toutes les cases intermédiaires ont été remplies avec la valeur du bit appropriée, comme le montre la figure 4.16.

**Figure 4.16**
*Table remplie avec les bits à 0 et 1 selon l'étape 7.*

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| / 24 | 1 | | | | | | | | | | | | |
| / 25 | 0 | 1 | | | | | | | | | | | |
| / 22 | 0 | 0 | 1 | | | | | | | | | | |
| / 26 | 0 | 0 | 0 | 1 | | | | | | | | | |
| / 32 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | |
| / 30 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | |
| / 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | |

### Huitième étape

Tous les sous-réseaux sont énumérés en utilisant le bloc de bits qui leur a été attribué, et les adresses qui en sont dérivées, sont rangées dans la figure 4.17.

**Figure 4.17**
*Résultats de toutes les adresses sous-réseaux générées.*

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| / 24 | **1** | | | | | | | | | | | | | |
| Segment 2 | 1 | 0 | 0 | 0 | 1 | | | | | | | | | 200.170.177.0/24 |
| Segment 3 | 1 | 0 | 0 | 1 | 0 | | | | | | | | | 200.170.178.0/24 |
| / 25 | **0** | **1** | | | | | | | | | | | | |
| Segment 6 | 0 | 1 | 0 | 0 | 0 | 1 | | | | | | | | 200.170.168.128/25 |
| Segment 7 | 0 | 1 | 0 | 0 | 1 | 0 | | | | | | | | 200.170.169.0/25 |
| Segment 8 | 0 | 1 | 0 | 0 | 1 | 1 | | | | | | | | 200.170.169.128/25 |
| / 22 | **0** | **0** | **1** | | | | | | | | | | | |
| Segment 1 | 0 | 0 | 1 | | | | | | | | | | | 200.170.164.0/22 |
| / 26 | **0** | **0** | **0** | **1** | | | | | | | | | | |
| Segment 4 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | | | | | | | 200.170.162.64/26 |
| Segment 5 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | | | | | | | 200.170.162.128/26 |
| / 32 | **0** | **0** | **0** | **0** | **0** | **1** | | | | | | | | |
| Routeur R1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 200.170.160.129/32 |
| Routeur R2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 200.170.160.130/32 |
| Routeur R3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 200.170.160.131/32 |
| Routeur R4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 200.170.160.132/32 |
| Routeur R5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 200.170.160.133/32 |
| Routeur R6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 200.170.160.134/32 |
| Routeur R7 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 200.170.160.135/32 |
| Routeur R8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 200.170.160.136/32 |
| / 30 | **0** | **0** | **0** | **0** | **0** | **0** | **1** | | | | | | | |
| Liaison P2P 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | | 200.170.160.68/30 |
| Liaison P2P 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | | | 200.170.160.72/30 |
| / 28 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | | | | | |
| Réseau F/R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | 200.170.160.16/28 |

## Configuration de RIP version 2

Le protocole RIP version 2 n'est pas vraiment un protocole à part, mais plutôt une version améliorée de RIP version 1. Il possède une nouvelle fonction importante qui permet d'envoyer les préfixes réseau en même temps que leurs longueurs (ou masques) dans les mises à jour de

© copyright Éditions Eyrolles

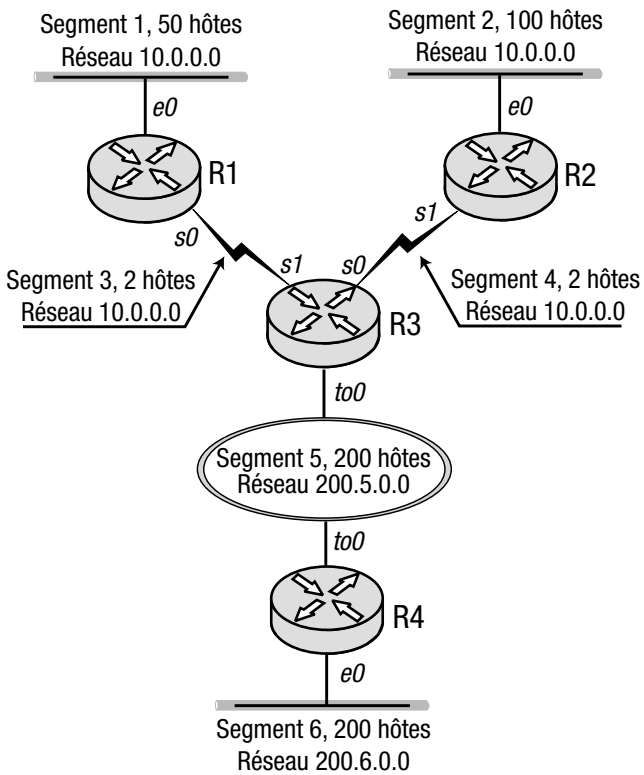
routage à une adresse multicast (224.0.0.9), à la place de l'adresse de diffusion générale (255.255.255.255).

Pour mettre en évidence le caractère sans classe de RIP version 2, apportons les modifications nécessaires au réseau (cf. figure 4.5) que nous avons utilisé pour la version 1. Ce réseau modifié est illustré sur la figure 4.18. Appliquons maintenant la méthode VLSM pour définir les adresses IP à attribuer pour chaque segment du réseau 10.0.0.0, en suivant les mêmes étapes que dans le cas précédent :

Nous n'avons aucune prévision concernant l'extension future de ce réseau, n'est-ce pas ? Passons donc à l'étape suivante.

**Figure 4.18**

*Schéma physique du réseau avec le nombre maximum d'hôtes par segment.*




Les segments réseau sont en fait réduits à quatre, avec en plus les trois adresses individuelles des routeurs. Les résultats de cette étape se trouvent dans la table 4.9.

À l'aide des tables 4.3, 4.4 et 4.9, nous pouvons déduire toutes les valeurs nécessaires de tous les segments du réseau, telles qu'elles figurent dans la table 4.10.

Le calcul des $T_S$ pour chaque masque de sous-réseau donne les valeurs de la table 4.11. Comme il n'est constaté aucun conflit d'égalité de valeur parmi ces $T_S$, nous pouvons passer à l'étape 5.

L'espace d'adressage tout entier du réseau 10.0.0.0 a été alloué, dont la longueur de préfixe est 8. Le $max(T_S) + 1$, dans notre cas vaut 7 + 1 = 8, qui est inférieur à $32 - L_A$, qui vaut 24 ; la condition d'inégalité est donc remplie.

© copyright Éditions Eyrolles

6, 7 et 8. Vu le faible nombre de segments, nous pouvons combiner les trois dernières étapes en une seule, dont les résultats se trouvent sur la figure 4.19.

**Table 4.9. Types de segments réseau selon la figure 4.18.**

| Type de réseau | Nombre maximum d'hôtes |
|---|---|
| Segment 2 (Eg) | 100 |
| Segment 1 (Ep) | 50 |
| Segments 3 et 4 (PàP) | 2 |
| Adresses IP individuelles (I) | 3 |

**Figure 4.19**
*Résultat des étapes 6, 7 et 8.*

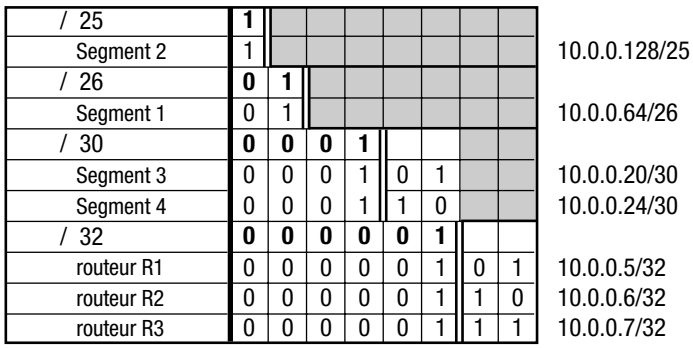



**Table 4.10. Résultats de l'étape 3.**

| Type de réseau | Nombre d'hôtes | Masque de sous-réseau | Nombre de segments | $32 - L_S$ | $N_S$ |
|---|---|---|---|---|---|
| Eg | 100 | /25 | 1 | 7 | 0 |
| Ep | 50 | /26 | 1 | 6 | 0 |
| PàP | 2 | /30 | 2 | 2 | 2 |
| I | 3 | /32 | 3 | 0 | 2 |

**Table 4.11. Valeurs des $T_S$ calculées à l'étape 4.**

| Type de réseau | Nombre d'hôtes | Masque de sous-réseau | Nombre de segments | $32 - L_S$ | $N_S$ | $T_S$ |
|---|---|---|---|---|---|---|
| Eg | 100 | /25 | 1 | 7 | 0 | 7 |
| Ep | 50 | /26 | 1 | 6 | 0 | 6 |
| PàP | 2 | /30 | 2 | 2 | 2 | 4 |
| I | 3 | /32 | 3 | 0 | 2 | 2 |

Les adresses IP du réseau étant définies, nous pouvons maintenant procéder à la configuration des routeurs selon les listings 4.64 à 4.67.

© copyright Éditions Eyrolles

**Listing 4.64. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.0.0.65 255.255.255.192

interface Serial0
 ip address 10.0.0.21 255.255.255.252

router rip
 version 2
 network 10.0.0.0
```

**Listing 4.65. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.6 255.255.255.255
!
interface Ethernet0
 ip address 10.0.0.129 255.255.255.128

interface Serial1
 ip address 10.0.0.25 255.255.255.252

router rip
 version 2
 network 10.0.0.0
```

**Listing 4.66. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.26 255.255.255.252

interface Serial1
 ip address 10.0.0.22 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 version 2
 network 10.0.0.0
 network 200.5.0.0
```

**Listing 4.67. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0
 shutdown

interface TokenRing0
```

© copyright Éditions Eyrolles

```
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 version 2
 network 200.5.0.0
 network 200.6.0.0
```

Le listing 4.68 donne la sortie de la table de routage du routeur R1. Nous pouvons y voir tous les sous-réseaux configurés avec le bon masque. Mais la table de routage du routeur R4 sur le listing 4.69 donne un aperçu du réseau qui est différent.

**Listing 4.68. Table de routage du routeur R1.**

```
R1#show ip route
...
   10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
R     10.0.0.6/32 [120/2] via 10.0.0.22, 00:00:01, Serial0
R     10.0.0.7/32 [120/1] via 10.0.0.22, 00:00:01, Serial0
C     10.0.0.5/32 is directly connected, Loopback0
R     10.0.0.24/30 [120/1] via 10.0.0.22, 00:00:01, Serial0
C     10.0.0.20/30 is directly connected, Serial0
C     10.0.0.64/26 is directly connected, Ethernet0
R     10.0.0.128/25 [120/2] via 10.0.0.22, 00:00:01, Serial0
R  200.5.0.0/24 [120/1] via 10.0.0.22, 00:00:01, Serial0
R  200.6.0.0/24 [120/2] via 10.0.0.22, 00:00:01, Serial0
```

**Listing 4.69. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    10.0.0.0/8 [120/1] via 200.5.0.1, 00:00:03, TokenRing0
```

Tout en fonctionnant sous un protocole sans classe, le routeur R3 pratique néanmoins l'agrégation des routes en n'annonçant que le préfixe réseau lors de l'envoi de la mise à jour de routage le concernant *via* l'interface Token Ring dont l'adresse IP n'appartient pas à ce préfixe réseau, comme le met en évidence le listing 4.69 de la table de routage du routeur R4.

### Désactivation de l'auto-agrégation dans RIP version 2

Les deux versions de RIP pratiquent l'auto-agrégation par défaut, mais on peut la désactiver dans la version 2.

**REMARQUE** La version 1 de RIP est un protocole de routage à classe, ce qui l'oblige à pratiquer l'auto-agrégation ; la version 2 qui est sans classe peut, quant à elle, dépasser certaines restrictions telle que l'auto-agrégation.

Étant un protocole sans classe, le RIP version 2 peut envoyer les préfixes sous-réseau en même temps que leurs longueurs (ou masques) dans ses mises à jour de routage. Ce qui lui permet d'annoncer les sous-réseaux *via* une interface dont l'adresse IP n'appartient pas au même préfixe réseau que ceux-ci.

© copyright Éditions Eyrolles

Pour permettre à RIP en version 2 d'envoyer les mises à jour sans tenir compte du préfixe réseau auquel appartient l'interface de sortie, il faut désactiver l'auto-agrégation par la commande **no auto-summary** en mode de configuration routeur (**router rip**).

Observons ce qui se passe si cette commande est entrée sur le routeur R3 de la section précédente. Nous constatons dans la table de routage du routeur R4 que celui-ci reçoit bien la mise à jour complète de tous les sous-réseaux ayant pour préfixe 10.0.0.0 (cf. listing 4.70). Le routeur R3 ne pratique donc plus l'auto-agrégation.

**Listing 4.70. Table de routage du routeur R4.**

```
R4#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
   10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
R    10.0.0.6/32 [120/2] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.7/32 [120/1] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.5/32 [120/2] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.24/30 [120/1] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.20/30 [120/1] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.64/26 [120/2] via 200.5.0.1, 00:00:05, TokenRing0
R    10.0.0.128/25 [120/2] via 200.5.0.1, 00:00:05,TokenRing0
```

Nous pouvons signaler en passant, que RIP version 2 peut aussi annoncer des super-réseaux qui seront traités au chapitre 6.

### Utilisation simultanée de RIP en version 1 et 2

L'implémentation de RIP dans le système IOS de Cisco permet l'utilisation simultanée des deux versions 1 et 2. En outre, cela peut se faire interface par interface. Deux commandes sont disponibles pour mettre en œuvre cette fonction. La première concerne l'envoi des mises à jour selon la version désirée, par **ip send version** {**1**|**2**} ; la deuxième concerne leur réception selon la version, par la commande **ip receive version** {**1**|**2**}. Ces deux commandes doivent être entrées en mode de configuration interface.

**REMARQUE** La commande pour gérer les versions RIP a pour format complet **ip rip version** {**1**|**2**} [{**1**|**2**}], ce qui permet d'utiliser les deux versions de RIP simultanément sur une interface si on la configure en renseignant le paramètre en option. Cette dernière doit être utilisée avec précaution car le routeur configuré sur une interface avec les deux versions simultanément doit envoyer sa mise à jour pour chacune d'elles, ce qui double son travail.

Quelle que soit la version (1 par défaut) active de RIP sous le mode de configuration routeur par **router rip** ou suivie de **version** pour passer à 2, les deux commandes **ip send**/**receive** priment sur les interfaces où elles ont été introduites.

Nous allons modifier la configuration du routeur R4 dans l'exemple consacré à RIP version 2 pour le faire passer à 1, et nous allons faire en sorte que le routeur R3 tourne sous les deux versions en même temps. Pour ce dernier, seule l'interface Token Ring sera configurée pour envoyer et recevoir les mises à jour en RIP version 1, les autres restant en version 2. Les modifications à apporter au routeur R3 se trouvent sur le listing 4.71.

© copyright Éditions Eyrolles

**Listing 4.71. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.26 255.255.255.252

interface Serial1
 ip address 10.0.0.22 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ip rip send version 1
 ip rip receive version 1
 ring-speed 16

router rip
 version 2
 network 10.0.0.0
 network 200.5.0.0
```

Le listing 4.72 de la table de routage R4, montre un envoi de mise à jour provenant du routeur R3, qui ne comporte que le préfixe réseau 10.0.0.0. Ce qui prouve que ce dernier tourne en RIP version 1 pour son interface Token Ring.

**Listing 4.72. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    10.0.0.0/8 [120/1] via 200.5.0.1, 00:00:02, TokenRing0
```

### Configuration de EIGRP

Le protocole de routage EIGRP est plus fiable que les protocoles à vecteur de distance. Dans le cadre de cet ouvrage, sans en donner une couverture complète, nous nous limiterons à ses caractéristiques essentielles.

Contrairement à RIP version 2 qui est une amélioration de RIP version 1, EIGRP est un protocole intermédiaire entre les protocoles élémentaires à vecteur de distance et ceux, plus élaborés, à état des liens. EIGRP fait des emprunts à ces deux catégories. Il est donc bien plus qu'une amélioration (*Enhanced* IGRP) par rapport à IGRP. Il n'envoie pas de mises à jour régulières comme les protocoles à vecteur de distance, mais échange des messages de « salut » (*hello*) avec ses voisins pour connaître leur existence et savoir s'ils sont toujours actifs.

Les mises à jour sont envoyées par EIGRP uniquement dans le cas d'événements qui modifient la topologie du réseau. Par exemple, si l'une des interfaces d'un routeur tombe en panne ou si celui-ci manque trois messages hello consécutifs d'un de ses voisins, déclaré de ce fait indisponible. Tout changement de topologie entraîne le déroulement d'un algorithme spécial dit de « diffusion de mise à jour » ou DUAL (*Diffusing Update ALgorithm*) qui permet de passer par des chemins de secours (conservés en réserve) pour les préfixes réseau affectés. Cet algorithme

© copyright Éditions Eyrolles

permet de réduire le temps de convergence qui passe ainsi de quelques minutes pour RIP et IGRP à quelques dizaines de secondes seulement pour EIGRP.

Le protocole EIGRP utilise aussi certaines des techniques des protocoles à vecteur de distance telles que le clivage d'horizon, le temporisateur de maintien de route et les mises à jour déclenchées.

La configuration de base de EIGRP, malgré son caractère de protocole sans classe, est la même que celle de IGRP. Tout comme pour ce dernier, la commande **network** ne permet d'assigner aux processus de routage, que les préfixes réseau auxquels appartiennent les adresses IP des interfaces du routeur, sans qu'on puisse y préciser les sous-réseaux.

Les commandes **neighbor**, **distance** et **passive-interface** sont également disponibles avec les mêmes fonctionnalités que dans RIP et IGRP.

Prenons un cas pratique pour illustrer le fonctionnement de EIGRP. Pour ce faire, nous allons utiliser le même schéma de réseau que pour RIP version 2 de la figure 4.18. Les deux protocoles étant sans classe, nous pouvons conserver le même adressage. Les configurations respectives des routeurs R1, R2, R3 et R4 se trouvent sur les listings de 4.73 à 4.76.

**Listing 4.73. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.0.0.65 255.255.255.192

interface Serial0
 ip address 10.0.0.21 255.255.255.252

router eigrp 15
 network 10.0.0.0
```

**Listing 4.74. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.6 255.255.255.255

interface Ethernet0
 ip address 10.0.0.129 255.255.255.128

interface Serial1
 ip address 10.0.0.25 255.255.255.252

router eigrp 15
 network 10.0.0.0
```

**Listing 4.75. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.0.0.26 255.255.255.252

interface Serial1
 ip address 10.0.0.22 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router eigrp 15
 network 10.0.0.0
 network 200.5.0.0
```

**Listing 4.76. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router eigrp 15
 network 200.5.0.0
 network 200.6.0.0
```

Les listings 4.77 et 4.78 montrent les tables de routage des routeurs R1 et R4.

**Listing 4.77. Table de routage du routeur R1 avec présence de routes apprises via EIGRP.**

```
R1#show ip route
...
  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
D  10.0.0.6/32 [90/2809856] via 10.0.0.22, 00:03:52, Serial0
D  10.0.0.7/32 [90/2297856] via 10.0.0.22, 00:03:52, Serial0
C  10.0.0.5/32 is directly connected, Loopback0
D  10.0.0.24/30 [90/2681856] via 10.0.0.22, 00:03:52, Serial0
C  10.0.0.20/30 is directly connected, Serial0
C  10.0.0.64/26 is directly connected, Ethernet0
D  10.0.0.128/25 [90/2707456] via 10.0.0.22, 00:03:52,Serial0
D 200.5.0.0/24 [90/2185984] via 10.0.0.22, 00:03:52, Serial0
D 200.6.0.0/24 [90/2211584] via 10.0.0.22, 00:03:34, Serial0
```

**Listing 4.78. Table de routage du routeur R4.**

```
R4#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
D  10.0.0.0/8 [90/304128] via 200.5.0.1, 00:08:18, TokenRing0
```

© copyright Éditions Eyrolles

Ces deux tables de routage ressemblent à celles qu'on avait déjà vues dans le cas de RIP version 2. Pas de surprise donc. La seule différence qu'on peut y constater, c'est le changement de code pour les routes apprises en dynamique, qui devient « D » pour EIGRP au lieu de « R » pour RIP.

### EIGRP et sa métrique

La métrique calculée par EIGRP est basée sur la même formule que celle de IGRP, avec une multiplication du résultat par 256, ce qui donne :

$$M_{EIGRP} = M_{IGRP} \times 256$$

Pour désactiver la fonction d'auto-agrégation on procède de la même façon que pour RIP, par la commande **no auto-summary** en mode de configuration routeur, pour empêcher le protocole d'agréger les routes sur une frontière de classe en n'annonçant que le préfixe réseau.

Si nous entrons cette commande sur le routeur R3 sous **router eigrp 15** et si nous examinons la table de routage du routeur R4, nous verrons le contenu du listing 4.79 qui montre que ce routeur voit les mêmes routes que le routeur R1.

**Listing 4.79. Table de routage du routeur R4.**

```
R4#show ip route
...
C 200.5.0.0/24 is directly connected,TokenRing0
C 200.6.0.0/24 is directly connected,Ethernet0
  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
D  10.0.0.6/32 [90/231984] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.7/32 [90/304128] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.5/32 [90/231984] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.24/30 [90/218984] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.20/30 [90/218984] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.64/26 [90/221584] via 200.5.0.1,00:00:02,TokenRing0
D  10.0.0.128/25 [90/221584]via 200.5.0.1,00:00:02,TokenRing0
```

### Configuration de l'agrégation de route avec EIGRP

Une commande particulièrement utile dans EIGRP est l'agrégation manuelle de route que l'on introduit par **ip summary-address eigrp** *<numéro de système autonome> <adresse IP/masque de sous-réseau>* en mode de configuration d'interface. Le routeur cesse d'envoyer les mises à jour de routage normales et diffuse à leur place, l'adresse agrégée qui est renseigné en deuxième paramètre.

Par exemple, procédons au changement des adresses sur les routeurs R1, R2 et R3 de la figure 4.80, en substituant au réseau 10.0.0.X celui de 210.0.0.X. Modifions ensuite la configuration du routeur R3 comme indiqué en italique sur le listing 4.80.

**Listing 4.80. Configuration du routeur R3.**

```
interface Loopback0
 ip address 210.0.0.7 255.255.255.255

interface Serial0
 ip address 210.0.0.26 255.255.255.252
```

© copyright Éditions Eyrolles

```
interface Serial1
 ip address 210.0.0.22 255.255.255.252
interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ip summary-address eigrp 15 210.0.0.0 255.255.0.0
 ring-speed 16

router eigrp 15
 network 200.5.0.0
 network 210.0.0.0
```

La table de routage du routeur R4 (cf. listing 4.81) confirme que le routeur R3 effectue l'agrégation manuelle de route lors de l'envoi des mises à jour EIGRP *via* l'interface de Token Ring 0.

**Listing 4.81. Table de routage du routeur R4.**

```
R4#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
D  210.0.0.0/16 [90/304128] via 200.5.0.1,00:01:18,TokenRing0
```

**ASTUCE** Il est conseillé d'accompagner la commande d'agrégation manuelle de route par celle de **ip classless**.

Si nous examinons maintenant la table de routage du routeur R3, nous y verrons comme sur le listing 4.82, en plus des autres mises à jour de routage (non représentées), l'agrégation telle qu'elle est diffusée.

**Listing 4.82. Agrégation de route par EIGRP dans la table de routage du routeur R3.**

```
R3#show ip route
...
D    210.0.0.0/16 is a summary, 00:24:13, Null0
```

La première chose que l'on peut noter, c'est que la route pointe sur une interface nulle, c'est-à-dire que les paquets qui lui sont destinés seront mis au rebut. Nous devons nous rappeler cependant que le routeur, quand il consulte la table de routage, cherche la correspondance la plus longue pour une destination donnée. Ce qui va l'amener à utiliser une route au préfixe plus long que celle qui pointe vers l'interface Null, faisant de cette dernière, une candidate parfaite à l'agrégation. Les paquets dont la destination n'a aucune correspondance plus longue seront donc mis au rebut évitant ainsi de saturer le réseau. L'agrégation de route permet néanmoins au routeur de l'annoncer dans ses mises à jour en utilisant le protocole qui l'a installé dans la table de routage.

### Configuration de EIGRP sur un réseau Frame Relay non intégralement maillé

Comme nous l'avons vu auparavant, le protocole RIP utilisé sur un réseau Frame Relay non intégralement maillé peut avoir un comportement imprévisible dû à la désactivation par défaut du clivage d'horizon. Le protocole EIGRP se comporte différemment en masquant mutuellement les réseaux de routeurs non connectés *via* des CVP.

© copyright Éditions Eyrolles

Pour une raison inconnue Cisco a décidé qu'il n'était pas utile de désactiver par défaut le clivage d'horizon sur les interfaces configurées en Frame Relay. La commande qui permet de désactiver cette fonction a un format (différent de celui de RIP) qui est **no ip split-horizon eigrp** <*numéro de système autonome*>. Nous allons prendre le même schéma de réseau que pour RIP qui se trouve à la figure 4.10, en changeant le protocole à EIGRP.

Les listings 4.83 à 4.85 montrent les nouvelles configurations des routeurs.

**Listing 4.83. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.200.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.2 102 broadcast
 frame-relay map ip 200.200.0.3 103 broadcast
 frame-relay lmi-type ansi

router eigrp 15
 network 200.1.0.0
 network 200.200.0.0
```

**Listing 4.84. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.200.0.2 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.1 201 broadcast
 frame-relay lmi-type ansi

router eigrp 15
 network 200.2.0.0
 network 200.200.0.0
```

**Listing 4.85. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.200.0.3 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 200.200.0.1 301 broadcast
 frame-relay lmi-type ansi

router eigrp 15
 network 200.3.0.0
 network 200.200.0.0
```

© copyright Éditions Eyrolles

La sortie de la table de routage du routeur R2 sur le listing 4.86 confirme ce qu'on attendait : la règle du clivage d'horizon appliquée sur les lignes série configurées en Frame Relay du routeur R1 l'empêche de voir le segment 3 du routeur R3.

Désactivons la fonction de clivage d'horizon sur le routeur R1 par la commande **no ip split-horizon eigrp 15** en mode de configuration interface ligne série 1 pour voir si les choses changent (cf. listing 4.87).

La table de routage du routeur R2 sur le listing 4.88 affiche maintenant le segment 3 du routeur R3 qui était masqué auparavant.

**Listing 4.86. Table de routage du routeur R2.**

```
R2#show ip route
...
C  200.200.0.0/24 is directly connected, Serial0
D  200.1.0.0/24 [90/2195456] via 200.200.0.1,00:06:24,Serial0
C  200.2.0.0/24 is directly connected, Ethernet0
```

**Listing 4.87. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.200.0.1 255.255.255.0
 encapsulation frame-relay
 no ip split-horizon eigrp 15
 frame-relay map ip 200.200.0.2 102 broadcast
 frame-relay map ip 200.200.0.3 103 broadcast
 frame-relay lmi-type ansi

router eigrp 15
 network 200.1.0.0
 network 200.200.0.0
```

**Listing 4.88. Table de routage du routeur R1.**

```
R2#show ip route
...
C  200.200.0.0/24 is directly connected, Serial0
D  200.1.0.0/24 [90/2195456] via 200.200.0.1,00:01:09,Serial0
C  200.2.0.0/24 is directly connected, Ethernet0
D  200.3.0.0/24 [90/2707456] via 200.200.0.1,00:00:14,Serial0
```

© copyright Éditions Eyrolles

# 5

# Routage dynamique : protocoles à état des liens

**Solutions de configuration présentées dans ce chapitre**



Les protocoles à état des liens appartiennent à une catégorie bien à part, basée sur des algorithmes de routage dynamique totalement différents de ceux des protocoles à vecteur de distance. Contrairement à ces derniers, les protocoles à état des liens sont complètement informés de la topologie de la partie (ou même de la totalité, s'il n'y a pas découpage logique) du réseau sur lequel ils opèrent.

© copyright Éditions Eyrolles

Les protocoles à état des liens sont implémentés dans le système IOS de Cisco selon deux normes différentes, d'une part avec le protocole dit « d'ouverture prioritaire du plus court chemin » ou OSPF (*Open Shortest Path First*) défini par l'organisme Internet, et d'autre part le protocole dit « de système intermédiaire à système intermédiaire » ou IS-IS (*Intermediate System to Intermediate System*) de l'ISO. Le protocole OSPF est décrit dans la RFC 2328.

Depuis son introduction, OSPF est devenu le plus répandu des protocoles de routage dynamique. Plusieurs raisons expliquent ce succès, parmi lesquelles un temps de convergence rapide, une capacité à s'adapter à des réseaux de grande dimension et le fait qu'il s'agisse d'une norme ouverte. Le protocole IS-IS, bien que possédant toutes ces qualités, est rarement utilisé ; nous ne traiterons dans cet ouvrage que du premier.

## Protocole OSPF

Dans les sections suivantes, on ne donne qu'un aperçu succinct du protocole OSPF et des principes de base qui le régissent. S'agissant d'un sujet très vaste, son traitement complet sortirait du cadre de cet ouvrage. Pour l'approfondir, le meilleur moyen est de se reporter à la RFC 2328 intitulée « OSPF version 2 ».

**REMARQUE** La RFC 2328 n'est malheureusement disponible que sous forme textuelle. Or la compréhension du protocole OSPF est grandement facilitée par les schémas graphiques. Les nombreux schémas disponibles (en format Postscript) dans la RFC 1583, ancienne version d'OSPF, restant valables pour la RFC 2328 ; il est recommandé de s'y référer.

### *Aperçu du protocole*

Étant un protocole à état des liens, OSPF possède une vision complète soit de la topologie du réseau entier, soit d'une partie spécifique appelée « aire OSPF ». Bien évidemment, la table de routage à elle seule ne peut apporter cette connaissance ; le protocole doit donc tenir à jour un ensemble d'informations appelé base de données d'état des liens ou LSD (*Link State Database*) qui sert à alimenter la table de routage.

La structure de la LSD est conçue pour stocker les informations de topologie tandis que celle de la table de routage facilite la recherche d'adresses IP. Pour remplir cette table, les informations de la LSD sont d'abord converties par l'algorithme de Dijkstra décrit dans la section suivante. Cet algorithme s'exécute à chaque changement intervenu dans la LSD pour remettre à jour la table de routage.

Les informations de topologie stockées dans la LSD doivent être transmises à tous les routeurs qui participent au protocole de routage OSPF. La procédure de communication est la suivante :

1. Les routeurs OSPF utilisent un protocole de communication appelée « OSPF Hello », d'abord pour se découvrir mutuellement et ensuite pour maintenir l'état de leurs liens par une surveillance réciproque.
2. Deux routeurs impliqués dans une découverte synchronisent leur LSD en éliminant les incohérences qu'elles peuvent contenir de façon à disposer en permanence de données à jour. Pour un routeur qui démarre, cette procédure permet d'être informé rapidement de la topologie courante.

© copyright Éditions Eyrolles

3. Le routeur OSPF annonce la partie du réseau à laquelle il est directement connecté, à intervalles réguliers (prédéfini à 30 minutes), envers tous ses « voisins de proximité » (*adjacent neighbors*) (cette notion sera explicitée dans la section sur les types de réseau OSPF). Ces voisins de proximité propagent à leur tour les annonces reçues vers leur propres voisins de proximité, et ainsi de suite. C'est ainsi que la topologie du réseau entier est portée à la connaissance de tous les routeurs.
4. En sus de ces annonces régulières, les routeurs se communiquent ponctuellement les changements dès qu'ils surviennent.

Le protocole OSPF est véritablement sans classe (*classless*), c'est-à-dire qu'il ne préjuge pas de la classe (A, B ou C) à laquelle peut appartenir une adresse IP ou le masque de sous-réseau qui lui est dévolu par défaut.

Pour finir, sachons que chaque routeur OSPF est identifié par un numéro unique (*OSPF router ID* ou simplement *router ID*). Si le système autonome OSPF comprend plusieurs aires de routage (décrites dans la section « modèle de routage hiérarchique »), chacune d'elles sera de même identifiée par un numéro d'aire (*area ID*).

## *Algorithme Dijkstra du plus court chemin*

L'algorithme du chemin le plus court, fondé sur la théorie des graphes, fut mis au point par Edsger Dijkstra en 1959. Depuis lors, grâce à son efficacité et à sa relative simplicité, il a été utilisé dans bon nombre d'applications dans le domaine de l'informatique et des réseaux. Parmi celles-ci, nous pouvons citer les protocoles d'arbre de recouvrement et à état des liens.

Une explication sommaire de son fonctionnement est donnée ci-après. L'algorithme opère sur un graphe orienté comportant des nœuds reliés par des arcs pondérés. Le problème à résoudre consiste à trouver pour un nœud du graphe (appelé nœud racine) le chemin le plus court vers tous les nœuds accessibles (ce chemin est celui dont le cumul des poids de tous les arcs vers ces nœuds destination est minimal ; nous appellerons distance cette valeur cumulée).

L'algorithme utilise deux ensembles de données composés de deux champs : l'identification du nœud et sa distance au nœud racine. Le premier ensemble appelé « chemin le plus court » ou SP (*Shortest Path*) est destiné à contenir les nœuds pour lesquels le plus court chemin a déjà été trouvé. Le deuxième ensemble appelé « chemin candidat le plus court » ou CSP (*Candidate Shortest Path*) contient les nœuds pour lesquels le chemin le plus court reste à confirmer. Au démarrage l'algorithme range le nœud racine dans le SP (la valeur du chemin est à 0, s'agissant du nœud racine) laissant la CSP vide. Par itérations successives l'algorithme déroule les étapes suivantes :

- Il calcule la distance du nœud S vers tous ses voisins. S'il s'agit de l'initialisation, le nœud S est le noeud racine ; sinon, c'est celui défini à l'étape 4 qui est pris comme nœud source.
- Il range tous les nœuds voisins de S dans la CSP en leur associant la distance minimale.
- Il choisit le nœud ayant la distance minimale parmi tous ceux de la CSP pour le ranger dans la SP.
- Il prend le nœud suivant de la CSP comme nœud source pour boucler à partir de l'étape1. L'itération se termine à l'épuisement des nœuds continus dans la CSP

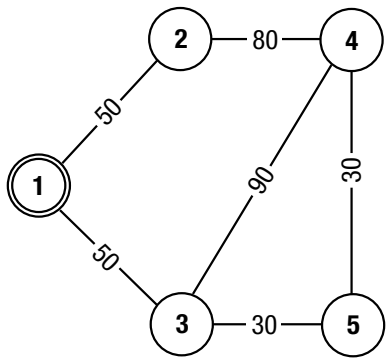
**REMARQUE** Il est important de préciser que dans les itérations successives, la distance dont il est question est la distance au nœud racine et non au nœud source.

© copyright Éditions Eyrolles

La version étendue de l'algorithme de Dijkstra utilisée dans le protocole OSPF (ou dans les protocoles d'état des liens) consiste à calculer le plus court chemin pour **tous** les nœuds du graphe vers les autres nœuds.

Pour mieux comprendre le mécanisme de l'algorithme, prenons l'exemple du graphe illustré sur la figure 5.1. Les nœuds sont représentés par des cercles numérotés. Les arcs sont représentés par des lignes, chacune avec le poids associé.

**Figure 5.1**
*Exemple de graphe orienté.*




La figure 5.2 déroule les étapes de l'algorithme. Les nœuds qui sont rangés dans la CSP sont reliés par des lignes en pointillé, tandis que ceux qui viennent d'être rangés dans la SP le sont par des lignes continues. Le chiffre voisin de chaque nœud correspond à la distance au nœud source.

Des démonstrations animées peuvent être visualisées sur le web par l'exécution interactive de l'algorithme de Dijkstra, soit pas à pas soit en séquence continue. Les paramètres tels que le nombre de nœuds, la charge des arcs que comporte un chemin peuvent être modifiés à chaque exécution de cet algorithme. L'un de ces logiciels de démonstration se trouve implémenté sous forme d'applet Java et accessible sur le site http://carnap.ss.uci.edu/java/dijkstra/DijkstraApplet.html.

## Types de réseau OSPF

Les réseaux implantés en pratique sont très éloignés des graphes orientés sur lesquels se déroule l'algorithme de Dijkstra. Dans le but de l'appliquer aux cas réels, certains aménagements ont dû être apportés à la représentation de ces graphes, du moins dans les structures de données du protocole OSPF, pour pouvoir exécuter l'algorithme.

Les spécifications de OSPF donnent certaines recommandations pour la représentation logique des composants physiques d'un réseau, tels que les routeurs, les réseaux point à point, les réseaux à accès multiple, etc. Nous en donnons ci-dessous les éléments principaux :

- Les routeurs OSPF sont toujours représentés par des nœuds dans un graphe.
- Les liaisons point à point peuvent bien évidemment être représentées en tant qu'arcs du graphe.
- Les réseaux à accès multiple de type LAN et NBMA comme Frame Relay posent problème. Ces deux types de réseau permettent de relier plusieurs routeurs *via* un seul support physique qui ne peut pas être représenté par un arc du graphe.

Dans le cas des LAN, OSPF représente le support physique comme étant lui-même un nœud réseau auquel chaque routeur (qui est aussi un nœud) est relié par son interface réseau qui

© copyright Éditions Eyrolles

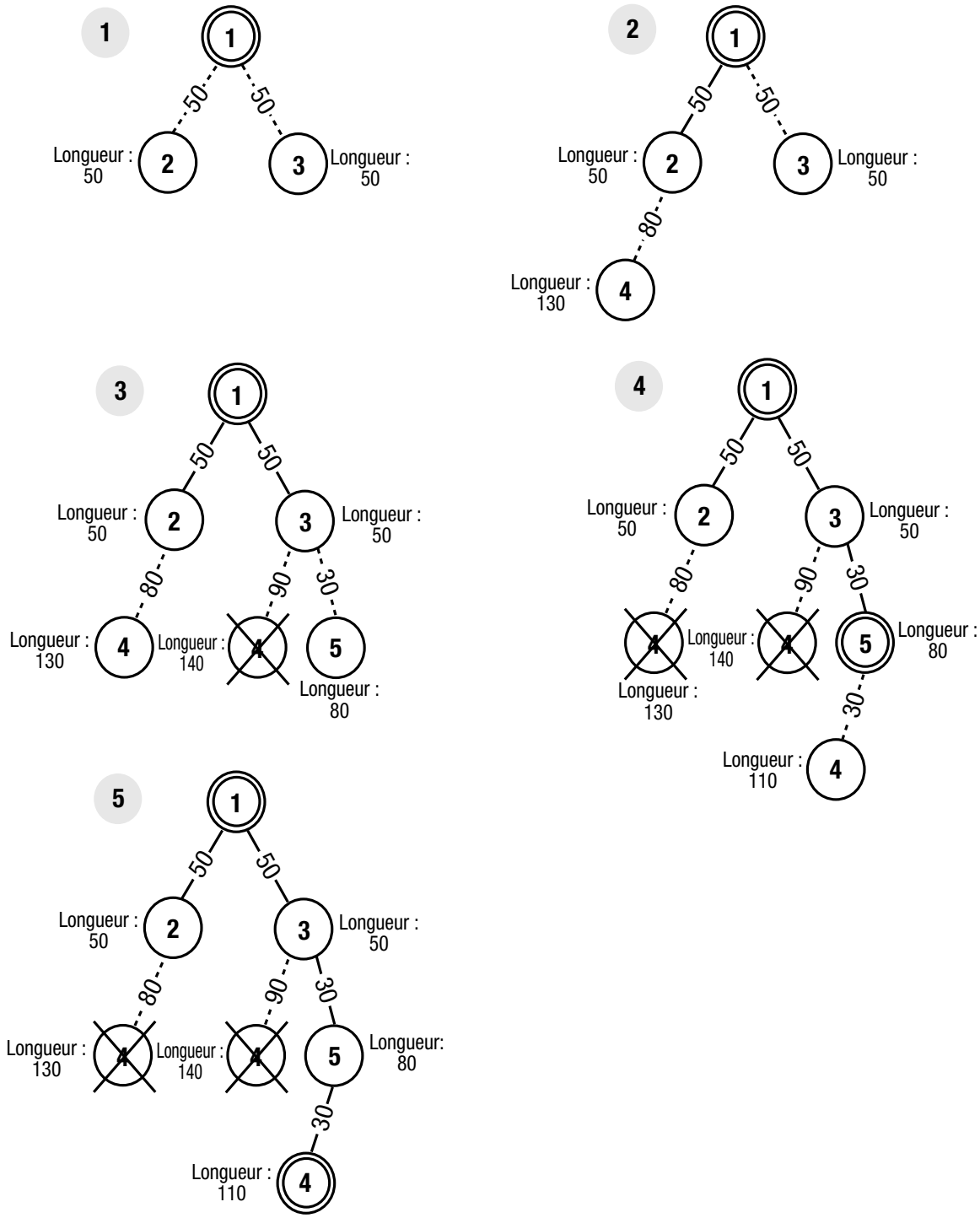



**Figure 5.2**

*Déroulement de l'algorithme de Dijkstra pour le calcul du plus court chemin de chaque nœud du graphe.*

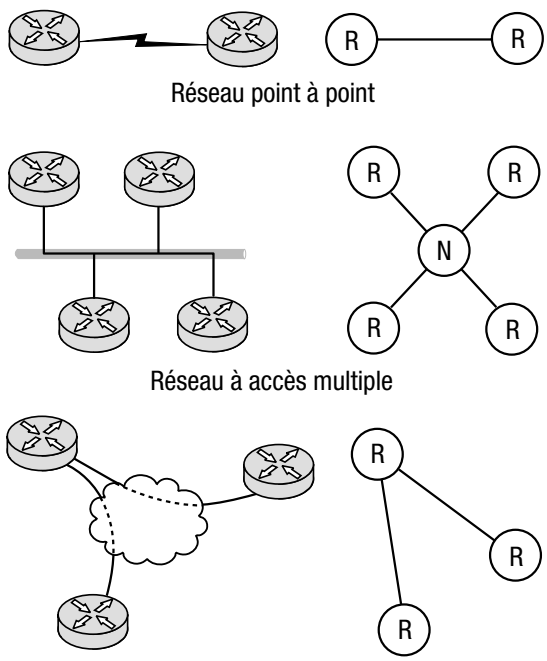
© copyright Éditions Eyrolles

devient l'arc du graphe. Comme le nœud réseau représentant le support physique ne peut pas participer activement au routage OSPF, l'un des routeurs se substitue à lui. Ce routeur prend le nom de routeur désigné ou DR (*Designated Router*). La spécification de OSPF prévoit également un routeur de secours appelé routeur désigné suppléant ou BDR (*Backup Designated Router*).

Les réseaux NBMA peuvent être représentés soit comme une liaison à accès multiples (similaire au LAN) si tous les routeurs sont reliés deux à deux (réseaux intégralement maillés), soit comme plusieurs liaisons point à point (réseaux non intégralement maillés).

La figure 5.3 illustre les différentes représentations logiques des réseaux physiques tels qu'ils sont traduits dans OSPF, où les lettres « R » et « N »(Network) se rapportent aux nœuds routeur et réseau respectivement.

**Figure 5.3**
*Représentation dans OSPF des réseaux à diffusion générale (broadcast).*




Le fonctionnement de OSPF nécessite que les routeurs forment une relation particulière appelée « proximité » (*adjacency*). Pour en faire une représentation logique, on relie par un arc du graphe les deux nœuds que sont les routeurs. OSPF applique les règles suivantes à la création d'une relation de proximité :

- Seuls deux routeurs directement connectés peuvent former une proximité.
- Dans le cas de deux routeurs reliés en point à point, ceux-ci sont toujours à proximité.
- Dans le cas des réseaux à accès multiple, chaque routeur forme une proximité avec aussi bien le routeur désigné que son suppléant.

La relation de proximité est utilisée dans OSPF pour propager les informations de topologie à tout le système autonome.

© copyright Éditions Eyrolles

## Un modèle de routage hiérarchique

Comme nous l'avons dit auparavant, le protocole OSPF fut spécifiquement conçu pour couvrir des réseaux de grande dimension. Le découpage hiérarchique de leur espace d'adressage y est donc inclus.

Le routage hiérarchique prévoit qu'un domaine de routage appelé aussi système autonome soit divisible en aires plus petites, chacune d'elles possédant son propre routage. De l'extérieur, une aire n'est accessible que par une agrégation d'adresses ou plus, qui fournit une correspondance minimale pour toute adresse IP contenue dans cette aire. Le premier but d'une telle méthode est de réduire le nombre de préfixes dans les mises à jour en les agrégeant lors de leur échanges d'une aire à l'autre à l'intérieur du système autonome.

Dans une certaine mesure, le routage hiérarchique existe déjà dans les protocoles moins élaborés tels que RIP version 1, qui pratique l'agrégation quand il doit annoncer un préfixe réseau auquel n'appartient pas l'adresse IP de l'interface par laquelle il doit diffuser ce préfixe. On a déjà rencontré ce phénomène dans un chapitre précédent. Mais le routage hiérarchique ne prend toute son ampleur que dans les protocoles sans classe comme OSPF, EIGRP et RIP version 2. Par exemple, EIGRP fournit la commande **ip summary-address eigrp** <*Numéro de système autonome*><*Adresse IP/masque de sous-réseau*> qui permet de configurer le routage hiérarchique. OSPF est cependant le seul parmi ces protocoles à disposer d'outils évolués simplifiant grandement l'implémentation du routage hiérarchique qui doit remplir les conditions suivantes :

Le domaine de routage doit comporter une seule aire appelée dorsale avec l'identité 0.

- Toutes les autres aires doivent être connectées à l'aire dorsale et ne peuvent communiquer que *via* celle-ci.
- Les routeurs reliés à plus d'une aire doivent avoir une base de données d'état des liens séparée pour chacune. Toutes les opérations telles que l'arrosage (*flooding*) par des annonces d'état des liens ou LSA (*Link State Advertisement*) , l'exécution de l'algorithme du chemin le plus court ou SPF (*Shortest Path First*), etc., se feront également de façon séparée.
- Les routeurs qui relient les aires à la dorsale sont appelés routeurs de « bordure d'aire » ou ABR (*Area Border Router*).

Le protocole OSPF permet aux informations de routage en provenance d'autres protocoles d'être redistribuées dans leur système autonome. Cette fonction sera traitée en détail au chapitre 6, mais elle est mentionnée ici pour introduire un nouveau concept, les routeurs de « limite de système autonome » ou ASBR (*Autonomous System Boundary Router*) dont le rôle est précisément cette redistribution.

## Les LSA

Le protocole OSPF annonce les informations de topologie par des données structurées appelées LSA (*Link State Advertisement*) qui constituent des enregistrements dans sa base de données spécifique appelée LSD (*Link State Database*).

Les LSA contiennent des informations sur l'état local d'un routeur ou du réseau. Dans le cas d'un routeur il peut s'agir par exemple de l'état de ses interfaces ou de celui de ses voisins. Les LSA sont générées par les routeurs OSPF et propagées sur tous leurs voisins, sauf celui à l'origine de ces LSA. Si c'est le routeur lui-même qui est à l'origine d'une LSA, il la propage sur tous ses voisins.

© copyright Éditions Eyrolles

Tous les routeurs OSPF ne génèrent pas tous les types de LSA. Le tableau 5.1 dresse la liste de qui génère quoi.

**Tableau 5.1. Types de LSA générées par les routeurs OSPF.**

| Type de LSA | Nom de LSA | Description |
|---|---|---|
| Type 1 | Router-LSA | Générée par tous les routeurs. Décrit l'état des interfaces du routeur dans l'aire. Arrosée uniquement sur toute l'aire correspondante. |
| Type 2 | Network-LSA | Générée pour les réseaux à accès multiple et NBMA uniquement par les routeurs désignés respectifs. Arrosée uniquement sur toute l'aire correspondante. |
| Type 3 | Summary-LSA | Générée par les ABR uniquement. Décrit les routes vers les destinations situées en dehors de l'aire, mais à l'intérieur du système autonome (par exemple une route agrégée pour une autre aire). Arrosée sur toute l'aire uniquement. |
| Type 4 | Summary-LSA | Générée par les ABR uniquement. Décrit les routes vers les ASBR situés en dehors de l'aire. Arrosée sur toute l'aire uniquement. |
| Type 5 | AS-external-LSA | Générée par les ASBR uniquement. Décrit les routes externes qui peuvent être utilisées comme routes par défaut. Arrosée sur tout le système autonome. |

Pour récapituler les informations du tableau 5.1, nous pouvons dire que les `router-LSA` et les `network-LSA` décrivent les interconnexions entre routeurs et réseaux à l'intérieur d'une aire, que les `summary-LSA` décrivent les routes inter-aire, et que les AS-external-LSA décrivent les routes externes injectées dans le système autonome.

# Solutions de configuration

## Configuration de OSPF avec aire unique

Pour configurer OSPF avec une aire unique, nous devons suivre les étapes suivantes qui sont assez simples :

1. Créer une identité OSPF (*OSPF ID*) par la configuration d'une interface en rebouclage (*loopback*) et lui assigner une adresse IP.

**REMARQUE** Cette étape est facultative, bien que recommandée. Si l'identité OSPF n'est pas précisée, le routeur prend comme telle, l'adresse IP la plus élevée parmi celles configurées sur les interfaces actives. Si cette interface tombe en panne, elle sera remplacée par l'adresse IP la plus élevée parmi les interfaces restantes. Mais si des interfaces sont configurées en rebouclage, c'est l'adresse IP la plus haute de ces interfaces qui est prise pour l'identité OSPF.

L'avantage de l'interface de rebouclage, c'est qu'elle est interne au routeur, et vu son caractère logique, ne peut jamais tomber en panne.

Si la commande de création de l'interface en rebouclage est omise lors de la configuration de OSPF, et qu'elle est introduite par la suite, le routeur ne basculera pas l'identité OSPF sur l'adresse IP de cette interface. Un basculement de l'identité OSPF sur une interface de rebouclage créée après coup ne peut intervenir que lors du redémarrage du routeur lui-même ou de l'interface physique qui servait jusque là comme identité OSPF.

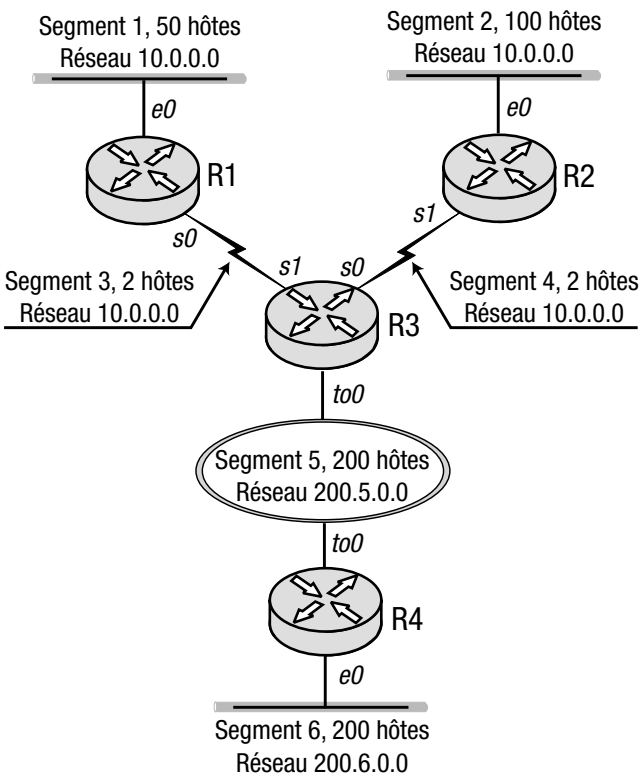
© copyright Éditions Eyrolles

2. Par la commande **router ospf** <*identité de processus*>, créer un processus OSPF sur les routeurs. Le paramètre peut être un chiffre quelconque entre 1 et 65535.
3. Par la commande **network** <*adresse IP/masque générique*> **area 0** en mode de configuration **router ospf**, activer le traitement des mises à jour sur les interfaces adéquates, c'est-à-dire sur toutes celles qui sont renseignées en paramètre de cette commande. Le masque générique (en notation décimale pointée) est constitué de bits dont chaque position à 1 signifie que le bit correspondant dans l'adresse IP doit être ignoré lors de son application. La disposition des bits du masque générique, contrairement au masque de sous-réseau, n'a pas à être continue.

**REMARQUE** Contrairement à la commande **router** de IGRP et de EIGRP, celle de OSPF ne comporte pas en paramètre le système autonome, mais plutôt l'identité du processus qui n'a qu'une signification locale au routeur. Cette identité n'étant pas transportée dans les PDU de OSPF, les routeurs distants sont incapables de savoir de quel processus elle provient. Le numéro de cette identité peut donc être quelconque d'un routeur à l'autre participant au routage OSPF, mais pour une question de cohérence, il est préférable de choisir le même numéro pour tous les routeurs OSPF.

Pour bien voir le fonctionnement de OSPF, prenons le schéma de réseau déjà utilisé dans l'exemple de RIP version 2 (chapitre 4), illustré en figure 5.4 avec le même adressage IP.

**Figure 5.4**
*Tous les routeurs ne sont connectés qu'à l'aire dorsale.*




Les listings 5.1 à 5.4 montrent les configurations des quatre routeurs.

© copyright Éditions Eyrolles

**Listing 5.1. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.0.0.65 255.255.255.192

interface Serial0
 ip address 10.0.0.22 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

**Listing 5.2. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.6 255.255.255.255

interface Ethernet0
 ip address 10.0.0.129 255.255.255.128

interface Serial1
 ip address 10.0.0.26 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

**Listing 5.3. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.25 255.255.255.252

interface Serial1
 ip address 10.0.0.21 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 network 200.0.0.0 0.255.255.255 area 0
```

**Listing 5.4. Configuration du routeur R4.**

```
interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
```

© copyright Éditions Eyrolles

```
 ring-speed 16

router ospf 1
 network 200.0.0.0 0.255.255.255 area 0
```

Notons le format de la commande **network** utilisé dans OSPF. Il nécessite un masque générique qui permet de déterminer les interfaces desservies par OSPF, tandis que les préfixes réseau à annoncer et leur longueur sont prélevés directement sur les interfaces correspondantes. Le mot clé **area** de la commande spécifie l'aire de routage à laquelle appartient le préfixe annoncé.

**ASTUCE** Pour plus de commodité, on peut utiliser les adresses IP d'hôtes pour enregistrer les interfaces à desservir par OSPF. Pour ce faire, la commande **network** doit être répétée autant de fois qu'il y a d'interfaces, avec pour chacune, son adresse IP exacte suivie du masque générique 0.0.0.0. On peut ainsi activer OSPF interface par interface sans avoir à déterminer le masque générique qui engloberait toutes ces interfaces. Le seul inconvénient, c'est la répétition de la commande **network**.

Le listing 5.5 affiche la sortie de la commande **show ip route** sur le routeur R4. On peut y voir que toutes les routes apprises par OSPF sont codées avec la lettre « O ». Ce routeur voit également tous les sous-réseaux qui appartiennent au réseau 10.0.0.0 bien qu'aucune de ses interfaces ne soit configurée avec une adresse IP de ce réseau. Il s'agit là d'une faculté importante de OSPF qui, contrairement aux protocoles à vecteur de distance, ne pratique pas l'auto-agrégation.

**ASTUCE** Si toutes les interfaces de tous les routeurs configurées avec OSPF appartiennent à la même aire, un numéro autre que 0 peut lui être attribué, même si le 0 est fortement recommandé.

**REMARQUE** OSPF pratique le partage de charge à coût égal comme tous les autres protocoles de routage dynamique et le routage statique.

**AVERTISSEMENT** Si des adresses IP secondaires sont utilisées sur des routeurs configurés avec OSPF et que celles-ci sont enregistrées dans son processus de routage, elles doivent appartenir à la même aire que l'adresse primaire

**Listing 5.5. Table de routage du routeur R4.**

```
R4#show ip route
...
  200.0.0.0/32 is subnetted, 1 subnets
C    200.0.0.1 is directly connected, Loopback0
C 200.5.0.0/24 is directly connected, TokenRing0
C 200.6.0.0/24 is directly connected, Ethernet0
  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O  10.0.0.6/32 [110/71] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.7/32 [110/7] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.5/32 [110/71] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.24/30 [110/70] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.20/30 [110/70] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.64/26 [110/80] via 200.5.0.1, 01:35:32, TokenRing0
O  10.0.0.128/25 [110/80] via 200.5.0.1, 01:35:33, TokenRing0
```

© copyright Éditions Eyrolles

## OSPF et son coût

Chaque interface individuelle enregistrée sous OSPF possède un coût qui intervient dans le calcul de son algorithme par la formule suivante :

$$C = 10^8 / B$$

*B* désigne le débit logique de l'interface mesuré en bits par seconde (bit/s). La valeur courante de l'interface peut être visualisée par la commande **show interfaces** <*interface*> <*numéro*>. La valeur du débit logique peut être modifiée par la commande **bandwidth** <*débit*> en mode de configuration d'interface.

**REMARQUE** Le débit logique ne nécessite un ajustement que sur les interfaces série.

| **Solutions apparentées :** | **Page :** |
|---|---|
| IGRP et sa métrique | 144 |
| EIGRP et sa métrique | 166 |

## Configuration de OSPF avec aires multiples

Nous savons déjà que le routage hiérarchique est implanté dans OSPF et permet de diviser un domaine de routage en aires multiples connectées à une aire unique qualifiée de « dorsale ».

Avant de configurer OSPF avec des aires multiples, nous devons au préalable découper les réseaux et concevoir le plan d'adressage IP adéquat. Celui-ci doit prévoir un préfixe agrégé ou plus, de longueur inférieure ou égale à celle du préfixe le plus court de chaque aire concernée, de façon à ce que celui-ci ait une correspondance minimale avec toute adresse IP s'y trouvant. Supposons par exemple qu'une aire possède les adresses réseau 10.0.128.0/17, 10.0.1.0/24 et 10.0.0.128/25. On pourrait prendre 10.0.0.0/16 comme préfixe agrégé car sa longueur inférieure à celle de tous les préfixes et sa correspondance minimale avec toute adresse IP de cette aire en font un choix approprié.

Nous pouvons maintenant configurer les routeurs en suivant les mêmes étapes que dans le cas précédent à une seule aire, en ajoutant l'étape supplémentaire 4 ci-dessous :

4. Par la commande **area** <*aire*> **range** <*adresse IP/masque de sous-réseau*>, définir pour chaque aire un préfixe agrégé ou plus sur les routeurs qui relient celle-ci à l'aire dorsale. Cette commande donne aux routeurs la consigne d'annoncer en direction de la dorsale (qui peut elle-même avoir des préfixes agrégés) ce nouveau préfixe plutôt que les préfixes réseau.

**REMARQUE** La commande **area** s'attend à un masque de sous-réseau et non pas un masque générique comme **network**.

**ASTUCE** On peut utiliser plusieurs commandes **area** si on veut définir plus d'un préfixe agrégé pour une même aire, en renseignant le paramètre après le mot clé **range** avec un argument différent. Tous ces préfixes agrégés ainsi définis seront annoncés par les routeurs de bordure d'aire ou ABR (*Area Border Router*) en direction des aires autres que celle qui est donnée en argument au paramètre <*aire*> de cette commande.

Modifions la topologie du réseau utilisée dans l'exemple précédent qui ne comprenait qu'une seule aire en la décomposant en deux comme sur la figure 5.5. En tenant compte de la carac-

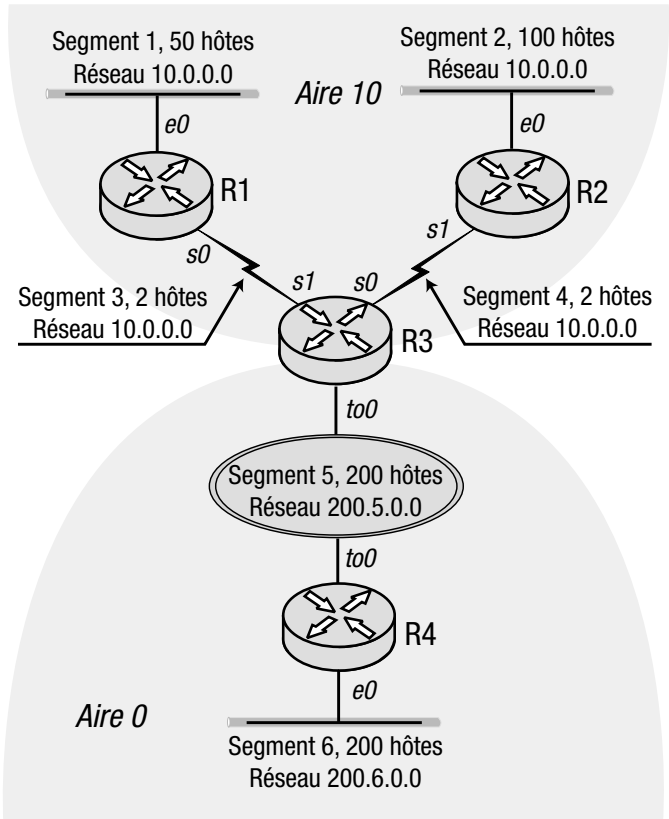
© copyright Éditions Eyrolles

téristique sans classe de OSPF nous pouvons définir les adresses agrégées de ces aires de la façon suivante :

- aire 0, adresse agrégée 200.0.0.0/8 ;
- aire 100, adresse agrégée 10.0.0.0/24.

**Figure 5.5**

*La topologie précédente du réseau comprend maintenant deux aires.*




Pour bien comprendre comment agit la commande **area** <*aire*> **range** <*adresse IP/masque de sous-réseau*>, configurons d'abord le routeur R3 (le seul d'ailleurs qui requière une telle commande), sans cette commande.

Les listings 5.6 à 5.8 montrent les configurations révisées des routeurs R1, R2 et R3 (le routeur R4 ne change pas de configuration car il reste dans la même aire 0).

**ASTUCE** Nous pouvons reconnaître qu'un routeur est un ABR si dans ses commandes **network** existent au moins deux numéros d'aire différents.

**Listing 5.6. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.0.0.65 255.255.255.192
```

© copyright Éditions Eyrolles

```
interface Serial0
 ip address 10.0.0.22 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
```

**Listing 5.7. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.6 255.255.255.255

interface Ethernet0
 ip address 10.0.0.129 255.255.255.128

interface Serial1
 ip address 10.0.0.26 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
```

**Listing 5.8. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.25 255.255.255.252

interface Serial1
 ip address 10.0.0.21 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 network 200.0.0.0 0.255.255.255 area 0
```

Pour voir ce qu'apportent ces configurations sur les routeurs R1, R2 et R3, affichons la table de routage du routeur R4 (cf. listing 5.9). La seule différence qu'on peut y constater par rapport à la table de routage du listing 5.5 du même routeur, c'est la présence du code « IA » qui veut dire I*nter*-A*rea* (inter-aires) devant toutes les routes qui appartiennent à une aire autre (dans ce cas 10) que celle du routeur R4 qui, quant à lui, appartient à l'aire dorsale. Tous nos efforts pour définir d'abord un plan d'adressage avec aires multiples et à configurer ensuite les routeurs en conséquence sont peine perdue pour ce maigre résultat.

Ajoutons maintenant la commande **area** <*aire*> **range** <*adresse IP/masque de sous-réseau*> à la configuration du routeur R3. Les arguments passés au paramètre de cette commande devraient créer des adresses agrégées pour les aires dorsale et 10, comme nous l'avons vu dans une section précédente. Le listing 5.10 montre la configuration correspondante du routeur R3.

© copyright Éditions Eyrolles

**Listing 5.9. Table de routage du routeur R4.**

```
R4#show ip route
...
  200.0.0.0/32 is subnetted, 1 subnets
C    200.0.0.1 is directly connected, Loopback0
C 200.5.0.0/24 is directly connected, TokenRing0
C 200.6.0.0/24 is directly connected, Ethernet0
  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O IA 10.0.0.6/32 [110/71] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.7/32 [110/7] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.5/32 [110/71] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.24/30 [110/70] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.20/30 [110/70] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.64/26 [110/80] via 200.5.0.1, 00:27:11,TokenRing0
O IA 10.0.0.128/25 [110/80] via 200.5.0.1,00:27:11,TokenRing0
```

**Listing 5.10. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.25 255.255.255.252

interface Serial1
 ip address 10.0.0.21 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 network 200.0.0.0 0.255.255.255 area 0
 area 0 range 200.0.0.0 255.0.0.0
 area 10 range 10.0.0.0 255.255.255.0
```

Une fois que la commande **area** a été introduite dans le routeur R3, examinons de nouveau la table de routage du routeur R4 sur le listing 5.11. Cette fois-ci, nos efforts de planification et de configuration des routeurs selon le modèle hiérarchique de OSPF trouvent enfin leur justification. Le routeur R4 ne voit plus qu'une route pour l'adresse agrégée de l'aire 10.

**Listing 5.11. Table de routage du routeur R4 après introduction de la commande `area` sur le routeur R3.**

```
R4#show ip route
...
   200.0.0.0/32 is subnetted, 1 subnets
C     200.0.0.1 is directly connected, Loopback0
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
   10.0.0.0/24 is subnetted, 1 subnets
O IA  10.0.0.0 [110/80] via 200.5.0.1, 00:23:10, TokenRing0
```

© copyright Éditions Eyrolles

Examinons maintenant la table de routage du routeur R1 (cf. 5.12). On peut y constater qu'il voit toutes les routes disponibles dans sa propre aire 10, mais n'a connaissance que d'une seule route pour l'aire dorsale dont l'adresse agrégée est 200.0.0.0/8.

**REMARQUE** La RFC 1879 recommande expressément d'inclure la commande **ip classless** en configurant OSPF, ce qui amène le routeur à se conformer à la RFC 1812. Ce document qui est la dernière spécification sur les routeurs en mode IP version 4, est par ailleurs très utile pour comprendre les principes du routage actuel.

**Listing 5.12. Table de routage du routeur R1.**

```
R1#show ip route
...
  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O    10.0.0.6/32 [110/129] via 10.0.0.21, 19:19:57, Serial0
O    10.0.0.7/32 [110/65] via 10.0.0.21, 19:19:57, Serial0
C    10.0.0.5/32 is directly connected, Loopback0
O    10.0.0.24/30 [110/128] via 10.0.0.21, 19:19:57, Serial0
C    10.0.0.20/30 is directly connected, Serial0
C    10.0.0.64/26 is directly connected, Ethernet0
O    10.0.0.128/25 [110/138] via 10.0.0.21, 19:19:57, Serial0
O IA 200.0.0.0/8 [110/80] via 10.0.0.21, 00:25:30, Serial0
```

## Annonce de la route par défaut

Un routeur peut être configuré pour annoncer une route par défaut dans OSPF. À cet effet, la commande **default-information originate always** doit être entrée en mode de configuration **router ospf**. Ajoutons cette commande à la configuration du routeur R4 (cf. listing 5.13). Observons-en le résultat sur la table de routage du routeur R1 (cf. listing 5.14), où l'on voit que celui-ci a connaissance d'un routeur par défaut (*gateway of last resort* ou *default gateway*), indiqué à la première ligne en italique, et que la route par défaut elle-même est enregistrée avec le code « E2 » (qui veut dire route externe OSPF de type 2) sur l'avant dernière ligne en italique. La signification des types 1 et 2 sera précisée au chapitre 6.

**Listing 5.13. Configuration du routeur R4.**

```
interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router ospf 1
 network 200.0.0.0 0.255.255.255 area 0
 default-information originate always
```

© copyright Éditions Eyrolles

**Listing 5.14. Table de routage du routeur R1.**

```
R1#show ip route
...

Gateway of last resort is 10.0.0.21 to network 0.0.0.0

  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O    10.0.0.6/32 [110/129] via 10.0.0.21, 00:30:51, Serial0
O    10.0.0.7/32 [110/65] via 10.0.0.21, 00:30:51, Serial0
C    10.0.0.5/32 is directly connected, Loopback0
O    10.0.0.24/30 [110/128] via 10.0.0.21, 00:30:51, Serial0
C    10.0.0.20/30 is directly connected, Serial0
C    10.0.0.64/26 is directly connected, Ethernet0
O    10.0.0.128/25 [110/138] via 10.0.0.21, 00:30:51, Serial0
O*E2 0.0.0.0/0 [110/1] via 10.0.0.21, 00:19:27, Serial0
O IA 200.0.0.0/8 [110/80] via 10.0.0.21, 00:19:27, Serial0
```

## Affichage de la base de données d'état des liens

Pour visualiser la base de données OSPF d'état des liens, la commande **show ip ospf database** est utilisée. Examinons la sortie de cette commande sur le routeur R4 du listing 5.15.

**Listing 5.15. Base de données OSPF du routeur R4.**

```
R4#show ip ospf database

   OSPF Router with ID (200.0.0.1) (Process ID 1)


          Router Link States (Area 0)

Link ID   ADV Router  Age     Seq#       Checksum Link count
10.0.0.7  10.0.0.7    1627    0x80000003 0x92D8   1
200.0.0.1 200.0.0.1   786     0x80000005 0x70C1   3

          Net Link States (Area 0)

Link ID   ADV Router  Age     Seq#       Checksum
200.5.0.1 10.0.0.7    1627    0x80000001 0xDDA1

          Summary Net Link States (Area 0)

Link ID   ADV Router  Age     Seq#       Checksum
10.0.0.0  10.0.0.7    1382    0x80000001 0x696E

          Type-5 AS External Link States

Link ID   ADV Router  Age     Seq#       Checksum Tag
0.0.0.0   200.0.0.1   787     0x80000001 0x26C2   1
```

© copyright Éditions Eyrolles

Chaque aire est affichée individuellement. Le routeur R4 n'étant connecté qu'à la seule aire 0, la commande ne sort que son contenu avec les annonces d'état des liens ou LSA (*Link State Advertisement*) groupées par types tels qu'ils sont définis dans la table 5.1. Nous voyons sur ce listing 5.15 les quatre types de LSA, chacun comprenant les informations sur l'identité du lien (*link ID*), l'identité du routeur annonceur (*router ID*), l'âge du lien (*age*), le numéro de séquence et la somme de contrôle (*checksum*).

Pour plus d'explications sur le contenu de la base de données LSA de OSPF, veuillez consulter la RFC 2428.

## Configuration d'aires confinées dans OSPF

Selon les spécifications de OSPF, quelques aires peuvent être configurées en confinement ou *stub.* Ces aires confinées ne sont pas habilitées à recevoir des LSA de type 5, d'où leur appellation. En revanche, elles reçoivent une route par défaut sous le code « IA » qui signifie route inter-aires.

**REMARQUE** Si un système autonome ne reçoit pas beaucoup de LSA de type 5, il n'est pas toujours utile de configurer certaines de ses aires en confinement. Cependant dans le cas d'un système autonome relié au réseau Internet, les protocoles inter-domaines du genre EGP ( *Exterior Gateway Protocol*) et plus particulièrement BGP (*Border Gateway Protocol*) utilisés redistribuent leurs routes vers le domaine OSPF, ce qui nécessite de configurer certaines des aires de ce dernier en confinement (*stub*).

Pour configurer une aire OSPF en confinement, tous les routeurs appartenant à celle-ci doivent avoir la même commande **area** <*aire*> **stub** sous le mode de configuration **router ospf**. Le paramètre doit être renseigné avec le numéro d'aire.

Dans l'exemple du réseau de la figure 5.5, configurons l'aire 10 en confinement. Les listings 5.16 à 5.18 montrent les configurations des routeurs R1, R2 et R3.

**Listing 5.16. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.0.0.65 255.255.255.192

interface Serial0
 ip address 10.0.0.22 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 area 10 stub
```

**Listing 5.17. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.6 255.255.255.255

interface Ethernet0
 ip address 10.0.0.129 255.255.255.128

interface Serial1
```

© copyright Éditions Eyrolles

```
 ip address 10.0.0.26 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 area 10 stub
```

**Listing 5.18. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.25 255.255.255.252

interface Serial1
 ip address 10.0.0.21 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 network 200.0.0.0 0.255.255.255 area 0
 area 0 range 200.0.0.0 255.0.0.0
 area 10 stub
 area 10 range 10.0.0.0 255.255.255.0
```

Notons que sur le routeur R3 qui est un ABR, seule l'aire 10 est configurée en confinement.

**REMARQUE** Si un seul des routeurs d'une aire confinée ne porte pas la commande **area** <*aire*> **stub**, il ne verra aucune route disponible *via* OSPF.

Examinons la table de routage du routeur R1 sur le listing 5.19 pour constater que la route par défaut n'est plus codée comme externe.

**Listing 5.19. Table de routage du routeur R1.**

```
R1#show ip route
...
Gateway of last resort is 10.0.0.21 to network 0.0.0.0

  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O    10.0.0.6/32 [110/129] via 10.0.0.21, 00:02:35, Serial0
O    10.0.0.7/32 [110/65] via 10.0.0.21, 00:02:35, Serial0
C    10.0.0.5/32 is directly connected, Loopback0
O    10.0.0.24/30 [110/128] via 10.0.0.21, 00:02:35, Serial0
C    10.0.0.20/30 is directly connected, Serial0
C    10.0.0.64/26 is directly connected, Ethernet0
O    10.0.0.128/25 [110/138] via 10.0.0.21, 00:02:35, Serial0
O*IA 0.0.0.0/0 [110/65] via 10.0.0.21, 00:02:35, Serial0
O IA 200.0.0.0/8 [110/80] via 10.0.0.21, 00:02:35, Serial0
```

© copyright Éditions Eyrolles

Le système IOS de Cisco procure aussi un moyen de configurer une aire en confinement total (*totally stubby area*) qui est une extension propriétaire de Cisco. Une aire ainsi configurée ne recevra aucune route sous le code « IA », mais seulement une seule route par défaut pour toutes les adresses IP situées en dehors de cette aire.

**REMARQUE** La configuration de certaines aires en confinement total réduit le nombre de LSA que les routeurs auront à traiter.

La commande **area** <*aire*> **stub no-summary** doit être introduite sur le routeur ABR pour rendre une aire totalement confinée. Le listing 5.20 montre le routeur R3 configuré avec l'aire 10 en confinement total.

**Listing 5.20. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.7 255.255.255.255

interface Serial0
 ip address 10.0.0.25 255.255.255.252

interface Serial1
 ip address 10.0.0.21 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 10
 network 200.0.0.0 0.255.255.255 area 0
 area 0 range 200.0.0.0 255.0.0.0
 area 10 stub no-summary
 area 10 range 10.0.0.0 255.255.255.0
```

Si on examine la table de routage du routeur R1, on peut y voir cette fois-ci, qu'il n'y a qu'une seule route (celle par défaut) sous le code « IA » (cf. listing 5.21).

**Listing 5.21. Table de routage du routeur R1.**

```
R1#show ip route
...
Gateway of last resort is 10.0.0.21 to network 0.0.0.0

  10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
O    10.0.0.6/32 [110/129] via 10.0.0.21, 00:01:13, Serial0
O    10.0.0.7/32 [110/65] via 10.0.0.21, 00:01:13, Serial0
C    10.0.0.5/32 is directly connected, Loopback0
O    10.0.0.24/30 [110/128] via 10.0.0.21, 00:01:13, Serial0
C    10.0.0.20/30 is directly connected, Serial0
C    10.0.0.64/26 is directly connected, Ethernet0
O    10.0.0.128/25 [110/138] via 10.0.0.21, 00:01:13, Serial0
O*IA 0.0.0.0/0 [110/65] via 10.0.0.21, 00:01:13, Serial0
```

© copyright Éditions Eyrolles

## Liaisons virtuelles OSPF pour restaurer un réseau dorsal sectionné

Selon les spécifications de OSPF, il ne peut y avoir qu'une seule aire dorsale à laquelle devront être reliées toutes les autres aires par des ABR (*Area Border Router*). Mais ces spécifications prévoient cependant le cas où, pour une raison quelconque, l'aire dorsale est sectionnée. Pour y remédier, une technique spéciale connue sous le nom de « liaisons virtuelles » (*virtual links*) doit être utilisée pour reformer la connectivité initiale entre les différentes sections de l'aire dorsale.

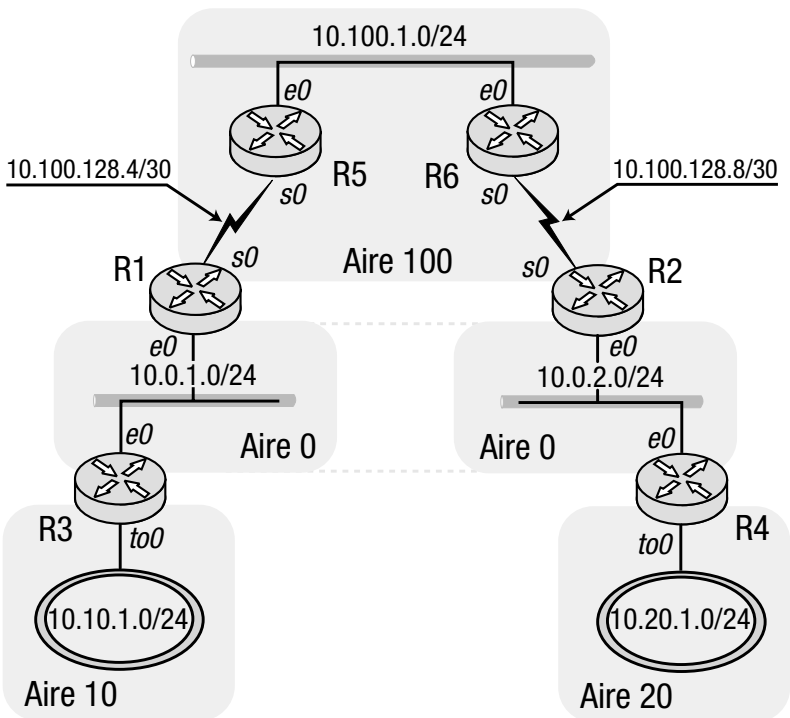
Pour utiliser les liaisons virtuelles, il doit y avoir une aire unique qui relie deux sections de l'aire dorsale. Une fois que cette aire a été définie, nous devons trouver deux routeurs dont chacun est relié à une section de l'aire dorsale et à un routeur de l'aire commune. Nous pouvons ensuite configurer sur ces deux routeurs les liaisons virtuelles par la commande **area** *<aire>* **virtual-link** *<OSPF ID>* pour restaurer la connectivité de l'aire dorsale. Le premier paramètre désigne l'aire commune, et le deuxième l'identité OSPF (*OSPF ID*) du routeur de l'autre section de l'aire dorsale.

**REMARQUE** L'identité OSPF n'est pas une adresse IP du routeur de l'autre section, mais il s'agit du numéro le plus élevé prélevé en priorité parmi les adresses IP configurées en rebouclage (*loopback*) sur ce routeur ou si celles-ci n'existent pas, parmi les adresses IP des interfaces de celui-ci.

Prenons le cas de la topologie du réseau illustrée sur la figure 5.6.

La dorsale OSPF (aire 0) est sectionnée en deux sections chacune reliée à une aire commune (aire 100) par les routeurs R1 et R2.

**Figure 5.6**
*Aire dorsale sectionnée.*




© copyright Éditions Eyrolles

Les règles d'adressage IP dans cet exemple sont les suivantes :

- Tous les réseaux ont un préfixe de longueur supérieure ou égale à 24.
- Chaque numéro d'aire est codé sur le deuxième octet de toutes les adresses IP qui lui appartiennent.
- Une adresse IP avec le troisième octet à 0 ne peut être assignée qu'à une interface en rebouclage.

Pour mesurer l'importance de sauvegarder la connectivité d'une aire dorsale, voyons d'abord ce qui se passe si les routeurs R1 et R2 sont configurés sans liaisons virtuelles entre leurs sections respectives. Les listings 5.22 à 5.27 montrent leur configuration actuelle.

**Listing 5.22. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
 ip address 10.100.128.5 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 100 range 10.100.0.0 255.255.0.0
```

**Listing 5.23. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Serial0
 ip address 10.100.128.9 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 100 range 10.100.0.0 255.255.0.0
```

**Listing 5.24. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Ethernet0
 ip address 10.0.1.2 255.255.255.0

interface TokenRing0
 ip address 10.10.1.1 255.255.255.0
```

© copyright Éditions Eyrolles

```
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
```

**Listing 5.25. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.0.2.2 255.255.255.0

interface TokenRing0
 ip address 10.20.1.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.20.0.0 0.0.255.255 area 20
 area 0 range 10.0.0.0 255.255.0.0
 area 20 range 10.20.0.0 255.255.0.0
```

**Listing 5.26. Configuration du routeur R5.**

```
interface Loopback0
 ip address 10.100.0.5 255.255.255.255

interface Ethernet0
 ip address 10.100.1.1 255.255.255.0

interface Serial0
 ip address 10.100.128.6 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 100
```

**Listing 5.27. Configuration du routeur R6.**

```
interface Loopback0
 ip address 10.100.0.6 255.255.255.255

interface Ethernet0
 ip address 10.100.1.2 255.255.255.0

interface Serial0
 ip address 10.100.128.10 255.255.255.252

router ospf 1
 network 10.0.0.0 0.255.255.255 area 100
```

Affichons la table de routage du routeur R3 par la commande **show ip route** dont la sortie se trouve sur le listing 5.28. C'est sans surprise que nous y constatons que le routeur R3 ne peut

© copyright Éditions Eyrolles

avoir connaissance du réseau tout entier. Par exemple l'aire 20 (adresse agrégée 10.20.0.0/16) est absente de sa table. Le sectionnement de l'aire dorsale en est la cause.

**Listing 5.28. Table de routage du routeur R3.**

```
R3#show ip route
...
  10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C    10.10.1.0/24 is directly connected, TokenRing0
C    10.0.0.3/32 is directly connected, Loopback0
O    10.0.0.1/32 [110/11] via 10.0.1.1, 00:14:18, Ethernet0
C    10.0.1.0/24 is directly connected, Ethernet0
O IA 10.100.0.0/16 [110/74] via 10.0.1.1, 00:14:18, Ethernet0
```

Introduisons la commande **area** <*aire*> **virtual-link** <*OSPF ID*> (en mode **router ospf 1**) sur les routeurs R1 et R2 pour restaurer la connectivité de l'aire dorsale. Les listings 5.29 et 5.30 montrent les configurations contenant cette commande.

**Listing 5.29. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
 ip address 10.100.128.5 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 100 range 10.100.0.0 255.255.0.0
 area 100 virtual-link 10.0.0.2
```

**Listing 5.30. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Serial0
 ip address 10.100.128.9 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 100 range 10.100.0.0 255.255.0.0
 area 100 virtual-link 10.0.0.1
```

Si nous examinons la table de routage sur le listing 5.31, nous voyons que le routeur R3 a maintenant connaissance des adresses agrégées de toutes les aires.

© copyright Éditions Eyrolles

**Listing 5.31. Table de routage du routeur R3 après configuration des liaisons virtuelles OSPF sur les routeurs R1 et R2.**

```
R3#show ip route
...
  10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
C    10.10.1.0/24 is directly connected, TokenRing0
O    10.0.0.2/32 [110/149] via 10.0.1.1, 01:55:46, Ethernet0
O    10.0.2.0/24 [110/158] via 10.0.1.1, 01:55:46, Ethernet0
C    10.0.0.3/32 is directly connected, Loopback0
O    10.0.0.1/32 [110/11] via 10.0.1.1, 01:55:46, Ethernet0
C    10.0.1.0/24 is directly connected, Ethernet0
O    10.0.0.4/32 [110/159] via 10.0.1.1, 01:55:46, Ethernet0
O IA 10.20.0.0/16 [110/164] via 10.0.1.1, 01:55:46, Ethernet0
O IA 10.100.0.0/16 [110/74] via 10.0.1.1, 01:55:46, Ethernet0
```

La commande **show ip ospf virtual-links** peut être utilisée pour vérifier l'état d'une liaison virtuelle comme le montre le listing 5.32. La ligne en italique y indique l'état actif (*up*) ou inactif (*down*) de la liaison. Pour plus de détails, se reporter à la documentation de Cisco.

**Listing 5.32. Sortie de la commande `show ip ospf virtual-links` sur le routeur R1.**

```
R1#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 10.0.0.2 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 100, via interface Serial0, Cost of using 138
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
Retransmit 5
    Hello due in 00:00:07
    Adjacency State FULL (Hello suppressed)
```

**REMARQUE** Bien que les liaisons virtuelles fassent partie des spécifications OSPF, leur utilisation doit se limiter uniquement aux cas d'urgence ou de déploiement à large échelle. Jamais en tant que solution permanente.

Voici un exemple de risque potentiel qui peut être associé à l'utilisation des liaisons virtuelles. Sans doute avez-vous remarqué dans l'exemple de la figure 5.6 que l'adresse agrégée de l'aire dorsale manquait dans les configurations des routeurs R1 et R2 (cf. listings 5.29 et 5.30). Nous allons savoir pourquoi, en ajoutant cette adresse agrégée aux configurations des deux routeurs comme sur les listings 5.33 et 5.34.

**Listing 5.33. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
 ip address 10.100.128.5 255.255.255.252
```

© copyright Éditions Eyrolles

```
router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 0 range 10.0.0.0 255.255.0.0
 area 100 range 10.100.0.0 255.255.0.0
 area 100 virtual-link 10.0.0.2
```

**Listing 5.34. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Serial0
 ip address 10.100.128.9 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.100.0.0 0.0.255.255 area 100
 area 0 range 10.0.0.0 255.255.0.0
 area 100 range 10.100.0.0 255.255.0.0
 area 100 virtual-link 10.0.0.1
```

Affichons à présent la table de routage du routeur R3 dont la sortie sur le listing 5.35 ressemble étrangement à celle d'avant. Pour mieux voir ce qu'il en est exactement, nous allons lancer un ping sur l'adresse de rebouclage du routeur R1. Le listing 5.36 en affiche le résultat qui paraît tout à fait normal.

**Listing 5.35. Table de routage du routeur R3.**

```
R3#show ip route
...
  10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
C    10.10.1.0/24 is directly connected, TokenRing0
O    10.0.0.2/32 [110/149] via 10.0.1.1, 00:02:43, Ethernet0
O    10.0.2.0/24 [110/158] via 10.0.1.1, 00:02:43, Ethernet0
C    10.0.0.3/32 is directly connected, Loopback0
O    10.0.0.1/32 [110/11] via 10.0.1.1, 00:02:43, Ethernet0
C    10.0.1.0/24 is directly connected, Ethernet0
O    10.0.0.4/32 [110/159] via 10.0.1.1, 00:02:43, Ethernet0
O IA 10.20.0.0/16 [110/164] via 10.0.1.1, 00:02:43, Ethernet0
O IA 10.100.0.0/16 [110/74] via 10.0.1.1, 00:02:43, Ethernet0
```

**Listing 5.36. Sortie de la commande `ping 10.0.0.1` sur le routeur R3.**

```
R3#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
4/4/4 ms
```

© copyright Éditions Eyrolles

Lançons maintenant un ping sur l'interface de rebouclage du routeur R2. Le résultat est affiché sur le listing 5.37.

**Listing 5.37. Sortie de la commande `ping 10.0.0.2` sur le routeur R3.**

```
R3#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is
2 seconds:
.....
Success rate is 0 percent (0/5)
```

Le problème ne semble pas provenir du routeur R3 qui possède bien une route pointant vers l'adresse 10.0.0.2. Utilisons la commande **traceroute** pour cette adresse sur ce routeur afin d'en savoir plus. Le résultat est affiché sur le listing 5.38.

**Listing 5.38. Sortie de la commande `traceroute 10.0.0.2` sur le routeur R3.**

```
R3#traceroute 10.0.0.2

Type escape sequence to abort.
Tracing the route to 10.0.0.2

  1 10.0.1.1 4 msec 4 msec 4 msec
  2 10.100.128.6 20 msec 16 msec 20 msec
  3 10.100.128.5 16 msec 20 msec 16 msec
  4 10.100.128.6 32 msec 32 msec 32 msec
  5 10.100.128.5 32 msec 32 msec 32 msec
  6 10.100.128.6 48 msec 44 msec 44 msec
  7 10.100.128.5 44 msec 48 msec 44 msec
  8 10.100.128.6 60 msec 60 msec 60 msec
  9 10.100.128.5 60 msec 60 msec 60 msec
 10 10.100.128.6 76 msec 76 msec 72 msec
...
```

On peut constater ci-dessus que les paquets générés par **traceroute** semblent boucler sur la ligne série qui relie les routeurs R1 et R5. Examinons la table de routage de ce dernier sur le listing 5.39.

**Listing 5.39. Table de routage du routeur R5.**

```
R5#show ip route
...
  10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA 10.10.0.0/16 [110/80] via 10.100.128.5, 00:50:30,Serial0
O IA 10.0.0.0/16 [110/65] via 10.100.128.5, 00:18:19,Serial0
O IA 10.20.0.0/16 [110/90] via 10.100.1.2, 03:04:14,Ethernet0
O    10.100.0.6/32 [110/11] via 10.100.1.2,03:04:37,Ethernet0
C    10.100.0.5/32 is directly connected, Loopback0
C    10.100.1.0/24 is directly connected, Ethernet0
O    10.100.128.8/30 [110/74] via 10.100.1.2, 03:04:37,
Ethernet0
C    10.100.128.4/30 is directly connected, Serial0
```

© copyright Éditions Eyrolles

La ligne en italique de la table de routage ci-dessus nous révèle sans surprise que le routeur R5 n'a qu'une seule route vers l'aire dorsale qui passe, par le routeur R1 plus proche de cette aire. Tout paquet en provenance du routeur R5 destiné à une adresse de l'aire dorsale doit passer par R1, même quand il s'agit de l'autre section, ce qui crée une boucle de routage, mise en évidence par la sortie de la commande **traceroute** du listing 5.38.

Le routeur R6 est dans le même cas vis à vis du routeur R2.

Si nous faisons un retour en arrière en enlevant des configurations des routeurs R1 et R2 l'adresse agrégée de l'aire dorsale, nous pouvons constater en lançant un ping du routeur R3 vers l'adresse de rebouclage du routeur R2, que cette commande aboutit bien comme le montre le listing 5.40.

**Listing 5.40. Résultat de la commande `ping 10.0.0.2` après retrait de l'adresse agrégée de l'aire dorsale dans les configurations des routeurs R1 et R2.**

```
R3#ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
56/61/72 ms
```

Bien que cette modification de la configuration puisse paraître minime sur les routeurs R1 et R2, nous devons en mesurer les conséquences. Il introduit une incohérence dans la configuration des routeurs. R3 et R4 possèdent une adresse agrégée de l'aire dorsale alors que les routeurs R1 et R2 n'en ont pas. Plus grave encore, le retrait de la configuration de ces derniers de l'adresse agrégée, annule la raison d'être du routage hiérarchique.

Si nous examinons la table de routage du routeur R5 sur le listing 5.41 (après retrait de l'adresse agrégée de l'aire dorsale des routeurs R1 et R2), nous y voyons apparaître toutes les routes de l'aire dorsale sans que ce routeur soit directement connecté à celle-ci. Cela vient du fait que le routeur R1, n'étant plus configuré pour agréger les routes de l'aire dorsale, les envoie toutes vers l'aire 100 du routeur R5. Le modèle hiérarchique devient donc caduc.

**Listing 5.41. Table de routage du routeur R5 après retrait de l'adresse agrégée des routeurs R1 et R2.**

```
R5#show ip route
...
     10.0.0.0/8 is variably subnetted, 13 subnets, 4 masks
O IA    10.10.0.0/16 [110/80] via 10.100.128.5, 01:14:17, Serial0
O IA    10.0.2.0/24 [110/84] via 10.100.1.2, 00:08:58, Ethernet0
O IA    10.0.0.2/32 [110/75] via 10.100.1.2, 00:08:58, Ethernet0
O IA    10.0.0.3/32 [110/75] via 10.100.128.5, 00:08:58, Serial0
O IA    10.0.1.0/24 [110/74] via 10.100.128.5, 00:08:58, Serial0
O IA    10.0.0.1/32 [110/65] via 10.100.128.5, 00:08:58, Serial0
O IA    10.0.0.4/32 [110/85] via 10.100.1.2, 00:08:58, Ethernet0
O IA    10.20.0.0/16 [110/90] via 10.100.1.2, 03:28:02, Ethernet0
O       10.100.0.6/32 [110/11] via 10.100.1.2, 03:28:24, Ethernet0
C       10.100.0.5/32 is directly connected, Loopback0
```

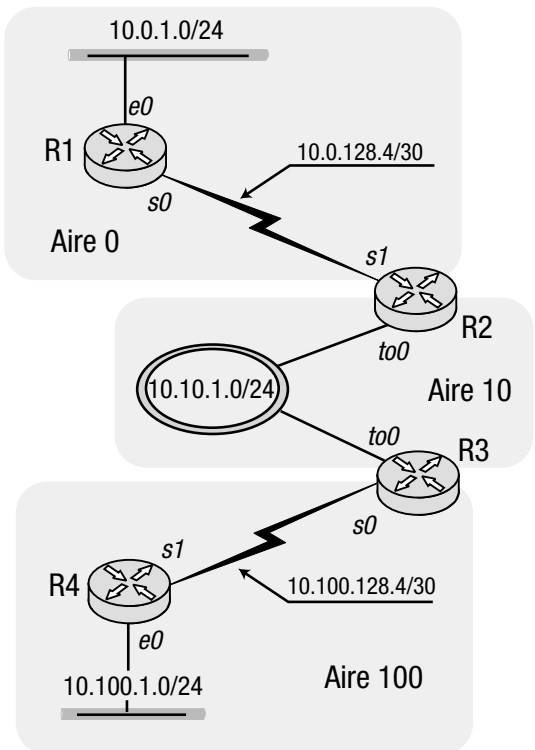
© copyright Éditions Eyrolles

```
C       10.100.1.0/24 is directly connected, Ethernet0
O       10.100.128.8/30 [110/74] via 10.100.1.2, 03:28:24, Ethernet0
C       10.100.128.4/30 is directly connected, Serial0
```

## Liaisons virtuelles OSPF pour relier des aires isolées

Les liaisons virtuelles OSPF peuvent aussi être utilisés pour relier des aires qui ne peuvent être connectées qu'en passant par l'aire dorsale. Une liaison virtuelle entre un ABR et un routeur d'aire isolée qui ont tous les deux une connexion à une aire intermédiaire, permet l'extension de l'aire dorsale *via* l'aire intermédiaire vers l'aire isolée. Dans ce cas, le routeur d'aire isolée connecté à l'aire intermédiaire devient à son tour un ABR. Voir figure 5.7.

**Figure 5.7**
*Aire 100 reliée à l'aire dorsale que via l'aire 10.*




L'aire 100 ne peut être reliée à l'aire dorsale que *via* l'aire 10. Voyons ce qui se passe si les routeurs R2 et R3 ne sont pas configurés avec une liaison virtuelle. Les listings 5.42 à 5.45 montrent les configurations de tous ces routeurs.

**Listing 5.42. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.0.128.5 255.255.255.252

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
```

**Listing 5.43. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Serial1
 ip address 10.0.128.6 255.255.255.252

interface TokenRing0
 ip address 10.10.1.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
```

**Listing 5.44. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.10.0.3 255.255.255.255

interface Serial0
 ip address 10.100.128.5 255.255.255.252

interface TokenRing0
 ip address 10.10.1.2 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.10.0.0 0.0.255.255 area 10
 network 10.100.0.0 0.0.255.255 area 100
 area 10 range 10.10.0.0 255.255.0.0
 area 100 range 10.100.0.0 255.255.0.0
```

**Listing 5.45. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.100.0.4 255.255.255.255

interface Ethernet0
 ip address 10.100.1.1 255.255.255.0

interface Serial1
 ip address 10.100.128.6 255.255.255.252

router ospf 1
 network 10.100.0.0 0.0.255.255 area 100
```

© copyright Éditions Eyrolles

Les tables de routage des routeurs R1 et R4 sont affichées sur les listings 5.46 et 5.47.

**Listing 5.46. Table de routage du routeur R1.**

```
R1#show ip route
...
  10.0.0.0/8 is variably subnetted, 5 subnets, 4 masks
O IA 10.10.0.0/16 [110/71] via 10.0.128.6, 00:13:01, Serial0
O    10.0.0.2/32 [110/65] via 10.0.128.6, 00:13:01, Serial0
C    10.0.1.0/24 is directly connected, Ethernet0
C    10.0.0.1/32 is directly connected, Loopback0
C    10.0.128.4/30 is directly connected, Serial0
```

**Listing 5.47. Table de routage du routeur R4.**

```
R4#show ip route
...
  10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C    10.100.0.4/32 is directly connected, Loopback0
C    10.100.1.0/24 is directly connected, Ethernet0
C    10.100.128.4/30 is directly connected, Serial1
```

Visiblement le routage ne se fait pas correctement sans la liaison virtuelle. Le routeur R1 ignore l'existence de l'aire 100 tout entière, tandis que le routeur R4 ne peut voir que les réseaux qui lui sont directement connectés.

Ajoutons maintenant une liaison virtuelle entre les routeurs R2 et R3 comme indiqué sur les listings 5.48 et 5.49.

**Listing 5.48. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Serial1
 ip address 10.0.128.6 255.255.255.252

interface TokenRing0
 ip address 10.10.1.1 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
 area 10 virtual-link 10.10.0.3
```

**Listing 5.49. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.10.0.3 255.255.255.255

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.100.128.5 255.255.255.252

interface TokenRing0
 ip address 10.10.1.2 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.10.0.0 0.0.255.255 area 10
 network 10.100.0.0 0.0.255.255 area 100
 area 10 range 10.10.0.0 255.255.0.0
 area 10 virtual-link 10.0.0.2
 area 100 range 10.100.0.0 255.255.0.0
```

Si nous affichons maintenant la table de routage des routeurs R1 et R4, elles auront l'aspect des listings 5.50 et 5.51.

**Listing 5.50. Table de routage du routeur R1.**

```
R1#show ip route
...
  10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
O IA 10.10.0.0/16 [110/71] via 10.0.128.6, 00:02:12, Serial0
O    10.0.0.2/32 [110/65] via 10.0.128.6, 00:02:12, Serial0
C    10.0.1.0/24 is directly connected, Ethernet0
C    10.0.0.1/32 is directly connected, Loopback0
O IA 10.100.0.0/16 [110/144] via 10.0.128.6, 00:02:12,Serial0
C    10.0.128.4/30 is directly connected, Serial0
```

**Listing 5.51. Table de routage du routeur R4.**

```
R4#show ip route
...
  10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
O IA 10.10.0.0/16 [110/70] via 10.100.128.5, 00:01:58,Serial1
O IA 10.0.0.2/32 [110/71] via 10.100.128.5, 00:01:44,Serial1
O IA 10.0.0.1/32 [110/135] via 10.100.128.5, 00:01:44,Serial1
O IA 10.0.1.0/24 [110/144] via 10.100.128.5, 00:01:44,Serial1
C    10.100.0.4/32 is directly connected, Loopback0
C    10.100.1.0/24 is directly connected, Ethernet0
O IA 10.0.128.4/30 [110/134]via 10.100.128.5,00:01:44,Serial1
C    10.100.128.4/30 is directly connected, Serial1
```

Le routage semble mieux fonctionner, sans être parfait. La table de routage du routeur R1 est correcte, avec une route d'adresse agrégée vers l'aire 100. Il reste à améliorer encore la table du routeur R4 qui est remplie avec chaque route individuelle de l'aire dorsale sans que ce routeur y soit directement connecté. On peut expliquer facilement ce phénomène si on se rappelle qu'une liaison virtuelle est une extension de l'aire dorsale qui va, dans ce cas, jusqu'au routeur R3 qui ne pratique pas l'agrégation de route pour cette aire. Ce qui nous amène à la conclusion que la commande **area 0 range** doit être introduite dans la configuration du routeur R3. Ajoutons cette commande sur ce routeur comme indiqué sur le listing 5.52.

La table de routage du routeur R4 sur le listing 5.53 elle aussi est correcte.

© copyright Éditions Eyrolles

**Listing 5.52. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.10.0.3 255.255.255.255

interface Serial0
 ip address 10.100.128.5 255.255.255.252

interface TokenRing0
 ip address 10.10.1.2 255.255.255.0

 ring-speed 16

router ospf 1
 network 10.10.0.0 0.0.255.255 area 10
 network 10.100.0.0 0.0.255.255 area 100
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0
 area 10 virtual-link 10.0.0.2
 area 100 range 10.100.0.0 255.255.0.0
```

**Listing 5.53. Table de routage du routeur R4.**

```
R4#show ip route
...
  10.0.0.0/8 is variably subnetted, 5 subnets, 4 masks
O IA 10.10.0.0/16 [110/70] via 10.100.128.5, 00:31:38,Serial1
O IA 10.0.0.0/16 [110/144] via 10.100.128.5, 00:04:38,Serial1
C    10.100.0.4/32 is directly connected, Loopback0
C    10.100.1.0/24 is directly connected, Ethernet0
C    10.100.128.4/30 is directly connected, Serial1
```

## Configuration de OSPF sur des réseaux NBMA

La configuration des routeurs Cisco avec OSPF, reliés par des réseaux de type NBMA tel que Frame Relay, a toujours été considérée comme une tâche ardue, alors qu'elle le serait moins, si avant d'y procéder, on se rappelait les principes de fonctionnement de ce protocole. Nous allons aborder ci-après différentes méthodes pour configurer OSPF sur des routeurs Cisco interconnectés *via* Frame Relay.

### Configuration de OSPF sur un réseau NBMA intégralement maillé

Les sections suivantes proposent deux méthodes pour configurer OSPF sur un réseau Frame Relay intégralement maillé (*fully meshed*). On se servira à ce propos du schéma de réseau de la figure 5.8.

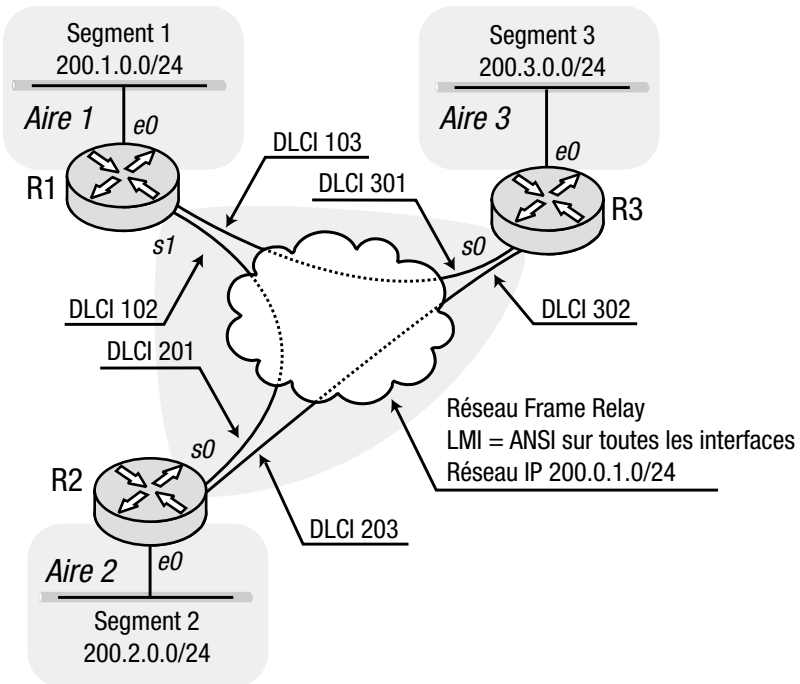
Comme nous l'avons dit dans l'introduction de ce chapitre, les réseaux NBMA, et plus particulièrement Frame Relay, sont représentés dans la base de données OSPF d'état des liens, comme des réseaux de diffusion générale à accès multiples, tout comme un LAN. En d'autres termes, un routeur désigné et son suppléant (DR et BDR) sont élus et servent de porte-parole aux autres routeurs sur le réseau qui est lui-même un nœud du graphe.

© copyright Éditions Eyrolles

Il y cependant un inconvénient à cette représentation car les routeurs OSPF s'appuient sur le protocole de communication *hello* pour se découvrir mutuellement et former un lien de proximité. Ce protocole utilise l'adressage par diffusion multidestinataire (*multicast*), qui peut ne pas être disponible dans un réseau NBMA, auquel cas, les routeurs sont incapables de former le lien de proximité pour un routage correct. Nous allons voir ci-après comment corriger cette déficience.

**Figure 5.8**

*Routeurs connectés par un réseau Frame Relay intégralement maillé.*




### La commande *neighbor*

Pour aider les routeurs OSPF dans un réseau NBMA à former un lien de proximité entre eux, nous pouvons configurer les adresses IP de leurs voisins sur chacun, ce qui leur permettrait de ne plus dépendre de la diffusion multidestinataire.

Sous le mode de configuration **router ospf**, nous devons utiliser la commande **neighbor** *<adresse IP distante>* autant de fois que ce routeur a de voisins.

**REMARQUE** Tous les routeurs voisins ne forment pas un lien de proximité ; cela est particulièrement vrai quand, dans un seul réseau à accès multiples, on a plus de deux routeurs connectés.

Les listings 5.54 à 5.56 montrent les configurations des routeurs selon le schéma de réseau de la figure 5.8. Chaque routeur est configuré avec deux voisins (accessibles *via* Frame Relay).

**Listing 5.54. Configuration du routeur R1.**

```
ip subnet-zero

interface Loopback0
```

© copyright Éditions Eyrolles

```
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.0.1.1 255.255.255.248
 encapsulation frame-relay
 frame-relay map ip 200.0.1.2 102 broadcast
 frame-relay map ip 200.0.1.3 103 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.1.0.0 0.0.255.255 area 1
 neighbor 200.0.1.3
 neighbor 200.0.1.2
 area 0 range 200.0.0.0 255.255.0.0
 area 1 range 200.1.0.0 255.255.0.0

ip classless
```

**Listing 5.55. Configuration du routeur R2.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.2 255.255.255.255

interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.2 255.255.255.248
 encapsulation frame-relay
 frame-relay map ip 200.0.1.1 201 broadcast
 frame-relay map ip 200.0.1.3 203 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.2.0.0 0.0.255.255 area 2
 neighbor 200.0.1.3
 neighbor 200.0.1.1
 area 0 range 200.0.0.0 255.255.0.0
 area 2 range 200.2.0.0 255.255.0.0

     ip classless
```

**Listing 5.56. Configuration du routeur R3.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.3 255.255.255.255
```

© copyright Éditions Eyrolles

```
interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.3 255.255.255.248
 encapsulation frame-relay
 frame-relay map ip 200.0.1.1 301 broadcast
 frame-relay map ip 200.0.1.2 302 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.3.0.0 0.0.255.255 area 3
 neighbor 200.0.1.1
 neighbor 200.0.1.2
 area 0 range 200.0.0.0 255.255.0.0
 area 3 range 200.3.0.0 255.255.0.0

ip classless
```

**REMARQUE** Toutes les configurations actuelles contiennent deux commandes jamais utilisées auparavant. Il s'agit de **ip classless** à laquelle on a déjà fait allusion ; elle est nécessaire pour passer de l'algorithme de routage à classe à celui de sans classe, quand des super-réseaux doivent être agrégés en 200.0.0.0/16, par exemple. L'autre commande **ip subnet-zero** est introduite pour utiliser des adresses de sous-réseau ne comportant que des zéros. Nous avons besoin de cette commande par exemple pour le réseau 200.0.1.0/29 auquel appartiennent les interfaces Frame Relay des routeurs de la figure 5.8.

La table de routage du routeur R2 est affichée sur le listing 5.57.

**Listing 5.57. Table de routage du routeur R2.**

```
R2#show ip route
...
200.0.0.0/32 is subnetted, 3 subnets
O       200.0.0.1 [110/65] via 200.0.1.1, 00:07:55, Serial0
C       200.0.0.2 is directly connected, Loopback0
O       200.0.0.3 [110/65] via 200.0.1.3, 00:07:55, Serial0
C    200.2.0.0/24 is directly connected, Ethernet0
     200.0.1.0/29 is subnetted, 1 subnets
C       200.0.1.0 is directly connected, Serial0
O IA 200.1.0.0/16 [110/74] via 200.0.1.1, 00:07:55, Serial0
O IA 200.3.0.0/16 [110/74] via 200.0.1.3, 00:07:55, Serial0
```

Comme nous pouvions nous y attendre, le routeur R2 voit maintenant les adresses agrégées des aires 1 et 3. La commande **show ip ospf neighbor** permet de vérifier sur un routeur l'état de disponibilité de ses voisins. Les listings 5.58 à 5.60 montrent la sortie de cette commande sur les routeurs R1, R2 et R3.

**Listing 5.58. Voisins OSPF du routeur R1.**

```
R1#show ip ospf neighbor

Neighbor ID  Pri  State        Dead Time Address    Interface
200.0.0.3      1  FULL/DR      00:01:53  200.0.1.3  Serial1
200.0.0.2      1  FULL/BDR     00:01:49  200.0.1.2  Serial1
```

© copyright Éditions Eyrolles

**Listing 5.59. Voisins OSPF du routeur R2.**

```
R2#show ip ospf neighbor

Neighbor ID  Pri  State        Dead Time Address    Interface
200.0.0.3      1  FULL/DR      00:01:33  200.0.1.3  Serial0
200.0.0.1      1  FULL/DROTHER 00:01:58  200.0.1.1  Serial0
```

**Listing 5.60. Voisins OSPF du routeur R3.**

```
R3#show ip ospf neighbor

Neighbor ID  Pri  State        Dead Time Address    Interface
200.0.0.1      1  FULL/DROTHER 00:01:48  200.0.1.1  Serial0
200.0.0.2      1  FULL/BDR     00:01:50  200.0.1.2  Serial0
```

Notons sur les listings ci-dessus comment est mis en évidence le rôle fonctionnel des routeurs voisins. DR (*Designated Router*) est le routeur désigné ; BDR (*Backup DR*) est son suppléant. DROTHER signifie que ce routeur voisin est « autre », c'est-à-dire ni DR, ni BDR, et qu'il doit former un lien de proximité avec eux.

**REMARQUE** Pour de plus amples informations sur les états possibles des routeurs voisins OSPF, se reporter aux spécifications de OSPF (cf. RFC 2328) et à la documentation de Cisco.

### La commande ip ospf network broadcast

Bien que les réseaux NBMA ne permettent pas l'adressage par diffusion générale ou par diffusion multidestinataire à la couche accès réseau, les routeurs peuvent toujours utiliser la diffusion générale au moins à la couche Internet.

Nous avons déjà eu l'occasion d'appliquer la commande **frame-relay map ip** *<adresse IP distante> <DLCI>* qui comporte l'option [**broadcast**]. Si cette dernière est utilisée, le routeur envoie le trafic destiné à la diffusion générale (adresse : 255.255.255.255) sur le CVP (Circuit Virtuel Permanent) normalement utilisé pour le trafic monodestinataire. Si plusieurs CVP sont configurés sur une interface série ou sa sous-interface, ce trafic est envoyé sur tous les CVP dont le numéro est suivi par le mot clef **broadcast** dans cette commande. Celui-ci tout seul ne suffit cependant pas à OSPF pour commencer à traiter l'interface comme étant connectée à un réseau de diffusion générale à accès multiples. Pour ce faire, la commande supplémentaire, **ip ospf network broadcast** doit être ajoutée (en mode de configuration d'interface) à la place de **neighbor (**en mode de configuration **router)** du cas précédent. Les configurations révisées des routeurs se trouvent sur les listings 5.61 à 5.63.

**Listing 5.61. Configuration du routeur R1.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.1.0.1 255.255.255.0
```

© copyright Éditions Eyrolles

```
interface Serial1
 ip address 200.0.1.1 255.255.255.248
 encapsulation frame-relay
 ip ospf network broadcast
 frame-relay map ip 200.0.1.2 102 broadcast
 frame-relay map ip 200.0.1.3 103 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.1.0.0 0.0.255.255 area 1
 area 0 range 200.0.0.0 255.255.0.0
 area 1 range 200.1.0.0 255.255.0.0

ip classless
```

**Listing 5.62. Configuration du routeur R2.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.2 255.255.255.255

interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.2 255.255.255.248
 encapsulation frame-relay
 ip ospf network broadcast
 frame-relay map ip 200.0.1.1 201 broadcast
 frame-relay map ip 200.0.1.3 203 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.2.0.0 0.0.255.255 area 2
 area 0 range 200.0.0.0 255.255.0.0
 area 2 range 200.2.0.0 255.255.0.0

ip classless
```

**Listing 5.63. Configuration du routeur R3.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.3 255.255.255.255

interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.3 255.255.255.248
 encapsulation frame-relay
```

© copyright Éditions Eyrolles

```
 ip ospf network broadcast
 frame-relay map ip 200.0.1.1 301 broadcast
 frame-relay map ip 200.0.1.2 302 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.3.0.0 0.0.255.255 area 3
 area 0 range 200.0.0.0 255.255.0.0
 area 3 range 200.3.0.0 255.255.0.0

ip classless
```

Dans tous ces listings, les lignes et les mots en italique indiquent les emplacements de la nouvelle commande et du mot clef optionnel.

Le listing 5.64 affiche la table de routage du routeur R2 qui est la même que celle du listing 5.57 du cas précédent.

**Listing 5.64. Table de routage du routeur R2.**

```
R2#show ip route
...
     200.0.0.0/32 is subnetted, 3 subnets
O       200.0.0.1 [110/65] via 200.0.1.1, 00:07:34, Serial0
C       200.0.0.2 is directly connected, Loopback0
O       200.0.0.3 [110/65] via 200.0.1.3, 00:07:34, Serial0
C    200.2.0.0/24 is directly connected, Ethernet0
     200.0.1.0/29 is subnetted, 1 subnets
C       200.0.1.0 is directly connected, Serial0
O IA 200.1.0.0/16 [110/74] via 200.0.1.1, 00:07:34, Serial0
O IA 200.3.0.0/16 [110/74] via 200.0.1.3, 00:07:34, Serial0
```

Les listings 5.65 à 5.67 affichent la sortie de la commande **show ip ospf neighbor** sur les trois routeurs. Aucun changement par rapport au cas précédent.

**Listing 5.65. Voisins OSPF du routeur R1.**

```
R1#show ip ospf neighbor

Neighbor ID Pri  State        Dead Time  Address    Interface
200.0.0.3     1  FULL/DR      00:00:39   200.0.1.3  Serial1
200.0.0.2     1  FULL/BDR     00:00:38   200.0.1.2  Serial1
```

**Listing 5.66. Voisins OSPF du routeur R2.**

```
R2#show ip ospf neighbor

Neighbor ID Pri  State        Dead Time  Address    Interface
200.0.0.1     1  FULL/DROTHER 00:00:39   200.0.1.1  Serial0
200.0.0.3     1  FULL/DR      00:00:36   200.0.1.3  Serial0
```

© copyright Éditions Eyrolles

**Listing 5.67. Voisins OSPF du routeur R3.**

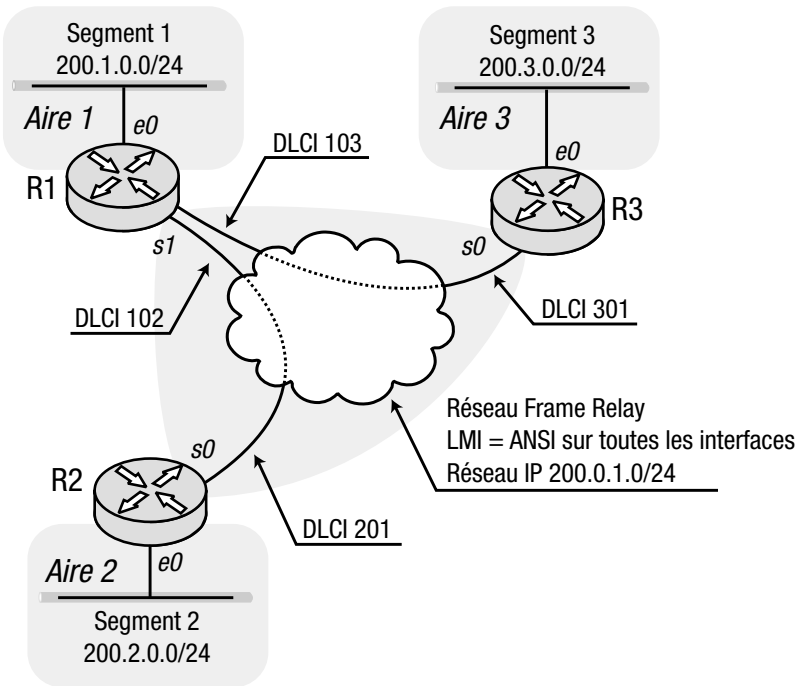
```
R3#show ip ospf neighbor

Neighbor ID Pri  State        Dead Time  Address    Interface
200.0.0.1     1  FULL/DROTHER 00:00:32   200.0.1.1  Serial0
200.0.0.2     1  FULL/BDR     00:00:39   200.0.1.2  Serial0
```

### Configuration de OSPF sur un réseau NBMA non intégralement maillé

Pour faciliter la tâche de configuration des routeurs Cisco avec OSPF sur un réseau NBMA non intégralement maillé, considérée comme la plus difficile, il est utile d'avoir bien assimilé les règles de base sur lesquelles s'appuie ce protocole.

Les trois sections suivantes proposent chacune une méthode pour configurer OSPF sur des routeurs Cisco, en se référant au schéma de réseau illustré à la figure 5.9.

**Figure 5.9**
*Routeurs connectés par un réseau Frame Relay non intégralement maillé.*




#### Utilisation de sous-interfaces

Cette méthode est la plus simple, sans doute aussi la meilleure. Qu'il s'agisse de réseau intégralement ou non intégralement maillé, il est toujours possible d'assigner des CVP à des sous-interfaces de routeurs, chacune ayant une adresse IP appartenant à un sous-réseau séparé. Du point de vue de OSPF, chaque paire de sous-interfaces connectée *via* un CVP de Frame Relay devient une liaison point à point.

Les listings 5.68 à 5.70 montrent les configurations des trois routeurs de la figure 5.9.

© copyright Éditions Eyrolles

**Listing 5.68. Configuration du routeur R1.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial1.2 point-to-point
 description connects R1 to R2 via PVC 102
 ip address 200.0.1.1 255.255.255.252
 frame-relay interface-dlci 102

interface Serial1.3 point-to-point
 description connects R1 to R3 via PVC 103
 ip address 200.0.1.5 255.255.255.252
 frame-relay interface-dlci 103

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.1.0.0 0.0.255.255 area 1
 area 0 range 200.0.0.0 255.255.0.0
 area 1 range 200.1.0.0 255.255.0.0

ip classless
```

**Listing 5.69. Configuration du routeur R2.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.2 255.255.255.255

interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 200.0.1.2 255.255.255.252
 frame-relay interface-dlci 201

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.2.0.0 0.0.255.255 area 2
 area 0 range 200.0.0.0 255.255.0.0
 area 2 range 200.2.0.0 255.255.0.0

ip classless
```

© copyright Éditions Eyrolles

**Listing 5.70. Configuration du routeur R3.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.3 255.255.255.255

interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 200.0.1.6 255.255.255.252
 frame-relay interface-dlci 301

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.3.0.0 0.0.255.255 area 3
 area 0 range 200.0.0.0 255.255.0.0
 area 3 range 200.3.0.0 255.255.0.0

ip classless
```

Le listing 5.71 affiche la table de routage du routeur R2. Ce routeur peut voir les deux aires distantes comme avant mais cette fois-ci *via* la même sous-interface du routeur R1 à laquelle il est relié.

**Listing 5.71. Table de routage du routeur R2.**

```
R2#show ip route
...
     200.0.0.0/32 is subnetted, 3 subnets
O      200.0.0.1 [110/65] via 200.0.1.1, 00:12:55, Serial0.1
C      200.0.0.2 is directly connected, Loopback0
O      200.0.0.3 [110/129] via 200.0.1.1, 00:12:55, Serial0.1
C    200.2.0.0/24 is directly connected, Ethernet0
     200.0.1.0/30 is subnetted, 2 subnets
C      200.0.1.0 is directly connected, Serial0.1
O      200.0.1.4 [110/128] via 200.0.1.1, 00:12:55, Serial0.1
O IA 200.1.0.0/16 [110/74] via 200.0.1.1, 00:12:55, Serial0.1
O IA 200.3.0.0/16 [110/138] via 200.0.1.1, 00:12:55,Serial0.1
```

Les sorties de la commande **show ip ospf neighbor** (cf. listings 5.72 à 5.74) donnent l'état des voisins de chaque routeur qui est FULL/ -, ce qui signifie qu'aucun d'eux n'a été désigné comme DR ou BDR, conséquence des liaisons point à point.

**Listing 5.72. Voisins OSPF du routeur R1.**

```
R1#show ip ospf neighbor

Neighbor ID  Pri   State     Dead Time   Address    Interface
200.0.0.2      1   FULL/  -  00:00:30    200.0.1.2  Serial1.2
200.0.0.3      1   FULL/  -  00:00:30    200.0.1.6  Serial1.3
```

© copyright Éditions Eyrolles

**Listing 5.73. Voisins OSPF du routeur R2.**

```
R2#show ip ospf neighbor

Neighbor ID  Pri   State     Dead Time   Address    Interface
200.0.0.1      1   FULL/  -  00:00:35    200.0.1.1  Serial0.1
```

**Listing 5.74. Voisins OSPF du routeur R3.**

```
R3#show ip ospf neighbor

Neighbor ID  Pri   State     Dead Time   Address    Interface
200.0.0.1      1   FULL/  -  00:00:36    200.0.1.5  Serial0.1
```

**REMARQUE** Les sous-interfaces peuvent être configurées en multipoint au lieu du point à point. Mais il faut ajouter à la configuration, comme dans le cas des interfaces simples, soit la commande **neighbor**, soit la commande **ip ospf network broadcast**.

### La commande ip ospf network point-to-multipoint

La méthode ci-après ne nécessite pas de sous-interfaces. Néanmoins elle repose sur la même astuce selon laquelle chaque CVP est considéré comme une liaison point à point. Pour que le processus OSPF du routeur traite une interface Frame Relay configurée avec plusieurs CVP, comme autant de liaisons point à point, la commande **ip ospf network point-to-multipoint** doit être introduite en mode de configuration d'interface. Ces liaisons point à point n'existent que dans les structures internes au processus OSPF. Cette procédure est semblable à celle qui utilise la commande **ip ospf network broadcast**, telle qu'on peut la voir sur les listings 5.75 à 5.77.

**Listing 5.75. Configuration du routeur R1.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.0.1.1 255.255.255.248
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 frame-relay map ip 200.0.1.2 102 broadcast
 frame-relay map ip 200.0.1.3 103 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.1.0.0 0.0.255.255 area 1
 area 0 range 200.0.0.0 255.255.0.0
 area 1 range 200.1.0.0 255.255.0.0

ip classless
```

© copyright Éditions Eyrolles

**Listing 5.76. Configuration du routeur R2.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.2 255.255.255.255

interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.2 255.255.255.248
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 frame-relay map ip 200.0.1.1 201 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.2.0.0 0.0.255.255 area 2
 area 0 range 200.0.0.0 255.255.0.0
 area 2 range 200.2.0.0 255.255.0.0

ip classless
```

**Listing 5.77. Configuration du routeur R3.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.3 255.255.255.255

interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.3 255.255.255.248
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 frame-relay map ip 200.0.1.1 301 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.3.0.0 0.0.255.255 area 3
 area 0 range 200.0.0.0 255.255.0.0
```

Le listing 5.78 de la table de routage du routeur R2 ainsi que les listings 5.79 à 5.80 qui montrent l'état des voisins de chaque routeur, sont les mêmes que précédemment ; ces derniers indiquent en outre qu'il s'agit de liaisons point à point (aucun DR ou BDR élu).

© copyright Éditions Eyrolles

**Listing 5.78. Table de routage du routeur R2.**

```
R2#show ip route
...
  200.0.0.0/32 is subnetted, 3 subnets
O    200.0.0.1 [110/65] via 200.0.1.1, 00:39:15, Serial0
C    200.0.0.2 is directly connected, Loopback0
O    200.0.0.3 [110/129] via 200.0.1.1, 00:39:15, Serial0
C 200.2.0.0/24 is directly connected, Ethernet0
  200.0.1.0/24 is variably subnetted, 3 subnets, 2 masks
O    200.0.1.1/32 [110/64] via 200.0.1.1, 00:39:15, Serial0
C    200.0.1.0/29 is directly connected, Serial0
O    200.0.1.3/32 [110/128] via 200.0.1.1, 00:39:15, Serial0
O IA 200.1.0.0/16 [110/74] via 200.0.1.1, 00:38:37, Serial0
O IA 200.3.0.0/16 [110/138] via 200.0.1.1, 00:38:37, Serial0
```

**Listing 5.79. Sortie de la commande `show ip ospf neighbor` sur le routeur R1.**

```
R1#show ip ospf neighbor

Neighbor ID Pri  State     Dead Time   Address    Interface
200.0.0.2     1  FULL/  -  00:01:40    200.0.1.2  Serial1
200.0.0.3     1  FULL/  -  00:01:50    200.0.1.3  Serial1
```

**Listing 5.80. Sortie de la commande `show ip ospf neighbor` sur le routeur R2.**

```
R2#show ip ospf neighbor

Neighbor ID Pri  State     Dead Time   Address    Interface
200.0.0.1     1  FULL/  -  00:01:57    200.0.1.1  Serial0
```

**REMARQUE** La commande qu'on vient de décrire est valable dans n'importe quel réseau NBMA. La topologie de la figure 5.9 avec deux routeurs connectés par des CVP à un routeur central n'est qu'un cas parmi d'autres.

### Priorités dans les routeurs OSPF

Un réseau NBMA n'a pas à être considéré comme ayant de multiples liaisons point à point. Même quand il n'est pas intégralement maillé, il est possible de lui affecter un routeur en tant que DR.

Le trait principal d'un DR, c'est le lien de proximité que tous les autres routeurs doivent former avec lui. De ce point de vue, le routeur R1 de la figure 5.9 semble le parfait candidat pour devenir ce routeur désigné.

Mais le DR doit être élu par les autres routeurs. Et *a priori* nous ne savons pas lequel parmi les trois de la figure 5.9 sera choisi.

Fort opportunément, les spécifications de OSPF tiennent compte de ce problème en donnant la possibilité d'attribuer aux routeurs une priorité qui influence leur choix. C'est le routeur à la plus haute priorité, valeur configurable par l'administrateur réseau, qui l'emporte. Le système IOS de Cisco en conformité avec ces spécifications dispose de la commande **ip ospf priority** <*priorité*> pour influencer le choix d'un routeur plutôt qu'un autre. Si nous ne voulons pas qu'un routeur soit désigné, nous devons lui assigner la priorité 0. Dans notre cas,

© copyright Éditions Eyrolles

les routeurs R2 et R3 auront donc cette priorité, de façon à favoriser le routeur R1 qui, quant à lui, aura la priorité 10 pour être choisi comme DR.

Les réseaux à accès multiples qui ne disposent pas de la diffusion générale doivent préciser les voisins OSPF manuellement. Nous utilisons à cet effet la commande **neighbor** en mode de configuration **router ospf** (cf. listing 5.81), comme nous l'avons déjà fait dans le cas des réseaux NBMA intégralement maillés.

Dans les listings de configuration 5.81 à 5.83 les nouvelles commandes sont marquées par des lignes en italique.

**Listing 5.81. Configuration du routeur R1.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.1 255.255.255.255

interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial1
 ip address 200.0.1.1 255.255.255.248
 encapsulation frame-relay
 ip ospf network non-broadcast
 ip ospf priority 10
 frame-relay map ip 200.0.1.2 102 broadcast
 frame-relay map ip 200.0.1.3 103 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.1.0.0 0.0.255.255 area 1
 neighbor 200.0.1.2
 neighbor 200.0.1.3
 area 0 range 200.0.0.0 255.255.0.0
 area 1 range 200.1.0.0 255.255.0.0

ip classless
```

**Listing 5.82. Configuration du routeur R2.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.2 255.255.255.255

interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.2 255.255.255.248
 encapsulation frame-relay
```

© copyright Éditions Eyrolles

```
 ip ospf network non-broadcast
 ip ospf priority 0
 frame-relay map ip 200.0.1.1 201 broadcast
 frame-relay map ip 200.0.1.3 201 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.2.0.0 0.0.255.255 area 2
 area 0 range 200.0.0.0 255.255.0.0
 area 2 range 200.2.0.0 255.255.0.0
 ! neighbor 200.0.1.1  is no longer required.

ip classless
```

**Listing 5.83. Configuration du Routeur R3.**

```
ip subnet-zero

interface Loopback0
 ip address 200.0.0.3 255.255.255.255

interface Ethernet0
 ip address 200.3.0.1 255.255.255.0

interface Serial0
 ip address 200.0.1.3 255.255.255.248
 encapsulation frame-relay
 ip ospf priority 0
 frame-relay map ip 200.0.1.1 301 broadcast
 frame-relay map ip 200.0.1.2 301 broadcast
 frame-relay lmi-type ansi

router ospf 1
 network 200.0.0.0 0.0.255.255 area 0
 network 200.3.0.0 0.0.255.255 area 3
 area 0 range 200.0.0.0 255.255.0.0
 area 3 range 200.3.0.0 255.255.0.0
 ! neighbor 200.0.1.1  is no longer required.

ip classless
```

Les listings 5.84 à 5.86 de la commande **show ip ospf neighbor** sur les routeurs indiquent l'absence du BDR qui n'a pas été désigné. Et nous pouvons y voir aussi que le routeur R1 a bien été élu comme DR, ce que nous souhaitions.

**Listing 5.84. Sortie de la commande `show ip ospf neighbor` sur le routeur R1.**

```
R1#show ip ospf neighbor

Neighbor ID Pri  State         Dead Time Address    Interface
200.0.0.2     0  FULL/DROTHER  00:01:45  200.0.1.2  Serial1
200.0.0.3     0  FULL/DROTHER  00:01:47  200.0.1.3  Serial1
```

© copyright Éditions Eyrolles

**Listing 5.85. Sortie de la commande `show ip ospf neighbor` sur le routeur R2.**

```
R2#show ip ospf neighbor

Neighbor ID Pri  State         Dead Time Address    Interface
200.1.0.1    10  FULL/DR       00:01:45  200.0.1.1  Serial0
```

**Listing 5.86. Sortie de la commande `show ip ospf neighbor` sur le routeur R3.**

```
R3#show ip ospf neighbor

Neighbor ID Pri  State         Dead Time Address    Interface
200.1.0.1    10  FULL/DR       00:01:50  200.0.1.1  Serial0
```

La table de routage du routeur R2 sur le listing 5.87 montre que la connectivité globale est assurée comme avant. Bien que l'aire 3 soit toujours accessible, cette fois-ci c'est le routeur R3 qui est celui du saut suivant, et non plus le routeur R1. Celui-ci, en tant que DR, se base sur un réseau censé relier directement tous les routeurs, et fait transiter l'annonce des routes disponibles dans l'aire 3 par le routeur R3 lui-même.

Pour que le routage se fasse correctement, nous devons donc fournir cette connexion « directe », en ajoutant la commande supplémentaire **frame-relay map ip** *<adresse IP distante> <DLCI>*. Voir listings 5.82 et 5.83 (lignes en italique).

**Listing 5.87. Table de routage du routeur R2.**

```
R2#show ip route
...
     200.0.0.0/32 is subnetted, 3 subnets
O       200.0.0.1 [110/65] via 200.0.1.1, 00:16:52, Serial0
C       200.0.0.2 is directly connected, Loopback0
O       200.0.0.3 [110/65] via 200.0.1.3, 00:16:52, Serial0
C    200.2.0.0/24 is directly connected, Ethernet0
     200.0.1.0/29 is subnetted, 1 subnets
C       200.0.1.0 is directly connected, Serial0
O IA 200.1.0.0/16 [110/74] via 200.0.1.1, 00:16:52, Serial0
O IA 200.3.0.0/16 [110/74] via 200.0.1.3, 00:16:52, Serial0
```

**AVERTISSEMENT** Cette dernière méthode pour configurer OSPF sur un réseau NBMA ne doit jamais être utilisée dans un réseau opérationnel. Elle n'est pas conforme aux spécifications OSPF qui nécessitent que soient assurées dans les réseaux à accès multiples, la connectivité directe entre tous les routeurs, la désignation d'un BDR, etc. Dans cet ouvrage, cette méthode n'est incluse que dans un but pédagogique.

© copyright Éditions Eyrolles

# 6

# Maîtrise du flux de données et des mises à jour de routage

**Solutions de configuration présentées dans ce chapitre**



© copyright Éditions Eyrolles

Il est parfois nécessaire d'implanter plusieurs protocoles de routage sur un même routeur. Faire communiquer deux réseaux d'entreprise utilisant respectivement IGRP et RIP nécessite par exemple d'installer des routeurs multiprotocole entre ces réseaux. Autre cas de figure nécessitant des routeurs multiprotocole, celui d'un réseau basé sur IGRP englobant des serveurs Unix qui requièrent d'obtenir les informations de routage complètes directement, au lieu de les obtenir par le routeur par défaut (*default gateway*). La plupart des hôtes capables d'effectuer du routage opèrent avec des protocoles ouverts tels que RIP. Les informations de routage IGRP qui leur parviennent doivent donc être préalablement converties en format RIP par des routeurs.

Pour permettre l'échange et la conversion d'informations de routage entre différents protocoles exécutés sur le même routeur, celui-ci doit être explicitement configuré à cet effet. Un tel échange n'est pas automatique pour la simple raison que les protocoles qui se déroulent dans un même routeur ont des usages différents ; qu'ils s'échangent des informations n'est ni prévu ni souhaitable. En outre, les protocoles calculent leur métrique différemment, souvent d'une manière incompatible avec les autres. RIP par exemple se base sur le nombre de sauts pour sa métrique, tandis que des protocoles plus perfectionnés tels que IGRP et EIGRP utilisent une formule complexe, dans laquelle interviennent des paramètres de débit, de délai, etc. Ces métriques étant incompatibles, il faut convertir leurs valeurs. Ajoutons enfin qu'il est impossible – du fait de la codification des adresses IP – de passer d'un protocole sans classe à un protocole à classe sans conversion, l'inverse étant possible.

## Redistribution d'informations de routage

La conversion des informations de routage pour les transmettre entre différentes sources d'informations de routage s'appelle la « redistribution d'informations de routage » ou plus simplement « redistribution ». Il peut s'agir de protocoles de routage dynamique, statique ou de routage d'interfaces connectées. Ces deux dernières sources peuvent redistribuer leurs informations vers le protocole de routage dynamique sans que l'inverse soit possible. Chaque fois qu'une information de routage passe d'un protocole à un autre, elle est dite « redistribuée » vers ce protocole.

Pour qu'une redistribution soit effective, on doit préciser le protocole source des informations de routage, le protocole d'arrivée, et la méthode de conversion de métrique de la source avant son transfert. La redistribution n'est pas toujours bidirectionnelle, les informations de routage peuvent passer d'un protocole à un autre sans que l'inverse soit vrai.

Quand des machines Unix, par exemple, sont connectées à un réseau où elles ont besoin d'informations de routage précises, celles-ci doivent être redistribuées vers le protocole qu'elles sont susceptibles de comprendre, RIP. En même temps, il n'est pas du tout souhaitable que les routeurs de ce réseau apprennent en retour les informations de routage que ces machines peuvent diffuser. Si celles-ci sont multidomiciliées (possédant plusieurs interfaces réseau), elles peuvent assurer par défaut une fonction de routage inter-réseaux, souvent ignorée, et également peu performante s'agissant de machines de traitement non dédiées. Elles pourront de ce fait annoncer des routes *semblant* meilleures que celles des vrais routeurs, et à tort prises en compte. Pour toutes ces raisons, interdire aux routeurs la prise en compte des informations de routage en provenance de ces sources est obligatoire.

La conversion de métrique est un autre élément important dans une redistribution. Le système IOS de Cisco n'implémente que deux protocoles (IGRP et EIGRP) qui peuvent s'échanger leurs métriques sans perte du coût calculé. Les métriques de tous les autres protocoles sont incompatibles entre elles et ne peuvent donc pas être converties. La redistribution de toute

© copyright Éditions Eyrolles

route dans ces derniers cas se fait par assignation d'une valeur statique à la métrique, faisant perdre celle qui était accumulée dans le protocole source, lors du transfert vers le protocole récepteur.

Dans cet ouvrage, un réseau continu au sein duquel un protocole donné propage les mises à jour de routage et dont la métrique augmente, sera appelé son « domaine de métrique ». Si cette métrique doit être revalorisée, on a atteint la frontière de ce domaine de métrique.

Par exemple, des réseaux interconnectés dont les adresses de sous-réseau appartiennent à un même réseau desservi par un protocole à classe et où les routes apprises ont une métrique qui exprime un coût adéquat, relèvent du domaine de métrique de ce protocole. Une aire du protocole OSPF est un autre exemple de domaine de métrique propre à celui-ci.

Nous pouvons reformuler cette importante propriété des métriques comme suit :

*Hormis le cas particulier de la redistribution entre IGRP et EIGRP, les domaines de métrique des protocoles de routage sont délimités par les routeurs qui pratiquent la redistribution.*

## Redistribution d'informations de routage filtrées

La redistribution doit souvent se faire de manière contrôlée en déterminant les routes autorisées à passer d'un protocole vers un autre. C'est là qu'intervient le filtrage des informations de routage selon des critères que doit satisfaire toute route pour être éligible à la redistribution.

Le filtrage le plus simple consiste à comparer le préfixe réseau de la route avec un agencement de bits avant de la passer à un autre protocole. Des filtrages plus élaborés peuvent impliquer la comparaison d'autres facteurs tels que la métrique, l'étiquette de route, etc.

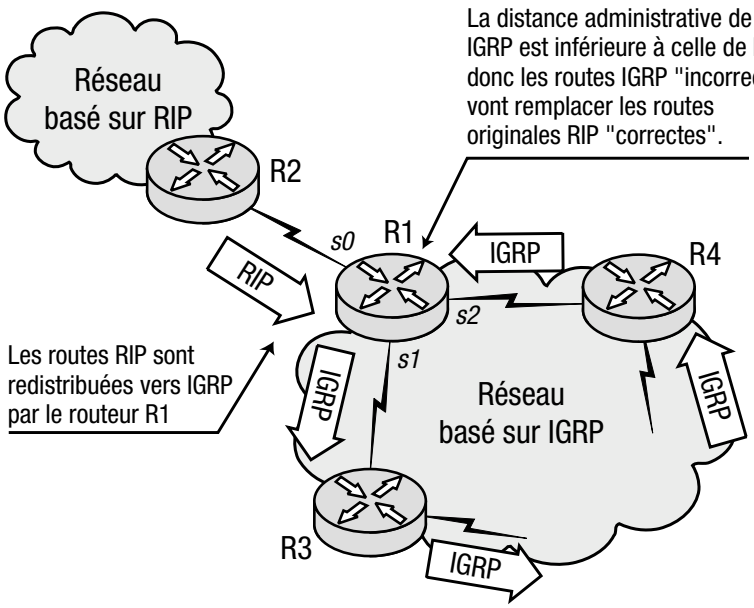
## Redistribution et son risque potentiel

Bien que la redistribution soit souvent inévitable, elle n'est pas toujours souhaitable. De nombreux problèmes peuvent en résulter, le pire étant la boucle de routage.

Pour mieux comprendre ce phénomène, considérons la situation illustrée sur la figure 6.1.

**Figure 6.1**

*Boucle de routage induite par la redistribution de RIP vers IGRP.*




© copyright Éditions Eyrolles

Le réseau est composé de deux domaines, l'un desservi par RIP, et l'autre par IGRP. Supposons que le routeur R2 du domaine RIP échange des informations de routage *via* ce protocole avec R1 qui les redistribue en IGRP dans son propre domaine basé sur ce dernier protocole. Si le routeur R1 y possède plus d'une connexion, une boucle de routage peut se former. Les mises à jour de routage IGRP que ce routeur envoie par l'une de ses interfaces peuvent lui revenir par une deuxième interface après avoir traversé le domaine IGRP. À l'origine, le routeur R1 a appris les routes transportées dans ces mises à jour par RIP, et elles ont été enregistrées dans sa table de routage avec la distance administrative correspondante. Quand ces mêmes routes lui reviennent, elles portent la distance administrative de IGRP, qui est inférieure à celle de RIP. Ces routes de retour vont donc supplanter celles de RIP qui se trouvaient dans la table du routeur R1, pointant maintenant vers le domaine IGRP lui-même, et supprimant ainsi la connectivité avec celui de RIP.

La boucle n'est pas perpétuelle car l'un des routeurs du domaine IGRP finira par augmenter la métrique de ces routes factices, provoquant une avalanche de mises à jour déclenchées qui feront passer la métrique de ces routes à la valeur infinie, ce qui les placera sous temporisation de maintien. Une fois celle-ci écoulée, la connectivité avec le domaine RIP sera rétablie pour un laps de temps et le cycle recommencera tant qu'aucune mesure ne sera prise pour le prévenir.

Il existe aussi d'autres circonstances de redistribution qui peuvent entraîner ce phénomène de boucle et nous en donnerons un certain nombre d'exemples dans la suite de ce chapitre.

La boucle de routage n'est pas un phénomène inéluctable et peut ne jamais survenir. Dans l'introduction du chapitre 4, nous avons passé en revue les règles appliquées par les routeurs lors de l'annonce de leurs mises à jour. L'une d'elles stipule qu'un protocole à classe qui n'est pas à l'origine d'une route installée dans la table de routage ne doit pas la diffuser. Dans le cas de la figure 6.1, la mise à jour de IGRP à son retour ne parviendra probablement jamais sur l'interface serial 2 du routeur R1. Quand la mise à jour IGRP distante d'un saut atteint le routeur R4, celui-ci a déjà une route installée pour ce préfixe dans sa table, apprise *via* IGRP en provenance du routeur R1. Cette route existante ayant une métrique inférieure à celle de la route candidate, ne sera pas remplacée par cette dernière. Ce qui devrait empêcher le routeur R4 d'annoncer des routes factices vers le routeur R1. Mais celui-ci peut tomber en panne et avoir oublié lors de son redémarrage les routes apprises quand il était en activité, l'incitant à prendre en compte les mauvaises routes venant du routeur R4, pour produire ce phénomène de boucle.

La redistribution est à éviter autant que possible ou si elle est incontournable, une bonne préparation devrait éviter les problèmes évoqués ci-dessus.

# Solutions de configuration

## Filtrage de trafic avec listes d'accès

Une liste d'accès est une série d'expressions conditionnelles d'autorisation (*permit*) ou d'interdiction (*deny*) qui, appliquées à certaines caractéristiques du trafic, produit un résultat logique vrai ou faux.

La liste d'accès peut servir à différents usages comme, par exemple, filtrer le trafic de données. Par exemple, les datagrammes IP transportant des données utilisateur sont mis au rebut s'ils font l'objet d'une condition d'interdiction qui est vérifiée dans la liste d'accès.

© copyright Éditions Eyrolles

Plusieurs lignes d'expressions conditionnelles peuvent être élaborées pour constituer une liste d'accès avec chacune comportant une autorisation ou une interdiction. Quand le routeur applique cette liste sur un trafic, il trouve une correspondance vérifiée pour l'une des lignes de la liste et ignore le reste immédiatement.

### Listes d'accès standard

La liste d'accès standard ne permet de comparer que l'adresse IP source à un agencement de bits particulier. Pour configurer une telle liste, les étapes sont les suivantes :

1. Création d'une liste d'accès par la commande **access-list** *<numéro de liste entre 1 et 99>* {**permit**|**deny**} *<adresse IP source/masque générique>*. Le masque générique (en notation décimale pointée) est constitué de bits dont chaque position à 1 signifie que le bit correspondant dans l'adresse IP source doit être ignoré lors de l'application de ce masque. Il s'agit d'une convention inverse à celle d'un masque de réseau ou sous-réseau. Le dernier paramètre peut être remplacé par le mot clef **host** suivi du paramètre *<adresse IP>* qui équivaut à *<adresse IP>* **0.0.0.0** correspondant à une adresse IP individuelle. Si l'adresse IP source est indifférente, le mot clef **any** peut être utilisé, ayant la même signification que **0.0.0.0 255.255.255.255**, qui correspond à toute adresse IP.

   Après une liste d'accès contenant ou non plusieurs lignes définies par la commande **access-list** *<numéro de liste entre 1 et 99>***,** le routeur insère automatiquement cette même commande suivie des mots clefs **deny any** (« exclure tout »), mais non visible dans la configuration.

   Le routeur passe en revue une liste d'accès jusqu'à trouver la première correspondance, après laquelle il arrête sa recherche. Il renvoie alors la condition **permit** ou **deny** correspondante.

**AVERTISSEMENT** Une liste d'accès qui comprend plusieurs lignes ne peut être effacée que globalement par la commande inverse **no access-list** *<numéro de liste à effacer>*. Avant de modifier une liste d'accès, toute référence la concernant doit être désactivée sur les interfaces. La référence à une liste d'accès inexistante peut produire un résultat imprévisible suivant le contexte d'exécution du système IOS de Cisco.

2. En mode de configuration d'interface, par la commande **ip access-group** *<numéro de liste>* [{**in**|**out**}], affecter la liste d'accès à l'interface choisie. Les mots clefs **in** et **out** permettent de filtrer le trafic en entrée ou en sortie, respectivement. Si les mots clefs sont omis, la liste s'applique à la sortie.

**REMARQUE** Il est indispensable de définir les lignes dont la correspondance s'avère vraie plus souvent en tête de liste d'accès, de façon à diligenter l'exécution. Par exemple, si l'on ne doit autoriser qu'un seul hôte d'un sous-réseau particulier à accéder *via* un routeur, à un service situé sur un autre sous-réseau, la liste doit commencer par l'adresse individuelle de cet hôte précédée du mot clef **permit**. L'interdiction se fait ensuite par le mot clef **deny** pour toutes les autres adresses du sous-réseau de cet hôte.

En tenant compte de la remarque ci-dessus, examinons sur le listing 6.1 comment le routeur R de la figure 6.2 est configuré pour n'autoriser que l'hôte H3 à accéder à l'hôte H1 situé sur le Token Ring.

© copyright Éditions Eyrolles

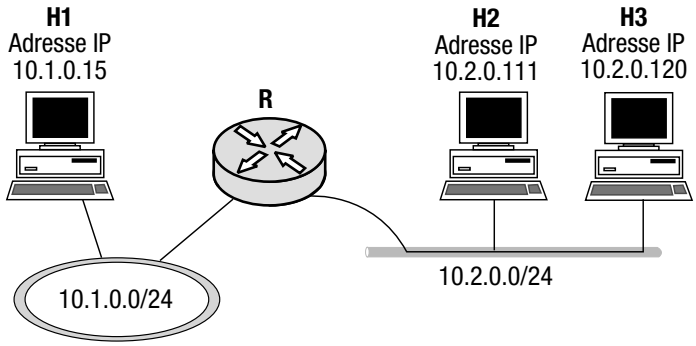



**Figure 6.2**
*Routeur R configuré pour n'autoriser que l'hôte H3 à communiquer avec l'hôte H1.*

**Listing 6.1. Configuration du routeur R.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface TokenRing0
 ip address 10.1.0.1 255.255.255.0
 ip access-group 1 out
 ring-speed 16

access-list 1 permit 10.2.0.120
access-list 1 deny   10.2.0.0 0.0.0.255
access-list 1 permit any
```

La dernière ligne du listing 6.1 n'est pas utile, mais elle a été néanmoins incluse au cas où d'autres segments connectés au routeur R auraient besoin d'accéder à l'hôte H1.

La commande **ping** envers l'hôte H1 à partir de l'hôte H2 n'aboutit pas. Comme on peut le voir sur le listing 6.2, les paquets ping ne sont pas simplement mis au rebut. Si c'était le cas, le message d'erreur serait « *Request timed out* » ( temporisation de requête écoulée) au lieu de « *destination unreachable* » (destination inaccessible) qui est affiché en sortie.

**Listing 6.2. Sortie de la commande `ping 10.1.0.15` sur l'hôte H2.**

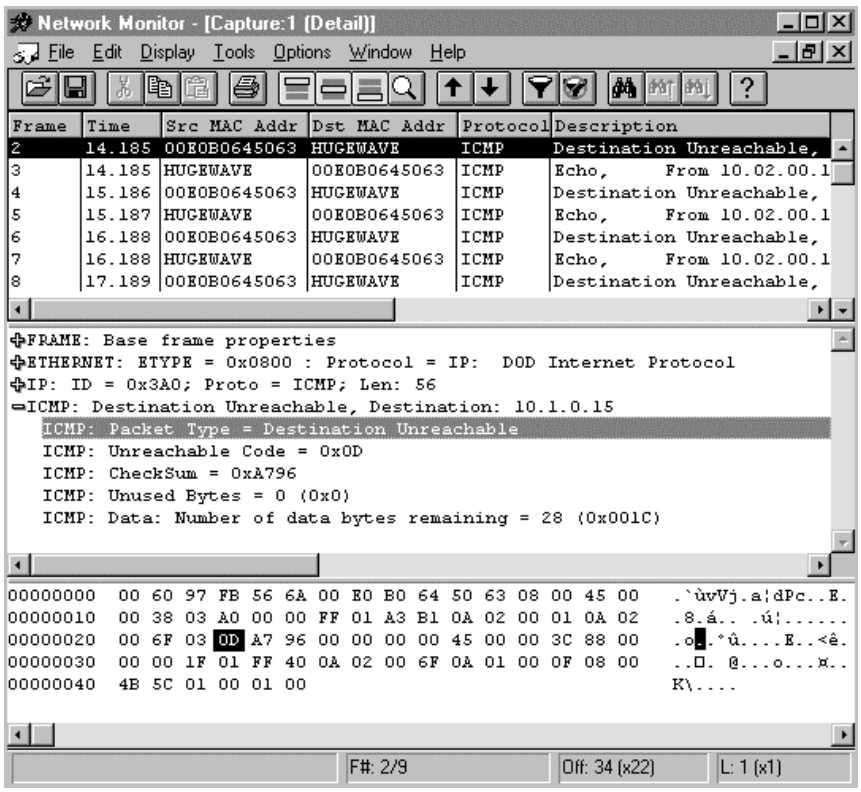
```
C:\>ping 10.1.0.15

Pinging 10.1.0.15 with 32 bytes of data:

Reply from 10.2.0.1: Destination net unreachable.
Reply from 10.2.0.1: Destination net unreachable.
Reply from 10.2.0.1: Destination net unreachable.
Reply from 10.2.0.1: Destination net unreachable.
```

Sur la figure 6.3, on peut voir sur la fenêtre de l'analyseur de protocole (*Microsoft Network Monitor*) utilisé en même temps que l'exécution de la commande **ping** sur l'hôte H2, qu'il reçoit en retour à ses paquets ping, le message d'erreur de type 3 du protocole ICMP qui correspond à « *destination unreachable* » (destination inaccessible), et dont le code hexadécimal 0x0D (13 en décimal) ne correspond à aucune valeur répertoriée dans la table 1.5 du chapitre 1.

© copyright Éditions Eyrolles

**Figure 6.3**

*La fenêtre principale du Network Monitor affiche la trace de la trame ICMP qui contient le message d'erreur de type 3 « destination unreachable » envoyé par le routeur R à l'hôte H2.*




### Listes d'accès étendues

Pour un contrôle d'accès plus granulaire, les listes d'accès étendues permettent de filtrer les paquets au niveau de la couche transport du modèle OSI (les ports source et destination), tout en précisant l'adresse IP de destination, ainsi que d'autres paramètres.

La procédure est la même que pour la liste d'accès standard, avec la syntaxe suivante :

**access-list** *<numéro de liste entre 100 et 199>* {**permit**|**deny**} *<protocole sur IP> <adresse IP source/masque générique>* [*<opérateur>*[*<port>*]] *<adresse IP destination/masque générique>* [*<opérateur>* [*<port>*]] [**established**] [**log**]

Le protocole sur IP est un nombre entre 0 et 255 ou son nom équivalent parmi ceux du tableau 6.1 ; l'opérateur est une fonction conditionnelle sous la forme **eq** (égal à), **lt** (plus petit que), **gt** (plus grand que), **ne** (non égal à) ou **range** (inclusif). Si ce dernier mot clef est utilisé, il doit être suivi de deux numéros de port. Le mot clef **established** n'est valable que pour TCP ; le dernier mot clef **log** permet de préciser si les correspondances trouvées pour la liste d'accès doivent être consignées sur le serveur syslog.

**REMARQUE** Le numéro de port peut aussi être remplacé par son nom. Par exemple, telnet au lieu de 23.

Seuls les protocoles qui utilisent directement IP sont pris en compte. Par exemple, RIP n'apparaît pas dans le tableau 6.1 car il utilise UDP.

Pour les options propres à chaque protocole, se reporter à la documentation de Cisco.

© copyright Éditions Eyrolles

**Tableau 6.1. Nom de protocoles pour liste d'accès étendue.**

| Nom | Description |
|---|---|
| Ip | Tout protocole (c'est-à-dire que le champ protocole ne sera pas analysé) |
| tcp | Transmission Control Protocol |
| udp | User Datagram Protocol |
| icmp | Internet Control Message Protocol |
| igrp | Interior Gateway Routing Protocol de Cisco |
| eigrp | Enhanced IGRP (version améliorée de IGRP) |
| ospf | Open Shortest Path First protocol |
| gre | Protocole de mise en tunnel Cisco |
| ipinip | Protocole de mise en tunnel ip dans ip |
| igmp | Inernet Gateway Message Protocol |
| nos | Protocole de mise en tunnel ip dans ip compatible KA9Q NOS |
| ahp | Protocole d'authentification d'en-tête (Athentification Header Protocol) |
| esp | Encapsulation sécuritaire de la charge utile (Encapsulation Security Payload) |
| pcp | Protocole de compression de charge utile (Payload Compression Protocol) |

Comme dans le cas de la liste d'accès standard, le mot clef **host** désigne l'adresse individuelle d'un hôte et le mot clef **any** équivaut à toute adresse ; une ligne implicite **access-list ip deny any** se trouve également en fin de liste.

**REMARQUE** Le tableau 6.1 est extrait du Cisco IOS 12.0 (2a). En fonction de la version utilisée, la liste des mots clés peut être différente.

Nous allons maintenant modifier la liste d'accès du réseau de la figure 6.2 pour n'autoriser que la session telnet et les services de jour (*daytime*) de l'hôte H1 vers l'hôte H3. Toute autre communication dans le réseau doit être complètement disponible.

Le listing 6.3 donne un moyen parmi d'autres de définir la liste d'accès (les quatre dernières lignes en italique) selon les directives citées plus haut. Elle est appliquée en entrée du trafic de l'interface Token Ring par la commande **ip access-group** <*numéro de liste*> **in**.

**Listing 6.3. Configuration du routeur R.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface TokenRing0
 ip address 10.1.0.1 255.255.255.0
 ip access-group 100 in
 ring-speed 16

access-list 100 permit tcp any host 10.2.0.120 eq telnet
access-list 100 permit tcp any host 10.2.0.120 eq daytime
access-list 100 deny   ip any host 10.2.0.120
access-list 100 permit ip any any
```

© copyright Éditions Eyrolles

La commande **ping** vers l'hôte H3 à partir de l'hôte H1 échoue, comme on peut le voir sur le listing 6.4.

**Listing 6.4. Liste d'accès étendue appliquée au trafic entrant sur l'interface Token Ring du routeur R qui interdit la commande `ping` de l'hôte H1 vers l'hôte H3.**

```
C:\>ping 10.2.0.120

Pinging 10.2.0.120 with 32 bytes of data:

Reply from 10.1.0.1: Destination net unreachable.
Reply from 10.1.0.1: Destination net unreachable.
Reply from 10.1.0.1: Destination net unreachable.
Reply from 10.1.0.1: Destination net unreachable.
```

Mais si nous ouvrons une session telnet de l'hôte H1 vers l'hôte H3, la communication est possible, comme le montre le listing 6.5.

**Listing 6.5. Pas de blocage du trafic telnet entre les hôtes H1 et H3.**

```
C:\>telnet 10.2.0.120
Trying 10.2.0.120...
Connected to 10.2.0.120.  Escape key is Ctrl-].

Welcome to the Telnet Service on THUNDER

Username:
```

Ni la session telnet, ni la commande **ping** ne sont bloquées, quand elles sont utilisées de l'hôte H1 vers l'hôte H2 (cf. listing 6.6).

**Listing 6.6. Session telnet et commande `ping` accessibles à partir de l'hôte H1 vers l'hôte H2.**

```
C:\>ping 10.2.0.111

Pinging 10.2.0.111 with 32 bytes of data:

Reply from 10.2.0.111: bytes=32 time<10ms TTL=127
Reply from 10.2.0.111: bytes=32 time<10ms TTL=127
Reply from 10.2.0.111: bytes=32 time<10ms TTL=127
Reply from 10.2.0.111: bytes=32 time<10ms TTL=127

C:\>telnet 10.2.0.111
Trying 10.2.0.111...
Connected to 10.2.0.111.  Escape key is Ctrl-].

Welcome to the Telnet Service on HUGEWAVE

Username:
```

© copyright Éditions Eyrolles

### Listes d'accès nommées

Un nouveau format de liste d'accès a été introduit à partir de la version 11.2 du système IOS de Cisco, appelé liste nommée, ce qui signifie qu'un nom peut être attribué à la liste tout entière au lieu d'un numéro. Les fonctionnalités restent cependant les mêmes.

Pour créer une liste d'accès nommée, on doit procéder aux étapes suivantes :

1. La commande **ip access-list** {**standard**|**extended**} <*nom de liste*> permet d'entrer en mode de configuration de liste.
2. Une fois qu'on se trouve sous le mode de configuration de liste, on définit une ou plusieurs lignes de contrôle d'accès suivant le format approprié qui dépend de l'option choisie dans la première étape : standard ou étendue (*extended*).

**REMARQUE** Une ligne implicite **access-list ip deny any** (**any**) est placée automatiquement en fin de liste comme pour les listes standard et étendue.

Nous utiliserons dans les exemples qui suivent, aussi bien les listes d'accès par numéro que par nom.

**REMARQUE** Il est possible de donner un numéro à la place d'un nom comme argument à la commande **ip access-list** ; dans ce cas, IOS reformate la liste en tant que liste standard ou étendue, suivant le numéro, avant de la stocker en RAM et NVRAM.

## Contrôle des mises à jour de routage

Les listes d'accès constituent un outil facilement adaptable à différentes situations. Nous avons évoqué plus haut leur utilisation pour filtrer le trafic de données. Ces mêmes listes peuvent aussi servir à filtrer les informations de routage, ce que nous allons voir dans les paragraphes suivants.

Le filtrage de mises à jour de routage permet de comparer un préfixe réseau en entrée ou en sortie d'une interface à la liste d'accès correspondante qui peut comporter une autorisation ou une interdiction. Suivant le cas, la mise à jour entrante est admise ou non ; celle sortante, est transmise ou non.

Pour le filtrage des mises à jour de routage, on procède de la manière suivante :

1. Créer une liste d'accès en standard ou étendue comme décrit auparavant.
2. En mode de configuration de routeur, par la commande **distribute-list** <*numéro de liste*> {**in**|**out**} <*interface*>, appliquer la liste définie précédemment à l'interface donnée en argument. Les mots clefs **in** (entrée) et **out** (sortie) permettent de déterminer le sens du filtre.

**REMARQUE** Il n'est pas conseillé d'appliquer un filtre en sortie pour des mises à jour envoyées par des protocoles à état des liens tel que OSPF car celui-ci nécessite d'envoyer la base de données d'état des liens dans son intégralité à ses voisins de proximité.

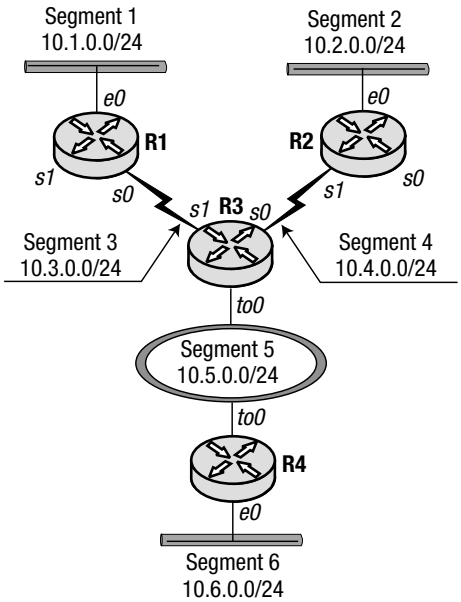
Prenons l'exemple du réseau illustré sur la figure 6.4. Tous les routeurs y sont configurés avec le protocole IGRP. Notre but est d'interdire au routeur R3 d'envoyer ses mises à jour concernant les liaisons série qui le relient aux routeurs R1 et R2 vers le routeur R4. Les listings 6.7 à 6.10 montrent la configuration de chacun de ces routeurs. Sur le listing 6.9, les lignes en

© copyright Éditions Eyrolles

italique mettent en évidence la définition de la liste d'accès et son enregistrement dans le processus de routage IGRP.

**Figure 6.4**

*Filtrage des mises à jour du routeur R3 en sortie de son interface Token Ring pour les réseaux 10.3.0.0/24 et 10.4.0.0/24.*




**Listing 6.7. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.8. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.9. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
```

© copyright Éditions Eyrolles

```
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 10.5.0.1 255.255.255.0
 ring-speed 16

router igrp 10
 network 10.0.0.0
 distribute-list 1 out TokenRing0

access-list 1 deny   10.3.0.0 0.0.255.255
access-list 1 deny   10.4.0.0 0.0.255.255
access-list 1 permit any
```

**Listing 6.10. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 10.6.0.1 255.255.255.0

interface TokenRing0
 ip address 10.5.0.2 255.255.255.0
 ring-speed 16

router igrp 10
 network 10.0.0.0
```

Si nous examinons la table de routage du routeur R4 sur le listing 6.11, nous y constatons qu'il ne reçoit aucune mise à jour pour les préfixes réseau 10.3.0.0/24 et 10.4.0.0/24.

**Listing 6.11. Table de routage du routeur R4.**

```
R4#show ip route
...
   10.0.0.0/24 is subnetted, 4 subnets
I     10.2.0.0 [100/8639] via 10.5.0.1, 00:00:35, TokenRing0
I     10.1.0.0 [100/8639] via 10.5.0.1, 00:00:35, TokenRing0
C     10.6.0.0 is directly connected, Ethernet0
C     10.5.0.0 is directly connected, TokenRing0
```

Si nous entrons la commande **debug ip igrp transactions** sur le routeur R3, nous voyons qu'il n'envoie pas les préfixes réseau des liaisons série comme en atteste l'extrait sur le listing 6.12.

**Listing 6.12. Sortie de la commande `debug ip igrp` transactions sur le routeur R3 indique que les préfixes 10.3.0.0/24 et 10.4.0.0/24 sont exclus des mises à jour envoyées sur l'interface Token Ring.**

```
...
IGRP: sending update to 255.255.255.255 via TokenRing0
(10.5.0.1)
      subnet 10.2.0.0, metric=8576
      subnet 10.1.0.0, metric=8576
```

© copyright Éditions Eyrolles

# La redistribution

Plusieurs sources d'informations de routage peuvent être redistribuées l'une vers l'autre. Les principales sont les protocoles dynamiques, les routes statiques et les routes d'interfaces connectées. Nous allons étudier comment se comportent ces sources quand il y a redistribution, la façon d'en contrôler les informations de routage, ainsi que les problèmes potentiels de boucle qui en résultent.

## La redistribution de base

Nous avons vu dans la première partie de ce chapitre que la redistribution consiste à convertir une information de routage d'une source de protocole à une autre, et inversement, le cas échéant. Cependant, certaines sources ne peuvent être redistribuées que dans un sens. Il s'agit de routes statiques et celles d'interfaces connectées dont la redistribution se fait vers un protocole de routage dynamique. La réciproque étant sans objet.

La commande pour la redistribution de base se fait par **redistribute** <*source de l'information de routage*> **metric** <*métrique spécifique au protocole*>, en mode de configuration routeur. Le premier paramètre désigne la source du protocole dont les mots clefs se trouvent dans le tableau 6.2.

**REMARQUE** Le tableau 6.2 est extrait du Cisco IOS 12.0 (2a). Suivant la version utilisée les mots clefs disponibles peuvent être différents.

Le paramètre <*métrique spécifique*> possède un format variable qui peut comporter plusieurs éléments suivant le protocole dans lequel l'information de routage se trouve redistribuée. Pour IGRP ceux-ci sont le débit (0 ou le nombre de Kbit/s), le délai (en unité de 10µs), la fiabilité (0 à 255), la charge (0 à 255) et la MTU de 1500 octets.

**REMARQUE** Le mot clef **metric** et son paramètre associé sont optionnels. S'il est omis, la métrique spécifique prend comme valeur par défaut 0 ; sauf si la commande **default-metric** est utilisée. Celle-ci est décrite dans les sections suivantes.

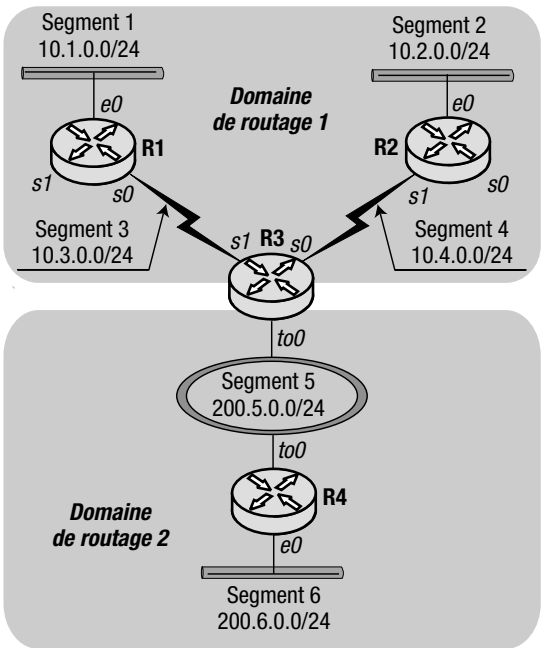
**Tableau 6.2. Mots clefs disponibles pour le paramètre <*source de l'information de routage*> de la commande `redistribute`.**

| Mot clef | Source de l'information de routage |
|---|---|
| bgp | Protocole de routage inter-domaines ou BGP (*Border Gateway Protocol*) |
| connected | Connecté |
| egp | Protocole de routage inter-domaines ou EGP (*Exterior Gateway Protocol*) |
| eigrp | Protocole de routage intra-domaine amélioré ou EIGRP (*Enhanced* IGRP) |
| igrp | Protocole de routage intra-domaine ou IGRP (*Interior Gateway Routing Protocol*) |
| iso-igrp | IGRP pour réseaux ISO |
| isis | Protocole de routage intra-domaine ISO ou IS-IS (*Intermediate System to Intermediate System*) |
| odr | Routes d'aire confinée (*stub* area) sur demande (*On Demand stub Routes*) |
| ospf | Protocole ouvert au plus court chemin ou OSPF (*Open Shortest Path First*) |
| rip | Protocole d'information de routage ou RIP (*Routing Information Protocol*) |
| static | Routes statiques |
| mobiles | Routes mobiles |

© copyright Éditions Eyrolles

Prenons l'exemple de la figure 6.5 où le réseau est composé de deux domaines de routage. Les routeurs R1 et R2 sont du domaine 1 (IGRP), le routeur R4 est du domaine 2 (RIP), tandis que le routeur R3, appartenant aussi bien à l'un qu'à l'autre, est chargé par conséquent de la redistribution d'informations de routage entre ces deux protocoles.

**Figure 6.5**
*Redistribution d'informations de routage entre les deux domaines RIP et IGRP par le routeur R3.*




Les listings 6.13 à 6.16 contiennent les configurations des quatre routeurs ; les lignes en italique du listing 6.15 mettent en évidence la commande **redistribute** sur le routeur R3.

**Listing 6.13. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.14. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 6.15. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 redistribute igrp 10 metric 1
 network 200.5.0.0

router igrp 10
 redistribute rip metric 10000 1 255 1 1500
 network 10.0.0.0
```

**Listing 6.16. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 network 200.5.0.0
 network 200.6.0.0
```

Si nous examinons la table de routage du routeur R4 sur le listing 6.17, nous n'y voyons qu'une seule route apprise *via* RIP (codée avec la lettre « R »), qui est 10.0.0.0. Cette route visiblement provient de la redistribution mutuelle entre les deux domaines RIP et IGRP, dont le routeur R3 a la charge. De même, la table de routage du routeur R1 sur le listing 6.18 contient deux routes codées avec la lettre « I » pour IGRP, qui sont 200.5.0.0/24 et 200.6.0.0/24.

**Listing 6.17. Table de routage du routeur R4.**

```
R4#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
R  10.0.0.0/8 [120/1] via 200.5.0.1, 00:00:10, TokenRing0
```

**Listing 6.18. Table de routage du routeur R1.**

```
R1#show ip route
...
10.0.0.0/24 is subnetted, 4 subnets
I     10.2.0.0 [100/10576] via 10.3.0.1, 00:00:04, Serial0
C     10.3.0.0 is directly connected, Serial0
C     10.1.0.0 is directly connected, Ethernet0
I     10.4.0.0 [100/10476] via 10.3.0.1, 00:00:04, Serial0
I  200.5.0.0/24 [100/8539] via 10.3.0.1, 00:00:04, Serial0
I  200.6.0.0/24 [100/8477] via 10.3.0.1, 00:00:04, Serial0
```

© copyright Éditions Eyrolles

La table de routage du routeur R3, quant à elle, contient toutes les routes de tous les réseaux apprises *via* les sources de protocole d'origine (IGRP, RIP et interfaces connectées).

**Listing 6.19. Table de routage du routeur R3.**

```
R3#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
R  200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:14, TokenRing0
   10.0.0.0/24 is subnetted, 4 subnets
I     10.2.0.0 [100/8576] via 10.4.0.2, 00:00:20, Serial0
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.3.0.2, 00:01:00, Serial1
C     10.4.0.0 is directly connected, Serial0
```

**ASTUCE** Plusieurs commandes **redistribute** peuvent être introduites en mode de configuration routeur pour le même protocole, chacune pour une source différente d'informations de routage.

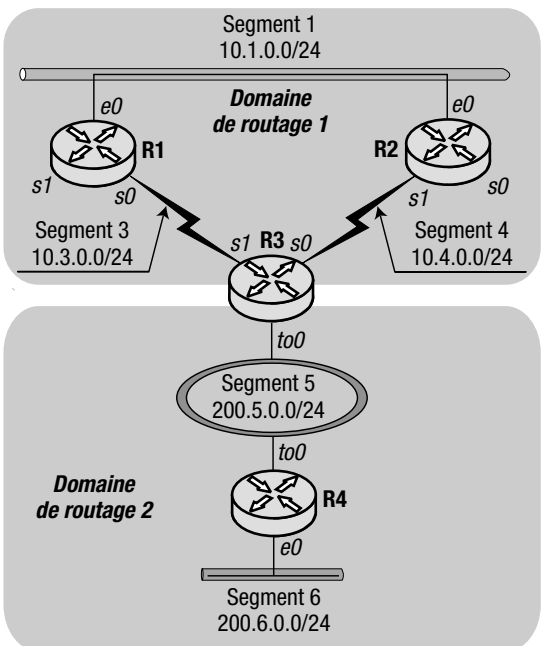
### Phénomène de boucle résultant de la redistribution

Nous avons déjà évoqué au début de ce chapitre une situation typique où un phénomène de boucle de routage peut se produire lors de la redistribution. Nous allons l'illustrer dans ce qui suit par un exemple concret qui montre également son caractère oscillatoire.

Le réseau de la figure 6.6 a été légèrement modifié. Les segments 1 et 2 ont été fusionnés pour ne faire qu'un seul qui relie à présent les routeurs R1 et R2. Ils vont donc échanger des mises à jour de routage du protocole IGRP. Le reste de la topologie est celui de la figure 6.5. Voyons maintenant ce qui induit la boucle de routage.

**Figure 6.6**

*Domaine de routage 1 avec protocole IGRP se trouve coupé du domaine de routage 2 avec protocole RIP en cas de boucle.*




La configuration du routeur R2, la seule modifiée, se trouve sur le listing 6.20.

© copyright Éditions Eyrolles

**Listing 6.20. Configuration modifiée du routeur R2.**

```
interface Ethernet0
 ip address 10.1.0.2 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

Le routeur R1 est connecté en premier au segment 1, puis au segment 3. Il reçoit de ce fait la mise à jour en provenance du routeur R2 en premier. Vérifions la table de routage du routeur R3 sur le listing 6.21 qui présente les symptômes du phénomène de boucle. Le préfixe réseau 200.6.0.0/24 pointe sur les routeurs R1 et R2 alors que c'est le routeur R3 lui-même qui peut y accéder.

**Listing 6.21. Phénomène de boucle visible dans la table de routage du routeur R3.**

```
R3#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
I  200.6.0.0/24 [100/10577] via 10.4.0.2, 00:04:43, Serial0
                [100/10577] via 10.3.0.2, 00:01:12, Serial1
   10.0.0.0/24 is subnetted, 3 subnets
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.4.0.2, 00:00:30, Serial0
               [100/8576] via 10.3.0.2, 00:01:13, Serial1
C     10.4.0.0 is directly connected, Serial0
```

Le contenu de la table de routage du routeur R3 vient de subir un changement comme on peut la voir sur le listing 6.22. Apparemment, l'un des routeurs impliqués dans cette « duperie » a augmenté la métrique de la route pour le réseau 200.6.0.0/24 provoquant ainsi une avalanche de mises à jour déclenchées faisant passer la métrique à une valeur infinie. La route se trouve mise sous temporisation de maintien (*hold down*) comme l'atteste l'indication « *possibly down* » de la table de routage la concernant.

**Listing 6.22. Table de routage du routeur R3 indiquant que la route pour le réseau 200.6.0.0 est sous temporisation de maintien.**

```
R3#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
I  200.6.0.0/24 is possibly down,routing via 10.4.0.2,Serial0
   10.0.0.0/24 is subnetted, 3 subnets
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.4.0.2, 00:00:02, Serial0
               [100/8576] via 10.3.0.2, 00:00:57, Serial1
C     10.4.0.0 is directly connected, Serial0
```

Procédons maintenant au vidage de la table de routage du routeur R3 par la commande **clear ip route *** et affichons de nouveau son contenu. Il semble retourner à la normale, mais pas pour longtemps.

© copyright Éditions Eyrolles

**Listing 6.23. Table de routage du routeur R3 après son vidage.**

```
R3#clear ip route *
R3#show ip route
...
C  200.5.0.0/24 is directly connected, TokenRing0
R  200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:07, TokenRing0
   10.0.0.0/24 is subnetted, 3 subnets
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.4.0.2, 00:00:07, Serial0
               [100/8576] via 10.3.0.2, 00:00:07, Serial1
C     10.4.0.0 is directly connected, Serial0
```

Affichons les tables de routage des routeurs R1 et R2 (cf. listings 6.24 et 6.25) ; celle du routeur R1 paraît correcte, mais celle du routeur R2 présente une anomalie. En effet, la route pour le réseau 200.6.0.0/24 y est mise pour le moment sous temporisation de maintien, bien qu'elle soit tout à fait accessible *via* le routeur R3 ; ceci est une conséquence de la boucle qui avait disparu auparavant.

**Listing 6.24. Table de routage du routeur R1.**

```
R1#show ip route
...
   10.0.0.0/24 is subnetted, 3 subnets
C     10.3.0.0 is directly connected, Serial0
C     10.1.0.0 is directly connected, Ethernet0
I     10.4.0.0 [100/8576] via 10.1.0.2, 00:00:17, Ethernet0
I  200.5.0.0/24 [100/8539] via 10.3.0.1, 00:00:35, Serial0
I  200.6.0.0/24 [100/8477] via 10.3.0.1, 00:04:42, Serial0
```

**Listing 6.25. Table de routage du routeur R2 indiquant que la route pour le réseau 200.6.0.0 est sous temporisation de maintien *(possibly down).***

```
R2#show ip route
...
   10.0.0.0/24 is subnetted, 3 subnets
I     10.3.0.0 [100/8576] via 10.1.0.1, 00:00:48, Ethernet0
C     10.1.0.0 is directly connected, Ethernet0
C     10.4.0.0 is directly connected, Serial1
I  200.5.0.0/24 [100/8539] via 10.4.0.1, 00:00:12, Serial1
I  200.6.0.0/24 is possibly down,routing via 10.4.0.1,Serial1
```

Aussitôt que la temporisation de maintien aura expiré pour la route pointant vers le réseau 200.6.0.0, le routeur R2 enverra une mise à jour positive au routeur R3 qui va supplanter la route apprise *via* RIP, qui était correcte. Ceci va créer une boucle temporaire jusqu'à ce que l'un des routeurs augmente la métrique de cette route, provoquant une avalanche de mises à jour déclenchées, le passage de sa métrique à la valeur infinie (0xFFFFFFFF en hexadécimal), sa mise sous temporisation de maintien pour 280 secondes. Celle-ci, arrivée à échéance va déclencher à nouveau le même cycle.

© copyright Éditions Eyrolles

**REMARQUE** Tant que la temporisation de maintien pour une route n'est pas écoulée, le routeur annonce celle-ci avec une métrique de valeur infinie, mais n'en accepte aucune mise à jour. Par conséquent, si un protocole autre que celui qui a placé cette route sous temporisation, mais dont la distance administrative est plus grande, tente de la remplacer par une route correcte vers la même destination, cette route ne sera pas prise en compte.

### Redistribution de routes statiques et d'interfaces connectées

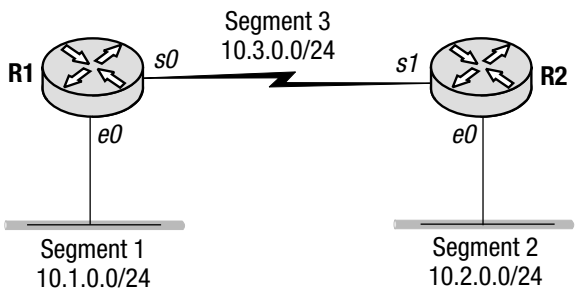
En mode de configuration routeur du protocole dynamique vers lequel on veut redistribuer les routes statiques et d'interfaces connectées, les commandes **redistribute static** et **redistribute connected** sont à utiliser, respectivement.

Il existe cependant un cas particulier qui concerne les routes statiques pointant sur une interface. Ces routes sont automatiquement redistribuées vers le protocole dynamique si leur préfixe réseau correspond à l'argument de la commande **network** introduite en mode de configuration routeur de ce protocole. Dans le cas contraire, une commande explicite **redistribute static** est nécessaire.

Sur la figure 6.7 les routeurs R1 et R2 utilisent le protocole de routage IGRP, en plus de certaines routes statiques configurées sur le premier dont deux pointent sur des interfaces. Les listings 6.26 et 6.27 montrent leur configuration.

**Figure 6.7**

*Exemple de réseau où le routeur R1 possède plusieurs routes statiques redistribuées vers le protocole de routage dynamique IGRP.*




**Listing 6.26. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 100
 network 10.0.0.0

ip route 10.100.0.0 255.255.255.0 Ethernet0
ip route 10.110.0.0 255.255.255.0 10.1.0.2
ip route 172.16.1.0 255.255.255.0 Ethernet0
```

**Listing 6.27. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

router igrp 100
 network 10.0.0.0
```

La table de routage du routeur R2 sur le listing 6.28 ne possède qu'une seule route statique (10.100.0.0/24 pointant sur l'interface Ethernet 0) redistribuée parmi les trois du routeur R1 car elle est la seule à correspondre à la commande **network** introduite sous le mode de configuration routeur du protocole IGRP sur ce routeur R1, confirmant la règle énoncée plus haut..

**Listing 6.28. Table de routage du routeur R2.**

```
R2#show ip route
...
   10.0.0.0/24 is subnetted, 5 subnets
C     10.2.0.0 is directly connected, Ethernet0
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.3.0.2, 00:00:08, Serial1
C     10.4.0.0 is directly connected, Serial0
I     10.100.0.0 [100/8576] via 10.3.0.2, 00:00:08, Serial1
```

Si nous ajoutons la commande **redistribute static metric 10000 1 255 1 1500** en mode de configuration routeur (commande **router igrp 100**) sur le routeur R1, nous verrons que ses trois routes statiques vont apparaître avec le code « I » sur la table de routage du routeur R2 (cf. listing 6.29).

**Listing 6.29. Table de routage du routeur R2.**

```
R2#show ip route
...
I  172.16.0.0/16 [100/8477] via 10.3.0.2, 00:00:28, Serial1
   10.0.0.0/24 is subnetted, 6 subnets
C     10.2.0.0 is directly connected, Ethernet0
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.3.0.2, 00:00:29, Serial1
C     10.4.0.0 is directly connected, Serial0
I     10.110.0.0 [100/8477] via 10.3.0.2, 00:00:29, Serial1
I     10.100.0.0 [100/8477] via 10.3.0.2, 00:00:29, Serial1
```

**REMARQUE** Les routes statiques pointant sur des interfaces ne sont pas automatiquement redistribuées vers le protocole OSPF même si leurs préfixes réseau correspondent aux commandes OSPF **network**.

### Redistribution avec métrique par défaut

Il est possible d'attribuer une métrique par défaut si on omet le mot clef **metric** et son paramètre associé dans la commande **redistribute.** Une deuxième commande **default-metric** *<métrique spécifique au protocole>* doit être introduite à la suite, toujours en mode de configuration routeur du protocole concerné, et son paramètre renseigné selon le format adéquat.

© copyright Éditions Eyrolles

La modification de la configuration du routeur R3 de la figure 6.5 d'après ce qui est dit plus haut se trouve sur le listing 6.30.

**Listing 6.30. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 redistribute igrp 10
 network 200.5.0.0
 default-metric 3

router igrp 10
 redistribute rip
 network 10.0.0.0
 default-metric 10000 1 255 1 1500
```

Nous pouvons constater sur le listing 6.31 de la table de routage du routeur R2 (caractère en italique) que la métrique pour la route du réseau 10.0.0.0/8 apprise *via* IGRP, n'est plus 1 comme auparavant (cf. listing 6.17) mais 3 qui est l'argument de la commande **default-metric** sur le routeur R3.

**Listing 6.31. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    10.0.0.0/8 [120/3] via 200.5.0.1, 00:00:03, TokenRing0
```

### Redistribution à sens unique

Comme nous l'avons évoqué au tout début de ce chapitre, il arrive que des hôtes dans un réseau aient besoin d'informations de routage par un protocole dynamique, au lieu d'être configurés avec une adresse de passerelle par défaut. Dans la plupart de ces cas, il s'agit d'un protocole de routage dit « ouvert » comme RIP. Pour les routeurs, il peut s'agir d'un autre protocole non compatible tel que IGRP. Les routes apprises par ce dernier doivent donc être redistribuées vers le protocole RIP sur tous les segments où résident les hôtes, et non l'inverse.

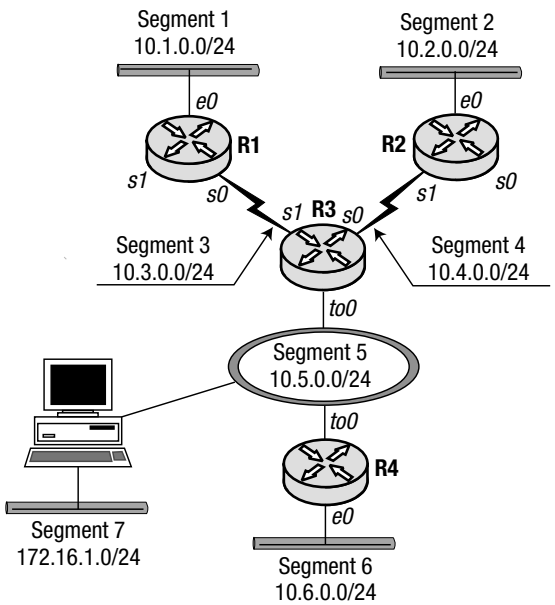
Lors d'une telle redistribution à sens unique, il serait préférable de s'assurer que les routeurs ne prennent pas en compte les mises à jour RIP qu'ils reçoivent de ces hôtes, en retour. En mode de configuration routeur du protocole RIP, par la commande **distance 255** on peut mettre ces mises à jour au rebut. Cette action est indispensable si l'on veut empêcher que la table de routage de ces routeurs soit remplie par ces informations.

© copyright Éditions Eyrolles

Prenons l'exemple du réseau de la figure 6.8 où l'hôte H1 multidomicilié reçoit les mises à jour par RIP tandis que les routeurs exécutent le protocole IGRP. C'est un cas qui ressemble à celui du chapitre 4 de la section portant sur « la discrimination des mises à jour entrantes », à la différence que l'hôte H1 et les routeurs y exécutaient le même protocole RIP. Notre but à présent est d'éliminer les mises à jour concernant le préfixe réseau du segment 7 en provenance de l'hôte H1 vers le routeur tout en chargeant celui-ci de la tâche de redistribution des informations de IGRP vers RIP.

**Figure 6.8**

*Hôte H1 multidomicilié dont le protocole RIP reçoit les mises à jour de routage redistribuées par les routeurs connectés au segment Token ring.*




Les listings 6.32 à 6.35 contiennent les configurations des quatre routeurs.

**Listing 6.32. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.33. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 6.34. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 10.5.0.1 255.255.255.0

router rip
 redistribute igrp 10 metric 3
 passive-interface Serial0
 passive-interface Serial1
 network 10.0.0.0
 distance 255

router igrp 10
 network 10.0.0.0
```

**Listing 6.35. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 10.6.0.1 255.255.255.0

interface TokenRing0
 ip address 10.5.0.2 255.255.255.0
 ring-speed 16

router rip
 redistribute igrp 10 metric 3
 passive-interface Ethernet0
 network 10.0.0.0

router igrp 10
 network 10.0.0.0
```

Notons au passage que le routeur R4 ne contient pas la commande **distance 255** en mode de configuration routeur du protocole RIP (**routeur rip**). C'est une omission volontaire qui a pour but de montrer que sa table de routage va contenir du fait de l'absence de cette commande les mises à jour en provenance de l'hôte H1. On peut voir sur le listing 6.36 que le routeur R4 a effectivement une route pour le segment 7 (ligne en italique).

**Listing 6.36. Table de routage du routeur R4.**

```
R4#show ip route
...
R  172.16.0.0/16 [120/2] via 10.5.0.15, 00:00:07, TokenRing0
   10.0.0.0/24 is subnetted, 6 subnets
I     10.2.0.0 [100/8639] via 10.5.0.1, 00:01:10, TokenRing0
I     10.3.0.0 [100/8539] via 10.5.0.1, 00:01:10, TokenRing0
I     10.1.0.0 [100/8639] via 10.5.0.1, 00:01:10, TokenRing0
C     10.6.0.0 is directly connected, Ethernet0
I     10.4.0.0 [100/8539] via 10.5.0.1, 00:01:10, TokenRing0
C     10.5.0.0 is directly connected, TokenRing0
```

© copyright Éditions Eyrolles

Par contre, la table de routage du routeur R3 sur le listing 6.37 n'a aucune route pour le préfixe réseau 172.16.0.0/16 car la commande **distance 255** a bien été introduite sur ce routeur.

**Listing 6.37. Table de routage du routeur R3.**

```
R3#show ip route
...
   10.0.0.0/24 is subnetted, 6 subnets
I     10.2.0.0 [100/8576] via 10.4.0.2, 00:01:05, Serial0
C     10.3.0.0 is directly connected, Serial1
I     10.1.0.0 [100/8576] via 10.3.0.2, 00:00:10, Serial1
I     10.6.0.0 [100/1163] via 10.5.0.2, 00:00:08, TokenRing0
C     10.4.0.0 is directly connected, Serial0
C     10.5.0.0 is directly connected, TokenRing0
```

| **Solution apparentée :** | **À la page :** |
|---|---|
| Discrimination des mises à jour entrantes | 131 |

### Redistribution avec filtrage des mises à jour de routage par listes d'accès

Nous avons déjà évoqué plus haut la nécessité d'une redistribution plus contrôlée. On pourrait envisager de ne redistribuer que certaines routes précisément au moyen d'une liste d'accès.

La procédure à suivre est la suivante :

- Par la commande **access-list** définir la liste d'accès comportant les lignes qui autorisent ou refusent les préfixes réseau, suivant le mot clef **permit** ou **deny**.
- Par la commande **distribute-list** <*numéro de liste d'accès*> **out** <*source de l'information de routage*>, appliquer la liste définie à l'étape précédente au protocole correspondant à l'argument du dernier paramètre qui, le cas échéant, peut requérir un numéro de processus ou de système autonome. Cette source peut être une interface connectée (*connected*) ou une route statique (*static*).

Prenons comme exemple le réseau de la figure 6.5 en ajoutant, cette fois-ci, aux routeurs R1 et R2 des interfaces de rebouclage (*loopback*) dont les adresses IP sont 172.16.0.0/24 et 172.17.0.0/24, respectivement. Ces routeurs sont configurés pour diffuser les mises à jour de ces préfixes *via* IGRP. Nous allons donc les filtrer sur le routeur R3 de façon à ce qu'ils ne soient pas redistribués dans RIP.

Les listings 6.38 à 6.41 montrent le contenu des configurations pour ces routeurs. Notons que la liste d'accès est nommée au lieu d'être identifiée par un numéro.

**Listing 6.38. Configuration du routeur R1.**

```
interface Loopback0
 ip address 172.16.1.1 255.255.255.0

interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
 network 172.16.0.0
```

© copyright Éditions Eyrolles

**Listing 6.39. Configuration du routeur R2.**

```
interface Loopback0
 ip address 172.17.1.1 255.255.255.0

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
 network 172.17.0.0
```

**Listing 6.40. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 redistribute igrp 10 metric 1
 network 200.5.0.0
 distribute-list no172 out igrp 10

router igrp 10
 redistribute rip metric 10000 1 255 1 1500
 network 10.0.0.0

ip access-list standard no172
 deny   172.16.0.0 0.15.255.255
 permit any
```

**Listing 6.41. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 network 200.5.0.0
 network 200.6.0.0
```

© copyright Éditions Eyrolles

La table de routage du routeur R4 sur le listing 6.42 ne contient aucune route pour les adresses IP d'interfaces de rebouclage définies sur les routeurs R1 et R2, suite à l'application du filtre.

**Listing 6.42. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    10.0.0.0/8 [120/1] via 200.5.0.1, 00:00:05, TokenRing0
```

Si on consulte la table de routage du routeur R3 sur le listing 6.43, on peut bien y voir les deux préfixes réseau d'interface de rebouclage dont les routes ont été apprises *via* IGRP.

**Listing 6.43. Table de routage du routeur R3.**

```
R3#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
R    200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:05, TokenRing0
I    172.17.0.0/16 [100/8976] via 10.4.0.2, 00:01:18, Serial0
I    172.16.0.0/16 [100/8976] via 10.3.0.2, 00:00:09, Serial1
     10.0.0.0/24 is subnetted, 4 subnets
I       10.2.0.0 [100/8576] via 10.4.0.2, 00:01:18, Serial0
C       10.3.0.0 is directly connected, Serial1
I       10.1.0.0 [100/8576] via 10.3.0.2, 00:00:09, Serial1
C       10.4.0.0 is directly connected, Serial0
```

### Redistribution avec filtrage des mises à jour de routage par séquence conditionnelle

Un autre moyen de filtrer les mises à jour consiste à définir une séquence conditionnelle (*route map*) comportant une clause (ou plusieurs) d'expressions dont chacune, si elle est vérifiée, permet de modifier par une action (*set*) ou plus, les caractéristiques des routes redistribuées. Dans une clause, chaque ligne doit produire un résultat logique vrai pour que la route correspondante soit redistribuée en exécutant un jeu d'actions ou une seule selon le cas. Si une clause de la séquence conditionnelle ne se vérifie pas, le routeur passe à la suivante, jusqu'à en trouver une dont chaque ligne produit un résultat vrai. Si aucune ne se vérifie, la route concernée n'est pas redistribuée. Une clause qui ne comporte aucune expression donne l'admission à toutes les routes.

**REMARQUE** La définition sur la séquence conditionnelle donnée dans le texte ne présente qu'un aspect de celle-ci en tant que filtre de mises à jour. D'autres fonctions y sont prévues mais ne sont pas mentionnées pour faciliter la compréhension dans le contexte d'utilisation qui nous concerne.

Par défaut, chaque séquence conditionnelle est censée ne produire un résultat logique vrai ou faux que sur une autorisation qui peut être explicite (*permit*). On peut également définir une clause d'interdiction (*deny*) qui, si elle s'avère vraie, fera en sorte que la route correspondante ne soit pas redistribuée, tout en terminant la consultation de la séquence qui ne sera pas poursuivie plus loin.

© copyright Éditions Eyrolles

Pour utiliser la séquence conditionnelle en tant que filtre de mises à jour, les étapes sont les suivantes :

1. Par la commande **route-map** <*nom d'en-tête de séquence*> [{**permit**|**deny**}] <*numéro de clause*>, il faut créer un nom d'en-tête avec un numéro de clause.
2. Par la commande **match** (comparaison) suivie des paramètres adéquats, définir la condition à remplir, ce qui se fait souvent par **match ip address** {<*numéro de liste d'accès*>|<*nom de liste d'accès*>}. Une comparaison sera faite entre le préfixe réseau et la plage d'adresses IP contenue dans la liste d'accès. Pour les autres formats de la commande **match**, se reporter à la documentation de Cisco.
3. Introduire par la commande **set** l'action à accomplir autant de fois que c'est nécessaire. Pour plus de renseignements sur les autres paramètres de cette commande, consulter la documentation de Cisco.
4. Si la séquence conditionnelle comporte plusieurs clauses, répéter les étapes 1 à 3 en renseignant le paramètre <*nom d'en-tête de séquence*> avec le même nom, mais en donnant un numéro de clause différent à chaque fois. La clause au chiffre le plus petit est vérifiée en premier.
5. En mode de configuration routeur du protocole vers lequel se fait la redistribution, renseigner dans la commande **redistribute,** le mot clef **route-map** suivi du nom de l'en-tête de séquence défini à l'étape 1.

Procédons à la révision de la configuration du routeur R3 du réseau de l'exemple de la figure 6.5 pour qu'il ne filtre plus les préfixes réseau 172.16.0.0/16 et 172.17.0.0/16 mais modifie leur métrique.

Le listing 6.44 montre la nouvelle configuration du routeur R3, avec les lignes en italique qui concernent l'application de la séquence conditionnelle elle-même, ainsi que la définition de ses deux clauses.

**Listing 6.44. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 redistribute igrp 10 metric 1 route-map setm172
 network 200.5.0.0

router igrp 10
 redistribute rip metric 10000 1 255 1 1500
 network 10.0.0.0

ip access-list standard net172-16
 permit 172.16.0.0 0.0.255.255
```

© copyright Éditions Eyrolles

```
ip access-list standard net172-17
 permit 172.17.0.0 0.0.255.255

route-map setm172 permit 10
 match ip address net172-16
 set metric +5

route-map setm172 permit 20
 match ip address net172-17
 set metric +10
```

En examinant la table de routage du routeur R4 sur le listing 6.45, nous pouvons y constater que les deux préfixes d'interface de rebouclage des routeurs R1 et R2 apparaissent bien, cette fois-ci, redistribués sur RIP, en provenance de IGRP.

**REMARQUE** Si dans la définition d'une clause de séquence conditionnelle par **route-map**, la commande **match** retourne un résultat vrai, et qu'une action par **set** est prévue, celle-ci peut modifier la métrique comme dans l'exemple proposé dans le texte. Sinon, c'est soit la métrique associée au mot clef **metric**, soit celle de la commande **default-metric** qui sera prise en compte. Faute d'avoir été renseignée par l'un de ces moyens, la métrique de la route sera redistribuée avec la valeur infinie.

### Agrégation de routes avec l'interface Null

Dans le chapitre 4, nous avons déjà évoqué la possibilité d'annoncer des routes de super-réseau *via* RIP version 2. Toutefois, l'implémentation de ce protocole dans les routeurs Cisco ne fournit aucun moyen commode pour configurer des adresses de super-réseau comme les commandes **area** <*aire*> **range** de OSPF et **ip summary-address eigrp** de EIGRP. Nous allons cependant proposer une solution portant sur la redistribution d'une route statique de super-réseau vers RIP v2. En tant que protocole sans classe, celui-ci est en mesure d'annoncer une telle route. Si un paquet arrive pour une destination existante appartenant à l'un des préfixes du super-réseau, il est acheminé ; autrement, c'est la mise au rebut.

Pour pouvoir mettre au rebut les datagrammes adressés à des préfixes réseau inexistants, une interface logique particulière appelée Null doit être configurée. Tout paquet destiné à cette interface se trouve mis au rebut.

La solution proposée serait parfaite si RIP v2 n'annonçait pas les préfixes réseau spécifiques en même temps que le préfixe du super-réseau. Mais comment filtrer ces préfixes tout en admettant celui du super-réseau, le seul qualifié à être annoncé ? Cette question est importante car le préfixe de super-réseau correspond par définition à n'importe quel préfixe spécifique. En voulant filtrer ce dernier par une liste d'accès qui correspond à la plage des préfixes auquel il appartient, on tombera forcément sur l'équivalence avec le préfixe de super-réseau qui sera filtré lui aussi. Pour sortir de ce dilemme, on évitera d'assigner l'adresse de super-réseau avec un masque de sous-réseau spécifique à tout segment existant. Ce faisant, on peut composer une liste d'accès qui filtre tous les préfixes réseau spécifiques tout en autorisant le passage au seul préfixe de super-réseau, de la manière suivante :

1. La liste d'accès doit comporter, comme première condition, la correspondance exacte pour l'adresse de super-réseau à annoncer, par le mot clef **permit** et le masque générique 0.0.0.0.
2. La deuxième condition de la liste d'accès doit interdire tout préfixe spécifique appartenant au super-réseau par le mot clef **deny** avec le masque générique 0.255.255.255.

© copyright Éditions Eyrolles

3. La troisième condition de la liste d'accès doit inclure les mots clefs **permit any**.

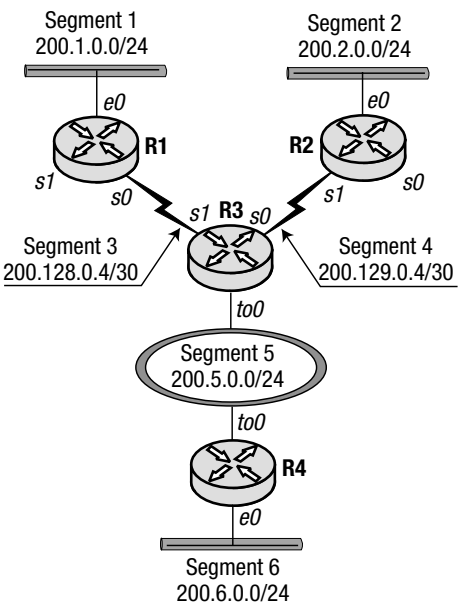
Pour la configuration de routes de super-réseaux dans RIP v2, effectuer les actions suivantes :

1. En mode de configuration globale, entrer la commande **ip classless**.
2. En mode de configuration **router rip**, introduire les commandes **version 2** et **no auto-summary**.
3. Pour chaque adresse de super-réseau, créer une route statique pointant sur l'interface Null 0.
4. En suivant la procédure de composition de la phase précédente (décrite plus haut), créer une seule liste d'accès pour chaque super-réseau.
5. En introduisant la commande **redistribute static**, la route statique sera redistribuée dans RIP v2.
6. Pour chaque interface à travers laquelle RIP v2 doit annoncer l'adresse du super-réseau, créer un filtre de mises à jour par la commande **distribute-list** <*numéro de liste*> **out** <*interface*>. Le premier paramètre est le numéro donné à la liste d'accès de la phase précédente et le second paramètre correspond à l'interface sur laquelle cette liste doit être appliquée.
7. Cette étape optionnelle permet de faire le filtrage inverse. Pour interdire à certaines interfaces dont l'adresse IP appartient au super-réseau d'annoncer celui-ci, il faut créer une liste d'accès séparée avec les instructions inverses de la phase précédente et l'appliquer aux interfaces concernées.

Prenons comme exemple pratique le réseau de la figure 6.9. Le routeur R3 n'y est autorisé à annoncer que le préfixe de super-réseau à travers son interface Token Ring.

**Figure 6.9**

*Annonce via l'interface Token Ring du super-réseau 200.0.0.0/8 par le routeur R3.*




Les listings 6.46 à 6.49 montrent les configurations des quatre routeurs. Noter particulièrement les lignes en italique de la configuration du routeur R3 qui mettent en œuvre la solution préconisée.

© copyright Éditions Eyrolles

**Listing 6.45. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    172.17.0.0/16 [120/10] via 200.5.0.1, 00:00:07, TokenRing0
R    172.16.0.0/16 [120/5] via 200.5.0.1, 00:00:07, TokenRing0
```

**Listing 6.46. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 200.1.0.1 255.255.255.0

interface Serial0
 ip address 200.128.0.6 255.255.255.252

router rip
 version 2
 network 200.128.0.0
 network 200.1.0.0
 no auto-summary

ip classless
```

**Listing 6.47. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 200.2.0.1 255.255.255.0

interface Serial1
 ip address 200.129.0.6 255.255.255.252

router rip
 version 2
 network 200.129.0.0
 network 200.2.0.0
 no auto-summary

ip classless
```

**Listing 6.48. Configuration du routeur R3.**

```
interface Serial0
 ip address 200.129.0.5 255.255.255.252

interface Serial1
 ip address 200.128.0.5 255.255.255.252

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router rip
 version 2
```

© copyright Éditions Eyrolles

```
 redistribute static metric 1
 network 200.5.0.0
 network 200.128.0.0
 network 200.129.0.0
 distribute-list 1 out TokenRing0
 distribute-list 2 out Serial0
 distribute-list 2 out Serial1
 no auto-summary

ip classless

ip route 200.0.0.0 255.0.0.0 Null0

access-list 1 permit 200.0.0.0 0.0.0.0
access-list 1 deny   200.0.0.0 0.255.255.255
access-list 1 permit any

access-list 2 deny   200.0.0.0 0.0.0.0
access-list 2 permit 200.0.0.0 0.255.255.255
access-list 2 permit any
```

**Listing 6.49. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router rip
 version 2
 network 200.5.0.0
 network 200.6.0.0
 no auto-summary

ip classless
```

Dans la table de routage du routeur R4 sur le listing 6.50, on ne voit effectivement que le préfixe du super-réseau et non les préfixes spécifiques des réseaux situés derrière le routeur R3.

**Listing 6.50. Table de routage du routeur R4.**

```
R4#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
C    200.6.0.0/24 is directly connected, Ethernet0
R    200.0.0.0/8 [120/1] via 200.5.0.1, 00:00:25, TokenRing0
```

La table de routage du routeur R3 sur le listing 6.51, quant à elle, possède des routes pour tous les préfixes spécifiques, en particulier une pour le préfixe de super-réseau, pointant sur l'interface Null 0.

© copyright Éditions Eyrolles

**Listing 6.51. Table de routage du routeur R3.**

```
R3#show ip route
...
    200.128.0.0/30 is subnetted, 1 subnets
C       200.128.0.4 is directly connected, Serial1
    200.129.0.0/30 is subnetted, 1 subnets
C       200.129.0.4 is directly connected, Serial1
R   200.1.0.0/24 [120/1] via 200.128.0.6, 00:00:26, Serial1
R   200.2.0.0/24 [120/1] via 200.129.0.6, 00:00:26, Serial0
C   200.5.0.0/24 is directly connected, TokenRing0
R   200.6.0.0/24 [120/1] via 200.5.0.2, 00:00:05, TokenRing0
S   200.0.0.0/8 is directly connected, Null0
```

**REMARQUE** Notons que la route agrégée du préfixe de super-réseau pointant sur l'interface Null0 ressemble à celle déjà vue au chapitre 4 portant sur la même fonction dans EIGRP, à cette différence près que le code est différent : « S » pour agrégation (*summary*) au lieu de « D » (cf. listing 4.82).

## Redistribution avec EIGRP

Étant suffisamment perfectionné, le protocole EIGRP peut faire la distinction entre des routes apprises dans son propre processus sur un routeur, et celles transmises *via* un protocole différent (tel que RIP) d'un autre routeur, et redistribuées par ce dernier. De telles routes sont codées « D EX » pour EIGRP externe. Leur distance administrative affectée de la valeur 170 possède une priorité faible par rapport à celle des autres protocoles de routage dynamique.

Il existe cependant des cas particuliers où les routes externes EIGRP ont une priorité supérieure comparée à celles qui pointent sur la même destination tout en provenant de sources de distance administrative inférieure à 170. L'un de ces cas concerne la redistribution automatique entre IGRP et EIGRP d'un même système autonome. Depuis les versions 11.2.X de l'IOS de Cisco, cette règle s'applique également au cas de la redistribution manuelle (commande **redistribute**) entre ces deux protocoles, mais faisant partie de systèmes autonomes différents.

Ces deux cas seront présentés dans les paragraphes suivants. Le cas de la redistribution manuelle concerne les versions 11.1.X de l'IOS ou celles qui sont antérieures.

### Redistribution entre EIGRP et IGRP dans un même système autonome

La redistribution automatique s'effectue entre les protocoles IGRP et EIGRP sans qu'il soit besoin de la configurer par la commande **redistribute** en modes **router igrp** <*numéro de système autonome*> et **router eigrp** <*numéro de système autonome*>, où les paramètres ont le même numéro.

Les métriques de ces deux protocoles sont compatibles, elles peuvent donc être transmises de l'un vers l'autre sans perte de la valeur initiale accumulée. En quelque sorte, il s'agit d'une fusion des domaines métriques respectifs de ces deux protocoles.

**AVERTISSEMENT** Bien que les métriques entre les protocoles EIGRP et IGRP soient compatibles, les préfixes réseau ne le sont pas. Car le premier est un protocole sans classe tandis que le second est à classe. De ce fait, lors du passage des préfixes réseau de longueur variable de EIGRP, tous ceux qui ne correspondent pas à la longueur fixe définie dans IGRP, seront perdus.

© copyright Éditions Eyrolles

La distance administrative des routes externes de EIGRP est de 170, supérieure à celle de IGRP (100). Mais quand un conflit se produit entre une route candidate d'une source de distance administrative inférieure à celle de la route externe (déjà inscrite en provenance de EIGRP ou IGRP), c'est la métrique la plus petite qui prévaudra dans un routeur déroulant les processus IGRP et EIGRP. Néanmoins lors d'une réception normale de mises à jour de source EIGRP, ses routes auront la primauté sur celles externes de source IGRP ou EIGRP, pour une même destination.

Pour bien assimiler cette règle, prenons l'exemple concret illustré sur la figure 6.10. Il s'agit d'une topologie comparable à celle de la figure 6.6 qui avait tendance à se mettre en boucle.

Dans le cas présent, il s'agit de routes redistribuées du processus IGRP vers celui de EIGRP dans le routeur R4 et affectées d'une distance administrative de 170. On peut s'interroger sur le risque d'un phénomène de boucle si le routeur R3 redistribue à son tour ces routes aux routeurs R1 et R2 comme étant de source IGRP, à savoir avec la distance administrative 100. Que se passera-t-il si ces routeurs, en retour, annoncent ces mêmes routes avec la même distance vers le routeur R3 ?

Les listings 6.52 à 6.55 montrent les configurations des quatre routeurs.

**Listing 6.52. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 ip address 10.3.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.53. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.1.0.2 255.255.255.0

interface Serial1
 ip address 10.4.0.2 255.255.255.0

router igrp 10
 network 10.0.0.0
```

**Listing 6.54. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router eigrp 10
 network 200.5.0.0

router igrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 6.55. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router eigrp 10
 network 200.5.0.0

router igrp 10
 network 200.6.0.0
```

En vérifiant la table de routage du routeur R3 sur le listing 6.56, nous pouvons constater qu'elle ne présente aucun signe du phénomène de boucle. Remarquez le code par lequel le préfixe réseau 200.6.0.0/24 est désigné (« D EX » pour EIGRP externe), et la distance administrative de sa route qui vaut 170. Ces deux éléments sont en italique sur la deuxième ligne.

**Listing 6.56. Table de routage du routeur R3.**

```
R3#show ip route
...
C    200.5.0.0/24 is directly connected, TokenRing0
D EX 200.6.0.0/24 [170/176128] via 200.5.0.2, 00:32:36,
TokenRing0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.3.0.0 is directly connected, Serial1
I       10.1.0.0 [100/8576] via 10.4.0.2, 00:00:54, Serial0
                 [100/8576] via 10.3.0.2, 00:01:03, Serial1
C       10.4.0.0 is directly connected, Serial0
```

Voyons maintenant comment le routeur R3 traite les mises à jour IGRP en provenance des routeurs R1 et R2. La sortie de la commande correspondante se trouve sur le listing 6.57. Pour le moment, il ne reçoit aucune mise à jour pour le préfixe réseau 200.6.0.0/24 des deux autres.

**Listing 6.57. Sortie de la commande `debug ip igrp transactions`.**

```
R3#debug ip igrp transactions
IGRP protocol debugging is on
R3#
...
IGRP: received update from 10.3.0.2 on Serial1
      subnet 10.1.0.0, metric 8576 (neighbor 1100)
      subnet 10.4.0.0, metric 10576 (neighbor 8576)
IGRP: received update from 10.4.0.2 on Serial0
      subnet 10.3.0.0, metric 10576 (neighbor 8576)
      subnet 10.1.0.0, metric 8576 (neighbor 1100)
...
```

Procédons au vidage de la table de routage du routeur R1. Ce dernier va-t-il apprendre une route pour le préfixe réseau 200.6.0.0/24 du routeur R2 et l'annoncer à son tour au routeur

© copyright Éditions Eyrolles

R3 ? Celui-ci remplacera-t-il alors la route pour ce même préfixe qui est déjà dans sa table par la nouvelle dont la distance est inférieure ?

Le listing 6.58 affiche la sortie de la commande **debug ip igrp transactions** sur le routeur R1, immédiatement suivie de la commande **clear ip route *.

**Listing 6.58. Sortie de la commande `debug ip igrp transactions` sur le routeur R1, immédiatement suivie de la commande `clear ip route *`.**

```
R1#debug ip igrp transactions
IGRP protocol debugging is on
R1#clear ip route *
R1#
IGRP: broadcasting request on Ethernet0
IGRP: broadcasting request on Serial0
IGRP: received update from 10.1.0.2 on Ethernet0
      subnet 10.1.0.0, metric 1200 (neighbor 1100)
      subnet 10.4.0.0, metric 8576 (neighbor 8476)
      network 200.5.0.0, metric 8639 (neighbor 8539)
      network 200.6.0.0, metric 8639 (neighbor 8539)
IGRP: edition is now 13
IGRP: sending update to 255.255.255.255 via Ethernet0
(10.1.0.1)
      subnet 10.3.0.0, metric=8476
IGRP: sending update to 255.255.255.255 via Serial0
(10.3.0.2)
      subnet 10.1.0.0, metric=1100
      subnet 10.4.0.0, metric=8576
      network 200.5.0.0, metric=8639
      network 200.6.0.0, metric=8639
IGRP: received update from 10.3.0.1 on Serial0
      subnet 10.4.0.0, metric 10476 (neighbor 8476)
      network 200.5.0.0, metric 8539 (neighbor 688)
      network 200.6.0.0, metric 8539 (neighbor 688)
...
```

Le routeur R1 tente visiblement (cf. listing 6.58) d'annoncer une route factice pour le préfixe réseau 200.6.0.0/24 vers le routeur R3, mais ce dernier l'acceptera-t-il ? Le listing 6.59 montre la sortie de la commande **debug ip igrp transactions** sur le routeur R3 juste après le vidage de la table du routeur R1.

**Listing 6.59. Sortie de la commande `debug ip igrp transactions` sur le routeur R3 juste après la commande `clear ip route *` introduite sur le routeur R1.**

```
R3#debug ip igrp transactions
IGRP protocol debugging is on
R3#
IGRP: received request from 10.3.0.2 on Serial1
IGRP: sending update to 10.3.0.2 via Serial1 (10.3.0.1)
      subnet 10.4.0.0, metric=8476
      network 200.5.0.0, metric=688
      network 200.6.0.0, metric=688
```

© copyright Éditions Eyrolles

```
IGRP: received update from 10.3.0.2 on Serial1
      subnet 10.1.0.0, metric 8576 (neighbor 1100)
      subnet 10.4.0.0, metric 10576 (neighbor 8576)
      network 200.5.0.0, metric 10639 (neighbor 8639)
      network 200.6.0.0, metric 10639 (neighbor 8639)
...
```

On voit sur le listing 6.59 que le routeur R3 reçoit effectivement la route factice pour le préfixe réseau 200.6.0.0/24, mais sans la prendre en compte. La route codée « D EX » demeure dans sa table de routage qui reste la même que celle du listing 6.56. Le routeur R3 ne fait qu'appliquer la règle selon laquelle la distance ne doit pas intervenir dans la résolution de conflit entre une route candidate de distance même inférieure à celle d'une route externe de source IGRP ou EIGRP, déjà présente dans sa table de routage. Dans pareille situation, c'est la métrique la plus faible qui doit prévaloir. Dans notre cas, la route externe EIGRP a une métrique de 176128, tandis que celle de la route IGRP vaut 10639. Pour la comparaison des métriques IGRP et EIGRP, on multiplie celle-là par 256 ; ce qui donne comme résultat 2723584, chiffre supérieur à 176128. La route externe EIGRP n'est donc pas remplacée.

### Redistribution entre EIGRP et IGRP de systèmes autonomes différents

Quel que soit le mode de redistribution entre EIGRP et IGRP (automatique ou manuel) la métrique des routes, lors du transfert d'un protocole à l'autre, est toujours convertie au lieu d'être réaffectée d'une valeur prédéterminée, comme c'est le cas avec d'autres protocoles. Mais il y a une différence de comportement dans la redistribution manuelle suivant les versions du système IOS de Cisco (antérieures ou à partir de 11.2.X). Dans celles-ci, la règle portant sur la distance administrative est la même, qu'il s'agisse de redistribution manuelle ou automatique. Dans les anciennes versions, la redistribution manuelle est régie par la règle normale de distance administrative selon laquelle, en cas de conflit entre une route installée dans une table et une autre candidate à son remplacement, c'est celle possédant la plus petite distance administrative qui l'emporte.

Voyons maintenant ce qui se passe en réalité. Les routeurs de la figure 6.10 tournent en version 11.1.24. Les listings 6.60 et 6.61 montrent les configurations des routeurs R3 et R4.

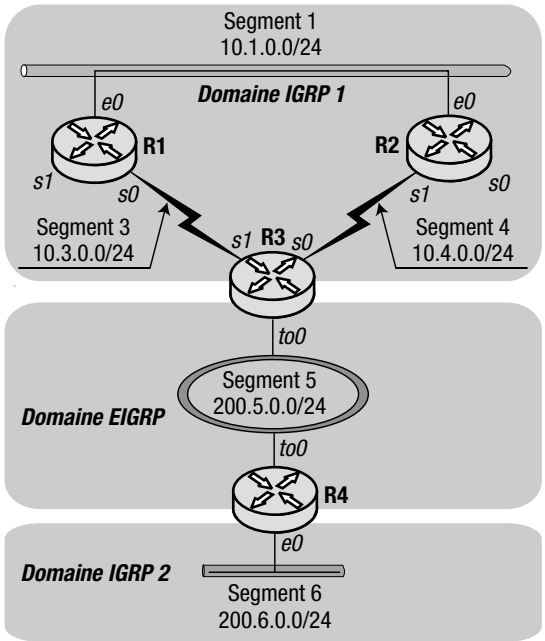
**Listing 6.60. Configuration du routeur R3.**

```
interface Serial0
 ip address 10.4.0.1 255.255.255.0

interface Serial1
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.1 255.255.255.0
 ring-speed 16

router eigrp 200
 redistribute igrp 10
 network 200.5.0.0

router igrp 10
 redistribute eigrp 200
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Figure 6.10**

*Les routeurs R3 et R4 redistribuent automatiquement les informations de routage entre les protocoles EIGRP et IGRP.*




**Listing 6.61. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 200.6.0.1 255.255.255.0

interface TokenRing0
 ip address 200.5.0.2 255.255.255.0
 ring-speed 16

router eigrp 200
 redistribute igrp 10
 network 200.5.0.0

router igrp 10
 redistribute eigrp 200
      network 200.6.0.0
```

La table de routage du routeur R4 sur le listing 6.62 affiche les trois routes redistribuées de IGRP à EIGRP sur le routeur R3, affectée d'une métrique calculée automatiquement sans que dans la commande **redistribute** le mot clef **metric** et son paramètre ne soient présents. Ce qui nous ramène au cas précédent d'extension automatique des domaines métriques, EIGRP et IGRP relevant d'un même système autonome.

**REMARQUE** Même si une métrique est affectée manuellement par la commande **redistribute** en renseignant le paramètre du mot clef **metric,** ou par la commande **default-metric**, les processus IGRP et EIGRP conserveront la métrique d'une route calculée dans leur domaine respectif avant son transfert de l'un vers l'autre, au lieu de la remplacer par la valeur affectée par l'une de ces deux commandes.

© copyright Éditions Eyrolles

**Listing 6.62. Table de routage du routeur R4.**

```
R4#show ip route
...
   10.0.0.0/24 is subnetted, 3 subnets
D EX 10.3.0.0 [170/176128] via 200.5.0.1,00:08:48,TokenRing0
D EX 10.1.0.0 [170/2211584] via 200.5.0.1,00:08:48,TokenRing0
D EX 10.4.0.0 [170/176128] via 200.5.0.1,00:08:49,TokenRing0
C  200.1.0.0/24 is directly connected, TokenRing0
C  200.5.0.0/24 is directly connected, TokenRing0
C  200.6.0.0/24 is directly connected, Ethernet0
C  200.7.0.0/24 is directly connected, Loopback0
```

Les versions 11.1.X de l'IOS et les précédentes sont toujours exposées au phénomène de boucle qui peut résulter de la redistribution manuelle entre IGRP et EIGRP. En l'état actuel, le routage s'étant stabilisé, la boucle n'a pas encore eu l'occasion de se manifester. Examinons à cet effet, la table de routage du routeur R3 sur le listing 6.63 qui, pour le moment, semble tout à fait correcte (ligne en italique).

**Listing 6.63. Table de routage du routeur R3 avant que la commande `clear ip route *` ne soit introduite dans le routeur R1.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 3 subnets
C       10.3.0.0 is directly connected, Serial1
I       10.1.0.0 [100/8576] via 10.3.0.2, 00:01:08, Serial1
                 [100/8576] via 10.4.0.2, 00:00:25, Serial0
C       10.4.0.0 is directly connected, Serial0
C    200.1.0.0/24 is directly connected, TokenRing0
C    200.5.0.0/24 is directly connected, TokenRing0
D EX 200.6.0.0/24 [170/176128] via 200.5.0.2, 00:03:15,
TokenRing0
```

Procédons maintenant au vidage de la table de routage du routeur R1, pour le contraindre à envoyer une requête IGRP sur toutes ses interfaces sur lesquelles ce protocole est actif. Comme auparavant, le routeur R1 recevra probablement une réponse en premier du routeur R2 du fait que le segment Ethernet qui le relie à ce dernier a un débit plus rapide que la ligne série le reliant au routeur R3. Dès qu'il reçoit une mise à jour pour le préfixe 200.6.0.0/24 du routeur R2, le routeur R1 va l'installer dans sa table de routage avec, comme routeur de saut suivant, l'expéditeur accessible par son interface Ethernet 0. Immédiatement après, ce routeur va annoncer à son tour ce même préfixe au routeur R3 avec une valeur de distance administrative de la route, inférieure à celle qui se trouve déjà dans la table de routage du routeur R3. Ce dernier va donc remplacer la bonne route externe EIGRP par celle qui est factice, en provenance du routeur R1. Le reste du phénomène de boucle est identique à ce qu'on a déjà évoqué dans la section qui lui était consacrée.

La table de routage du routeur R3 sur le listing 6.64 (ligne en italique) présente le signe déjà connu du phénomène de boucle, une fois qu'on a vidé la table de routage du routeur R1 pour le forcer à la remplir, provoquant ainsi l'annonce de la route factice vers le routeur R3.

© copyright Éditions Eyrolles

**Listing 6.64. Table de routage du routeur R3 après que la commande `clear ip route *` a été introduite sur le routeur R1.**

```
R3#show ip route
...
     10.0.0.0/24 is subnetted, 3 subnets
C       10.3.0.0 is directly connected, Serial1
I       10.1.0.0 [100/8576] via 10.3.0.2, 00:00:30, Serial1
                 [100/8576] via 10.4.0.2, 00:00:10, Serial0
C       10.4.0.0 is directly connected, Serial0
C    200.1.0.0/24 is directly connected, TokenRing0
C    200.5.0.0/24 is directly connected, TokenRing0
I    200.6.0.0/24 is possibly down, routing via 10.4.0.2, Serial0
```

## Redistribution avec OSPF

La redistribution de routes dans OSPF en provenance d'un autre protocole se fait sous forme d'annonces LSA (*Link State Advertisment*) de type 5. Il s'agit de routes externes qui se différencient encore en type de métrique 1 ou 2. Ces LSA sont ensuite arrosées sur tout le système autonome, les aires confinées mises à part (voir tableau 5.1).

Des LSA de type 7 sont également disponibles pour annoncer des routes externes, ce qui permet de transmettre les informations de routage apprises par d'autres sources qu'OSPF. Les LSA de type 7 ne sont pas décrites dans la RFC 2328 décrivant OSPF, mais dans un document séparé, la RFC 1587, consacrée à ce type de LSA.

Dans la suite de ce chapitre, nous allons étudier comment configurer la redistribution entre OSPF et d'autres protocoles de routage, en utilisant des LSA de type 5 et 7. Nous verrons également les différences qui existent entre ces deux types de LSA et leurs applications respectives.

### *Les routeurs ASBR et leur configuration*

Les routeurs chargés de la redistribution entre OSPF et les autres protocoles de routage sont des routeurs frontaliers du système autonome appelés routeurs *ASBR* (*Autonomous System Boundary Routers*).

Pour inclure des informations de routage à partir d'un autre système autonome ou à partir de l'Internet, il vous faut recourir aux routes externes OSPF normales annoncées par des LSA de type 5.

Les sections suivantes décrivent comment configurer un routeur Cisco pour assurer la redistribution des informations entre OSPF et les autres protocoles de routage, en utilisant les LSA de type 5.

#### Métrique des routes externes de type 1 et 2

La différence entre les types de métrique OSPF 1 et 2 est simple : les routes de type 1, étant propagées par des routeurs OSPF à travers un système autonome OSPF, voient leur métrique incrémentée de 1 à chaque saut tandis que les routes de type 2 conservent la valeur de métrique reçue au point de redistribution.

© copyright Éditions Eyrolles

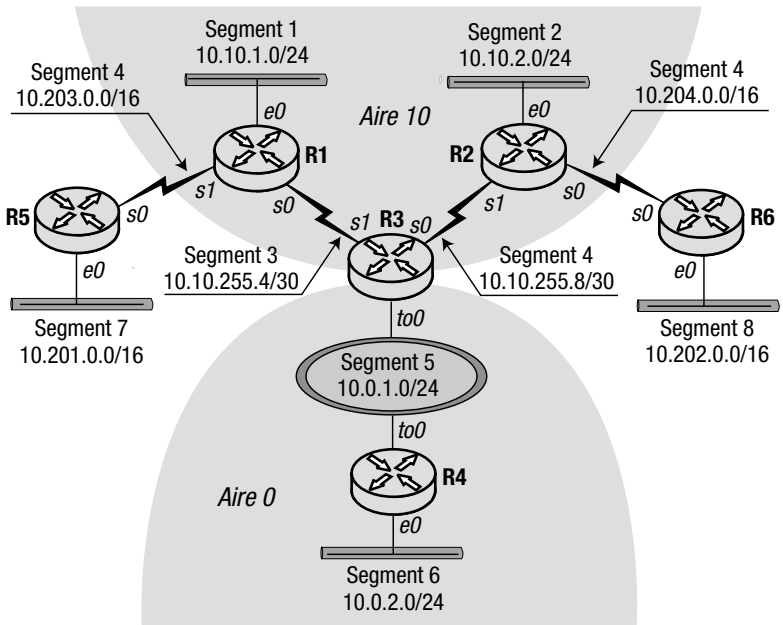
Une fois que le type de LSA (5 ou 7) à diffuser dans OSPF est choisi pour les routes en provenance d'un autre protocole, il faut utiliser la commande **redistribute** pour demander la redistribution des informations vers OSPF à partir de l'autre protocole. Il est possible de spécifier alors dans quel type de routes externes les routes redistribuées doivent être converties, en spécifiant le paramètre **metric-type {1|2}**. Le type de route par défaut est le type 2.

Autre paramètre pouvant être spécifié pour la commande **redistribute**, le mot clef **subnets** autorise expressément OSPF à recevoir des sous-réseaux spécifiques en provenance d'autres protocoles, car ils ne sont pas admis implicitement

La figure 6.11 illustre une topologie de réseau comportant un système autonome OSPF comprenant quatre routeurs – R1 à R4 –, interconnectés par des segments et deux routeurs externes – R5 et R6. La tâche ici est de configurer les routeurs R1 et R2 en tant que routeurs frontaliers ASBR de façon à ce qu'ils puissent annoncer les routes externes EIGRP de R5 et IGRP de R6 dans OSPF, en utilisant les deux types de métrique 1 et 2, respectivement.

**Figure 6.11**

*Les routeurs externes R5 et R6 sont connectés aux ASBR R1 et R2..*




Les listings 6.65 à 6.70 donnent la configuration pour les six routeurs.

**Listing 6.65. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.10.0.1 255.255.255.255

interface Ethernet0
 ip address 10.10.1.1 255.255.255.0

interface Serial0
 ip address 10.10.255.6 255.255.255.252

interface Serial1
```

© copyright Éditions Eyrolles

```
 ip address 10.203.0.1 255.255.0.0

router eigrp 10
 redistribute ospf 10 metric 10000 1 255 1 1500
 passive-interface Ethernet0
 passive-interface Loopback0
 passive-interface Serial0
 network 10.0.0.0

router ospf 10
 redistribute eigrp 10 metric 10 metric-type 1 subnets
 network 10.10.0.0 0.0.255.255 area 10
 distribute-list 10 out eigrp 10

ip classless

access-list 10 deny   10.10.0.0 0.0.255.255
access-list 10 permit any
```

**Listing 6.66. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.10.0.2 255.255.255.255

interface Ethernet0
 ip address 10.10.2.1 255.255.255.0

interface Serial0
 ip address 10.204.0.1 255.255.0.0

interface Serial1
 ip address 10.10.255.10 255.255.255.252

router ospf 10
 redistribute igrp 10 metric 10 subnets
 network 10.10.0.0 0.0.255.255 area 10
 distribute-list 10 out igrp 10

router igrp 10
 redistribute ospf 10 metric 10000 1 255 1 1500
 passive-interface Ethernet0
 passive-interface Loopback0
 passive-interface Serial1
 network 10.0.0.0

ip classless

access-list 10 deny   10.10.0.0 0.0.255.255
access-list 10 permit any
```

**Listing 6.67. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.10.255.9 255.255.255.252

interface Serial1
 ip address 10.10.255.5 255.255.255.252

interface TokenRing0
 ip address 10.0.1.1 255.255.255.0
 ring-speed 16

router ospf 10
 network 10.0.0.0 0.0.255.255 area 0
 network 10.10.0.0 0.0.255.255 area 10
 area 0 range 10.0.0.0 255.255.0.0
 area 10 range 10.10.0.0 255.255.0.0

ip classless
```

**Listing 6.68. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface TokenRing0
 ip address 10.0.1.2 255.255.255.0
 ring-speed 16

router ospf 10
 network 10.0.0.0 0.0.255.255 area 0

no ip classless
```

**Listing 6.69. Configuration du routeur R5.**

```
interface Ethernet0
 ip address 10.201.0.1 255.255.0.0

interface Serial0
 ip address 10.203.0.2 255.255.0.0

router eigrp 10
 network 10.0.0.0
```

**Listing 6.70. Configuration du routeur R6.**

```
interface Ethernet0
 ip address 10.202.0.1 255.255.0.0

interface Serial0
 ip address 10.204.0.2 255.255.0.0

router igrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

Examinons maintenant la table de routage du routeur R4 (listing 6.71). Les routes externes de OSPF y sont représentées suivies des codes « E1 » et « E2 » pour indiquer qu'elles sont de métrique de type 1 et 2 respectivement.

**Listing 6.71. Table de routage du routeur R4.**

```
R4#show ip route
...
  10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
O IA 10.10.0.0/16 [110/80] via 10.0.1.1, 00:21:27, TokenRing0
C    10.0.2.0/24 is directly connected, Ethernet0
O    10.0.0.3/32 [110/7] via 10.0.1.1, 00:21:27, TokenRing0
C    10.0.1.0/24 is directly connected, TokenRing0
C    10.0.0.4/32 is directly connected, Loopback0
O E1 10.201.0.0/16 [110/80] via 10.0.1.1, 00:21:08,TokenRing0
O E1 10.203.0.0/16 [110/80] via 10.0.1.1, 00:21:08,TokenRing0
O E2 10.202.0.0/16 [110/10] via 10.0.1.1, 00:16:31,TokenRing0
O E2 10.204.0.0/16 [110/10] via 10.0.1.1, 00:21:27,TokenRing0
```

Notez la différence qui existe entre les métriques des routes OSPF de type 1 et 2. Il suffit de reprendre la configuration des routeurs R1 et R2 (cf. listings 6.65 et 6.66) pour voir que les routes avaient été distribuées avec la même métrique – 10. Les routes de type 2 ont gardé cette métrique tandis que les routes de type 1 ont une valeur de métrique supérieure en raison des sauts qu'elles ont traversés avant d'atteindre le routeur R4.

Si nous analysons les configurations des routeurs R1 et R2 nous ne manquerons pas d'y remarquer qu'elles contiennent chacune une liste d'accès pour filtrer les informations, soit de EIGRP, soit de IGRP, selon le cas, avant leur redistribution dans OSPF. La raison en était la suivante : à la différence d'OSPF, ni IGRP ni EIGRP ne permettent de spécifier exactement sur quelles interfaces traiter la mise à jour de leurs informations de routage respectives. Ce qui oblige à utiliser la commande **network** suivie d'une adresse de réseau à classe pour demander les mises à jour sur toutes les interfaces de routeur relevant de cette adresse réseau. Cela signifie également que le processus de routage annonce le préfixe réseau configuré sur cette interface vers toutes les autres interfaces et le redistribue vers les autres protocoles pour lesquels la redistribution est configurée. Les listes d'accès assurent cette dernière opération. Elles empêchent la redistribution de tout préfixe IP appartenant au préfixe réseau agrégé de l'aire 10 (10.10.0.0/16) vers le processus de routage OSPF. À défaut, ces routes seraient enregistrées en tant que LSA de type 5 et arrosées sur tout le système autonome à l'exclusion des aires confinées.

Dans notre cas, les protocoles OSPF et EIGRP ou IGRP sont actifs sur toutes les interfaces appartenant au préfixe agrégé 10.10.0.0/16 des routeurs R1 et R2. Sans les listes d'accès, OSPF pense que les préfixes issus de ce réseau configurés sur ces interfaces sont aussi bien originaires de son propre processus que de provenance externe. Dans l'aire 10, les routes de ces préfixes seront inscrites dans la table de routage en tant que routes intra-aire et externes. Elles seront par conséquent annoncées sous forme de LSA de type 5 à tout le système autonome, entraînant leur prise en compte par les routeurs des autres aires comme des routes externes à OSPF.

© copyright Éditions Eyrolles

Observons ce qui se passe si l'on ôte la commande **distribute-list 10 out eigrp 10** de la configuration du routeur R1. Voyons avant cette action, l'affichage des LSA de type 5 dans la base d'état des liens du routeur R1 (listing 6.72).

**Listing 6.72. LSA de type 5 de la base d'état des liens du routeur R1.**

```
R1#show ip ospf database
...
          Type-5 AS External Link States

Link ID      ADV Router   Age      Seq#       Checksum Tag
10.201.0.0   10.10.0.1    1588     0x80000007 0x4BF0   0
10.202.0.0   10.10.0.2    1314     0x80000001 0xC8F6   0
10.203.0.0   10.10.0.1    1593     0x80000004 0x3904   0
10.204.0.0   10.10.0.2    1672     0x80000002 0xAE0E   0
```

Rien que de très normal dans cette table. Retirons à présent la commande **distribute-list** du routeur R1 pour constater le changement qui intervient dans la base LSD de ce routeur sur le listing 6.73.

**Listing 6.73. LSA de type 5 de la base du routeur R1, après retrait de la commande `distribute-list 10 out eigrp 10` dans la configuration du routeur en mode `router ospf 10`.**

```
R1#show ip ospf database
...
          Type-5 AS External Link States

Link ID      ADV Router   Age      Seq#       Checksum Tag
10.10.0.1    10.10.0.1    5        0x80000001 0x4AB6   0
10.10.1.0    10.10.0.1    5        0x80000001 0x49B7   0
10.10.255.4  10.10.0.1    5        0x80000001 0x1AE6   0
10.201.0.0   10.10.0.1    128      0x80000008 0x49F1   0
10.202.0.0   10.10.0.2    85       0x80000002 0xC6F7   0
10.203.0.0   10.10.0.1    130      0x80000005 0x3705   0
10.204.0.0   10.10.0.2    87       0x80000003 0xAC0F   0
```

Les trois routes en italique, codées « E1 » au début du listing 6.74, appartiennent visiblement à l'aire 10 de OSPF, mais elles se trouvent néanmoins redistribuées *via* EIGRP et donc stockées dans la base LSD du routeur R1 en tant que routes externes pour être diffusées en tant que LSA de type 5 de type de métrique 1 ; ce qui explique leur présence dans la table de routage du routeur R4 avec ce code. Nous pouvons y voir également une route pour le préfixe abrégé de cette même aire 10, codée « IA ».

**Listing 6.74. Table de routage du routeur R4.**

```
R4#show ip route
...
   10.0.0.0/8 is variably subnetted, 12 subnets, 4 masks
O IA 10.10.0.0/16 [110/80] via 10.0.1.1, 00:32:44, TokenRing0
O E1 10.10.1.0/24 [110/80] via 10.0.1.1, 00:02:28, TokenRing0
O E1 10.10.0.1/32 [110/80] via 10.0.1.1, 00:02:28, TokenRing0
```

© copyright Éditions Eyrolles

```
O E1 10.10.255.4/30 [110/80] via 10.0.1.1,00:02:29,TokenRing0
C    10.0.2.0/24 is directly connected, Ethernet0
O    10.0.0.3/32 [110/7] via 10.0.1.1, 00:32:44, TokenRing0
C    10.0.1.0/24 is directly connected, TokenRing0
C    10.0.0.4/32 is directly connected, Loopback0
O E2 10.202.0.0/16 [110/10] via 10.0.1.1, 00:27:48,TokenRing0
O E1 10.203.0.0/16 [110/80] via 10.0.1.1, 00:32:25,TokenRing0
O E1 10.201.0.0/16 [110/80] via 10.0.1.1, 00:32:25,TokenRing0
O E2 10.204.0.0/16 [110/10] via 10.0.1.1, 00:32:45,TokenRing0
```

### La commande `summary-address`

Le premier usage de la commande **summary-address** est l'agrégation des informations de routage, soit celles redistribuées par OSPF vers d'autres protocoles, soit celles externes importées dans ce protocole *via* un ASBR. Cette commande permet les opérations suivantes :

- l'agrégation de routes OSPF pour la redistribution vers un autre protocole de routage ;
- l'agrégation de routes externes injectées dans un système autonome OSPF *via* un routeur ASBR.

La syntaxe de cette commande est la suivante : **summary-address** <*adresse IP*> <*masque de sous-réseau*>. Les paramètres <*adresse IP*> et <*masque de sous-réseau*> spécifient le préfixe réseau agrégé. D'autres paramètres optionnels existent, que nous ne détaillerons pas ici. Ils sont décrits dans la documentation Cisco.

Nous allons maintenant voir comment cette commande peut être appliquée à la situation décrite sur la figure 6.11. Il convient d'abord d'étudier la table de routage du routeur R6 (cf. listing 6.75). On constate curieusement l'absence de routes relevant du préfixe réseau 10.0.0.0/16 (adresse CIDR de l'aire 0) ou du préfixe 10.10.0.0/16 (adresse CIDR de l'aire 10). Cela semble étrange puisque la longueur de préfixe réseau attendue par IGRP est /16, justement celle utilisée pour les deux adresses agrégées.

**Listing 6.75. Table de routage du routeur R6.**

```
R6#show ip route
...
   10.0.0.0/16 is subnetted, 5 subnets
C     10.202.0.0 is directly connected, Ethernet0
I     10.203.0.0 [100/8477] via 10.204.0.1, 00:00:24, Serial0
I     10.201.0.0 [100/8477] via 10.204.0.1, 00:00:24, Serial0
C     10.204.0.0 is directly connected, Serial0
```

Le protocole IGRP de par sa nature à classe ne peut distinguer aucun sous-réseau avec tous ses bits à zéro (*zero subnet*), ce qui explique que le préfixe de l'aire 0 est absent. Par contre, celui de l'aire 10 devrait lui être visible. Pourquoi ne l'est-il pas ?

Le routeur R2, auquel le routeur R6 est connecté, n'est pas un routeur ABR et ne peut donc opérer d'agrégation de route. Même s'il s'agissait d'un routeur ABR, R6 ne verrait toujours pas l'adresse agrégée de l'aire 10 car l'agrégation automatique de route ne se fait qu'entre routeurs ABR déroulant la processus OSPF. Pour les routeurs ABR, la commande explicite **summary-address** doit donc être utilisée sur des ASBR pour agréger des routes vers un autre protocole, en l'occurrence IGRP.

© copyright Éditions Eyrolles

Le listing 6.76 montre la nouvelle configuration du routeur R4, la ligne en italique mettant en évidence l'utilisation de la commande **summary-address**.

**Listing 6.76. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.10.0.2 255.255.255.255

interface Ethernet0
 ip address 10.10.2.1 255.255.255.0

interface Serial0
 ip address 10.204.0.1 255.255.0.0

interface Serial1
 ip address 10.10.255.10 255.255.255.252

router ospf 10
 summary-address 10.10.0.0 255.255.0.0
 redistribute igrp 10 metric 10 subnets
 network 10.10.0.0 0.0.255.255 area 10
 distribute-list 10 out igrp 10

router igrp 10
 redistribute ospf 10 metric 10000 1 255 1 1500
 passive-interface Ethernet0
 passive-interface Loopback0
 passive-interface Serial1
 network 10.0.0.0

ip classless

access-list 10 deny   10.10.0.0 0.0.255.255
access-list 10 permit any
```

La table de routage du routeur R6 montre bien la présence, cette fois-ci, d'une route pour l'adresse agrégée de l'aire 10 (cf. listing 6.77).

**Listing 6.77. Table de routage du routeur R6.**

```
R6#show ip route
...
   10.0.0.0/16 is subnetted, 5 subnets
I     10.10.0.0 [100/8477] via 10.204.0.1, 00:00:24, Serial0
C     10.202.0.0 is directly connected, Ethernet0
I     10.203.0.0 [100/8477] via 10.204.0.1, 00:00:24, Serial0
I     10.201.0.0 [100/8477] via 10.204.0.1, 00:00:24, Serial0
C     10.204.0.0 is directly connected, Serial0
```

© copyright Éditions Eyrolles

## Les aires de routage peu confinées (NSSA – Not-So-Stubby Area) et leur configuration

Il arrive qu'une aire soit une très bonne candidate au confinement, tout en nécessitant l'importation de routes externes en quantité limitée. Si une telle aire avait un statut normal, elle recevrait toutes les LSA de type 5 relevant du système autonome tout entier, sans nécessité. C'est pour de telles situations que fut conçue une aire au statut spécial « peu confinée » ou NSSA (*Not So Stubby Area*), avec un type de LSA particulier, le type 7.

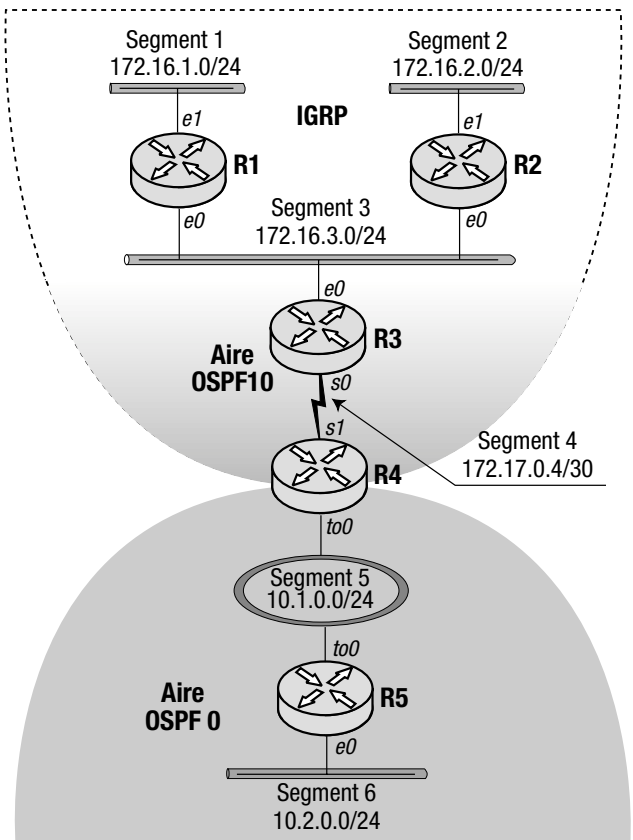
Une aire NSSA est très similaire à une aire confinée normale. Elle n'est pas autorisée à recevoir des annonces de type 5 mais peut héberger des routeurs exécutant des protocoles de routage dynamique autres que OSPF. Les informations de routage provenant de ces protocoles sont redistribuées vers OSPF, qui les annonce à son tour *via* des LSA de type 7.

Les LSA de type 7 sont très similaires aux LSA de type 5 en ce qu'elles décrivent aussi des routes externes. Mais elles en diffèrent sur les points suivants :

- Les LSA de type 7 sont arrosées uniquement dans l'aire NSSA, tandis que les LSA de type 5 sont envoyées vers tout le système autonome, aires confinées exceptées.
- Les routeurs ABR d'une aire NSSA traduisent les LSA de type 7 en LSA de type 5, pour ensuite les arroser vers l'ensemble du système autonome.

Sur la figure 6.12, vous pouvez voir un exemple de réseau dans lequel l'aire 10 peut être configurée comme aire peu confinée (NSSA). Le protocole prédominant dans cette aire est IGRP.

**Figure 6.12**
*L'aire 10 est une aire OSPF NSSA (peu confinée).*




© copyright Éditions Eyrolles

Les informations de routage IGRP sont redistribuées vers OSPF au niveau du routeur R3, puis annoncées au reste du réseau *via* un segment 4 de faible débit.

Voyons à présent comment configurer les routeurs pour faire de l'aire 10 une aire NSSA peu confinée. Les listings 6.78 à 6.82 donnent la configuration pour chacun des cinq routeurs.

**Listing 6.78. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 172.16.3.1 255.255.255.0

interface Serial0
 ip address 172.16.1.1 255.255.255.0

router igrp 172
 network 172.16.0.0
```

**Listing 6.79. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 172.16.3.2 255.255.255.0

interface Serial0
 ip address 172.16.2.1 255.255.255.0

router igrp 172
 network 172.16.0.0
```

**Listing 6.80. Configuration du routeur R3.**

```
interface Loopback0
 ip address 172.17.255.3 255.255.255.255

interface Ethernet0
 ip address 172.16.3.3 255.255.255.0

interface Serial0
 ip address 172.17.0.5 255.255.255.252

router ospf 10
 redistribute igrp 172 metric 10 metric-type 1 subnets
 network 172.17.0.0 0.0.255.255 area 10
 area 10 nssa

router igrp 172
 redistribute ospf 10
 network 172.16.0.0
 default-metric 10000 1 255 1 1500

ip classless
```

© copyright Éditions Eyrolles

**Listing 6.81. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Serial1
 ip address 172.17.0.6 255.255.255.252

interface TokenRing0
 ip address 10.1.0.1 255.255.255.0
 ring-speed 16

router ospf 10
 network 10.0.0.0 0.255.255.255 area 0
 network 172.0.0.0 0.255.255.255 area 10
 area 0 range 10.0.0.0 255.0.0.0
 area 10 nssa

ip classless
```

**Listing 6.82. Configuration du routeur R5.**

```
interface Loopback0
 ip address 10.0.0.5 255.255.255.255

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface TokenRing0
 ip address 10.1.0.2 255.255.255.0
 ring-speed 16

router ospf 1
 network 10.0.0.0 0.255.255.255 area 0

ip classless
```

**REMARQUE** Tout comme les routes OSPF externes normales, les routes externes NSSA peuvent être de type 1 ou de type 2.

La table de routage du routeur R3 ne présente pas encore de routes OSPF externes (cf. listing 6.83). Mais la base LSD de ce routeur que montre le listing 6.84 contient déjà les LSA de type 7.

**Listing 6.83. Table de routage du routeur R3.**

```
R3#show ip route
...
O IA 10.0.0.0/8 [110/80] via 172.17.0.6, 00:29:12, Serial0
   172.16.0.0/24 is subnetted, 3 subnets
I    172.16.1.0 [100/8576] via 172.16.3.1, 00:00:42,Ethernet0
I    172.16.2.0 [100/8576] via 172.16.3.2, 00:01:11,Ethernet0
C    172.16.3.0 is directly connected, Ethernet0
   172.17.0.0/16 is variably subnetted, 3 subnets, 3 masks
```

© copyright Éditions Eyrolles

```
C    172.17.255.3/32 is directly connected, Loopback0
C    172.17.0.4/30 is directly connected, Serial0
O    172.17.0.0/16 is a summary, 00:01:23, Null0
```

Remarquez que les LSA de type 7, à la différence de celles de type 5, continuent d'appartenir à leur aire d'origine.

**Listing 6.84. Partie de la base d'état des liens OSPF du routeur R3, contenant les informations des LSA de type 7.**

```
R3#show ip ospf database

       OSPF Router with ID (172.17.255.3) (Process ID 10)
...
          Type-7 AS External Link States (Area 10)

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.1.0   172.17.255.3    1115    0x80000001 0x9051   0
172.16.2.0   172.17.255.3    1115    0x80000001 0x855B   0
172.16.3.0   172.17.255.3    1115    0x80000001 0x7A65   0
```

Examinons maintenant la table de routage du routeur R4 (cf. listing 6.85). Les trois lignes en italique montrent les routes externes OSPF NSSA de type 1. À la différence des routes OSPF externes normales, celles-ci portent le code « N1 ». De même, le routes externes OSPF NSSA de type 2 sont étiquetées « N2 ».

**Listing 6.85. Table de routage du routeur R4.**

```
R4#show ip route
...
   172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O    172.17.255.3/32 [110/65] via 172.17.0.5,00:16:42,Serial1
C    172.17.0.4/30 is directly connected, Serial1
   172.16.0.0/24 is subnetted, 3 subnets
O N1 172.16.1.0 [110/75] via 172.17.0.5, 00:00:48, Serial1
O N1 172.16.2.0 [110/75] via 172.17.0.5, 00:00:48, Serial1
O N1 172.16.3.0 [110/75] via 172.17.0.5, 00:00:48, Serial1
   10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O    10.2.0.0/24 [110/16] via 10.1.0.2, 00:16:42, TokenRing0
C    10.1.0.0/24 is directly connected, TokenRing0
C    10.0.0.4/32 is directly connected, Loopback0
O    10.0.0.5/32 [110/7] via 10.1.0.2, 00:16:43, TokenRing0
```

Le routeur R4 étant un ABR de NSSA pour l'aire 10, il doit traduire les LSA de type 7 en LSA de type 5, ce qui apparaît dans la base d'état des liens du routeur R4 (cf. listing 6.86). Remarquez que cette base contient à la fois les LSA originales de type 7 et les LSA nouvelles de type 5, avec les mêmes identifiants de liens (*Link ID*).

**Listing 6.86. Partie de la base d'état des liens OSPF du routeur R4 contenant les informations des LSA de type 5 et de type 7.**

```
R4#show ip ospf database

      OSPF Router with ID (10.0.0.4) (Process ID 10)
```

© copyright Éditions Eyrolles

```
...
        Type-7 AS External Link States (Area 10)

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.1.0   172.17.255.3    1018    0x80000001 0x9051   0
172.16.2.0   172.17.255.3    1019    0x80000001 0x855B   0
172.16.3.0   172.17.255.3    1019    0x80000001 0x7A65   0

           Type-5 AS External Link States

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.1.0   10.0.0.4        970     0x80000001 0x5F3F   0
172.16.2.0   10.0.0.4        970     0x80000001 0x5449   0
172.16.3.0   10.0.0.4        970     0x80000001 0x4953   0
```

Enfin, la table de routage du routeur R5 (cf. listing 6.87) ne contient que des routes externes OSPF normales codées « E1 » annoncées à l'origine par le routeur R3 en LSA de type 7 et traduites en LSA de type 5 par le routeur R4 pour être reçues par ce routeur.

**Listing 6.87. Table de routage du routeur R5.**

```
R5#show ip route
...
   172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O IA 172.17.255.3/32 [110/71]via 10.1.0.1,00:13:48,TokenRing0
O IA 172.17.0.4/30 [110/70] via 10.1.0.1, 00:13:53,TokenRing0
   172.16.0.0/24 is subnetted, 3 subnets
O E1 172.16.1.0 [110/81] via 10.1.0.1, 00:13:47, TokenRing0
O E1 172.16.2.0 [110/81] via 10.1.0.1, 00:13:47, TokenRing0
O E1 172.16.3.0 [110/81] via 10.1.0.1, 00:13:47, TokenRing0
   10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C    10.2.0.0/24 is directly connected, Ethernet0
C    10.1.0.0/24 is directly connected, TokenRing0
O    10.0.0.4/32 [110/7] via 10.1.0.1, 00:14:01, TokenRing0
C    10.0.0.5/32 is directly connected, Loopback0
```

D'ailleurs, on ne verra que des routes externes de type 5 dans la base d'état des liens du routeur R5 (cf. listing 6.88) car ce routeur est situé dans l'aire 0, tandis que les LSA de type 7 provenaient de l'aire 10.

**Listing 6.88. La base d'état des liens OSPF du routeur R5.**

```
R5#show ip ospf database

      OSPF Router with ID (10.0.0.5) (Process ID 1)
...
           Type-5 AS External Link States

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.1.0   10.0.0.4        913     0x80000001 0x5F3F   0
172.16.2.0   10.0.0.4        914     0x80000001 0x5449   0
172.16.3.0   10.0.0.4        914     0x80000001 0x4953   0
```

© copyright Éditions Eyrolles

### Agrégation de routes externes dans OSPF avec la commande `summary-address`

Nous avons déjà utilisé la commande **summary-address** dans une précédente section quand il fallait envoyer un préfixe agrégé de OSPF vers un autre protocole. Dans le cas présent, il s'agit d'agréger des routes externes apprises à travers d'autres protocoles. Or, il n'est pas toujours possible de connaître tous les préfixes réseau disponibles dans les autres systèmes autonomes surtout que certains préfixes réseau peuvent être ajoutés ou supprimés par la suite. Cependant, une NSSA fait partie intégrante d'un même système autonome, même si elle utilise d'autres protocoles que OSPF pour leur routage dynamique. On n'aura donc souvent aucun mal à connaître tous les préfixes réseau des NSSA et donc à les agréger.

Dans le réseau illustré figure 6.12, le routeur R3 peut procéder à une telle agrégation pour les routes externes disponibles dans le domaine IGRP de l'aire 10 en utilisant le préfixe réseau 172.16.0.0/16. Le listing 6.89 montre la nouvelle configuration du routeur R3. La ligne en italique montre comment est utilisée la commande **summary-address**.

**Listing 6.89. Configuration du routeur R3, qui procède à l'agrégation des informations de routage redistribuées à partir de IGRP.**

```
interface Loopback0
 ip address 172.17.255.3 255.255.255.255

interface Ethernet0
 ip address 172.16.3.3 255.255.255.0

interface Serial0
 ip address 172.17.0.5 255.255.255.252

router ospf 10
 summary-address 172.16.0.0 255.255.0.0
 redistribute igrp 172 metric 10 metric-type 1 subnets
 network 172.17.0.0 0.0.255.255 area 10
 area 10 nssa

router igrp 172
 redistribute ospf 10
 network 172.16.0.0
 default-metric 10000 1 255 1 1500

ip classless
```

La base d'état des liens du routeur R3 ne contient qu'une LSA de type 7 qui décrit les préfixes agrégés de toutes les routes externes.

**Listing 6.90. LSA de type 7 de la base d'état des liens du routeur R3.**

```
R3#show ip ospf database

     OSPF Router with ID (172.17.255.3) (Process ID 10)
...
              Type-7 AS External Link States (Area 10)

Link ID      ADV Router      Age     Seq#       Checksum Tag
```

© copyright Éditions Eyrolles

```
172.16.0.0   172.17.255.3    153     0x80000001 0x9B47   0

           Type-5 AS External Link States

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.17.0.0   172.17.255.3    191     0x80000001 0x5D59   0
```

De même, la base d'état des liens du routeur R4 (cf. listing 6.91) contient une seule LSA de type 7 et une seule LSA de type 5.

**Listing 6.91. Base d'état des liens du routeur R4, qui contient les LSA décrivant les liens externes.**

```
R4#show ip ospf database

      OSPF Router with ID (10.0.0.4) (Process ID 10)
...
Type-7 AS External Link States (Area 10)

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.0.0   172.17.255.3    201     0x80000001 0x9B47   0

           Type-5 AS External Link States

Link ID      ADV Router      Age     Seq#       Checksum Tag
172.16.0.0   10.0.0.4        201     0x80000001 0x6A35   0
```

Bien entendu, la table de routage du routeur R4 (cf. listing 6.92) ne contient plus qu'une route externe OSPF NSSA pour le préfixe réseau agrégé.

**Listing 6.92. Table de routage du routeur R4.**

```
R4#show ip route
...
   172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O     172.17.255.3/32 [110/65] via 172.17.0.5, 00:14:23,
Serial1
C     172.17.0.4/30 is directly connected, Serial1
O N1 172.16.0.0/16 [110/75] via 172.17.0.5, 00:04:22, Serial1
   10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O     10.2.0.0/24 [110/16] via 10.1.0.2, 00:14:23, TokenRing0
C     10.1.0.0/24 is directly connected, TokenRing0
C     10.0.0.4/32 is directly connected, Loopback0
O     10.0.0.5/32 [110/7] via 10.1.0.2, 00:14:23, TokenRing0
```

Enfin, la table de routage du routeur R5 (cf. listing 6.93) contient maintenant une seule route externe OSPF pour le préfixe réseau 172.16.0.0/16, préfixe réseau agrégé pour tous les préfixes réseau gérés par IGRP disponibles dans l'aire 10.

© copyright Éditions Eyrolles

**Listing 6.93. Table de routage du routeur R5.**

```
R5#show ip route
...
   172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O IA  172.17.255.3/32 [110/71] via 10.1.0.1, 00:13:56,
TokenRing0
O IA  172.17.0.4/30 [110/70] via 10.1.0.1, 00:13:56,
TokenRing0
O E1 172.16.0.0/16 [110/81] via 10.1.0.1, 00:03:55,TokenRing0
   10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C     10.2.0.0/24 is directly connected, Ethernet0
C     10.1.0.0/24 is directly connected, TokenRing0
O     10.0.0.4/32 [110/7] via 10.1.0.1, 00:34:40, TokenRing0
C     10.0.0.5/32 is directly connected, Loopback0
```

© copyright Éditions Eyrolles

# 7

# Cas spéciaux de routage

**Solutions de configuration présentées dans ce chapitre**



## Introduction

Dans ce chapitre, nous aborderons différents cas particuliers de routage, considérés en général comme des variantes des procédures normales de routage. Il s'agit du routage sélectif (*policy-based routing*), de la traduction d'adresses réseaux (NAT), du protocole HSRP (*Hot Standby Router Protocol*) pour la tolérance aux pannes du routeur par défaut et du routage à la demande (sur ligne commutée – *Dial-On-Demand Routing*).

© copyright Éditions Eyrolles

## Le routage sélectif (*policy-based routing)*

Le procédé qui consiste à router des paquets à partir de règles bien définies, tenant compte de caractéristiques des datagrammes autres que leur destination, est appelé routage sélectif ou *policy-based routing*.

Le routage sélectif ne peut être que statique. S'il n'est pas planifié ni correctement implémenté, il peut avoir de très mauvaises conséquences sur le routage dynamique existant. Prenons un routeur R1 configuré pour du routage sélectif. Supposons que les règles implémentées dictent d'aiguiller un certain trafic vers un routeur R2 qui lui-même achemine les paquets vers l'adresse de destination disponible au travers de R1. Il adviendra que R2 redirige ce type de trafic sur R1 en retour ! Suivant la manière dont sont implémentées les règles de routages, cette situation peut aboutir à un excès de sauts (*hops*) sur une route, voire à un phénomène de boucle.

## La traduction d'adresses réseau (NAT)

La traduction d'adresses réseau NAT est une méthode qui consiste à remplacer l'adresse source ou l'adresse de destination originale dans un datagramme IP qui traverse un routeur configuré pour la NAT.

Trois types de problèmes peuvent êtres résolus en utilisant les techniques de NAT.

- **L'épuisement des adresses IP publiques disponibles –** Il est de plus en plus difficile de se voir attribuer des adresses IP publiques car l'espace disponible s'épuise. NAT aide en partie à résoudre ce problème en réutilisant certaines adresses en de multiples endroits – par exemple, il est possible d'utiliser des adresses IP privées (cf. RFC 1918, *Address Allocation for Private Internets*) sur des réseaux qui ne sont pas connectés à l'Internet, comme des réseaux privés d'entreprises.
- **La fusion de réseaux qui utilisent des espaces d'adresses IP qui se recouvrent** – Si deux compagnies utilisent des adresses IP privées ou si une société utilise des adresses IP qui appartiennent officiellement à une autre, et que l'on doive fusionner leurs réseaux, il pourra s'avérer trop coûteux d'élaborer un nouveau plan d'adressage. Dans de telles situations, la NAT peut être utilisée pour effectuer une traduction entre les réseaux dont les plages d'adresses se chevauchent.
- **L'équilibrage de charge** – Si plusieurs serveurs réalisent la même fonction, par exemple le service web, la NAT peut être appropriée pour équilibrer le trafic destiné à ce service entre les différentes machines. Vu de l'extérieur, les différentes machines apparaissent comme une seule adresse IP. Lorsque le premier paquet d'une connexion arrive au routeur réalisant ces fonctions de NAT, l'adresse de destination est remplacée par l'adresse d'une des machines serveurs. Tous les paquets suivants appartenant à cette connexion seront envoyés à cette machine.

La traduction d'adresses réseau NAT est documentée dans la RFC 1631, *NAT* (*The IP Network Address Translator*). L'utilisation de la NAT pour l'équilibrage de charge est documentée dans la RFC 2391, LSNAT (*Load Sharing Using IP Network Address Translation*).

© copyright Éditions Eyrolles

## Terminologie NAT

Du point de vue de la NAT, les réseaux se divisent en deux catégories : les réseaux internes et les réseaux externes. Les réseaux internes sont des réseaux dont les adresses IP ne sont pas *légitimes*, elles doivent être traduites en des adresses IP légitimes. Les réseaux externes sont des réseaux dont les adresses IP sont considérées comme légitimes.

**REMARQUE** Dans le contexte de la NAT, le concept de légitimité n'implique pas forcément le rattachement officiel à une autorité. Par exemple, si deux réseaux utilisant les mêmes plages d'adresses privées sont fusionnés, l'un de ces réseaux devient un réseau interne, l'autre le réseau externe. De ce fait, les adresses IP du premier réseau ne sont plus légitimes.

Les termes suivants sont utilisés pour décrire le type d'adresse IP dans le contexte de la NAT :

- Adresses internes locales – Ce sont les adresses IP des machines hôtes connectées sur le réseau interne. Ces adresses sont configurées sur les cartes réseaux des machines et doivent être traduites. Les adresses locales internes n'ont pas à être affectées officiellement et peuvent ne pas être connues du côté du réseau externe (ainsi, les routeurs connectés sur le réseau externe peuvent ne pas avoir de routes explicites vers les adresses internes locales).
- Adresses internes globales – Ce sont les adresses IP en lesquelles les adresses locales internes vont être traduites. Les adresses globales internes sont légitimes, et ainsi, doivent être connues du côté du réseau externe. Ceci suppose que les routeurs connectés sur le réseau externe doivent connaître les routes pour ces adresses internes globales.
- Adresses externes locales – Ce sont les adresses IP de l'espace d'adressage interne, par lesquelles sont connues les machines connectées au réseau externe. Ces adresses peuvent ne pas être légitimes. Elles doivent être routables dans l'espace d'adressage interne local.
- Adresses externes globales – Ce sont les adresses IP des machines connectées au réseau externe. Ces adresses doivent être légitimes et doivent être routables dans l'espace d'adressage global externe.

**REMARQUE** Toutes ces catégories d'adresses ne sont pas systématiquement utilisées dans une configuration NAT.

Les routeurs configurés pour la NAT tiennent à jour une table appelée table NAT. Chaque entrée de cette table NAT contient cinq champs, le protocole, l'adresse IP locale interne, l'adresse IP globale interne, l'adresse IP locale externe et l'adresse IP globale externe. La fonction des quatre derniers champs est de garder la correspondance entre les adresses. Le premier champ repère le protocole IP dont la connexion doit être traduite en utilisant les adresses IP contenues dans l'enregistrement. Selon la formule de NAT employée, ces champs peuvent être requis en partie ou en totalité.

Les entrées de la table NAT peuvent être remplies de deux façons différentes. La première est statique ; les entrées contenues dans la table sont configurées manuellement. Une fois les entrées enregistrées dans le routeur, elles apparaissent instantanément dans la table NAT. La deuxième est dynamique ; les entrées de la table NAT sont créées dynamiquement chaque fois qu'un datagramme IP, dont les caractéristiques satisfont les règles préalablement configurées dans le routeur, parvient à ce routeur.

© copyright Éditions Eyrolles

## Le protocol HSRP

HSRP est un protocole propriétaire Cisco, il offre un mécanisme de tolérance aux pannes de la passerelle par défaut (c'est-à-dire le routeur désigné dans les routes par défaut sur les postes du réseau) aux différentes machines du réseau qui sont incapables de découvrir dynamiquement les routeurs qui leur sont affectés.

**REMARQUE** Il est recommandé de ne pas utiliser HSRP si les machines peuvent découvrir dynamiquement leur routeur. Cependant, HSRP étant très répandu, est souvent préféré aux méthodes dynamiques comme *ICMP Router Discovery Protocol (IRDP)*. La popularité de HSRP peut être expliquée par sa facilité de configuration et le fait qu'il nécessite très peu ou pas du tout de modifications sur les machines.

HSRP est entièrement décrit dans la RFC 2281, *HSRP* (*Cisco Hot Standby Router Protocol*). Ce document est classé « pour information », ce qui veut dire que HSRP n'est pas un standard Internet.

L'idée sous-jacente à HSRP est simple. Supposons que deux routeurs au moins soient connectés à un même segment, et que l'un de ces routeurs serve comme passerelle par défaut pour l'ensemble des hôtes de ce segment. *A priori*, seul ce routeur peut envoyer le trafic sortant généré par les machines du segment. (Le trafic entrant peut, quant à lui, arriver par n'importe quel routeur de ce segment). Supposons que les autres routeurs surveillent l'activité du dit routeur jouant passerelle par défaut pour le réseau. Si ce routeur venait à *tomber*, un des autres routeurs prendrait alors son adresse IP. Les machines ne verraient ainsi aucun dysfonctionnement dans l'acheminement de leurs datagrammes vers l'extérieur.

Dans HSRP, le routeur prenant l'adresse IP du routeur défaillant, s'octroie aussi son adresse MAC. Ceci est important, car sans cette fonctionnalité, les machines doivent d'abord supprimer l'entrée ARP correspondant à l'ancien routeur dans leur table, puis ensuite initier une autre requête ARP pour retrouver la nouvelle adresse MAC du routeur suppléant.

Le routeur HSRP ayant le rôle de routeur par défaut pour les postes d'un segment est appelé routeur actif (*active router*). L'adresse IP que les machines hôtes utilisent pour leur routeur par défaut est appelée adresse IP virtuelle, elle est différente de l'adresse IP configurée sur l'interface réseau du routeur par défaut. L'adresse MAC correspondant à l'adresse IP virtuelle du routeur par défaut est appelée adresse MAC virtuelle, elle peut être identique à l'adresse MAC réelle mais ce n'est pas toujours le cas.

En sus du routeur actif, HSRP désigne aussi un routeur en attente (*standby router*) qui n'est autre que le routeur qui va prendre l'adresse MAC et l'adresse IP du routeur actif en cas de défaillance de celui-ci. Il ne peut y avoir qu'un seul routeur actif et un seul routeur en attente sur un même segment.

HSRP lui-même est un protocole très simple similaire au *hello protocol*. Le routeur actif et le routeur en attente envoient régulièrement à tous les autres routeurs HSRP du segment des paquets HSRP. Le principal motif de ces envois est d'avertir les autres routeurs HSRP de l'existence des routeurs actif et en attente. Si les autres routeurs HSRP ne reçoivent plus ces données HSRP pendant une certaine durée, un des routeurs HSRP est désigné comme remplaçant du routeur défaillant.

L'information véhiculée dans les paquets HSRP est la suivante :

- Priorité d'attente (*standby priority*) – désigne la priorité d'attente du routeur envoyant la trame ; c'est un entier compris entre 0 et 255 utilisé pour déterminer qui sera le routeur

© copyright Éditions Eyrolles

HSRP actif et qui sera le routeur en attente. Le routeur ayant la priorité d'attente la plus élevée devient le routeur actif. Le routeur ayant la priorité d'attente juste inférieure à celle du routeur actif mais supérieure à celle de tous les autres routeurs devient le routeur en attente.

**REMARQUE** Si deux routeurs HSRP ou plus ont la même priorité d'attente, c'est le routeur qui a la plus haute adresse IP sur l'interface HSRP activée qui a préséance.

- L'identifiant du groupe d'attente (*Standby group*) – désigne l'identifiant du groupe d'attente du routeur envoyant la trame. C'est un entier compris entre 0 et 2 sur les interfaces Token Ring, et entre 0 et 255 pour les autres médias. Les routeurs ayant le même identifiant de groupe d'attente simuleront la même adresse MAC virtuelle et une adresse IP virtuelle primaire et éventuellement des adresses IP virtuelles secondaires. Les nombreux routeurs connectés à un segment pouvant être configurés comme éléments de plusieurs groupes d'attente, il peut y avoir recouvrement, c'est à dire qu'un même routeur peut être membre de plusieurs groupes d'attente. L'élection d'un routeur actif et d'un routeur en attente se fait par groupe d'attente indépendamment les uns des autres. En d'autres termes, le champ groupe d'attente dans les paquets HSRP est utilisé pour démultiplexer les paquets HSRP entre les différents groupes d'attente. Chacun de ces groupes d'attente a son propre routeur actif et son propre routeur en attente. L'état d'un routeur en tant que membre d'un groupe d'attente est indépendant de son état en tant que membre d'un autre groupe d'attente.

Lorsque HSRP est configuré sur l'interface d'un routeur, celui-ci écoute les trames HSRP émanant du segment sur lequel est connectée l'interface. Si le routeur découvre qu'il est le seul routeur HSRP du segment, il devient le routeur actif. S'il découvre un routeur déjà actif avec une plus haute priorité que lui et pas de routeur en attente, le nouveau routeur devient le routeur en attente. Si le nouveau routeur découvre que sa priorité est plus élevée que celle du routeur actif ou du routeur en attente, il devient alors le nouveau routeur actif ou le nouveau routeur en attente, pour autant qu'il soit configuré avec les fonctions de préemption. Cette fonction de préemption est une valeur binaire qui indique si un routeur avec une priorité plus haute que le routeur actif ou que le routeur en attente peut les remplacer. Si le routeur remplacé n'est plus ni actif ni en attente, il arrête d'envoyer des trames HSRP.

HSRP est un protocole au-dessus de UDP dans le modèle DoD. Les trames HSRP sont encapsulées dans les datagrammes UDP, qui sont envoyés à l'adresse multicast 224.0.0.2 sur le port 1985. La valeur du champ TTL (*Time To Live*) dans le datagramme IP est mise à 1. En d'autres termes, ce datagramme reste local au segment sur lequel HSRP est mis en place.

En fonction du type de média de l'interface configurée pour HSRP, une adresse MAC spécifique est utilisée comme adresse MAC virtuelle. Les adresses MAC virtuelles dans un Token Ring sont C0-00-00-01-00-00, C0-00-00-02-00-00 ou C0-00-00-04-00-00, qui correspondent respectivement aux groupes 0, 1 ou 2. Sur les autres médias, les adresses MAC virtuelles doivent être 00-00-0C-07-AC-XX où XX désigne le numéro de groupe d'attente HSRP. Le routeur actif doit utiliser l'adresse MAC virtuelle comme adresse source pour les paquets de niveau *liaison* (*Data Link Layer*) transportant les datagrammes UDP, eux-mêmes contenant les trames HSRP. Aucun autre routeur HSRP, pas même le routeur en attente ne doit utiliser cette adresse MAC virtuelle comme adresse source dans les paquets émis. Ceci est nécessaire pour que HSRP puisse fonctionner dans le cas d'une installation avec des ponts transparents.

© copyright Éditions Eyrolles

## Le routage *composé à la demande*

Le routage composé à la demande (*Dial-On-Demand Routing*, DDR) est un vaste sujet qui nécessiterait un ouvrage spécifique. Ici, nous considèrerons seulement les deux configurations les plus classiques. Ces configurations sont le routage instantané (*snapshot routing*) et le secours commuté (*dial backup*). Toutes les informations concernant ces deux techniques se trouvent dans la section « Configuration du routage à la demande » en fin de section.

# *Solutions de configuration*

## La configuration du routage sélectif (*Policy-Based Routing)*

La configuration du routage sélectif sur les routeurs Cisco est basée sur le concept de séquences conditionnelles ou *route-maps*, qui sont appliquées comme des règles de routage sur les interfaces des routeurs. Ainsi les règles d'itinéraires sont appliquées sur chacun des datagrammes arrivant sur les interfaces.

Pour configurer le routage sélectif, il faut procéder aux étapes suivantes :

1. Créer une *route-map* dont la clause contient l'instruction **match ip address {**<*numéro de LA*>|<*nom de LA*>**}**, ainsi que l'un des groupes d'actions suivantes : **set interface** <*interface*> suivi de **set ip next-hop** <*adresse IP*> ou **set default interface** <*interface*> suivi de **set ip default next-hop** <*adresse IP*>. Utilisez la syntaxe décrite au chapitre 6 pour éditer une règle de routage.

   Le paramètre **{**<*numéro de LA*>|<*nom de LA*>**}** désigne le numéro ou le nom de la liste d'accès qui définit les datagrammes qui doivent subir les règles de routage. Si la liste d'accès renvoie **permit,** les actions entrées par **set** seront entreprises, sinon, les caractéristiques du datagramme seront comparées à la clause **match** suivante.

   Les actions **set interface** <*interface*> et **set ip next-hop** <*adresse IP*> sont utilisées par le routeur pour décider du routage applicable aux datagrammes validés par la clause **match**. Ces datagrammes sont routés *via* l'interface <*interface*> et au travers du prochain routeur dont l'adresse IP est <*adresse IP*>. L'action **set** contient le mot clé **default**, et seuls les datagrammes destinés aux adresses IP dont les routes sont absentes des tables de routage sont aiguillés selon les actions **set**.

**AVERTISSEMENT** Bien que les instructions **set interface** <*interface*> et **set ip next-hop** <*adresse IP*> soient facultatives, vérifiez que vous les utilisez ensemble. Si l'une ou l'autre est omise, le routeur remplace l'instruction absente par l'information correspondante dans sa table de routage. Cependant cette adresse IP peut ne pas être visible via l'interface configurée par l'action **set interface** <*interface*>. Dans ce cas, les datagrammes ne seront aiguillés par aucun routeur, et la connexion ne sera pas établie.

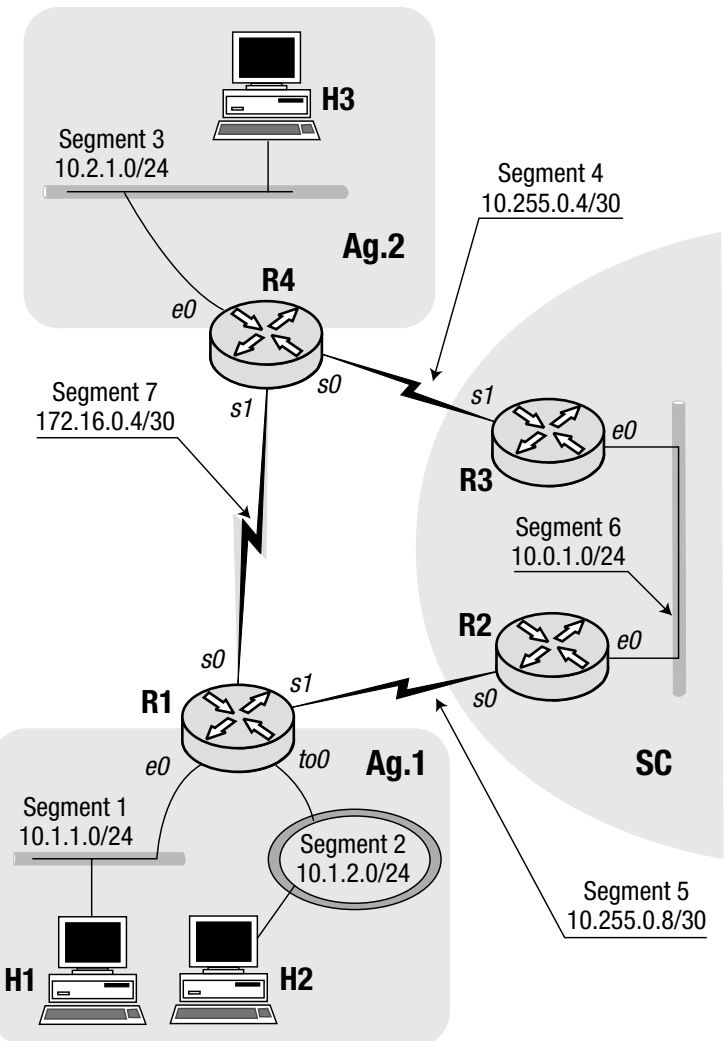
2. Appliquer la *route-map* sur l'interface appropriée en utilisant la commande **ip policy route-map** <*nom de la RM*>, où le paramètre <*nom de la RM*> désigne le nom de la *route-map*. Les datagrammes reçus sur cette interface sont comparés à la *route-map*. Si l'une des clauses de la *route-map* renvoie **permit**, le datagramme subit les règles de routage. Sinon, le datagramme est routé selon la table de routage.

© copyright Éditions Eyrolles

**REMARQUE** Les règles d'itinéraires (*route-maps*) en tant que règles de routage ne s'appliquent que sur le trafic entrant, contrairement aux listes d'accès

Le routage sélectif a de nombreuses applications, qui ne peuvent pas toutes être évoquées dans ce livre. Nous expliciterons son fonctionnement au travers de deux exemples basés sur la même topologie réseau. Ce réseau est composé d'un siège social avec son propre réseau (la partie grise appelée « SC ») et de nombreuses agences, dont deux seulement sont représentées sur la figure 7.1 (les zones grisées appelées « Ag.1 » et « Ag.2 »). Cette topologie est en étoile (*hub-and-spoke*), c'est-à-dire que les filiales sont interconnectées au travers du réseau central

**Figure 7.1**
*La topologie réseau d'un routage sélectif*




Cependant, deux agences ont mis en place une « maille » supplémentaire (appelée *segment 7* dans la figure 7.1), dont l'utilité sera précisée plus loin dans les exemples.

Nous choisissons ces deux exemples car ils sont caractéristiques des cas d'utilisation du routage sélectif.

© copyright Éditions Eyrolles

### Utilisation du routage sélectif *(policy-based routing)* pour du routage sur ligne spécialisée

Supposons que les agences Ag.1 et Ag.2 aient des utilisateurs se plaignant fréquemment des faibles performances de leurs machines pendant les heures chargées du siège social. Supposons aussi que ces utilisateurs soient répartis sur les deux agences et qu'ils doivent aussi communiquer entre eux. Les agences ont décidé d'installer une ligne spécialisée (segment 7) spécifiquement pour réunir les deux réseaux sans passer par le site central.

Supposons que les utilisateurs autorisés à utiliser le segment 7 sont branchés sur le segment 1 dans l'agence Ag.1 et sur le segment 3 dans l'agence Ag.2.

On peut croire de prime abord que l'on peut résoudre ce problème en utilisant un routage statique. Mais en y regardant de plus près, nous nous apercevons qu'un routage statique ne fera pas de distinction entre les trames provenant du segment 1 ou du segment 2 de l'agence Ag.1. Si elles sont destinées au segment 3 de l'agence Ag.2, un routage classique les fera passer par le segment 7. Dans notre cas, nous ne voulons router au travers du segment 7 que les trames émanant du segment 1 et tout autre trafic doit être routé selon les tables de routage.

En revanche, cette tâche peut être facilement résolue avec le routage sélectif. L'idée est de définir des règles de routage sur les interfaces Ethernet des routeurs R1 et R4 en utilisant des règles d'itinéraires (*route-map*) concordant avec les trames des segments 1 et 3.

Les listings 7.1 à 7.4 montrent les configurations des quatre routeurs. Le routage sélectif n'est en place que sur les routeurs R1 et R4. On voit les configurations spécifiques en caractères italiques.

**Listing 7.1. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip policy route-map Seg1-Seg6

interface Serial0
 ip address 172.16.0.5 255.255.255.252

interface Serial1
 ip address 10.255.0.10 255.255.255.252

interface TokenRing0
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16

router eigrp 10
 network 10.0.0.0

access-list 100 permit ip any 10.2.1.0 0.0.0.255

route-map Seg1-Seg6 permit 10
 match ip address 100
 set interface Serial0
 set ip next-hop 172.16.0.6
```

© copyright Éditions Eyrolles

**Listing 7.2. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
 ip address 10.255.0.9 255.255.255.252

router eigrp 10
 network 10.0.0.0
```

**Listing 7.3. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Ethernet0
 ip address 10.0.1.2 255.255.255.0

interface Serial1
 ip address 10.255.0.5 255.255.255.252

router eigrp 10
 network 10.0.0.0
```

**Listing 7.4. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.2.1.1 255.255.255.0
 ip policy route-map Seg6-Seg1

interface Serial0
 ip address 10.255.0.6 255.255.255.252

interface Serial1
 ip address 172.16.0.6 255.255.255.252

router eigrp 10
 network 10.0.0.0

access-list 100 permit ip any 10.1.1.0 0.0.0.255

route-map Seg6-Seg1 permit 10
 match ip address 100
 set interface Serial1
 set ip next-hop 172.16.0.5
```

© copyright Éditions Eyrolles

Bien que nous soyons intéressés par une réglementation du routage en fonction de la source des trames, notez que nous ne spécifions pas cette origine dans la liste d'accès (*access-list 100* sur les deux routeurs). Ceci est dû au fait que nous faisons du routage sélectif par interface. Dans notre cas, toutes les trames arrivant sur une certaine interface et à destination d'une autre interface doivent être routées de manière sélective. Ainsi, nous pouvons collecter toutes les trames en ne spécifiant que leur destination dans la liste d'accès. Mais si nous voulons réglementer plus finement, c'est à dire autoriser les trames de certaines machines seulement du segment 1 et non pas toutes, à passer au travers du segment 7 pour arriver sur le segment 3, il faudra expliciter leur adresse IP dans la liste d'accès et ne plus utiliser le mot clé **any**.

**AVERTISSEMENT** Les listes d'accès standard peuvent être utilisées dans la définition des itinéraires (*route-maps*) mais elles ne précisent que l'adresse source, et non l'adresse de destination.

**REMARQUE** Le routage sélectif est assez similaire au routage statique. Dans la plupart des cas, il sera donc nécessaire de configurer correctement tous les routeurs en jeu.

Le routage sélectif n'affecte en rien les tables de routage des routeurs en place. Le listing 7.5 montre la table de routage du routeur R1.

**Listing 7.5. La table de routage du routeur R1.**

```
R1#show ip route
...
 172.16.0.0/30 is subnetted, 1 subnets
C  172.16.0.4 is directly connected, Serial0
 10.0.0.0/8 is variably subnetted, 10 subnets, 3 masks
D  10.0.0.2/32 [90/2297856] via 10.255.0.9, 06:46:53, Serial1
C  10.1.2.0/24 is directly connected, TokenRing0
D  10.2.1.0/24 [90/2733056] via 10.255.0.9, 06:46:53, Serial1
D  10.0.0.3/32 [90/2323456] via 10.255.0.9, 06:46:53, Serial1
C  10.1.1.0/24 is directly connected, Ethernet0
D  10.0.1.0/24 [90/2195456] via 10.255.0.9, 06:46:53, Serial1
C  10.0.0.1/32 is directly connected, Loopback0
D  10.0.0.4/32 [90/2835456] via 10.255.0.9, 06:46:53, Serial1
D  10.255.0.4/30 [90/2707456] via 10.255.0.9,06:46:53,Serial1
C  10.255.0.8/30 is directly connected, Serial1
```

Afin de vérifier que le routage sélectif fonctionne correctement, utilisons la commande **traceroute** sur la machine H1 pour atteindre différentes destinations. Dans notre exemple, les machines sont des stations de travail Windows NT, et la commande **traceroute** est appelée **tracert**.

Le listing 7.6 montre la sortie écran de la commande **tracert -d 10.2.1.120**, où 10.2.1.120 est l'adresse IP de la machine H3. La commande confirme que les paquets envoyés par une machine du segment 1 vers une machine du segment 3 sont bien aiguillés au travers du segment 7 (remarquez les lignes en italique).

**Listing 7.6. Sortie écran de `tracert -d 10.2.1.120` sur la machine H1.**

```
C:\>tracert -d 10.2.1.120
```

© copyright Éditions Eyrolles

```
Tracing route to 10.2.1.120 over a maximum of 30 hops

  1    20 ms    10 ms    10 ms  10.1.1.1
  2    31 ms    20 ms    20 ms  172.16.0.6
  3    40 ms    40 ms    40 ms  10.2.1.120

Trace complete.
```

Le listing 7.7 montre la sortie écran de la commande **tracert -d 10.0.1.2**, où 10.0.1.2 est l'adresse IP de l'interface Ethernet0 du routeur R3. Cette fois, les paquets ne sont pas routés au travers du segment 7, car ils ne sont pas destinés au segment 3.

**Listing 7.7. Sortie écran de la commande `tracert -d 10.0.1.2` exécutée sur la machine H1.**

```
C:\>tracert -d 10.0.1.2

Tracing route to 10.0.1.2 over a maximum of 30 hops

  1    10 ms    10 ms    10 ms  10.1.1.1
  2    10 ms    10 ms    10 ms  10.255.0.9
  3    10 ms    10 ms    10 ms  10.0.1.2

Trace complete.
```

Enfin, le listing 7.8 montre la sortie écran de la commande **tracert -d 10.2.1.120**. Mais cette commande est exécutée sur la machine H2, qui est sur le segment 2 (l'anneau Token Ring). Ainsi que le montre la sortie écran, les règles de routage ne s'appliquent pas au trafic destiné au segment 3 émanant d'un segment autre que le segment 1.

**Listing 7.8. Sortie écran de la commande `tracert -d 10.2.1.120` exécuté sur la machine H2.**

```
C:\>tracert -d 10.2.1.120

Tracing route to 10.2.1.120 over a maximum of 30 hops

  1   <10 ms   <10 ms   <10 ms  10.1.2.1
  2   <10 ms    10 ms   <10 ms  10.255.0.9
  3   <10 ms    10 ms   <10 ms  10.0.1.2
  4   <10 ms    10 ms   <10 ms  10.255.0.6
  5   <10 ms    11 ms   <10 ms  10.2.1.120

Trace complete.
```

### L'utilisation du routage sélectif pour un routage de niveau applicatif

Supposons maintenant que les agences Ag.1 et Ag.2 ont des applications critiques en matière de trafic réseau (comme des bases de données, etc.) qui ne peuvent pas bien fonctionner durant les périodes d'occupation intense du réseau du site central. Cette fois-ci, les agences ont décidé d'utiliser le segment 7 uniquement pour ce type d'applications. Comme

© copyright Éditions Eyrolles

auparavant, les applicatifs sont situés sur le segment 1 de l'agence Ag.1 et sur le segment 3 de l'agence Ag.2.

Pour l'exemple, l'applicatif considéré sera telnet (port 23). Ainsi, notre tâche est ici, d'aiguiller uniquement le trafic telnet émanant du segment 1 à destination du segment 3 au travers du segment 7. Le trafic en retour, doit aussi utiliser le segment 7. Mais le trafic telnet émanant du segment 3 à destination du segment 1 ne doit pas utiliser cet itinéraire.

Nous rappelons ce que nous avons vu au chapitre 1, une connexion telnet est initiée par un client telnet, qui utilise *via* son système d'exploitation, un port TCP arbitraire. Ce port est utilisé comme port source par l'applicatif client. Le port TCP de destination est, quant à lui, le port *bien connu (well-known port)* 23. Le serveur telnet utilise le port 23 comme port TCP source.

Ainsi, nous devons modifier les listes d'accès sur les routeurs R1 et R2 comme suit :

- La liste d'accès sur le routeur R1 doit détecter les datagrammes contenant des segments TCP destinés au port TCP 23.
- La liste d'accès sur le routeur R4 doit détecter les datagrammes contenant des segments TCP avec un port TCP source 23.

Les listings 7.9 et 7.10 montrent les nouvelles configurations des routeurs R1 et R4. Les configurations des routeurs R2 et R3 sont inchangées.

**Listing 7.9. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip policy route-map Seg1-Seg6

interface Serial0
 ip address 172.16.0.5 255.255.255.252

interface Serial1
 ip address 10.255.0.10 255.255.255.252

interface TokenRing0
 ip address 10.1.2.1 255.255.255.0
 ring-speed 16

router eigrp 10
 network 10.0.0.0

ip access-list extended telnet172
 permit tcp any 10.2.1.0 0.0.0.255 eq telnet

route-map Seg1-Seg6 permit 10
 match ip address telnet172
 set interface Serial0
 set ip next-hop 172.16.0.6
```

© copyright Éditions Eyrolles

**Listing 7.10. Configuration du routeur R4.**

```
interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.2.1.1 255.255.255.0
 ip policy route-map Seg6-Seg1

interface Serial0
 ip address 10.255.0.6 255.255.255.252

interface Serial1
 ip address 172.16.0.6 255.255.255.252

router eigrp 10
 network 10.0.0.0

ip access-list extended telnet172
 permit tcp any eq telnet 10.1.1.0 0.0.0.255

route-map Seg6-Seg1 permit 10
 match ip address telnet172
 set interface Serial1
 set ip next-hop 172.16.0.5
```

Remarquez que nous avons utilisé des listes d'accès *nommées* au lieu de simples listes d'accès étendues. Ceci est optionnel.

Pour vérifier que le routage sélectif fonctionne correctement, nous ne pouvons plus utiliser la commande **traceroute**. La commande **traceroute** envoie du trafic UDP vers un port UDP inexistant et ne serait d'aucune utilité dans notre routage TCP ici. À la place, nous devons nous rabattre sur la commande **debug ip policy** pour visualiser comment le routeur effectue son routage sélectif.

Les listings 7.11 et 7.12 montrent les sorties écran des commandes **debug ip policy** sur les routeurs R1 et R4 respectivement, une fois que la commande **ping 10.2.1.120** est exécutée sur la machine H1. Du fait que le trafic ICMP de **ping** n'est détecté par la liste d'accès d' aucun routeur, ce trafic ICMP n'est pas routé selon les règles. (Remarquez les lignes en italique *policy rejected - normal forwarding*)

**Listing 7.11. Sortie écran de la commande `debug ip policy` sur le routeur R1, une fois entrée la commande `ping 10.2.1.120` sur la machine H1.**

```
R1#debug ip policy
Policy routing debugging is on
R1#
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial1), len 100,
 policy rejected -- normal forwarding
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial1), len 100,
 policy rejected -- normal forwarding
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial1), len 100,
 policy rejected -- normal forwarding
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial1), len 100,
 policy rejected -- normal forwarding
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial1), len 100,
 policy rejected -- normal forwarding
```

© copyright Éditions Eyrolles

**Listing 7.12. Sortie écran de la commande `debug ip policy` sur le routeur R4, une fois entrée la commande `ping 10.2.1.120` sur la machine H1.**

```
R4#debug ip policy
Policy routing debugging is on
R4#
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial0), len 100,
 policy rejected -- normal forwarding
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial0), len 100,
 policy rejected -- normal forwarding
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial0), len 100,
 policy rejected -- normal forwarding
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial0), len 100,
 policy rejected -- normal forwarding
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial0), len 100,
 policy rejected -- normal forwarding
```

Mais, si nous essayons la commande telnet depuis la machine H1 vers la machine H3, la sortie écran de la commande **debug ip policy** sur les deux routeurs montre que le trafic telnet est routé selon les règles (voir listing 7.13 et 7.14).

**Listing 7.13. Sortie écran de la commande `debug ip policy` sur le routeur R1, une fois entrée la commande `telnet 10.2.1.120` sur la machine H1.**

```
R1#debug ip policy
Policy routing debugging is on
R1#
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120, len 44,
 policy match
IP: route map Seg1-Seg6, item 10, permit
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial0),
 len 44, policy routed
IP: Ethernet0 to Serial0 172.16.0.6
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120, len 40,
 policy match
IP: route map Seg1-Seg6, item 10, permit
IP: s=10.1.1.10 (Ethernet0), d=10.2.1.120 (Serial0),
 len 40, policy routed
IP: Ethernet0 to Serial0 172.16.0.6
...
```

**Listing 7.14. Sortie écran de la commande `debug ip policy` sur le routeur R4, une fois entrée la commande `telnet 10.2.1.120` sur la machine H1.**

```
R4#debug ip policy
Policy routing debugging is on
R4#
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10, len 44,
 policy match
IP: route map Seg6-Seg1, item 10, permit
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial1),
 len 44, policy routed
```

© copyright Éditions Eyrolles

```
IP: Ethernet0 to Serial1 172.16.0.5
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10, len 40,
 policy match
IP: route map Seg6-Seg1, item 10, permit
IP: s=10.2.1.120 (Ethernet0), d=10.1.1.10 (Serial1), len 40,
 policy routed
IP: Ethernet0 to Serial1 172.16.0.5
...
```

La sortie de la commande **debug ip policy** est assez explicite. Si vous avez des questions sur une information particulière, référez-vous à la documentation de Cisco.

Une autre commande intéressante est **show route map**. Cette commande rappelle les clauses de l'itinéraire mais retrace aussi l'utilisation des règles de routage. La dernière ligne de la sortie écran de la commande comptabilise le nombre de détections et le nombre d'octets ainsi routés.

Exemple de sortie écran de la commande **show route-map** sur le listing 7.15.

**Listing 7.15. Sortie de la commande `show route-map` exécutée sur le routeur R4.**

```
R4#show route-map
route-map Seg6-Seg1, permit, sequence 10
  Match clauses:
    ip address (access-lists): telnet172
  Set clauses:
    interface Serial1
    ip next-hop 172.16.0.5
  Policy routing matches: 241 packets, 19386 bytes
```

## Configuration de la traduction d'adresses NAT (*Network Address Translation)*

La section suivante donne des indications pour la configuration du NAT sur les routeurs Cisco.

### Configuration de la NAT statique d'adresses IP internes

Il y a plusieurs configurations statiques de la NAT. La traduction statique d'adresses IP internes en est une parmi d'autres.

La traduction statique d'adresses IP internes permet, en configurant le routeur, de traduire une adresse interne locale donnée en une adresse interne (affectée en fait sur l'interface « extérieure » du routeur) connue de l'extérieur. Le routeur compare l'adresse IP source d'un datagramme reçu sur l'interface « intérieure » et destiné à l'extérieur avec les entrées de la table NAT. Si l'adresse IP est trouvée dans la table elle est alors remplacée par l'adresse IP correspondante connue de l'extérieur.

Pour configurer la traduction statique d'adresse IP, il faut suivre ces étapes :

1. Créer une correspondance NAT entre une adresse interne locale et une adresse interne connue de l'extérieur avec la commande **ip nat inside source** <*adresse IP locale*> <*adresse IP connue de l'extérieur*>. Le paramètre <*adresse IP locale*> est l'adresse IP interne locale qui va être traduite en l'adresse interne connue de l'extérieur passée dans le paramètre <*adresse IP connue de l'extérieur*>.

© copyright Éditions Eyrolles

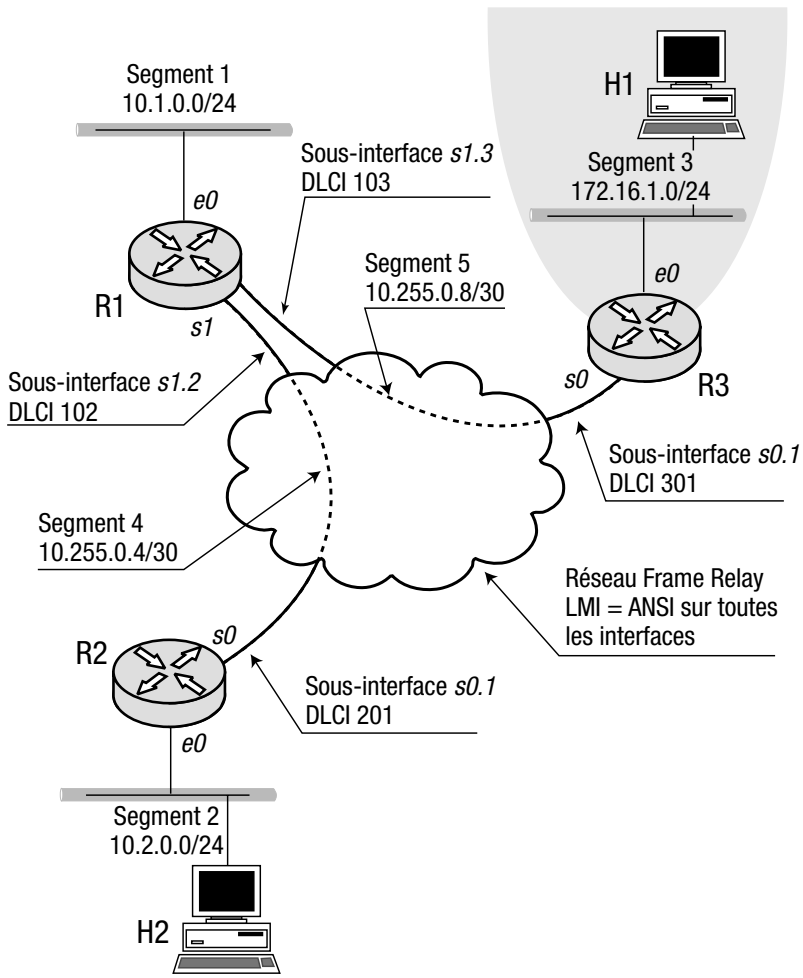
2. Activer la NAT sur les interfaces connectées aux segments d'adresses internes locales avec la commande **ip nat inside** dans le mode configuration de l'interface.
3. Activer la NAT sur les interfaces connectées aux segments d'adresses extérieurs avec la commande **ip nat outside** dans le mode configuration de l'interface.

**REMARQUE** L'adresse interne dite « globale » est une adresse interne connue de l'espace extérieur. En outre, les autres routeurs du réseau doivent pouvoir atteindre cette adresse interne connue de l'extérieur *via* le routeur NAT. Ceci peut être le fruit d'une déclaration statique de route dans ces routeurs ou d'une publication automatique de cette route par le routeur NAT.

La figure 7.2 décrit un exemple de réseau où le routeur R3 est connecté au segment 3 dont la plage d'adresses n'est pas légitime vue du reste du réseau. Aussi, le routeur R3 doit-il utiliser la NAT pour permettre à la machine H1 (du segment 3) de communiquer avec le reste du réseau.

**Figure 7.2**

*La zone grisée est un espace d'adresse illégal du point de vue du reste du réseau*




Les listings 7.16 à 7.18 expliquent la configuration des trois routeurs. Remarquez que seul le routeur R3 est configuré pour faire de la NAT. Les commandes spécifiques à la NAT sont en caractères italiques.

© copyright Éditions Eyrolles

**Listing 7.16. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial1
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial1.2 point-to-point
 ip address 10.255.0.5 255.255.255.252
 frame-relay interface-dlci 102

interface Serial1.3 point-to-point
 ip address 10.255.0.9 255.255.255.252
 frame-relay interface-dlci 103

router eigrp 10
 network 10.0.0.0
```

**Listing 7.17. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.6 255.255.255.252
 frame-relay interface-dlci 201

router eigrp 10
 network 10.0.0.0
```

**Listing 7.18. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301
```

© copyright Éditions Eyrolles

```
router eigrp 10
 network 10.0.0.0

ip nat inside source static 172.16.1.111 10.100.0.111
```

Le listing 7.19 montre qu'une session telnet de H1 sur H2 fonctionne.

**Listing 7.19. La session `telnet` du client H1 sur le serveur H2 fonctionne.**

```
C:\>telnet 10.2.0.120

Welcome to the Telnet Service on THUNDER

Username:
```

Comme prévu, la sortie écran de la commande **netstat -n** exécutée sur le serveur H2 (cf. listing 7.20) indique que la session telnet est active entre l'adresse IP du serveur H2 et l'adresse « interne » connue de l'extérieur (10.100.0.111) utilisée dans la configuration statique du NAT.

**Listing 7.20. La sortie écran de la commande `netstat -n` exécutée sur le serveur H2 montre que H2 a une connexion telnet active avec l'adresse 10.100.0.111.**

```
C:\WINDOWS\system32>netstat -n

Active Connections

Proto  Local Address          Foreign Address        State
TCP    10.2.0.120:23          10.100.0.111:1052      ESTABLISHED
TCP    127.0.0.1:1027         127.0.0.1:1028         ESTABLISHED
TCP    127.0.0.1:1028         127.0.0.1:1027         ESTABLISHED
```

Une connexion telnet depuis la machine H2 vers la machine H1 fonctionne tout aussi bien comme le montre le listing 7.21.

**Listing 7.21. La session telnet vers la machine H1 initiée depuis la machine H2 fonctionne.**

```
C:\>telnet 10.100.0.111

Welcome to the Telnet Service on HUGEWAVE

Username:
```

La sortie écran de la commande **netstat -n** exécutée sur la machine H1 (cf. listing 7.22) indique que la session telnet est active entre l'adresse IP de H1, qui est une adresse locale interne (illégale) et l'adresse IP de H2 qui est une adresse de la plage externe.

**Listing 7.22. La sortie écran de la commande `netstat -n` confirme que le routeur traduit l'adresse interne globale en l'adresse interne locale.**

```
C:\WINDOWS\system32>netstat -n

Active Connections

Proto  Local Address          Foreign Address        State
TCP    127.0.0.1:1026         127.0.0.1:1027         ESTABLISHED
TCP    127.0.0.1:1027         127.0.0.1:1026         ESTABLISHED
```

© copyright Éditions Eyrolles

```
TCP    172.16.1.111:23      10.2.0.120:1111      ESTABLISHED

C:\WINDOWS\system32>
```

Cependant, la sortie écran de la commande **netstat -n** exécutée sur la machine H2 (cf. listing 7.23) montre que la connexion telnet est active entre l'adresse IP de H2 et l'adresse interne connue de l'extérieur 10.100.0.111

**Listing 7.23. La sortie écran de la commande `netstat -n` sur la machine H2 montre que la connexion telnet est active avec l'adresse interne connue de l'extérieur.**

```
C:\>netstat -n

Active Connections

Proto  Local Address        Foreign Address      State
TCP    10.2.0.120:1111      10.100.0.111:23      ESTABLISHED
TCP    127.0.0.1:1027       127.0.0.1:1028       ESTABLISHED
TCP    127.0.0.1:1028       127.0.0.1:1027       ESTABLISHED
```

### Configuration de la NAT dynamique d'adresses IP internes

La traduction dynamique d'adresses IP internes est une des configurations dynamiques de la NAT, parmi d'autres.

La seule différence entre la traduction dynamique des adresses IP internes et la traduction statique est le processus de création de la table NAT. Dans la version statique, la table NAT est remplie manuellement par des instructions donnant les correspondances entre adresses appariées. Dans la version dynamique, c'est le routeur qui crée la table au fur et à mesure que les datagrammes satisfaisant à certains critères arrivent sur l'interface étiquetée « interne ». Les datagrammes arrivant sur l'interface « interne » du routeur depuis une adresse IP locale interne déjà existante sur la table NAT ne génèrent plus une nouvelle entrée de la table.

Les critères utilisés pour vérifier si le datagramme satisfait certaines règles et doit donner lieu à une entrée dans la table NAT sont exprimés sous forme de listes d'accès. Le datagramme doit satisfaire une règle de la liste d'accès et obtenir le résultat **permit** pour pouvoir créer une entrée dans la table NAT.

Pour créer des entrées dans la table NAT, le routeur utilise une plage d'adresses internes connues de l'extérieur (globales) mises à sa disposition lors de sa configuration. Pour chaque nouvelle entrée, le routeur consomme une adresse IP de la plage allouée dans un ordre croissant. Si cette plage ne contient plus d'adresse disponible, l'entrée dans la table n'est pas créée.

Pour configurer la NAT dynamique, il faut suivre ces étapes :

1. Définir une plage d'adresses IP internes connues de l'extérieur en utilisant la commande **ip nat pool** <*nom*> <*adresse IP de début*> <*adresse IP de fin*> **{netmask** <*masque de sous-réseau*>|**prefix-length** <*longueur de préfixe*>**}** dans le mode configuration globale. Le paramètre <*nom*> est utilisé pour identifier la plage d'adresses. Les paramètres <*adresse IP de début*> et <*adresse IP de fin*> définissent l'étendue de la plage que le routeur utilisera pour traduire les adresses IP internes locales. Les paramètres <*masque de sous-réseau*> et <*longueur de préfixe*> sont deux manières différentes de définir le masque de sous-réseau du réseau auquel appartiennent les adresses de la plage.

© copyright Éditions Eyrolles

La procédure précédente permet de définir une plage contiguë d'adresses NAT. Si vous désirez allouer des plages d'adresses discontinues, il faut enlever les paramètres <*adresse IP de début*> et <*adresse IP de fin*> de la commande précédente. Vous devez vous mettre en mode configuration de la plage NAT et définir plusieurs morceaux de plage d'adresses IP en suivant la syntaxe suivante : **address** <*adresse IP de début*> <*adresse IP de fin*>.

2. Définir une liste d'accès spécifiant les caractéristiques du trafic qui, arrivant sur l'interface « intérieure » l'autorise à générer une entrée dans la table NAT.

**REMARQUE** Vous pouvez utiliser des listes d'accès étendues. Cependant, la liste d'accès n'est prise en compte que si la table NAT n'a pas déjà une entrée coïncidant avec l'adresse IP source du datagramme postulant pour une traduction. Si l'adresse IP figure déjà (par exemple du fait d'une entrée statique dans la table), la liste d'accès n'est pas utilisée. Dans un tel cas, le datagramme subira la traduction d'adresse même si les autres caractéristiques ne devraient pas le permettre.

3. Établir une association entre les adresses internes locales susceptibles d'être traduites et la plage des adresses internes globales (connues de l'extérieur) avec la commande **ip nat inside source list {**<*numéro de LA*>|<*nom de LA*>**} pool** <*nom de la plage*>. Le paramètre <*numéro de LA*> est le nom ou le numéro de la liste d'accès définie à l'étape 2. Le paramètre <*nom de la plage*> est le nom de la plage d'adresses internes connues de l'extérieur définie à l'étape 1.
4. Activer la NAT sur les interfaces connectées aux segments d'adresses internes locales avec la commande **ip nat inside** dans le mode configuration de l'interface.
5. Activer la NAT sur les interfaces connectées aux segments d'adresses extérieures avec la commande **ip nat outside** dans le mode configuration de l'interface.

Modifions la procédure du cas précédent de traduction statique pour une seule machine H1 et optons pour une configuration dynamique de la NAT. Ici, les machines ayant une adresse dans 172.16.1.0/25 et initiant une connexion TCP vers un hôte du segment 2 sont aptes à voir leur adresse IP traduite en une adresse IP de la plage globale 10.100.0.50 à 10.100.0.100.

Le listing 7.24 montre la nouvelle configuration du routeur R3. Les autres routeurs gardent leur configuration précédente.

**Listing 7.24. Configuration du routeur R3.**

```
ip nat pool pool172 10.100.0.50 10.100.0.100 prefix-length 24
ip nat inside source list TCP172 pool pool172

interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
```

© copyright Éditions Eyrolles

```
ip nat outside
 frame-relay interface-dlci 301

router eigrp 10
 network 10.0.0.0
ip access-list extended TCP172
 permit tcp 172.16.1.0 0.0.0.127 10.2.0.0 0.0.0.255
```

Une commande utile pour surveiller la NAT est **show ip nat translations**. Cette commande affiche le contenu de la table NAT. Cependant, si le routeur est configuré pour de la traduction dynamique, il se peut que cette table soit encore vide, et la commande **show ip nat translations** ne génère aucune sortie écran.

Pour créer une entrée dans la table NAT, le routeur doit d'abord recevoir une trame satisfaisant aux règles de la liste d'accès. Si la liste d'accès est standard, n'importe quel trafic concordant avec celle-ci créera une entrée dans la table NAT. Si la liste d'accès est étendue, alors il faudra que les autres caractéristiques du trafic concordent avec la liste d'accès. Dans notre cas, le trafic devra non seulement émaner d'une machine de la plage 172.16.1.0/25 mais aussi être une trame TCP à destination d'une machine du segment 2.

Si nous essayons la commande **ping** vers une machine située du côté extérieur du routeur R3, celle-ci n'aboutira pas, pour la simple et bonne raison que **ping** utilise le protocole ICMP et pas TCP. Le listing 7.25 montre les résultats de la commande **ping** vers l'hôte *H2* et vers l'interface Ethernet du routeur R1.

**Listing 7.25. Résultats de la commande `ping` sur H2 et sur l'interface Ethernet du routeur R1 depuis la machine H2 avant que le routeur ne contienne une entrée pour la machine H1.**

```
C:\>ping 10.2.0.120

Pinging 10.2.0.120 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

C:\>ping 10.1.0.1

Pinging 10.1.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Une session telnet vers l'adresse IP de l'interface Ethernet du routeur R1 n'aboutira pas non plus.

En revanche, une session telnet depuis la machine H1 vers la machine H2 aboutira (cf. listing 7.26).

© copyright Éditions Eyrolles

**Listing 7.26. Telnet depuis H1 vers H2 aboutit.**

```
C:\>telnet 10.2.0.120

Welcome to the Telnet Service on THUNDER

Username:
```

Après cette commande, la table NAT du routeur R3 contient une entrée correspondant à l'adresse IP de H1 (172.16.1.111). Le listing 7.27 montre la sortie écran de la commande **show ip nat translations.**

**Listing 7.27. Sortie écran de la commande `show ip nat translations` sur le routeur R3.**

```
R3#show ip nat translations
Pro Inside global  Inside local  Outside local Outside global
--- 10.100.0.50    172.16.1.111  ---           ---
```

Une fois l'entrée créée, toutes les connexions depuis H1 vers le reste du réseau extérieur seront réussies (cf. listing 7.28 qui montre que la commande **ping** sur H1 ou R1 aboutit). Les connexions réussissent à présent car la table NAT ne contient que les paires d'adresses IP internes locale et interne globale. Elle ne contient pas les mêmes précisions en vigueur dans la liste d'accès utilisée lors de la création de l'entrée.

**Listing 7.28. Une fois l'entrée pour H1 ajoutée dans la table NAT, toutes les connexions depuis H1 vers le reste du réseau aboutissent.**

```
C:\>ping 10.2.0.120

Pinging 10.2.0.120 with 32 bytes of data:

Reply from 10.2.0.120: bytes=32 time=91ms TTL=125
Reply from 10.2.0.120: bytes=32 time=80ms TTL=125
Reply from 10.2.0.120: bytes=32 time=81ms TTL=125
Reply from 10.2.0.120: bytes=32 time=80ms TTL=125

C:\>ping 10.1.0.1

Pinging 10.1.0.1 with 32 bytes of data:

Reply from 10.1.0.1: bytes=32 time=60ms TTL=254
Reply from 10.1.0.1: bytes=32 time=50ms TTL=254
Reply from 10.1.0.1: bytes=32 time=50ms TTL=254
Reply from 10.1.0.1: bytes=32 time=50ms TTL=254
```

Une fois la session telnet vers la machine H2 établie sur H1, la sortie écran de la commande **netstat -n** exécutée sur H2 (cf. listing 7.29) montre que c'est l'adresse interne globale qui a remplacé l'adresse IP originale de H1.

**Listing 7.29. Sortie écran de la commande `netstat -n` exécutée sur H2.**

```
C:\WINDOWS\system32>netstat -n

Active Connections

Proto  Local Address          Foreign Address        State
TCP    10.2.0.120:23          10.100.0.50:1047       ESTABLISHED
TCP    127.0.0.1:1027         127.0.0.1:1028         ESTABLISHED
TCP    127.0.0.1:1028         127.0.0.1:1027         ESTABLISHED
```

© copyright Éditions Eyrolles

**ASTUCE** Vous pouvez utiliser la commande **clear ip nat translations *** pour effacer les entrées de la table NAT. À la place de l'astérisque, d'autres paramètres peuvent être entrés pour affiner les suppressions.

### NAT avec le protocole de routage OSPF

OSPF ajoute quelques subtilités à la configuration de NAT. Dans la section précédente, nous avons défini la plage des adresses internes globales en utilisant l'interface Loopback 0 du routeur R3. Nous comptions sur EIGRP, protocole de routage dynamique, pour diffuser une route vers cette plage d'adresses interne globale à l'ensemble du réseau. Mais si nous remplaçons EIGRP par OSPF dans les routeurs R1 à R3, nous remarquons que NAT ne fonctionne plus.

Les listings 7.30 à 7.32 montrent les configurations des routeurs avec OSPF à la place de EIGRP. Aucun autre changement n'a été effectué.

**Listing 7.30. Configuration du routeur R1.**

```
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial1
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial1.2 point-to-point
 ip address 10.255.0.5 255.255.255.252
 frame-relay interface-dlci 102

interface Serial1.3 point-to-point
 ip address 10.255.0.9 255.255.255.252
 frame-relay interface-dlci 103

router ospf 10
 network 10.0.0.0 0.255.255.255 area 0

ip classless
```

**Listing 7.31. Configuration du routeur R2.**

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.6 255.255.255.252
 frame-relay interface-dlci 201

router ospf 10
```

© copyright Éditions Eyrolles

```
 network 10.0.0.0 0.255.255.255 area 0

ip classless
```

**Listing 7.32. Configuration du routeur R3.**

```
ip nat pool pool172 10.100.0.50 10.100.0.100 prefix-length 24
ip nat inside source list TCP172 pool pool172

interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301

router ospf 10
 network 10.0.0.0 0.255.255.255 area 0

ip classless

ip access-list extended TCP172
 permit tcp 172.16.1.0 0.0.0.127 10.2.0.0 0.0.0.255
```

Une fois les changements effectués, la commande **telnet 10.2.0.120** entrée sur la machine H1 ne fonctionne plus. Cette commande n'opère pas car NAT, sur lequel nous nous appuyons ne fonctionne plus.

NAT ne fonctionne plus à cause du traitement réservé aux interfaces de rebouclage par OSPF. Si nous regardons les tables de routage du routeur R2 (cf. listing 7.33), nous remarquons qu'elles contiennent une route vers le réseau 10.100.0.1/32 et ceci en dépit du fait que nous avons configuré l'interface Loopback 0 du routeur R3 avec un masque de sous-réseau /24.

**Listing 7.33. La table de routage du routeur R2.**

```
R2#show ip route
...
 10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C  10.0.0.2/32 is directly connected, Loopback0
C  10.2.0.0/24 is directly connected, Ethernet0
O  10.1.0.0/24 [110/74] via 10.255.0.5, 00:12:40, Serial0.1
O  10.0.0.1/32 [110/65] via 10.255.0.5, 00:12:40, Serial0.1
O  10.100.0.1/32 [110/129] via 10.255.0.5, 00:12:40,Serial0.1
C  10.255.0.4/30 is directly connected, Serial0.1
O  10.255.0.8/30 [110/128] via 10.255.0.5, 00:12:40,Serial0.1
```

© copyright Éditions Eyrolles

Si nous entrons la commande **show ip ospf interface Loopback0** sur le routeur R3, nous remarquons que la dernière ligne de la sortie écran indique que les interfaces de rebouclage sont traitées (et donc publiées) comme des machines rattachées.

**Listing 7.34. Sortie écran de la commande `show ip ospf interface Loopback 0` sur le routeur R3.**

```
R3#show ip ospf interface Loopback0
Loopback0 is up, line protocol is up
  Internet Address 10.100.0.1/24, Area 0
  Process ID 10, Router ID 10.100.0.1, Network Type LOOPBACK,
 Cost: 1
  Loopback interface is treated as a stub Host
```

En outre, du fait que l'interface de bouclage Loopback0 a une adresse IP plus élevée que n'importe quelle autre adresse IP externe, elle est utilisée comme identificateur OSPF du routeur, ce qui est incompatible avec les identificateurs OSPF des autres routeurs.

La solution directe à ce problème est de définir une route statique pour l'espace d'adresse interne global pointant sur l'interface Null0 et de donner à l'interface Loopback 0 un identifiant OSPF adéquat. La route statique doit alors être rediffusée par le processus de OSPF.

Une autre solution est d'affecter à l'interface Loopback0 une adresse IP en tant que telle (dans notre exemple, 10.0.0.3/32) et ensuite de définir une interface Tunnel0 dont les adresses source et destination sont égales à cette nouvelle adresse IP. Comme les interfaces de rebouclage, les interfaces tunnel sont logiques. Cependant, ces dernières ne sont pas traitées par OSPF comme des machines rattachées. Ainsi, si l'adresse IP globale est définie en utilisant une interface tunnel, elle sera publiée par OSPF correctement. Cette solution évite aussi de rediffuser les routes statiques dans le processus OSPF du routeur R3.

Le listing 7.35 montre la nouvelle configuration du routeur R3.

**Listing 7.35. Configuration du routeur R3.**

```
ip nat pool pool172 10.100.0.50 10.100.0.100 prefix-length 24
ip nat inside source list TCP172 pool pool172

interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Tunnel0
 ip address 10.100.0.1 255.255.255.0
 tunnel source 10.0.0.3
 tunnel destination 10.0.0.3

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
```

© copyright Éditions Eyrolles

```
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301

router ospf 10
 network 10.0.0.0 0.255.255.255 area 0

ip classless

ip access-list extended TCP172
 permit tcp 172.16.1.0 0.0.0.127 10.2.0.0 0.0.0.255
```

La table de routage du routeur R2 (cf. listing 7.36) contient désormais une route vers l'espace d'adresse interne global connu de l'extérieur.

**Listing 7.36. Table de routage du routeur R2 une fois l'interface Tunnel 0 créée sur le routeur R3.**

```
R2#show ip route
...
 10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
C  10.0.0.2/32 is directly connected, Loopback0
C  10.2.0.0/24 is directly connected, Ethernet0
O  10.0.0.3/32 [110/129] via 10.255.0.5, 00:00:11, Serial0.1
O  10.1.0.0/24 [110/74] via 10.255.0.5, 00:00:11, Serial0.1
O  10.0.0.1/32 [110/65] via 10.255.0.5, 00:00:11, Serial0.1
O  10.100.0.0/24 [110/11239]via 10.255.0.5,00:00:11,Serial0.1
C  10.255.0.4/30 is directly connected, Serial0.1
O  10.255.0.8/30 [110/128] via 10.255.0.5, 00:00:11,Serial0.1
```

NAT fonctionne ainsi correctement.

### Utilisation du paramètre **type match-host** de la commande **ip nat pool**

Comme mentionné dans la section précédente, le routeur utilise pour l'adresse interne globale une adresse d'une plage réservée à cet usage prise dans un ordre croissant. Quelquefois, vous décidez de préserver la partie hôte de l'adresse IP (*hostid*) lors d'une traduction d'adresse réseau. Pour obtenir cette fonctionnalité, le paramètre optionnel **type match-host** peut être inséré à la fin de la commande **ip nat pool**.

Si ce paramètre est utilisé, le routeur calcule l'adresse IP interne globale avec cette formule :

$$IGA = IGNA + ( ILA - ILNA )$$

où IGA désigne l'adresse IP interne globale obtenue, IGNA la première adresse IP de la plage, ILA l'adresse IP interne locale (l'adresse qui doit être traduite), et ILNA la partie réseau (*network ID*) de l'adresse IP interne locale. Le masque de sous-réseau de la plage est défini soit par le paramètre **prefix-length** <*longueur*> soit par le paramètre ***netmask*** <*masque de sous-réseau*>. Tous les paramètres doivent être traités comme des adresses de 32 bits.

**AVERTISSEMENT** L'adresse IP interne globale ainsi calculée doit rester dans la plage définie par la commande **ip nat pool**. Sinon, la traduction NAT ne s'effectue pas.

© copyright Éditions Eyrolles

Remplaçons la plage d'adresses IP internes globales que nous avons utilisée dans la section précédente par une nouvelle accompagnée du paramètre **type match-host**. La nouvelle configuration du routeur R3 figure dans le listing 7.37.

**Listing 7.37. Configuration du routeur R3.**

```
ip nat pool pool172 10.100.0.96 10.100.0.127 prefix-length 24 type match-host

ip nat inside source list 1 pool pool172

interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301

router eigrp 10
 network 10.0.0.0

access-list 1 permit 172.16.1.0 0.0.0.127
```

La nouvelle table NAT figure dans le listing 7.38.

**Listing 7.38. La table NAT du routeur R3.**

```
R3#show ip nat translations
Pro Inside global  Inside local  Outside local Outside global
--- 10.100.0.111   172.16.1.111  ---           ---
```

**REMARQUE** Les machines situées à l'extérieur ne peuvent pas atteindre les machines situées à l'intérieur *via* leur adresse interne globale, tant que la table NAT ne contient pas une entrée correspondante. Dans le cas d'une NAT dynamique, l'entrée NAT est créée seulement si un trafic concordant avec les règles d'une liste d'accès a été routé vers l'extérieur par le routeur NAT. Dans le cas d'une NAT statique, l'entrée est toujours présente dans la table NAT.

### Configuration de la NAT avec un espace d'adresses IP internes globales surchargé

Dans certains cas, il n'est pas possible de définir une plage d'adresses IP internes globales suffisamment étendue. C'est par exemple le cas d'un réseau de classe privée qui doit être connecté à l'Internet. Si le réseau est important, et utilise ainsi une large plage d'adresses IP

© copyright Éditions Eyrolles

privées comme 10.0.0.0/8, il est impossible d'avoir une plage de taille identique du côté externe (côté relié à l'Internet). D'ailleurs, si cela avait été possible, on aurait pris cette plage d'adresses IP publique pour schéma d'adressage interne !

Une solution NAT est possible pour de telles situations, elle est référencée dans les RFC comme NAPT (*Network Address Port Translation*) et dans la documentation de Cisco comme PAT (*Port Address Translation*). NAPT permet à un grand nombre de machines connectées sur le réseau interne d'accéder au réseau extérieur en utilisant soit une seule adresse IP interne globale, soit un nombre restreint. Ceci est rendu possible en traduisant les identifiants du niveau transport – c'est à dire les ports TCP/UDP et les identifiants des requêtes ICMP – créés par les machines du réseau interne en des identifiants du niveau transport associé à la seule (ou les seules) adresse IP interne globale. Le routeur réalisant les fonctions de NAPT doit garder trace de ces identifiants transport ainsi que leur adresse IP interne locale correspondante.

Pour configurer NAPT, vous pouvez utiliser la procédure décrite dans la section « Configuration d'une traduction dynamique d'adresses IP internes » de ce chapitre à l'étape 3 mais en ajoutant le paramètre **overload** à la commande **ip nat inside source**.

Vous pouvez aussi utiliser la syntaxe suivante de la commande : **ip nat inside source list {**<*numéro de LA*>|<*nom de LA*>**} interface** <*interface*>. Dans ce cas, le routeur remplace les adresses IP internes locales par l'unique adresse IP configurée dans l'interface correspondante.

Le listing 7.39 montre la configuration du routeur R3, qui maintenant remplace les adresses IP internes locales par l'adresse IP de l'interface Serial0.1. Les configurations des routeurs R1 et R2 sont données dans les listings 7.16 et 7.17.

**Listing 7.39. Configuration du routeur R3.**

```
ip nat inside source list 1 interface Serial0.1 overload

interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301

router eigrp 10
 network 10.0.0.0

access-list 1 permit 172.16.1.0 0.0.0.127
```

© copyright Éditions Eyrolles

La table NAT du routeur R3 est dans le listing 7.40. Remarquez que tous les champs de la table NAT sont remplis.

**Listing 7.40. La table NAT du routeur R3.**

```
R3#show ip nat translations
Pro Inside global     Inside local       Outside local  Outside global
tcp 10.255.0.10:1054  172.16.1.111:1054  10.2.0.120:23  10.2.0.120:23
```

### Configuration de la NAT en cas de recouvrement des espaces d'adresses

Si deux réseaux utilisant les mêmes plages d'adresses IP sont fusionnés et que l'un de ces deux réseaux devient le réseau rattaché, il est possible d'utiliser NAT pour traduire les adresses recouvertes.

Cette solution NAT est basée sur l'utilisation du DNS (*Domain Name System*) par les machines du réseau rattaché pour résoudre les adresses IP des machines du réseau externe. Le routeur NAT intercepte les réponses du DNS. Si l'adresse IP retournée recouvre une adresse IP du réseau interne rattaché, le routeur la traduit en une adresse IP non ambiguë routable sur le réseau rattaché.

Suivez ces étapes pour configurer NAT en traducteur d'espaces d'adresses recouverts :

1. Utiliser la commande **ip nat pool**, et définir une plage d'adresses externes locales en lesquelles vont être traduites les adresses IP externes mentionnées dans les réponses DNS. Ces adresses doivent être routables à l'intérieur du réseau rattaché (le réseau interne local). La syntaxe de la commande **ip nat pool** est la même que la précédente.
2. Définir une liste d'accès spécifiant quel trafic arrivant sur l'interface étiquetée « extérieure » est autorisé à créer une entrée dans la table NAT.
3. Établir une correspondance entre les adresses externes globales et la plage d'adresses externes locales en utilisant la commande **ip nat outside source list {***<numéro de LA>*|*<nom de LA>***} pool** *<nom de la plage>*
4. Activer la NAT sur les interfaces connectées aux segments d'adresses internes locales avec la commande **ip nat inside** dans le mode configuration de l'interface.
5. Activer la NAT sur les interfaces connectées aux segments d'adresses externes globales avec la commande **ip nat outside** dans le mode configuration de l'interface.

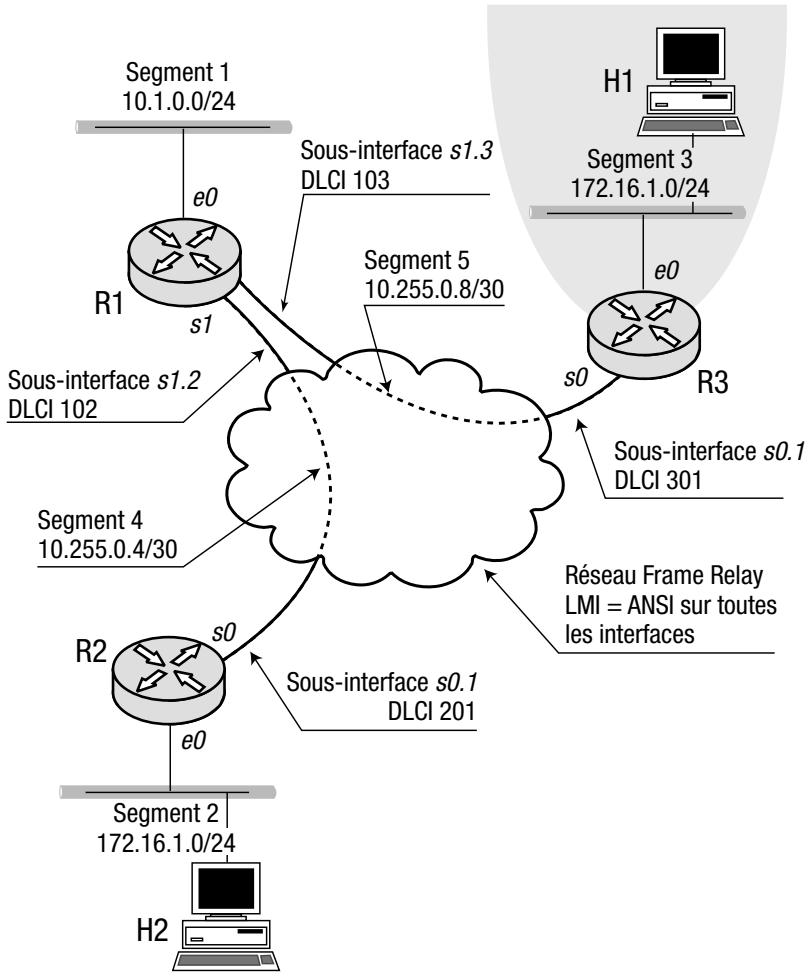
**REMARQUE** Les adresses IP constituant la plage d'adresses externes locales au routeur NAT doivent aussi être routables comme les adresses IP externes globales qu'elles remplacent. Du fait que les routeurs extérieurs n'ont aucune connaissance des adresses externes locales, ils ne peuvent publier ces adresses. Ainsi, est-il possible que le routage statique soit le seul moyen d'avertir du routage des adresses externes locales.

**ASTUCE** La traduction des adresses extérieures n'a de sens qu'associée avec une traduction des adresses internes.

La figure 7.3 donne un exemple de fusion de deux réseaux, dont l'un (la zone grisée) utilise le même préfixe réseau que l'autre. Par chance, le premier réseau n'est pas important et se voit rattaché au second réseau. NAT permet ainsi de résoudre le problème du recouvrement d'adresses.

© copyright Éditions Eyrolles

**Figure 7.3**
*La plage d'adresses IP utilisée dans la zone grisée recouvre une plage déjà existante sur le réseau (segment 2).*




Les configurations de tous les routeurs sont sur les listings 7.41 à 7.43.

**Listing 7.41. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial1.2 point-to-point
 ip address 10.255.0.5 255.255.255.252
 frame-relay interface-dlci 102

interface Serial1.3 point-to-point
 ip address 10.255.0.9 255.255.255.252
 frame-relay interface-dlci 103

router eigrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 7.42. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.6 255.255.255.252
 frame-relay interface-dlci 201

router eigrp 10
 network 10.0.0.0
 network 172.16.0.0
```

**Listing 7.43. Configuration du routeur R3.**

```
interface Loopback0
 ip address 10.100.0.1 255.255.255.0

interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
 ip nat inside

interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 301

router eigrp 10
 network 10.0.0.0

ip nat pool ext172 10.200.0.30 10.200.0.80 prefix-length 24
ip nat pool int172 10.100.0.50 10.100.0.100 prefix-length 24
ip nat inside source list 1 pool int172
ip nat outside source list 1 pool ext172

ip route 10.200.0.0 255.255.255.0 10.255.0.9

access-list 1 permit 172.16.1.0 0.0.0.255
```

**REMARQUE** Bien que deux plages différentes soient utilisées pour la traduction des adresses extérieures globales et des adresses intérieures locales, la liste d'accès utilisée pour la concordance avec les datagrammes IP ouvrant une nouvelle entrée dans la table NAT est la même dans les deux cas. Ceci est dû au fait que l'espace d'adresse IP traduit est le même (recouvert entre le segment 3 et le segment 2) et donc une seule liste d'accès suffit. Mais avoir une seule liste d'accès n'est pas une obligation pour la configuration de la NAT.

© copyright Éditions Eyrolles

La table NAT du routeur R3 figure sur le listing 7.44.

**Listing 7.44. La table NAT du routeur R3.**

```
R3#show ip nat translations
Pro Inside global  Inside local  Outside local Outside global
--- 10.100.0.50    172.16.1.111  ---           ---
--- ---            ---           10.200.0.31   172.16.1.1
--- ---            ---           10.200.0.30   172.16.1.120
--- 10.100.0.50    172.16.1.111  10.200.0.30   172.16.1.120
```

### La configuration de la NAT pour l'équilibrage de charge TCP

Comme expliqué dans l'introduction de ce chapitre, il est possible d'utiliser NAT pour l'équilibrage de charge. Cette fonction de la NAT est appelée LSNAT (*Load Sharing Using IP Network Address Translation*). La RFC 2391 décrit cette application.

Si plusieurs serveurs exécutent le même service sur le réseau, par exemple un serveur web et un serveur FTP, vous pouvez alors utiliser LSNAT pour permettre l'accès à tous ces serveurs via une unique adresse IP appelée *adresse IP virtuelle du serveur*. Un routeur configuré LSNAT redistribue les connexions entrant sur cette adresse IP virtuelle vers les adresses IP réelles des serveurs. Une fois la connexion établie, tous les paquets suivants seront acheminés sur le serveur sélectionné lors de la création de l'entrée dans la table NAT.

Le premier paquet de ces connexions est utilisé pour établir une entrée correspondante dans la table NAT du routeur. Les segments auxquels sont connectés les serveurs sont les réseaux internes. Les segments auxquels sont connectées les machines accédant au service *via* l'adresse IP virtuelle sont les réseaux externes.

Pour configurer LSNAT, suivez ces étapes :

1. Définir la plage des adresses IP réelles qui seront la traduction de l'adresse IP virtuelle du serveur. Chacune des adresses définies dans cette plage doit être l'adresse d'un serveur existant. Si les adresses IP réelles des serveurs ne sont pas contiguës, vous devez définir une plage non contiguë.

   Définir la plage en utilisant la même syntaxe de commande que celle décrite dans la section « Configuration d'une traduction dynamique d'adresses IP internes » de ce chapitre avec les mots clés **type rotary** ajoutés à la fin.

2. Créer une liste d'accès concordant avec l'adresse IP virtuelle du serveur donnant ainsi le résultat **permit**.

3. Établir une association entre l'adresse IP virtuelle du serveur et la plage en utilisant la commande **ip nat inside destination list {***<numéro de LA>*|*<nom de LA>***} pool** *<nom de la plage>*.

4. Activer la NAT sur les interfaces connectées aux segments sur lesquels résident les serveurs réels avec la commande **ip nat inside** dans le mode configuration de l'interface.

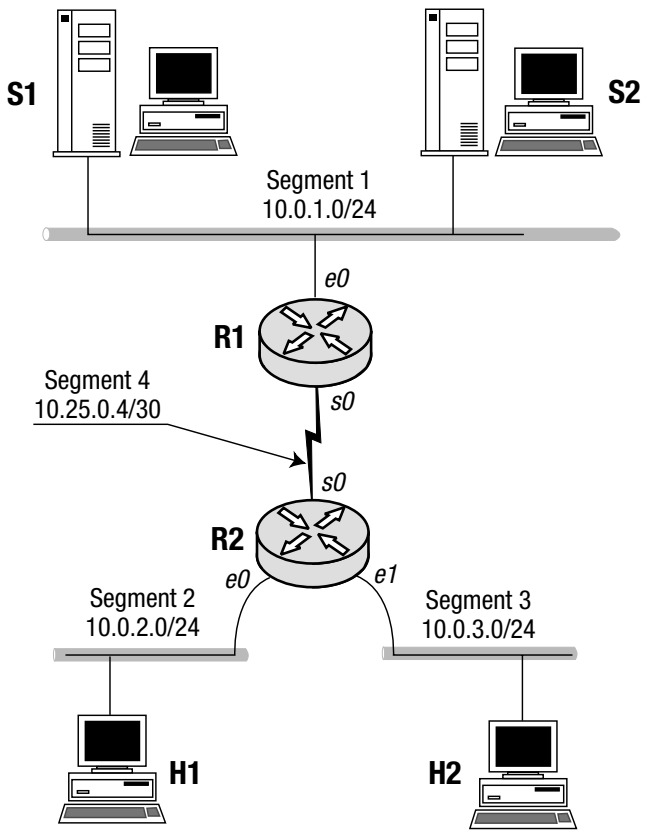
5. Activer la NAT sur les interfaces connectées aux segments d'adresses externes globales avec la commande **ip nat outside** dans le mode configuration de l'interface.

**REMARQUE** L'adresse IP virtuelle du serveur doit être accessible par le routeur LSNAT.

© copyright Éditions Eyrolles

La figure 7.4 montre un exemple de réseau pour le déploiement de LSNAT. Les serveurs S1 et S2 fournissent le service telnet, qui est supposé accessible par les machines H1 et H2 sur une unique adresse IP.

**Figure 7.4**
*Le routeur R1 équilibre le trafic des sessions TCP entre les serveurs S1 et S2*




Les listings 7.45 et 7.46 montrent les configurations des routeurs R1 et R2. Notez que nous utilisons une plage d'adresses non contiguës à cause des adresses IP des serveurs elles-mêmes non contiguës (10.0.1.11 et 10.0.1.222)

**Listing 7.45. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.0.1.1 255.255.255.0
 ip nat inside

interface Serial0
 ip address 10.255.0.5 255.255.255.252
 ip nat outside

router eigrp 10
 network 10.0.0.0

ip nat pool Servers prefix-length 24 type rotary
 address 10.0.1.111 10.0.1.111
```

© copyright Éditions Eyrolles

```
 address 10.0.1.222 10.0.1.222

ip nat inside destination list 1 pool Servers

access-list 1 permit 10.0.1.100
```

**Listing 7.46. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Ethernet1
 ip address 10.0.3.1 255.255.255.0

interface Serial0
 ip address 10.255.0.6 255.255.255.252

router eigrp 10
 network 10.0.0.0
```

Les listings 7.47 et 7.48 montrent les résultats d'une session telnet depuis les machines H1 et H2 respectivement, vers l'adresse IP virtuelle du serveur. La sortie écran indique clairement, bien que l'adresse IP destination soit la même, que les sessions sont engagées avec deux serveurs différents (HUGEWAVE et LITTLEWAVE).

**Listing 7.47. Session telnet depuis H1 vers l'adresse IP virtuelle du serveur.**

```
C:\>telnet 10.0.1.100

Welcome to the Telnet Service on HUGEWAVE

Username:
```

**Listing 7.48. Session telnet depuis H2 vers l'adresse IP virtuelle du serveur.**

```
C:\>telnet 10.0.1.100

Welcome to the Telnet Service on LITTLEWAVE

Username:
```

Les listings 7.49 et 7.50 donnent les sorties écran de la commande **netstat -n** exécutée sur les machines H1 et H2 respectivement. Les sorties écran donnent les adresses IP réelles qui ont remplacé l'adresse IP virtuelle du serveur.

**Listing 7.49. La sortie écran de la commande `netstat -n` montre que la session telnet est établie avec S1.**

```
C:\WINDOWS\system32>netstat -n

Active Connections

Proto  Local Address          Foreign Address        State
TCP    10.0.1.111:23          10.0.2.120:11004       ESTABLISHED
TCP    127.0.0.1:1025         127.0.0.1:1026         ESTABLISHED
TCP    127.0.0.1:1026         127.0.0.1:1025         ESTABLISHED
```

© copyright Éditions Eyrolles

**Listing 7.50. La sortie écran de la commande `netstat -n` montre que la session telnet est établie avec S2.**

```
C:\WINDOWS\system32>netstat -n

Active Connections

Proto  Local Address          Foreign Address        State
TCP    10.0.1.222:23          10.0.3.120:11005       ESTABLISHED
TCP    127.0.0.1:1025         127.0.0.1:1026         ESTABLISHED
TCP    127.0.0.1:1026         127.0.0.1:1025         ESTABLISHED
```

Enfin, le listing 7.51 donne la table NAT du routeur R1.

**Listing 7.51. La table NAT du routeur R1.**

```
R1#show ip nat translations
Pro Inside global  Inside local       Outside local      Outside global
tcp 10.0.1.100:23  10.0.1.222:23      10.0.3.120:11005   10.0.3.120:11005
tcp 10.0.1.100:23  10.0.1.111:23      10.0.2.120:11004   10.0.2.120:11004
```

## La configuration du protocole HSRP (*Hot Standby Router Protocol)*

Dans cette section, nous présentons deux configurations du protocole de tolérance aux pannes du routeur par défaut (HSRP) : la configuration de base de HSRP, et HSRP avec équilibrage de charge.

### Configuration de base de HSRP

Deux routeurs Cisco, ou plus, peuvent avoir leurs interfaces connectées à un segment et configurées pour supporter le protocole HSRP. Pour la configuration de base de HSRP, suivez ces étapes dans le mode configuration de l'interface.

1. Entrez la commande **standby** <*numéro du groupe*> **ip** <*adresse IP*>. Le paramètre <*numéro du groupe*> est le numéro du groupe d'attente (*standby group*), qui peut aller de 0 à 3 sur les interfaces Token Ring et de 0 à 255 sur les autres types d'interfaces réseau. Le paramètre <*adresse IP*> est l'adresse utilisée par les machines du réseau pour leur passerelle par défaut. (C'est donc l'adresse de la route par défaut que possèdent les machines du réseau. Elle est distincte de l'adresse IP réelle affectée aux interfaces, cf. listings plus bas).
2. (Cette étape est optionnelle) Entrez la commande **standby** <*numéro du groupe*> **priority** <*priorité*>. Le paramètre <*priorité*> définit la priorité accordée au routeur. Affectez la plus haute priorité au routeur que vous désirez voir élu comme routeur actif dans les conditions normales d'opérations. Si vous n'entrez pas cette commande, le routeur utilisera la priorité par défaut qui est 100.
3. (Cette étape est optionnelle.) Entrez la commande **standby** <*numéro de groupe*> **preempt** si vous désirez qu'un routeur en attente puisse devenir routeur actif dans le cas où le routeur actif courant diminuerait sa priorité par rapport à celle du routeur en attente. Entrer cette commande sur le routeur actif lui permet de recouvrer son état si toutefois son statut de routeur actif a été perdu du fait d'une indisponibilité passagère ou à cause d'une priorité en baisse.

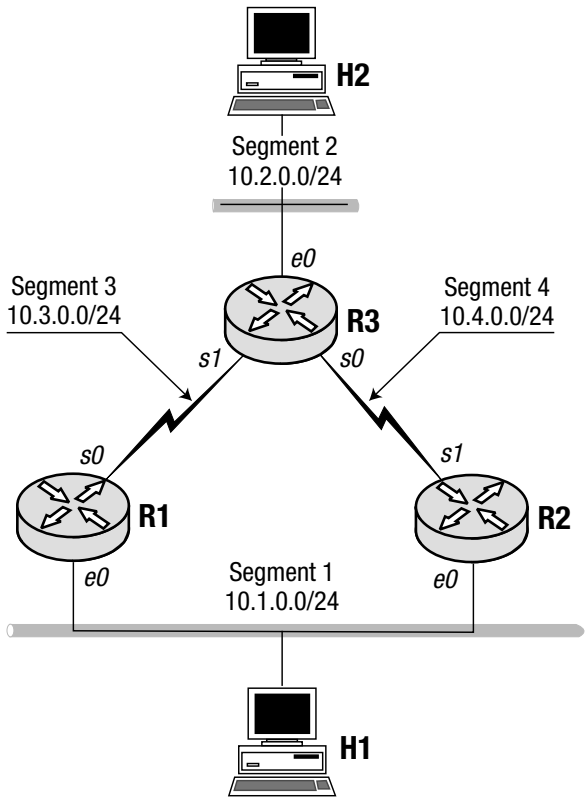
© copyright Éditions Eyrolles

4. (Cette étape est optionnelle.) Entrez la commande **standby** <*numéro de groupe*> **track** <*interface*> [<*décrément de priorité*>] si vous désirez que le routeur décrémente sa propre priorité d'attente en cas d'interruption de l'interface <*interface*>. Le paramètre <*décrément de priorité*>, s'il est utilisé, spécifie de combien cette priorité doit diminuer. La valeur par défaut de ce paramètre est 10.

Examinons la configuration réseau de la figure 7.5. Les routeurs R1 et R2 peuvent être configurés avec HSRP pour fournir un routeur par défaut redondant aux machines connectées au segment.

**Figure 7.5**

*Les routeurs R1 et R2 sont configurés avec le protocole HSRP afin de se relayer l'un, l'autre en cas de défaillance du prochain ou de l'une de leurs interfaces.*




Les listings 7.52 à 7.54 montrent la configuration de chacun de ces routeurs.

**Listing 7.52. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.2 255.255.255.0
 no ip redirects
 standby 10 priority 100
 standby 10 preempt
 standby 10 ip 10.1.0.1
 standby 10 track Serial0 50

interface Serial0
 ip address 10.3.0.1 255.255.255.0

router eigrp 1
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 7.53. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.1.0.3 255.255.255.0
 no ip redirects
 standby 10 priority 80
 standby 10 preempt
 standby 10 ip 10.1.0.1

interface Serial1
 ip address 10.4.0.1 255.255.255.0

router eigrp 1
 network 10.0.0.0
```

**Listing 7.54. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial0
 ip address 10.4.0.2 255.255.255.0

interface Serial1
 ip address 10.3.0.2 255.255.255.0

router eigrp 1
 network 10.0.0.0
```

La commande utilisée pour vérifier l'état de HSRP sur les routeurs est **show standby**. (Cette commande a des paramètres optionnels. Si vous désirez en savoir plus, référez-vous à la documentation de Cisco.)

Les listings 7.55 et 7.56 montrent les sorties écrans de la commande **show standby** exécutée sur les routeurs R1 et R2. La sortie écran de ces commandes est assez explicite.

**Listing 7.55. Sortie écran de la commande `show standby` exécutée sur le routeur R1.**

```
R1#show standby
Ethernet0 - Group 10
  Local state is Active, priority 100, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.056
  Hot standby IP address is 10.1.0.1 configured
  Active router is local
  Standby router is 10.1.0.3 expired
  Standby virtual mac address is 0000.0c07.ac0a
  Tracking interface states for 1 interface, 1 up:
    Up   Serial0 Priority decrement: 50
```

© copyright Éditions Eyrolles

**Listing 7.56. Sortie écran de la commande `show standby` exécutée sur le routeur R2.**

```
R2#show standby
Ethernet0 - Group 10
  Local state is Standby, priority 80, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.546
  Hot standby IP address is 10.1.0.1 configured
  Active router is 10.1.0.2 expires in 00:00:09
  Standby router is local
  Standby virtual mac address is 0000.0c07.ac0a
```

**REMARQUE** L'adresse MAC virtuelle, apparaissant à la dernière ligne du listing 7.56 correspond au modèle d'adresse MAC virtuelle pour des médias différents de Token Ring. Cf. Introduction de ce chapitre.

Comme mentionné dans l'introduction du chapitre, l'adresse MAC virtuelle utilisée par le routeur HSRP peut être identique ou différente de l'adresse MAC physique de l'interface selon le type de celle-ci. L'interface Ethernet0 du routeur R1 a un type qui force le routeur à utiliser son adresse MAC virtuelle pour son interface. De ce fait, et du fait du modèle utilisé dans le cas d'interfaces non Token Ring, le routeur devra changer l'adresse MAC visible en l'adresse MAC virtuelle et non pas garder l'adresse MAC physique gravée en dur.

Le listing 7.57 montre les deux premières lignes de la sortie écran de la commande **show interfaces Ethernet 0** exécutée sur le routeur R1. Le premier champ en italique sur le listing 7.57 montre l'adresse MAC courante de l'interface Ethernet0. Le second champ en italique montre l'adresse MAC physique originale gravée en dur (*burned-in address*).

**Listing 7.57. Sortie écran de la commande `show interfaces Ethernet 0` exécutée sur le routeur R1.**

```
R1#show interfaces Ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c07.ac0a (bia 00e0.b064.5063)
...
```

Néanmoins, le routeur R2 utilise encore son adresse MAC originale pour son adresse MAC visible (cf. listing 7.58).

**Listing 7.58. Sortie écran de la commande `show interfaces Ethernet 0` exécutée sur le routeur R2.**

```
R2#show interfaces Ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 00e0.b064.30a9 (bia 00e0.b064.30a9)
...
```

Si le routeur R1 était indisponible et que le routeur R2 devenait le routeur actif, il devrait alors changer son adresse MAC visible en la valeur de l'adresse MAC virtuelle. Ceci n'est faisable que si les caractéristiques matérielles de l'interface permettent d'avoir une adresse MAC différente de celle gravée en dur.

© copyright Éditions Eyrolles

Comme vous l'avez vu dans l'introduction du chapitre il ne peut y avoir sur un segment qu'un seul routeur actif et un seul routeur en attente (*standby router*). Ainsi, si on connecte un troisième routeur HSRP sur le segment 1 dont la configuration est sur le listing 7.59, son état ne sera ni *actif* ni *en attente*.

**Listing 7.59. Configuration de l'interface Ethernet 0 du routeur R4.**

```
interface Ethernet0
 ip address 10.1.0.4 255.255.255.0
 no ip redirects
 standby 1 priority 50
 standby 1 preempt
 standby 1 ip 10.1.0.1
```

Comme le montre le listing 7.60, le troisième routeur est en état d'écoute (*Listen*).

**Listing 7.60. Sortie écran de la commande `show standby` exécutée sur le routeur R4.**

```
R4#show standby
Ethernet0 - Group 10
  Local state is Listen, priority 50, may preempt
  Hellotime 3 holdtime 10
  Hot standby IP address is 10.1.0.1 configured
  Active router is 10.1.0.2 expires in 00:00:08
  Standby router is 10.1.0.3 expires in 00:00:07
```

Voyons maintenant ce qu'il se passe si le routeur R1 du segment 1 devient momentanément indisponible. Le premier message (cf. listing 7.61) indique que le routeur R2 est maintenant actif.

**Listing 7.61. Message sur la console du routeur R2 lorsqu'il change de l'état d'attente à l'état actif.**

```
R2#
02:13:27: %STANDBY-6-STATECHANGE: Standby: 10: Ethernet0 state Standby -> Active
```

Si nous vérifions la sortie écran de la commande **show interfaces Ethernet 0**, elle indique que l'adresse MAC de l'interface est devenue l'adresse MAC virtuelle. Autrement dit, pour devenir routeur actif, le routeur R2 a dû changer son adresse MAC visible et prendre l'adresse MAC virtuelle au lieu de garder son adresse MAC gravée en dur.

**Listing 7.62. Sortie écran de la commande `show interfaces Ethernet 0` illustrant le changement d'adresse MAC du routeur en l'adresse MAC virtuelle.**

```
R2#show interfaces Ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c07.ac0a (bia 00e0.b064.30a9)
...
```

Bien sûr, ce changement ne s'opère que si l'interface physique le permet.

Examinons maintenant le temps mis par le routeur R2 pour basculer son adresse MAC et son adresse IP. Pour ce faire, nous avons utilisé un script **perl** (dont le source figure en Annexe D) qui exécute la commande **ping** vers une adresse IP toutes les secondes et qui affiche les résultats.

© copyright Éditions Eyrolles

Le listing 7.63 montre la sortie écran du script **perl** lançant la commande **ping 10.2.0.120** exécutée sur la machine H1.

**Listing 7.63. Sortie écran de la commande `ping 10.2.0.120` lancée par le script `perl` sur la machine H1, qui affiche aussi l'horodatage.**

```
C:\>perl tping.pl 10.2.0.120
[41:25] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:26] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:27] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:28] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:30] Request timed out.
[41:32] Request timed out.
[41:34] Request timed out.
[41:36] Request timed out.
[41:37] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:38] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:39] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:40] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[41:41] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
```

Comme le montre ce listing, le routeur par défaut était indisponible pendant seulement 4 secondes.

**REMARQUE** La valeur de temporisation de convergence, dépend du temps à savoir celui mis par les routeurs à consolider leurs routes après un changement. Même si le routeur en attente a déjà basculé son adresse MAC et son adresse IP, le protocole de routage peut encore être dans la phase de consolidation des routes.

Si le routeur R1 redevient actif, il se peut qu'il n'y ait aucun dépassement de délai (cf. listing 7.64). Ceci provient du fait qu'il s'agit ici d'une transition entre deux routeurs avec les bonnes adresses.

**Listing 7.64. Le routeur R1 redevient disponible, alors que le script `perl` lance la commande `ping 10.2.0.120`. Comme on le voit, il n'y pas de dépassement de délai lorsque R1 redevient actif.**

```
C:\>perl tping.pl 10.2.0.120
[46:03] Reply from 10.2.0.120: bytes=32 time=20ms TTL=126
[46:04] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:05] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:06] Reply from 10.2.0.120: bytes=32 time=11ms TTL=126
[46:07] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:08] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:09] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:10] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:11] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:12] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:13] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:14] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:15] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
[46:16] Reply from 10.2.0.120: bytes=32 time=10ms TTL=126
```

© copyright Éditions Eyrolles

### Utilisation de MHSRP pour équilibrage de charge

Utiliser deux routeurs uniquement pour que l'un remplace l'autre en cas de défaillance est une mauvaise gestion des ressources de routage. Une meilleure idée consiste à équilibrer la charge du trafic requis par les machines locales entre ces deux routeurs.

Ce partage de ressources peut être obtenu en modifiant légèrement les configurations de base de HSRP. Cette fonctionnalité est appelée aussi MHSRP, *multigroup HSRP*. L'idée sous-jacente à MHSRP est simple : si deux routes sont configurées dans deux groupes d'attente (*standby group*) sur la même interface, le premier routeur peut être actif pour le premier groupe et le deuxième routeur pour le deuxième groupe. Ces deux groupes ne sont utilisés que pour prendre le relais l'un de l'autre en cas de défaillance. Ainsi, nous devrions diviser aussi les machines locales en deux groupes. Le premier groupe utilise comme routeur par défaut l'adresse IP du premier routeur, et le second groupe, celle du deuxième routeur. Le premier groupe de machine envoie donc le trafic sortant vers le premier routeur, et le deuxième groupe de machines, vers le deuxième routeur.

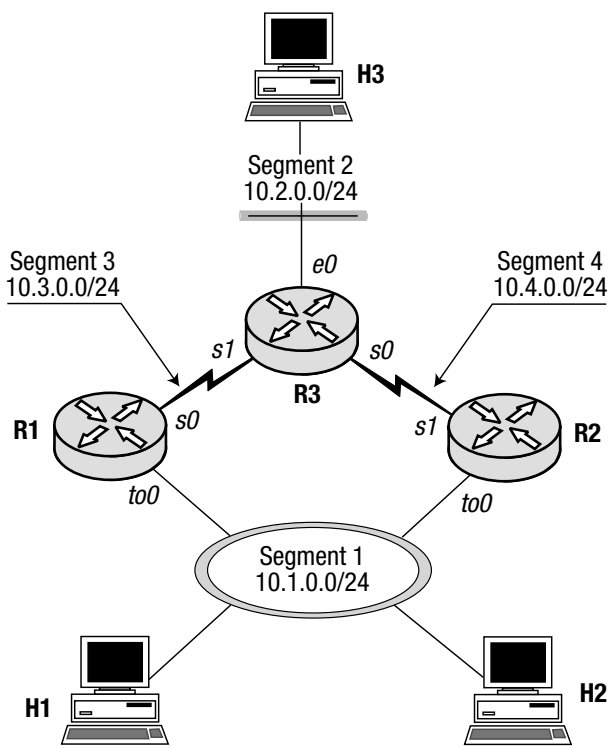
Les étapes nécessaires pour la configuration de MHSRP sur un routeur sont pratiquement les mêmes que dans le cas d'une configuration de base. La seule différence est que ce processus doit être fait pour chacun des groupes d'attente configurés sur la même interface.

Voyons comment les routeurs de la figure 7.6 doivent être configurés en MHSRP pour fournir deux routeurs redondants aux machines H1 et H2. La machine H1 utilise la première adresse IP redondante, et la machine H2, la deuxième.

Les listings 7.65 et 7.66 montrent les configurations des routeurs R1 et R2. La configuration du routeur R3 reste identique à celle de la section précédente.

**Figure 7.6**

*Les routeurs R1 et R2 en étant configurés pour deux groupes d'attente équilibrent le trafic entre eux. Le routeur R1 est le routeur par défaut du premier groupe et R2, celui du deuxième groupe.*




© copyright Éditions Eyrolles

**Listing 7.65. Configuration du routeur R1.**

```
interface Serial0
 ip address 10.3.0.1 255.255.255.0

interface TokenRing0
 ip address 10.1.0.3 255.255.255.0
 no ip redirects
 ring-speed 16
 standby 1 priority 100
 standby 1 preempt
 standby 1 ip 10.1.0.1
 standby 1 track Serial0 50
 standby 2 priority 80
 standby 2 preempt
 standby 2 ip 10.1.0.2

router eigrp 1
 network 10.0.0.0
```

**Listing 7.66. Configuration du routeur R2.**

```
interface Serial1
 ip address 10.4.0.1 255.255.255.0

interface TokenRing0
 ip address 10.1.0.4 255.255.255.0
 no ip redirects
 ring-speed 16standby 1 priority 80
 standby 1 preempt
 standby 1 ip 10.1.0.1
 standby 2 priority 100
 standby 2 preempt
 standby 2 ip 10.1.0.2
 standby 2 track Serial1 50

router eigrp 1
 network 10.0.0.0
```

Les listings 7.67 et 7.68 montrent les sorties écran de la commande **show standby** exécutée sur les routeurs R1 et R2. Remarquez que la commande **show standby** affiche maintenant deux groupes d'attente pour chacun des routeurs. Le routeur R1 est le routeur actif pour le groupe 1 et le routeur en attente pour le groupe 2, alors que le routeur R2 est le routeur actif pour le groupe 2 et le routeur en attente pour le groupe 1.

**Listing 7.67. Sortie écran de la commande `show standby` sur R1.**

```
R1#show standby
TokenRing0 - Group 1
  Local state is Active, priority 100, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:00.000
  Hot standby IP address is 10.1.0.1 configured
  Active router is local
  Standby router is 10.1.0.4 expired
  Standby virtual mac address is c000.0002.0000
```

© copyright Éditions Eyrolles

```
  Tracking interface states for 1 interface, 1 up:
    Up   Serial0 Priority decrement: 50
TokenRing0 - Group 2
  Local state is Standby, priority 80, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:02.486
  Hot standby IP address is 10.1.0.2 configured
  Active router is 10.1.0.4 expires in 00:00:08
  Standby router is local
  Standby virtual mac address is c000.0004.0000
```

**Listing 7.68. Sortie écran de la commande `show standby` sur R2.**

```
R2#show standby
TokenRing0 - Group 1
  Local state is Standby, priority 80, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.268
  Hot standby IP address is 10.1.0.1 configured
  Active router is 10.1.0.3 expires in 00:00:09
  Standby router is local
  Standby virtual mac address is c000.0002.0000
TokenRing0 - Group 2
  Local state is Active, priority 100, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:00.496
  Hot standby IP address is 10.1.0.2 configured
  Active router is local
  Standby router is 10.1.0.3 expired
  Standby virtual mac address is c000.0004.0000
  Tracking interface states for 1 interface, 1 up:
    Up   Serial1 Priority decrement: 50
```

Les listings 7.69 et 7.68 montrent la sortie écran de la commande **show interfaces TokenRing 0** exécutée sur les routeurs R1 et R2. Remarquez que l'interface physique tolère l'utilisation par le routeur d'une adresse virtuelle MAC différente de l'adresse MAC de l'interface.

**Listing 7.69. Sortie écran de la commande `show interfaces TokenRing 0` exécutée sur le routeur R1.**

```
R1#show interfaces TokenRing 0
TokenRing0 is up, line protocol is up
  Hardware is TMS380, address is 0007.0d26.0a46
(bia 0007.0d26.0a46)
...
```

**Listing 7.70. Sortie écran de la commande `show interfaces TokenRing 0` exécutée sur le routeur R2.**

```
R2#show interfaces TokenRing 0
TokenRing0 is up, line protocol is up
  Hardware is TMS380, address is 0007.0d26.0c15
(bia 0007.0d26.0c15)
...
```

© copyright Éditions Eyrolles

**REMARQUE** Comme mentionné plus haut, certains matériels ne peuvent avoir des adresses MAC virtuelles différentes de celle de leur interface. La raison en est que le matériel ne peut assigner plusieurs adresses MAC unicast à la même interface. En conséquence, MHSRP ne peut être implémenté sur un tel matériel, car différents groupes d'attente nécessitent différentes adresses MAC virtuelles.

## Configuration du routage à la demande (*Dial-On Demand Routing)*

Les deux sections suivantes décrivent les deux configurations les plus classiques du DDR (Dial-On Demand routing) : le routage instantané ou à la volée (*snapshot routing*) et le routage de secours (*dial backup).*

### Configuration du routage instantané

Le routage instantané (*snapshot routing*) est une solution économique qui permet de ne pas composer systématiquement pour accéder au réseau distant. Le routage instantané permet l'utilisation d'un protocole de routage à vecteur de distance pour établir un routage dynamique sur des lignes commutées. Les lignes en sortie ne sont pas systématiquement utilisées pour des mises à jour d'informations de routage.

Le routage instantané définit deux périodes : la période active et la période de sommeil (*quiet period*). Le protocole de routage peut activer la ligne en sortie pendant la période active. Pendant les périodes de sommeil, le protocole de routage dynamique doit compter sur l'information qu'il a récoltée durant la précédente période active. A contrario, le trafic sortant est autorisé à composer et à activer les lignes en sortie indépendamment du type de période. Il va de soi que si la ligne est activée par du trafic sortant, le protocole de routage peut en profiter pour échanger les mises à jour d'informations de routage.

En général, la période active est plutôt courte –de l'ordre de 5 à 10 minutes. La période de sommeil est plus longue–par exemple, 12 heures.

Le routage instantané suppose deux types de routeurs : les serveurs et les clients. Les routeurs clients initient les connexions distantes sur les lignes pendant les périodes d'activité ; les serveurs ne composent pas, ils sont en attente de connexion. En général, les routeurs clients sont les routeurs qui connectent les réseaux périphériques au réseau central *via* des lignes commutées. Les routeurs serveurs sont localisés sur les sites centraux et attendent les connexions émanant des réseaux périphériques ; ce sont des serveurs d'accès.

Le routage instantané ne fonctionne qu'avec les protocoles de routage RIP versions 1 et 2 ainsi qu'avec IGRP. Ainsi, pour déployer un réseau sans classe ou *classless* (agrégation de classe par préfixe commun) avec du routage instantané, vous devez opter pour RIP version 2.

La configuration du routage instantané repose sur ces étapes :

1. Configurer le routeur client avec la commande **snapshot client** <*période d'activité*> <*période de repos*> dans le menu de configuration de l'interface d'accès commuté (par exemple une interface RNIS BRI). Les paramètres <*période d'activité*> et <*période de sommeil*> sont des valeurs numériques spécifiant en minutes les périodes d'activité et les périodes de sommeil. Le paramètre <*période d'activité*> peut varier de 5 à 100, tandis que le paramètre <*période de repos*> peut varier de 8 à 100000.
2. Permettre au routeur client d'appeler le routeur serveur d'accès avec la commande **dialer map snapshot** <*numéro de séquence*> <*chaîne d'appel*>. Le paramètre <*numéro de séquence*> est une valeur numérique permettant l'identification de l'appelant. Elle peut

© copyright Éditions Eyrolles

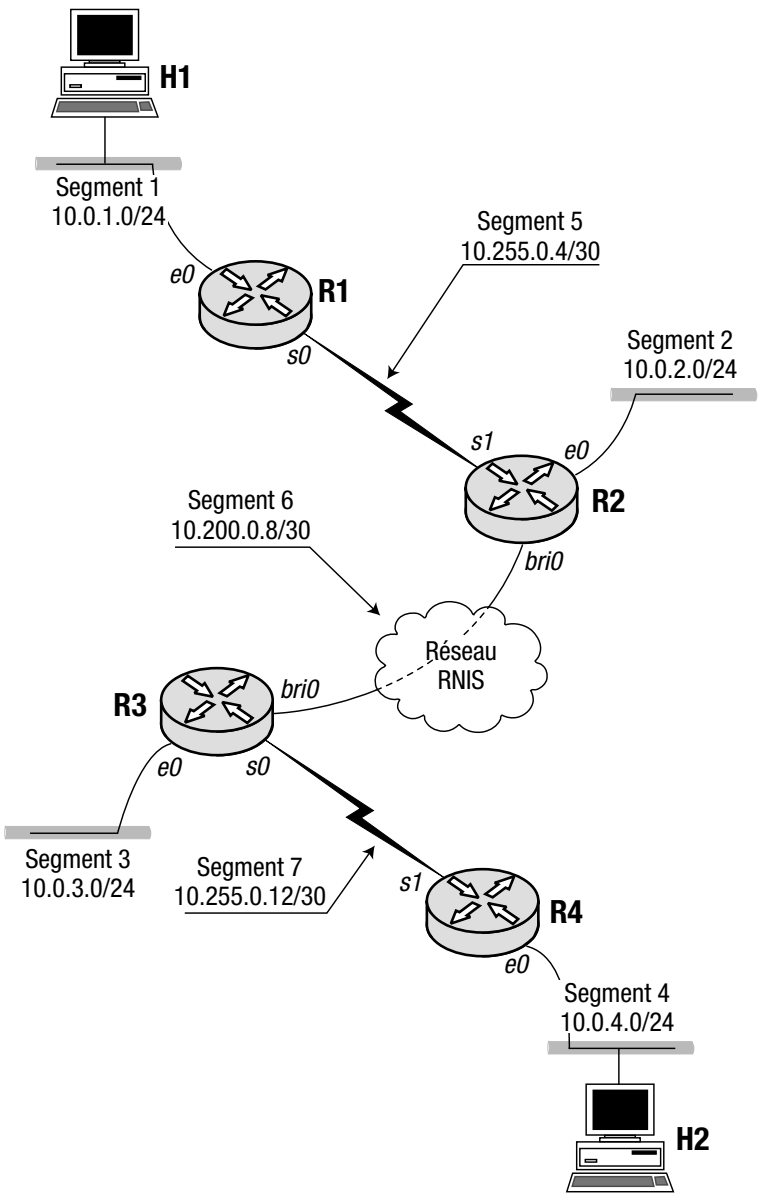
varier entre 1 et 254, inclus. Le paramètre <*chaîne d'appel*> est la chaîne utilisée lors de la composition du numéro pour se connecter au serveur d'accès.

3. Configurer le serveur d'accès avec la commande **snapshot server** <*période d'activité*>. Le paramètre <*période d'activité*> est une valeur numérique indiquant la période d'activité, elle varie entre 5 et 100.

La figure 7.7 montre un exemple de réseau avec du routage instantané. Les routeurs utilisent RIP version 2 comme protocole de routage.

**Figure 7.7**

*Les routeurs R2 et R3 sont configurés pour du routage instantané pour interconnecter l'ensemble du réseau.*




© copyright Éditions Eyrolles

Les listings 7.71 à 7.74 montrent les configurations des quatre routeurs. Seuls les routeurs R2 et R3 ont des configurations spécifiques pour le routage à la demande, elles sont mises en italique.

**Listing 7.71. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.0.1.1 255.255.255.0interface Serial0
ip address 10.255.0.5 255.255.255.252

router rip
 version 2
 network 10.0.0.0
```

**Listing 7.72. Configuration du routeur R2.**

```
username R3 password 0 cisco
isdn switch-type basic-ni

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Serial1
ip address 10.255.0.6 255.255.255.252

interface BRI0
 ip address 10.200.8.1 255.255.255.0
 encapsulation ppp
 dialer map snapshot 1 384020
 dialer map ip 10.200.8.2 name R3 broadcast 384020
 dialer-group 1
 isdn spid1 3840000001
 isdn spid2 3840000002
 snapshot client 5 20
 ppp authentication chap

router rip
 version 2
 network 10.0.0.0

ip classless

dialer-list 1 protocol ip permit
```

**Listing 7.73. Configuration du routeur R3.**

```
username R2 password 0 cisco

isdn switch-type basic-ni1

interface Ethernet0
 ip address 10.0.3.1 255.255.255.0

interface Serial0
 ip address 10.255.0.14 255.255.255.252

interface BRI0
 ip address 10.200.8.2 255.255.255.0
 encapsulation ppp
```

© copyright Éditions Eyrolles

```
 isdn spid1 3840200001
 isdn spid2 3840200002
 dialer map ip 10.200.8.1 name R2 broadcast 384000
 dialer-group 1
 snapshot server 5
 ppp authentication chap

router rip
 version 2
 network 10.0.0.0

dialer-list 1 protocol ip permit
```

**Listing 7.74. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 10.0.4.1 255.255.255.0

interface Serial1
 ip address 10.255.0.13 255.255.255.252

router rip
 version 2
 network 10.0.0.0
```

Une fois le routage dynamique stabilisé, les tables de routage de tous les routeurs devraient être les mêmes, indépendamment de l'état de la ligne commutée. Le listing 7.75 montre le contenu de la table de routage du routeur R4.

**Listing 7.75. La table de routage du routeur R4.**

```
R4#show ip route
...
   10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
R    10.0.2.0/24 [120/2] via 10.255.0.14, 00:00:04, Serial1
R    10.0.3.0/24 [120/1] via 10.255.0.14, 00:00:04, Serial1
R    10.0.1.0/24 [120/3] via 10.255.0.14, 00:00:04, Serial1
C    10.0.4.0/24 is directly connected, Ethernet0
R    10.200.8.0/24 [120/1] via 10.255.0.14, 00:00:04, Serial1
R    10.200.8.1/32 [120/1] via 10.255.0.14, 00:00:04, Serial1
R    10.255.0.4/30 [120/2] via 10.255.0.14, 00:00:04, Serial1
C    10.255.0.12/30 is directly connected, Serial1
```

Notez que le routeur R4 connaît les routes pour tous les segments du réseau.

Une commande utile pour connaître l'état du routage à la demande est **show snapshot**, qui affiche des résultats légèrement différents entre les clients et les serveurs.

Les listings 7.76 et 7.77 montrent les sorties écran de cette commande exécutée sur les routeurs R2 et R3, respectivement. La sortie écran est explicite. (Il apparaît une légère coquille d'affichage dans toutes les versions d'IOS, à savoir un retour à la ligne manquant sur la première ligne entre les mots **up** et **Snapshot**.)

**Listing 7.76. Sortie écran de la commande `show snapshot` exécutée sur le routeur R2.**

```
R2#show snapshot
BRI0 is up, line protocol is upSnapshot client line state
```

© copyright Éditions Eyrolles

```
down
 Length of active period:          5 minutes
 Length of quiet period:           20 minutes
 Length of retry period:           8 minutes
  Current state: active, remaining/exchange time: 3/2 minutes
  Updates received this cycle: ip
```

**Listing 7.77. Sortie écran de la commande `show snapshot` exécutée sur le routeur R3.**

```
R3#show snapshot
BRI0 is up, line protocol is upSnapshot server line state up
  Length of active period:          5 minutes
   For ip address: 10.200.8.1
    Current state: active, remaining time: 1 minute
```

### Configuration du lien de secours *(dial backup)*

Les liens de secours sont une fonctionnalité très puissante et très utile des routeurs Cisco. Deux routeurs connectés par une ligne spécialisée (LS), comme une T1, peuvent utiliser un lien de secours au cas où la LS tomberait.

Pour configurer le lien de secours, suivez ces étapes :

1. Dans le mode configuration de l'interface qui doit être secourue, entrez la commande **backup interface** <*interface commutée*>. Le paramètre <*interface commutée*> spécifie le nom de l'interface qui doit être utilisée en cas de secours.
2. En utilisant la commande **backup delay** <*délai de mise en route*> <*délai d'extinction*> dans le mode configuration de l'interface à secourir, vous définissez le délai pour que l'interface de secours se mette en route en cas de défaillance de l'interface principale et le délai pour que l'interface de secours se remette en inactivité une fois l'interface principale rétablie. Les deux paramètres sont des valeurs numériques mesurées en secondes dans la plage de 0 à 4294967294 ou la valeur conventionnelle **never**. Si cette dernière valeur est utilisée comme paramètre <*délai de mise en route*>, l'interface de secours ne se met jamais en marche. Si cette valeur est utilisée comme paramètre <*délai d'extinction*>, l'interface de secours reste active jusqu'à une extinction manuelle.

La configuration de secours suppose une configuration correcte des paramètres de composition. Mais cette configuration est différente d'une configuration d'accès distant (*dial-on demand routing*). Utiliser les paramètres classiques de configuration d'un accès distant sur une interface de secours peut être fatal à la bonne marche de celle-ci ou aboutir à des fonctionnalités superflues pour une ligne de secours. Vous trouverez ci-dessous une liste d'astuces qui peuvent vous aider à configurer *proprement* votre interface de secours :

- La configuration de secours ne se fait que sur l'un des deux routeurs. Le routeur en vis-à-vis doit avoir une configuration classique de DDR (*dial-on demand routing*)
- Le routeur en vis-à-vis ne doit pas initier la connexion si du trafic destiné à l'autre réseau lui arrive. Il faut tout de même avoir effectué la commande **dialer-group** <*numéro de groupe*> dans la configuration de numérotation. Il est même judicieux de définir explicitement qu'aucun trafic sortant destiné à l'autre réseau n'entraînera une composition du numéro en utilisant la commande **dialer-list** <*numéro de groupe*> **protocol ip deny**.
- Définir un délai de dépassement en cas de non activité sur le réseau (*idle timeout*) très long sur l'interface commutée du routeur en vis-à-vis. Du fait que le trafic sortant n'initie pas la connexion et que ce délai de dépassement est de 120 secondes par défaut, le routeur se déconnectera toutes les 3 minutes pendant la période de secours en cas de trafic plat.

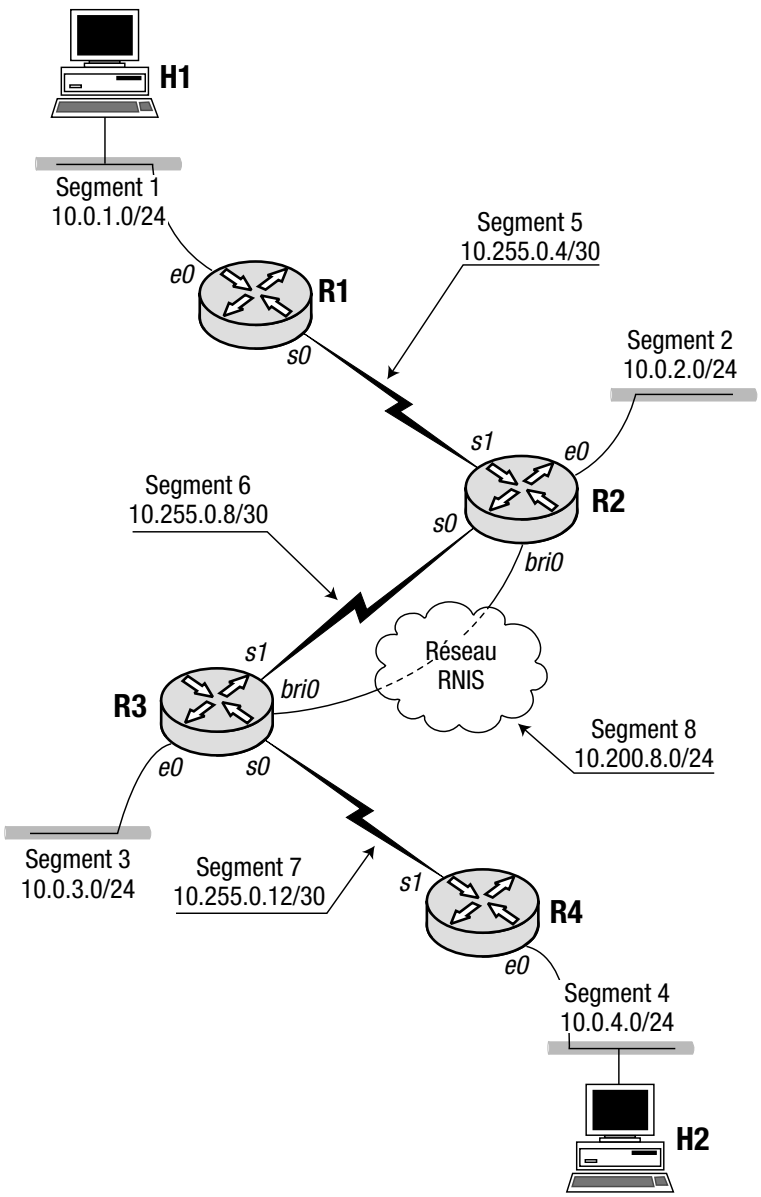
© copyright Éditions Eyrolles

- Vérifier le bon fonctionnement de l'accès distant avant de configurer la liaison de secours. Il serait plus difficile de diagnostiquer un problème de mauvaise configuration, ou d'authentification alors que la ligne de secours est déjà installée. Faire attention aux commandes d'authentification, comme **username** <*nom du routeur*> **password** <*mot de passe*>. En fonction du mode d'authentification PAP ou CHAP, cette commande peut nécessiter que les mots de passe soient identiques sur les deux routeurs.

Voyons comment la ligne de secours peut être utilisée dans le réseau de la figure 7.8. La différence avec le réseau de la section précédente est que les routeurs R2 et R3 sont ici connectés avec le segment 6. Le segment 6 est le lien principal tandis que le segment 8 est le lien de secours.

**Figure 7.8**

*Les routeurs R2 et R3 peuvent utiliser le segment 8, une connexion RNIS BRI, comme secours du segment 6.*




© copyright Éditions Eyrolles

Les listings 7.78 à 7.81 montrent les configurations des quatre routeurs. Les lignes de la configuration, spécifiques à la mise en place du lien de secours figurent en italique.

**Listing 7.78. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.0.1.1 255.255.255.0

interface Serial0
 ip address 10.255.0.5 255.255.255.252

router eigrp 10
 network 10.0.0.0
```

**Listing 7.79. Configuration du routeur R2.**

```
username R3 password 0 cisco

isdn switch-type basic-ni

interface Ethernet0
 ip address 10.0.2.1 255.255.255.0

interface Serial0
 ip address 10.255.0.9 255.255.255.252
 backup delay 3 20
 backup interface BRI0

interface Serial1
 ip address 10.255.0.6 255.255.255.252

interface BRI0
 ip address 10.200.8.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.200.8.2 name R3 broadcast 384020
 dialer-group 1
 isdn spid1 3840000001
 isdn spid2 3840000002
 ppp authentication chap

router eigrp 10
 network 10.0.0.0

dialer-list 1 protocol ip permit
```

**Listing 7.80. Configuration du routeur R3.**

```
username R2 password 0 cisco

isdn switch-type basic-ni1

interface Ethernet0
 ip address 10.0.3.1 255.255.255.0

interface Serial0
 ip address 10.255.0.14 255.255.255.252
```

© copyright Éditions Eyrolles

```
interface Serial1
 ip address 10.255.0.10 255.255.255.252

interface BRI0
 ip address 10.200.8.2 255.255.255.0
 encapsulation ppp
 isdn spid1 3840200001
 isdn spid2 3840200002
 dialer idle-timeout 2147483
 dialer map ip 10.200.8.1 name R2 broadcast 384000
 dialer-group 1
 ppp authentication chap

router eigrp 10
 network 10.0.0.0

dialer-list 1 protocol ip deny
```

**Listing 7.81. Configuration du routeur R4.**

```
interface Ethernet0
 ip address 10.0.4.1 255.255.255.0

interface Serial1
 ip address 10.255.0.13 255.255.255.252

router eigrp 10
 network 10.0.0.0
```

Pour vérifier que le lien de secours est en fonction, utilisons la commande **debug backup**. La sortie écran de cette commande exécutée sur le routeur R2 figure sur le listing 7.82. Notez que cette commande n'a de sens que sur le routeur R2 car c'est le seul qui est configuré pour le secours. Le routeur est aussi configuré avec la commande **service timestamps debug uptime**. Cette commande fait indiquer au routeur un horodatage depuis son allumage précédant les messages de débogage. Ceci nous permet de voir en combien de temps, l'interface de secours est mise en place lors d'un dysfonctionnement de l'interface principale.

**Listing 7.82. Sortie écran de la commande `debug backup` sur le routeur R2 alors que le lien principal entre R2 et R3 va être coupé.**

```
02:23:18: %LINK-3-UPDOWN: Interface Serial0, changed state to
          down
02:23:18: BACKUP(Serial0): event = primary went down
02:23:18: BACKUP(Serial0): changed state to "waiting to
          backup"
02:23:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface
          Serial0, changed state to down
02:23:21: BACKUP(Serial0): event = timer expired
02:23:21: %LINK-3-UPDOWN: Interface BRI0:1, changed state to
          down
02:23:21: %LINK-3-UPDOWN: Interface BRI0:2, changed state to
          down
```

© copyright Éditions Eyrolles

```
02:23:21: BACKUP(Serial0): secondary interface (BRI0) made
          active
02:23:21: BACKUP(Serial0): changed state to "backup mode"
02:23:21: %LINK-3-UPDOWN: Interface BRI0, changed state to up
02:23:21: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 65
          changed to up
02:23:23: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0, TEI 64
          changed to up
02:23:24: %LINK-3-UPDOWN: Interface BRI0:1, changed state to
          up
02:23:25: %LINEPROTO-5-UPDOWN: Line protocol on Interface
          BRI0:1, changed state to up
02:23:30: %ISDN-6-CONNECT: Interface BRI0:1 is now connected
          to 384020 R3
```

Le routeur a noté que l'interface principale est tombée à 02:23:18. Le routeur a initié la connexion de secours à 02:23:21. La différence de 3 secondes correspond exactement au délai indiqué dans la configuration *via* le paramètre <*délai de mise en route*>.

Si nous essayons d'atteindre la machine H1 depuis la machine H2 avec l'outil **ping** pendant toute la durée de l'opération, nous serons désagréablement surpris de voir que le temps de mise en route du secours est supérieur à 3 secondes. Le listing 7.83 indique les résultats de cette commande.

**Listing 7.83. Résultats de la commande `ping` sur H1 depuis H2.**

```
C:\>perl tping.pl 10.0.1.120
[35:01] Reply from 10.0.1.120: bytes=32 time=30ms TTL=124
[35:02] Reply from 10.0.1.120: bytes=32 time=30ms TTL=124
[35:04] Request timed out.
[35:06] Request timed out.
[35:07] Reply from 10.0.4.1: Destination host unreachable.
[35:08] Reply from 10.0.4.1: Destination host unreachable.
[35:09] Reply from 10.0.4.1: Destination host unreachable.
[35:10] Reply from 10.0.4.1: Destination host unreachable.
[35:11] Reply from 10.0.4.1: Destination host unreachable.
[35:12] Reply from 10.0.4.1: Destination host unreachable.
[35:13] Reply from 10.0.4.1: Destination host unreachable.
[35:14] Reply from 10.0.4.1: Destination host unreachable.
[35:15] Reply from 10.0.4.1: Destination host unreachable.
[35:16] Reply from 10.0.4.1: Destination host unreachable.
[35:18] Request timed out.
[35:19] Reply from 10.0.1.120: bytes=32 time=40ms TTL=124
[35:20] Reply from 10.0.1.120: bytes=32 time=30ms TTL=124
```

La période pendant laquelle le segment affecté par la panne n'est plus accessible est en fait de l'ordre de 16 secondes. Cette durée d'inaccessibilité est plus longue que la période spécifiée dans **backup delay** car le protocole de routage dynamique utilisé doit consolider ses informations lors de la mise en route de la connexion de secours, celle-ci n'étant pas active auparavant.

Nous avons utilisé EIGRP, qui a un temps de convergence court. D'autres protocoles de routage comme RIP ou IGRP pourraient prendre plus de temps lors de la mise en route de la

© copyright Éditions Eyrolles

liaison de secours. Il est donc important d'utiliser des protocoles de routage à temps de convergence rapide, ou d'introduire explicitement un routage statique pour rendre accessible le lien de secours rapidement.

Par chance, le processus inverse – c'est à dire, la remise en route de la liaison principale et la déconnexion du lien de secours – n'engendre pas de période d'inaccessibilité car les deux liens peuvent opérer en parallèle pendant un certain laps de temps. Ce temps dépend du temps de convergence du protocole de routage utilisé. Dans notre cas, le protocole est EIGRP, ainsi les 20 secondes configurées devraient suffire pour que EIGRP reconverge, et que la liaison de secours cesse sans engendrer de perte de connectivité entre les machines des deux réseaux.

© copyright Éditions Eyrolles

# 8

# Routage IP multicast

### Solutions de configuration présentées dans ce chapitre



Dans le chapitre 1, nous avons mentionné qu'un groupe d'adresses IP, appelé adresse multidestinataire (multicast) peut être utilisé pour atteindre un groupe de machines et que l'ensemble des adresses multicast est en fait la classe D des adresses IP. Les 4 bits de poids fort du premier octet de l'adresse IP valent 1110 (en binaire) dans une adresse IP de classe D. C'est une plage d'adresses qui va de 224.0.0.0 à 239.255.255.255 en notation décimale pointée.

Comme nous le savons, certaines de ces adresses sont utilisées pour d'autres besoins que les applications normales de multicast. Par exemple, certains protocoles de routage, comme EIGRP, OSPF et RIP version 2, utilisent des adresses IP multicast prédéfinies pour communiquer avec des routeurs voisins. Cette utilisation de l'adressage multicast est parfaitement justifiée, car seuls les routeurs intéressés recevront les informations de mise à jour des tables de routage, et ceci en une seule fois.

Le document décrivant les adresses IP multicast réservées (*preassigned multicast addresses*) est la RFC 1700, *Assigned Numbers*. Certaines de ces adresses figurent dans le tableau 8.1.

**REMARQUE** Certaines RFC du tableau 8.1 ont pu être mises à jour entre-temps. Il est prudent de le vérifier.

© copyright Éditions Eyrolles

**Tableau 8.1. Quelques adresses IP multicast réservées.**

| Adresse IP | Description | RFC associée |
|---|---|---|
| 224.0.0.0 | Adresse de base (réservée) | RFC 1112 |
| 224.0.0.1 | Tous les systèmes de ce réseau | RFC 1112 |
| 224.0.0.2 | Tous les routeurs de ce sous-réseau | N/A |
| 224.0.0.4 | Routeurs DVMRP | RFC 1075 |
| 224.0.0.5 | Tous les routeurs OSPF | RFC 2328 |
| 224.0.0.6 | Certains routeurs OSPF | RFC 2328 |
| 224.0.0.9 | Routeurs RIP2 | N/A |
| 224.0.0.10 | Routeurs IGRP | N/A |
| 224.0.1.1 | Protocol de Temps NTP (*network time protocol)* | RFC 1119 |

## Bases du routage multicast

Le routage IP multicast est complètement différent du routage IP unicast. Le mécanisme de routage IP unicast est basé sur la partie réseau de l'adresse IP. Dans le cas du multicast, ce mécanisme n'est plus applicable, car le groupe de machines concernées par l'adresse multicast peut être dispersé sur de multiples réseaux. Ainsi, l'adresse IP multicast ne peut être utilisée pour identifier ni des segments de réseaux ni des machines particulières. Aussi, d'autres règles doivent-elles être appliquées pour accomplir le routage des datagrammes destinés à une adresse multicast.

Avant d'étudier plus en détail les techniques de routage du trafic IP multicast, familiarisons-nous avec certains de ses aspects spécifiques.

De par leur nature, les adresses IP multicast sont souvent désignées sous le terme de groupes plutôt que d'adresses multicast.

Dans le routage IP unicast, une machine source envoyant du trafic à une machine destinataire, dans la plupart des cas, reçoit en retour un certain trafic. Dans la plupart des cas de routage IP multicast, cette règle ne s'applique pas. Certaines machines ne font qu'envoyer des données au groupe et ne reçoivent nullement de trafic en retour. D'autres machines, à l'inverse, ne font que recevoir du trafic multicast et ne sont pas en mesure d'émettre quoi que ce soit au groupe. Le premier type de machine est appelé *envoyeur* (*sender)* ou *source*, tandis que le deuxième est appelé *receveur* (*receiver*) ou *membre* du groupe. Le routage multicast utilise cette typologie d'*envoyeur* et de *receveur*, pour optimiser l'acheminement du trafic multicast entre les différents réseaux interconnectés.

### *Correspondance entre adresses IP multicast et adresses physiques* (MAC)

Un autre aspect important du routage multicast est la correspondance entre les adresses IP multicast et les adresses physiques (MAC). Dans le cas de l'IP multicast, le protocole de résolution d'adresses ARP n'est plus utilisé pour établir cette correspondance, même dans un environnement LAN. Les méthodes de résolutions de type ARP ne sont pas applicables dans un environnement multicast pour deux raisons. Tout d'abord, si une requête ARP est émise pour obtenir l'adresse MAC correspondant à l'adresse IP multicast, alors une multitude de réponses émaneraient des membres du groupe, ce qui engorgerait le réseau (outre qu'il y

© copyright Éditions Eyrolles

aurait difficulté à garder en mémoire et à traiter plusieurs adresses MAC se prétendant membres du groupe de destination). Ensuite, et c'est le pire, le protocole ARP ne permet pas d'utiliser les fonctionnalités intrinsèques des protocoles du niveau liaison qui permettent de définir des adresses de groupe au niveau MAC (c'est le cas pour les adresses multicast Ethernet).

Comme indiqué au chapitre 2, les adresses au niveau MAC permettent d'adresser un nombre important de stations en utilisant le bit *group-bit*. Si ce bit est mis à 1 dans l'adresse de destination MAC, alors cela indique que la trame est destinée à plusieurs destinataires. Ainsi, l'adresse IP multicast est en correspondance avec l'adresse MAC dont le bit *group-bit* est à 1 de la manière suivante : les 23 bits de poids faible de l'adresse IP multicast sont rangés dans les 23 bits de poids faible de l'adresse MAC 01-00-5E-00-00-00. Les couches physiques et FDDI permettent cette correspondance, avec une nuance dans le cas de FDDI : l'ordre des bits dans un octet doit être inversé.

Les adresses MAC des réseaux en anneaux à jetons (Token Ring) devraient permettre cette fonctionnalité, car elles disposent aussi d'un bit *group-bit* à l'instar de Ethernet et de FDDI. Mais cela peut ne pas être possible en raison des limitations de certains micro-contrôleurs Token Ring. Il reste alors deux possibilités. La première est de faire correspondre l'adresse IP multicast à l'adresse MAC de diffusion générale (*broadcast address*) qui devra être routée dans l'ensemble des anneaux de la topologie Token Ring dans un cas de routage par la source (*source-routing*). La seconde possibilité est de faire correspondre toutes les adresses IP multicast avec une adresse Token Ring spécifique localement administrée : C0-00-00-04-00-00. Cette dernière possibilité est préférable, mais elle n'est pas supportée par l'ensemble des machines. Aussi, sera-t-il nécessaire pour les machines qui supportent la correspondance avec une adresse Token Ring spécifique, de supporter aussi la correspondance avec l'adresse de diffusion générale.

**REMARQUE** L'adresse Token Ring C0-00-00-04-00-00 n'est pas réservée aux adresses IP multicast. Elle peut être utilisée par d'autres protocoles.

## Arbre de routage par la source

Comme indiqué auparavant, les méthodes de routage unicast ne peuvent pas être utilisées pour router des datagrammes destinés à des adresses IP multicast.

Une des possibilités pour router du trafic multicast est simplement d'*inonder* le réseau de routeur en routeur en le transférant sur toutes les interfaces sauf sur celle d'où émane le trafic. Une telle approche n'est, bien sûr, pas acceptable. Une telle inondation engendrée par l'envoi de données sur tous les routeurs, peut provoquer une saturation du réseau dans le cas de topologies avec mailles redondantes. En revanche, cette approche peut être retenue si dans un premier temps on élimine les chemins redondants, pour *arroser* ensuite les chemins restants par ce trafic multicast. Ceci évoque les algorithmes d'arbre de recouvrement (*spanning tree*) utilisés dans les réseaux locaux à multiples ponts transparents.

Une autre possibilité pour router le trafic multicast est de créer un arbre recouvrant l'ensemble du réseau dont la racine est la source du trafic multicast. Elle présente l'avantage, comme l'approche d'arbre de recouvrement classique, d'éviter les boucles dans les réseaux à mailles redondantes. Elle permet aussi d'optimiser l'acheminement du trafic en fonction de chacune

© copyright Éditions Eyrolles

des sources de trafic multicast. De tels arbres sont appelés *arbres de routage par la source (source-based trees).*

La manière dont sont construits de tels arbres dépend de l'implémentation du protocole de routage multicast. En général, la procédure en charge de la création de l'arbre de routage par la source est appelée mécanisme de découverte de topologie (*topology discovery mechanism*) ou plus simplement découverte de topologie (*topology discovery*). Il est intéressant de noter que ce mécanisme de découverte de topologie ne fait pas forcément partie intégrante du protocole de routage multicast lui-même.

Le mécanisme de découverte de topologie permet au routeur de connaître son voisinage, notamment de savoir si un routeur ou une machine sont en amont ou en aval par rapport à la source.

Les environnements réseau sont beaucoup plus dynamiques en configuration multicast qu'en unicast. Comme mentionné, le routage du trafic unicast est basé sur la partie réseau *networkid* de l'adresse IP. La partie réseau, ou préfixe réseau, correspond en général à un ensemble de réseaux ou à un seul réseau physique. Les réseaux physiques ne sont pas en perpétuelle création et destruction. Ce genre d'évènement survient en cas de déploiement de nouveaux réseaux ou en cas de dysfonctionnement de certains équipements. Dans le monde du multicast, de nouvelles machines peuvent rejoindre ou quitter un groupe fréquemment. Par conséquent, la découverte de topologie doit gérer les changements fréquents de constitution des groupes.

L'appartenance à un groupe est transmise *via* les messages *élaguer* ou *rejoindre* (*prune* ou *join messages*). Un routeur envoie un message *élaguer* sur les routeurs en amont vers la source s'il n'a plus de machine membre du groupe ou de routeur aval qui diffuse vers ce groupe. Les routeurs intermédiaires peuvent aussi envoyer un message *élaguer* vers l'amont si tous les voisins en aval ont envoyé un message *élaguer* de ce groupe multicast. Si au contraire, une machine (re)devient membre du groupe de diffusion multicast, alors le routeur auquel elle est rattachée va envoyer un message *rejoindre* pour récupérer le trafic multicast.

La méthode de routage multicast que l'on a décrite, est basée sur un type de protocole de routage multicast qualifié de dense (*dense-mode protocol*). Les protocoles de routage *denses* n'utilisent les messages *rejoindre* que si une machine rejoint un groupe précédemment *élagué*. Mais si une source commence à envoyer du trafic vers un groupe, elle arrosera tous les routeurs en aval. Si un routeur du voisinage n'a aucun membre actif du groupe, il utilisera le message *élaguer* pour se retrancher du groupe.

### *Arbres partagés*

Les protocoles en mode *dense* sont adéquats dans des environnements multicast où les groupes sont largement présents à l'intérieur du réseau ou dans un cas où la bande passante ne fait pas défaut. Ces protocoles sont, en revanche, inappropriés dans des environnements où les membres des groupes sont dispersés sur le réseau, comme sur l'Internet. Dans un tel cas, une technique modifiée de l'arbre routé par la source existe, il s'agit des arbres partagés (*shared trees*).

Un arbre routé par la source doit être construit pour chacune des sources qui émet pour un groupe. Un arbre partagé, lui, est bâti pour un groupe mais il est utilisé pour acheminer le trafic partant de n'importe quelle source.

© copyright Éditions Eyrolles

Les arbres basés sur la source prennent leur racine à la source du trafic multicast. Les arbres partagés prennent leur racine sur les routeurs, qui sont appelés *points de rendez-vous*. Ainsi, le trafic multicast de chaque source doit d'abord être acheminé vers le point de rendez-vous, d'où il va être ensuite distribué vers les membres du groupe en utilisant l'arbre partagé. Un seul point de rendez-vous existe par groupe multicast à un instant donné.

Les arbres partagés sont la base des protocoles de routages multicast *épars*. Les protocoles épars tolèrent la dispersion des membres de groupe sur le réseau contrairement aux protocoles denses. Les protocoles épars requièrent des messages *rejoindre* explicites de la part des membres d'un groupe. Ces messages sont alors propagés vers les routeurs en partant du point de rendez-vous. Si un message *rejoindre* n'est pas reçu d'un routeur pour tel groupe multicast, le trafic pour ce groupe n'est pas envoyé à ce routeur.

### Table de routage multicast

Les routeurs, pour leur algorithme de routage multicast, utilisent une structure de données appelée table de routage multicast. La table de routage multicast est une suite d'éléments contenant le groupe multicast, l'interface par laquelle le trafic arrive, les interfaces de sortie du trafic ainsi que d'autres champs.

Si un élément représente un routage multicast utilisant un arbre basé sur la source, il contient aussi l'adresse IP de la source. Un tel élément est souvent noté (S,G), où S désigne la source et G le groupe. Si un élément représente un routage multicast utilisant un arbre partagé, l'adresse de la source n'est pas incluse dans le couple. Il est noté (*,G), où l'astérisque, caractère générique, désigne le fait que l'arbre est partagé. Ainsi l'élément (*,G) est appelé élément générique.

### Algorithme de Reverse Path Forwarding

Indépendamment du type d'arbre utilisé pour router le trafic multicast, les routeurs utilisent toujours une procédure appelée algorithme RPF (*Reverse Path Forwarding*) pour prendre les décisions de routage.

L'algorithme RPF est décrit dans les lignes suivantes :

- L'algorithme dénomme l'interface utilisée pour acheminer le trafic unicast vers la source dans le cas d'arbre de routage par la source, ou vers le point de rendez-vous dans le cas d'arbre partagé, comme étant l'*interface RPF*.
- Si un datagramme arrive sur l'interface RPF, il est rerouté vers toutes les autres interfaces en sortie présentes dans les tables de routage multicast de ce groupe.
- Si un datagramme arrive sur une interface autre que l'interface RPF, il est tout simplement ignoré.

## Les protocoles IP multicast existants

Bien que le routage IP multicast soit encore en plein développement, certains protocoles ont le mérite d'exister, et la plupart d'entre eux sont implémentés dans le système IOS de Cisco. La section suivante décrit brièvement les protocoles multicast les plus connus disponibles actuellement.

© copyright Éditions Eyrolles

### Protocole de gestion de groupes IGMP (Internet Group Management Protocol)

L'un des plus importants protocoles de gestion de groupe multidestinataire est IGMP (*Internet Group Management Protocol*). Il en existe deux versions, la version 1 et 2 ; la version 3 est en cours de développement.

IGMP fournit au routeur la capacité de connaître les machines membres d'un groupe présentes sur les segments auquel ce routeur est directement connecté. Les routeurs utilisent IGMP pour tester l'état des membres d'un groupe périodiquement. IGMP permet de détecter qu'il n'y a plus de membre d'un groupe donné et qu'il est temps d'envoyer un message *élaguer*. Les machines membres d'un groupe annoncent qu'elles rejoignent le groupe ou qu'elles le quittent par les messages IGMP adéquats.

La version 1 de IGMP est décrite dans la RFC 1112, qui est un standard Internet. La version 2 de IGMP est documentée dans la RFC 2236 qui est juste une proposition de standard.

### Protocol Independent Multicast-Dense Mode (PIM-DM)

PIM-DM (*Protocol Independent Multicast Dense Mode*) est un protocole de routage multicast en mode dense. Il est indépendant des protocoles car il n'a pas son propre mécanisme de découverte de topologie. PIM-DM se base sur les protocoles de routage unicast sous-jacents et à l'algorithme RPF pour prendre les décisions de routage multicast.

Les routeurs PIM-DM qui n'ont plus de membres d'un certain groupe, ni plus de routeur membre en aval, envoient par eux-même un message *élaguer*. Les routeurs voisins qui reçoivent ce message cessent d'envoyer du trafic multicast destiné à ce groupe sur le routeur qui s'est retranché. Les messages *élaguer* ne sont pas envoyés régulièrement. Les routeurs qui reçoivent ce message enclenchent une temporisation. Au terme de la temporisation, les routeurs reprendront les envois de trafic multicast du groupe élagué vers les routeurs qui se sont retranchés. Ce cycle de diffusion/sélection (*broadcast and prune*) est le cycle classique des protocoles en mode dense.

PIM-DM utilise le protocole « *hello* » pour découvrir et surveiller l'activité des routeurs voisins.

Les détails de PIM-DM sont décrits dans les spécifications techniques *Protocol Independent Multicast Version 2 Dense Mode Specification* disponibles sur l'Internet à l'adresse *http://www.ietf.org/ids.by.wg/pim.html*.

### Protocol Independent Multicast-Sparse Mode (PIM-SM)

PIM-SM (*Protocol Independent Multicast Sparse Mode*) est plus un protocole de routage multicast nouveau qu'une simple évolution de PIM-DM. Toutefois, PIM-SM garde certaines fonctionnalités de PIM-DM. Ainsi, PIM-SM n'intègre pas non plus de mécanisme intrinsèque de découverte de topologie et se base sur le protocole de routage unicast et sur l'algorithme RPF (*Reverse Path Forwarding* – en fait son adaptation pour les arbres partagés) pour prendre les décisions de routage.

Voici les fonctionnalités clés de PIM-SM :

- Il crée un arbre partagé pour chaque groupe multicast. La racine de l'arbre est au point de *rendez-vous*.
- Il crée un arbre basé sur la source lorsque celle-ci diffuse un trafic qui le justifie.

© copyright Éditions Eyrolles

- Il requiert un message *rejoindre* explicite pour distribuer le trafic vers les membres du groupe. Le routeur PIM-SM propage alors le message *rejoindre* depuis le membre du groupe qui l'a émis vers le point de *rendez-vous* du groupe considéré.

Le protocole PIM-SM est documenté dans la RFC 2362, *(PIM-SM, Protocol Independent Multicast-Sparse Mode, Protocol Specification)*. Ce document a le statut d'expérimental.

### *Autres protocoles de routage multicast*

Tous les protocoles présentés jusqu'ici sont implémentés dans l'IOS de Cisco. Il en existe deux autres très populaires dont le premier est partiellement inclus dans l'IOS, alors que le second ne l'est pas du tout. Ces protocoles sont DVMRP (*Distance Vector Multicast Routing Protocol*) et le protocole d'extension multicast de OSPF (*MOSPF*)

DVMRP est le protocole conçu pour et implémenté dans le *backbone* multicast de l'Internet : MBONE. Les routeurs Cisco implémentent suffisamment de fonctionnalités DVMRP pour inter-opérer avec les autres routeurs supportant ce protocole de routage. DVMRP est décrit dans la RFC 1075, *Distance Vector Multicast Routing Protocol.*

MOSPF est décrit dans la RFC 1584, *Multicast Extensions to OSPF.*

Il existe aussi d'autres protocoles de routage multicast mais leur déscription sort du cadre de cet ouvrage.

# *Solutions de configuration*

Cette section donne quelques indications pour configurer le routage multicast sur les routeurs Cisco. Une compréhension plus approfondie du routage multicast nécessiterait plusieurs chapitres et va bien au-delà du classique routage IP qui est l'objet du présent ouvrage. Néanmoins, les configurations présentées ici sont un bon début pour mieux appréhender les mécanismes de routage multicast et la manière de les mettre en place sur des routeurs Cisco.

## Le programme MCASTER

Comme vous l'avez remarqué, **ping** est l'un des outils fondamentaux pour diagnostiquer un problème de routage IP. Malheureusement, cet outil n'est pas d'une grande aide dans un contexte de routage multicast.

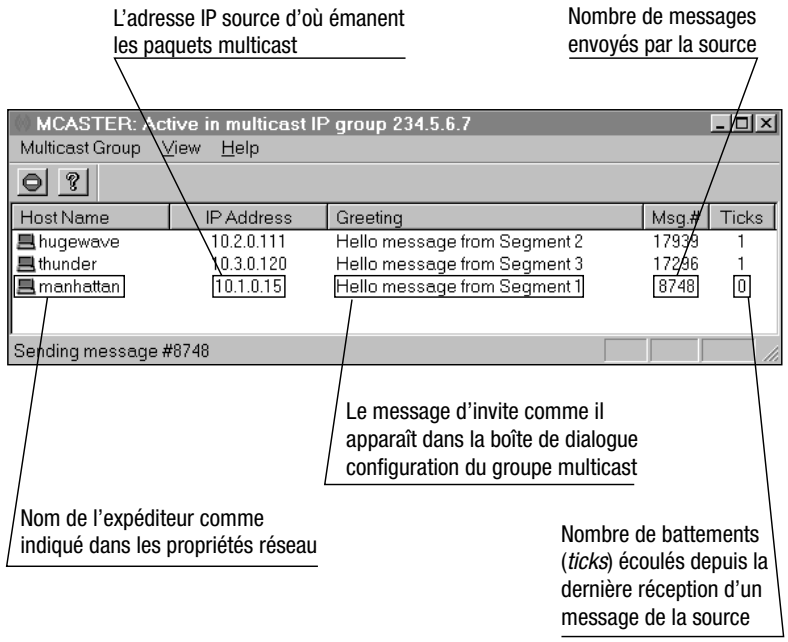
N'ayant pu trouver un outil aussi simple et aussi clair en terme de résultats que **ping**, nous avons créé notre propre programme de diagnostic appelé MCASTER qui sera utilisé comme outil de diagnostic dans ce chapitre.

MCASTER s'exécute sur une plate-forme Windows NT (il ne tourne pas sous Windows 95). MCASTER permet de rejoindre un groupe multicast donné, de recevoir des trames UDP de trafic multicast de la même manière que **ping** et d'envoyer des paquets vers un groupe. Il permet aussi d spécifier une adresse IP de groupe, un port UDP, et bien d'autres paramètres.

La figure 8.1 montre la fenêtre principale de MCASTER et explique ses principaux éléments.

© copyright Éditions Eyrolles

**Figure 8.1**
*Copie écran de la fenêtre MCASTER*




Nous avons essayé de rendre MCASTER aussi convivial et facile d'utilisation que possible. MCASTER est gratuit et disponible sur l'Internet à l'adresse *http://www.hugewave.com/blacktools*. N'hésitez pas à nous signaler les erreurs éventuelles ou à nous suggérer des améliorations.

## Configuration de PIM-DM

La configuration de PIM-DM est facile. Il vous suffit d'activer PIM-DM sur les interfaces qui vont faire partie de la zone de trafic multicast avec la commande **ip pim dense-mode**. Cette commande active automatiquement IGMP sur l'interface considérée.

**AVERTISSEMENT** La nature de PIM (DM et SM) nécessite une bonne configuration du routage *unicast* dans la zone desservie par PIM.

Pour comprendre comment configurer PIM-DM et comment il fonctionne, travaillons sur l'exemple de réseau de la figure 8.2.

Les listings 8.1 à 8.3 montrent les configurations des trois routeurs. Noter que EIGRP est le protocole de routage unicast utilisé avec PIM-DM

**Listing 8.1. Configuration du routeur R1.**

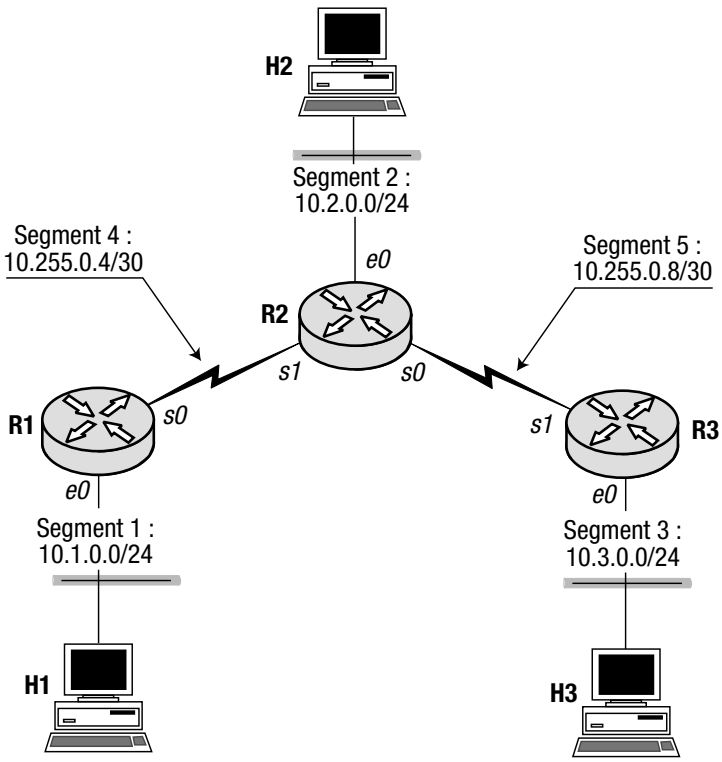
```
ip multicast-routing

interface Ethernet0
 ip address 10.1.0.1 255.255.255.0
 ip pim dense-mode

interface Serial0
```

© copyright Éditions Eyrolles

```
 ip address 10.255.0.6 255.255.255.252
 ip pim dense-mode

router eigrp 10
 network 10.0.0.0
```

**Figure 8.2**

*Les routeurs sont configurés avec PIM-DM pour que les machines communiquent en IP multicast*




**Listing 8.2. Configuration du routeur R2.**

```
ip multicast-routing

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0
 ip pim dense-mode

interface Serial0
 ip address 10.255.0.9 255.255.255.252
 ip pim dense-mode

interface Serial1
 ip address 10.255.0.5 255.255.255.252
 ip pim dense-mode

router eigrp 10
 network 10.0.0.0
```

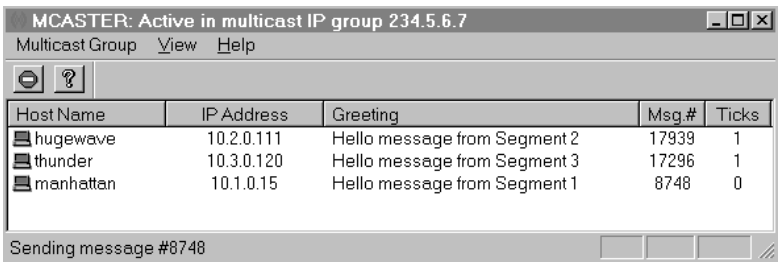
© copyright Éditions Eyrolles

**Listing 8.3. Configuration du routeur R3.**

```
ip multicast-routing

interface Ethernet0
 ip address 10.3.0.1 255.255.255.0
 ip pim dense-mode

interface Serial1
 ip address 10.255.0.10 255.255.255.252
 ip pim dense-mode

router eigrp 10
 network 10.0.0.0
```

Les trois machines sont des stations Windows NT dont les noms sont MANHATTAN (H1), HUGEWAVE (H2) et THUNDER (H3). MCASTER est configuré sur les trois stations pour rejoindre le groupe multicast 234.5.6.7 et utiliser le port UDP 3456 pour l'émission et la réception de données multicast. La fenêtre principale de MCASTER de la machine H1 est montrée figure 8.3.

**Figure 8.3**
*Copie écran de la fenêtre MCASTER sur la machine H1*




Comme vous pouvez le voir, la machine H1 peut voir les deux autres machines. Vous auriez le même type de résultat sur les fenêtres MCASTER des deux autres machines.

La commande utilisée pour afficher la table de routage multicast est **show ip mroute**, commande similaire à **show ip route**. La sortie écran de cette commande exécutée sur le routeur R1 est sur le listing 8.4.

**Listing 8.4. Sortie écran de la commande `show ip mroute` exécutée sur le routeur R1.**

```
R1#show ip mroute
IP Multicast Routing Table
Flags: D-Dense, S-Sparse, C-Connected, L-Local, P-Pruned
     R-RP-bit set, F-Register flag, T-SPT-bit set, J-Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 05:30:47/00:00:00, RP 0.0.0.0, flags: DJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Dense, 05:30:47/00:00:00
    Serial0, Forward/Dense, 05:30:47/00:00:00

(*, 234.5.6.7), 05:30:47/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
```

© copyright Éditions Eyrolles

```
  Outgoing interface list:
    Ethernet0, Forward/Dense, 02:29:40/00:00:00
    Serial0, Forward/Dense, 05:30:47/00:00:00

(10.1.0.15, 234.5.6.7), 02:29:39/00:02:59, flags: CT
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 02:29:39/00:00:00

(10.2.0.111, 234.5.6.7), 02:32:47/00:02:59, flags: CT
  Incoming interface: Serial0, RPF nbr 10.255.0.5
  Outgoing interface list:
    Ethernet0, Forward/Dense, 02:29:47/00:00:00

(10.3.0.120, 234.5.6.7), 02:32:46/00:02:59, flags: CT
  Incoming interface: Serial0, RPF nbr 10.255.0.5
  Outgoing interface list:
    Ethernet0, Forward/Dense, 02:29:47/00:00:00
```

La sortie de la commande **show ip mroute** est la suite de lignes correspondant aux entrées de la table de routage multicast. On y retrouve les couples source et groupe multicast, sous la forme (S,G) ou (*,G), S étant l'adresse IP de la source et G l'adresse IP du groupe multidestinataire. Plus loin, nous trouvons l'interface d'arrivée, la liste des interfaces en sortie, des indicateurs sur le mode de création de l'entrée – mode dense ou mode épars – et bien d'autres informations.

Notez que bien que le couple (*,234.5.6.7) avec le caractère générique * a été créé avec une interface d'arrivée qui est Null (ceci indique, attention, qu'il n'y a pas d'interface d'arrivée, et cela n'a aucun rapport avec l'interface symbolique Null disponible sur les routeurs Cisco). En effet, dans le cas d'un arbre partagé, on utilise le caractère générique * pour le groupe. Comme nous le savons, PIM-DM n'utilise pas d'arbre partagé.

Vous pouvez utiliser la commande **debug ip igmp** pour voir les transactions IGMP et la commande **debug ip pim** pour voir les transactions PIM.

**ASTUCE** La commande **debug ip pim** est utilisée pour PIM-DM et PIM-SM.

Le listing 8.5 montre la sortie écran de ces deux commandes pendant que l'on arrête MCASTER sur la machine H1 (MANHATTAN, adresse IP 10.1.0.15). Lorsque MCASTER s'arrête, il envoie un message IGMP indiquant que la machine quitte le groupe multicast.

**Listing 8.5. Sortie écran des commandes `debug ip igmp` et `debug ip pim` sur les routeurs R1.**

```
R1#debug ip igmp
IGMP debugging is on
R1#debug ip pim
PIM debugging is on
R1#
PIM: Send v2 Hello on Serial0
IGMP: Received Leave from 10.1.0.15 (Ethernet0) for 234.5.6.7
IGMP: Send v2 Query on Ethernet0 to 234.5.6.7
```

© copyright Éditions Eyrolles

```
IGMP: Send v2 Query on Ethernet0 to 234.5.6.7
PIM: Send v2 Hello on Ethernet0
IGMP: Deleting 234.5.6.7 on Ethernet0
PIM: Send v2 Prune on Serial0 to 10.255.0.5 for
 (10.2.0.111/32, 234.5.6.7)
PIM: Send v2 Prune on Serial0 to 10.255.0.5 for
 (10.3.0.120/32, 234.5.6.7)
PIM: Send v2 Prune on Serial0 to 10.255.0.5 for
 (10.2.0.111/32, 234.5.6.7)
PIM: Send v2 Prune on Serial0 to 10.255.0.5 for
 (10.3.0.120/32, 234.5.6.7)
```

Notez qu'une fois que la machine H1 a quitté le groupe, le routeur R1 envoie plusieurs messages *élaguer* sur l'interface Serial0, car la machine H1 était la seule machine membre du groupe 234.5.6.7.

Le listing 8.6 montre la sortie écran de la commande **show ip mroute** entrée sur le routeur R1, une fois que la machine H1 a quitté le groupe et que le routeur R1 s'est retranché du groupe. Notez que les entrées pointant sur les sources localisées sur des segments distants (c'est-à-dire 10.2.0.111 et 10.3.0.120) ont maintenant l'attribut P (*prune*).

**Listing 8.6. Sortie écran de la commande `show ip mroute` une fois H1 (MANHATTAN) retranché du groupe.**

```
R1#show ip mroute
...
(*, 234.5.6.7), 05:35:08/00:02:58, RP 0.0.0.0, flags: DJ
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 05:35:08/00:00:00

(10.1.0.15, 234.5.6.7), 02:33:59/00:02:35, flags: T
  Incoming interface: Ethernet0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 02:33:59/00:00:00

(10.2.0.111, 234.5.6.7), 02:37:00/00:02:38, flags: PT
  Incoming interface: Serial0, RPF nbr 10.255.0.5
  Outgoing interface list: Null

(10.3.0.120, 234.5.6.7), 02:37:01/00:02:37, flags: PT
  Incoming interface: Serial0, RPF nbr 10.255.0.5
  Outgoing interface list: Null
```

Regardons maintenant ce qu'il se passe lorsque la machine H1 rejoint le groupe au travers du listing 8.7, montrant les sorties écrans des commandes **debug ip igmp** et **debug ip pim** sur le routeur R1. Notez que le routeur R1 utilise maintenant le message *greffer*, qui est analogue au message *rejoindre* mais qui accélère le changement d'état concernant le groupe précédemment détaché 234.5.6.7 sur le routeur R1.

© copyright Éditions Eyrolles

**Listing 8.7. Sortie écran des commandes `debug ip igmp` et `debug ip pim` sur le routeur R1 une fois que H1 (MANHATTAN) rejoint le groupe multicast.**

```
R1#debug ip igmp
IGMP debugging is on
R1#debug ip pim
PIM debugging is on
R1#
PIM: Received v2 Hello on Serial0 from 10.255.0.5
IGMP: Send v2 Query on Ethernet0 to 224.0.0.1
IGMP: Set report delay time to 3.0 seconds for 224.0.1.40
 on Ethernet0
IGMP: Received v2 Report from 10.1.0.15 (Ethernet0) for
 234.5.6.7
PIM: Building Graft message for 234.5.6.7, Ethernet0:
 no entries
PIM: Building Graft message for 234.5.6.7, Serial0:
     10.2.0.111/32 10.3.0.120/32
PIM: Send v2 Graft to 10.255.0.5 (Serial0)
PIM: Received v2 Graft-Ack on Serial0 from 10.255.0.5
     Group 234.5.6.7:
     10.2.0.111/32 10.3.0.120/32
```

## Configuration de PIM-SM

La configuration de PIM-SM n'est pas plus difficile que celle de PIM-DM. Il s'agit d'entrer les commandes **ip pim sparse-mode** sur les interfaces supposées transférer le trafic IP multicast. Il faut aussi configurer l'adresse IP du point de rendez-vous sur les routeurs qui connectent les réseaux en aval, c'est à dire ceux qui ont des membres de groupes multicast attachés à leurs interfaces.

**ASTUCE** Il ne faut pas configurer un routeur comme point de rendez-vous. Il découvrira automatiquement son état au travers de la configuration effectuée sur les autres routeurs des réseaux en aval.

Pour configurer l'adresse IP du point de rendez-vous, utilisez la commande **ip pim rp-address** *<adresse IP>* [{*<numero LA>*|*<nom LA>*}]. Le paramètre *<adresse IP>* est l'adresse IP du point de rendez-vous. Le paramètre optionnel est la liste d'accès qui peut être utilisée pour limiter ce point de rendez-vous à certains groupes multicast seulement. Seuls les groupes sur lesquels la liste d'accès renvoie **permit** seront pris en considération sur le point de rendez-vous.

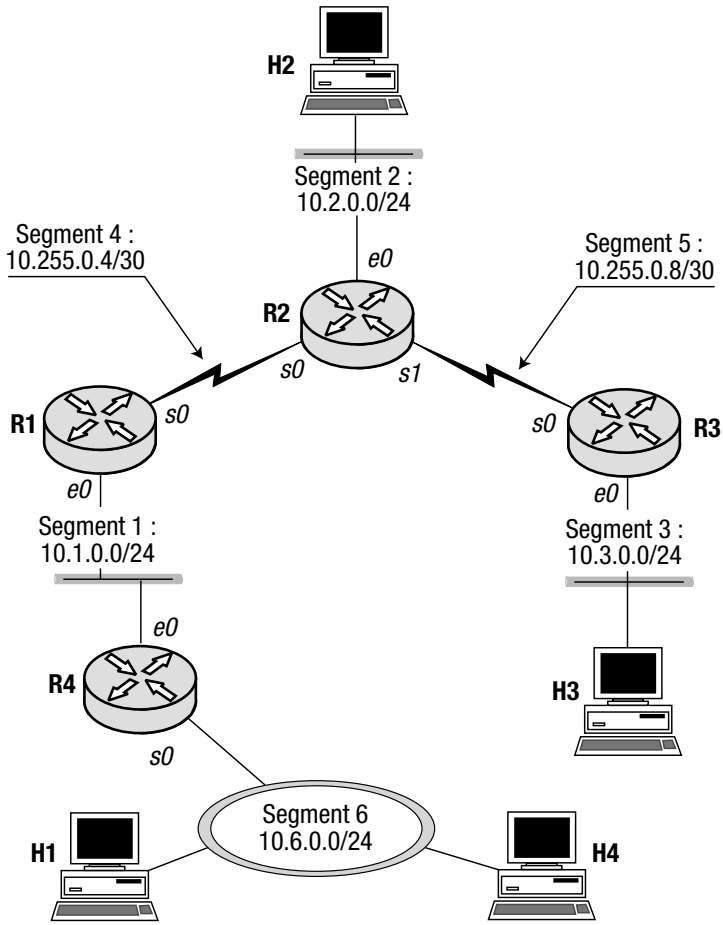
Étudions la topologie réseau de la figure 8.4. Supposons que les routeurs doivent être configurés en PIM-SM pour fournir un service multicast à tout le réseau. Les routeurs ont pour protocole de routage unicast EIGRP et utilisent leurs interfaces de rebouclage (*loopback*) avec une adresse IP qui vaut 10.0.0.X/32, où X est le numéro du routeur.

Utilisons l'adresse IP de l'interface Loopback0 du routeur R2 comme point de rendez-vous pour tous les groupes multicast. Les listings 8.8 à 8.11 montrent les configurations des quatre routeurs.

© copyright Éditions Eyrolles

**Figure 8.4**

*Les routeurs sont configurés avec PIM-SM pour permettre aux machines de communiquer en IP multicast.*




**Listing 8.8. Configuration du routeur R1.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-mode

interface Serial0
 ip address 10.255.0.6 255.255.255.252
 ip pim sparse-mode

router eigrp 10
 network 10.0.0.0
```

© copyright Éditions Eyrolles

**Listing 8.9. Configuration du routeur R2.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0
 ip pim sparse-mode

interface Serial0
 ip address 10.255.0.5 255.255.255.252
 ip pim sparse-mode

interface Serial1
 ip address 10.255.0.9 255.255.255.252
 ip pim sparse-mode

router eigrp 10
 network 10.0.0.0
```

**Listing 8.10. Configuration du routeur R3.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Ethernet0
 ip address 10.3.0.1 255.255.255.0
 ip pim sparse-mode

interface Serial0
 ip address 10.255.0.10 255.255.255.252
 ip pim sparse-mode

router eigrp 10
 network 10.0.0.0

ip pim rp-address 10.0.0.2
```

**Listing 8.11. Configuration du routeur R4.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.1.0.2 255.255.255.0
 ip pim sparse-mode

interface TokenRing0
 ip address 10.6.0.1 255.255.255.0
 ip pim sparse-mode
 ring-speed 16

router eigrp 10
 network 10.0.0.0

ip pim rp-address 10.0.0.2
```

© copyright Éditions Eyrolles

Le listing 8.12 montre la sortie écran de la commande **show ip mroute** exécutée sur le routeur R4. Notez que le couple (*, 234.5.6.7) dans la table de routage qui a cette fois-ci une interface d'entrée définie (Incoming interface: Ethernet0), est identifié comme épars (flags: S...) et a un point de rendez-vous (RP 10.0.0.2). Ceci est rendu possible par la nature même de PIM-SM, épars et utilisant des arbres partagés.

**Listing 8.12. Sortie écran de la commande `show ip mroute` exécutée sur le routeur R4.**

```
R4#show ip mroute
IP Multicast Routing Table
Flags: D-Dense, S-Sparse, C-Connected, L-Local, P-Pruned
   R-RP-bit set, F-Register flag, T - SPT-bit set, J-Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:11:08/00:00:00, RP 10.0.0.2, flags: SJPCL
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list: Null

(*, 234.5.6.7), 00:11:08/00:02:59, RP 10.0.0.2, flags: SJC
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse, 00:11:08/00:02:23

(10.2.0.111, 234.5.6.7), 00:11:08/00:02:59, flags: CT
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse, 00:11:08/00:02:23

(10.3.0.120, 234.5.6.7), 00:11:08/00:02:59, flags: CT
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse, 00:11:08/00:02:23

(10.6.0.10, 234.5.6.7), 00:11:09/00:02:59, flags: CT
  Incoming interface: TokenRing0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 00:10:27/00:02:26

(10.6.0.15, 234.5.6.7), 00:11:09/00:02:59, flags: CT
  Incoming interface: TokenRing0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 00:10:27/00:02:26
```

## Configuration de PIM-SM et PIM-DM sur la même interface simultanément

En entrant les commandes **ip pim dense-mode** et **ip pim sparse-mode**, vous choisissez un mode exclusif pour l'interface. Dans certaines situations, vous devez choisir PIM-DM pour certains groupes et PIM-SM pour d'autres groupes. Il est possible de réaliser une telle tâche avec la commande **ip pim sparse-dense-mode** à la place des deux précédentes commandes, ce qui permet d'utiliser PIM-SM pour tous les groupes qui disposent d'un point de *rendez-vous* et PIM-DM pour les autres.

Ainsi, il est possible d'utiliser cette commande conjointement avec la commande **ip pim rp-address** *<adresse IP>* [{*<numéro de LA>*|*<nom de LA>*}] sur les routes qui connectent les

© copyright Éditions Eyrolles

réseaux en aval. Les listes d'accès vous permettent de définir les groupes utilisant PIM-SM et ceux utilisant PIM-DM.

## Configuration de PIM-SM sur des réseaux NBMA

Comme d'habitude, les réseaux NBMA (*Non-Broadcast Multiple Access Network*) représentent un cas spécial pour le routage multicast. Si un routeur est connecté à un réseau NBMA non intégralement maillé, avec une seule interface, il devrait pouvoir distribuer du trafic au reste du réseau par la même interface. En général, ceci n'est pas possible car une interface d'entrée ne peut être aussi une interface en sortie.

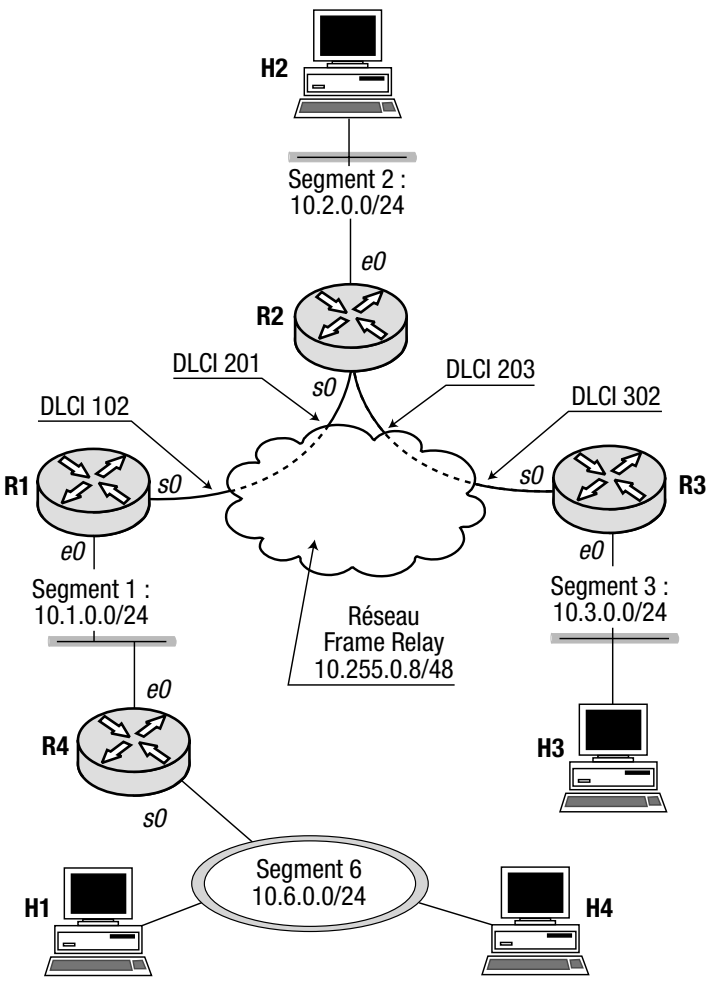
Pour effectuer un routage multicast dans un tel cas, PIM-SM possède un mode désigné mode NBMA. Dans le mode NBMA, PIM-SM gère des listes d'interface en sortie pour chacun des routeurs voisins accessibles *via* le réseau NBMA.

Pour disposer du mode NBMA, il faut entrer la commande **ip pim nbma-mode** dans le mode de configuration de l'interface qui connecte le routeur au réseau NBMA.

La figure 8.5 montre une version modifiée du réseau précédemment étudié pour les configurations PIM-SM. Dans cette version, le routeur R2 est connecté à deux circuits virtuels

**Figure 8.5**

*Le routeur R2 doit prendre en compte qu'il n'est pas connecté à un réseau frame-relay entièrement maillé.*




© copyright Éditions Eyrolles

Frame Relay *via* son interface Serial0. Il est obligatoire d'utiliser le mode NBMA pour créer du trafic multicast sur le réseau.

Configurons tous les routeurs avec la commande **ip pim sparse-dense-mode** et lançons deux instances de MCASTER – une pour le groupe multicast 234.5.6.7 et l'autre pour le groupe 234.9.9.9 – sur les machines H1, H2 et H3.

**ASTUCE** Si vous réalisez ces expérimentations, il faut choisir deux numéros de ports UDP différents.

Pour le groupe 234.5.6.7, nous définirons un point de rendez-vous avec la commande **ip pim rp-address 10.0.0.2 1**, où 10.0.0.2 est l'adresse IP de l'interface Loopback0 du routeur R2, et 1 est le numéro de la liste d'accès standard qui laisse passer uniquement le groupe 234.5.6.7.

Ainsi, PIM-SM sera le protocole du groupe 234.5.6.7 (présence d'un point de rendez-vous) et PIM-DM du groupe 234.9.9.9.

Les listings 8.13 à 8.16 montrent les configurations des quatre routeurs.

**Listing 8.13. Configuration du routeur R1.**

```
ip multicast-routing
interface Loopback0
 ip address 10.0.0.1 255.255.255.255

interface Ethernet0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-dense-mode

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.248
 ip pim sparse-dense-mode
frame-relay interface-dlci 102

router eigrp 10
 network 10.0.0.0
```

**Listing 8.14. Configuration du routeur R2.**

```
ip dvmrp route-limit 20000

interface Loopback0
 ip address 10.0.0.2 255.255.255.255

interface Ethernet0
 ip address 10.2.0.1 255.255.255.0
 ip pim sparse-dense-mode

interface Serial0
 ip address 10.255.0.9 255.255.255.248
 ip pim nbma-mode
 ip pim sparse-dense-mode
encapsulation frame-relay
 no ip split-horizon eigrp 10
```

© copyright Éditions Eyrolles

```
 bandwidth 64
 frame-relay map ip 10.255.0.10 201 broadcast
 frame-relay map ip 10.255.0.11 203 broadcast
 frame-relay lmi-type ansi

router eigrp 10
 network 10.0.0.0
```

**Listing 8.15. Configuration du routeur R3.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.3 255.255.255.255

interface Ethernet0
 ip address 10.3.0.1 255.255.255.0
 ip pim sparse-dense-mode

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 multipoint
 ip address 10.255.0.11 255.255.255.248
 ip pim sparse-dense-mode
 frame-relay map ip 10.255.0.9 302 broadcast

router eigrp 10
 network 10.0.0.0

ip pim rp-address 10.0.0.2 1

access-list 1 permit 234.5.6.7
```

**Listing 8.16. Configuration du routeur R4.**

```
ip multicast-routing

interface Loopback0
 ip address 10.0.0.4 255.255.255.255

interface Ethernet0
 ip address 10.1.0.2 255.255.255.0
 ip pim sparse-dense-mode

interface TokenRing0
 ip address 10.6.0.1 255.255.255.0
 ip pim sparse-dense-mode
 ring-speed 16

router eigrp 10
 network 10.0.0.0

ip pim rp-address 10.0.0.2 1

access-list 1 permit 234.5.6.7
```

© copyright Éditions Eyrolles

Le listing 8.17 montre la sortie écran de la commande **show ip mroute** exécutée sur le routeur R4. Notez que maintenant il y a deux ensembles d'entrées séparés, un pour le groupe 234.5.6.7 et un pour le groupe 234.9.9.9. Notez aussi que le couple (*,234.5.6.7) est étiqueté comme opérant sous PIM-SM (flags: S...), tandis que le couple (*,234.9.9.9) est étiqueté comme opérant sous PIM-DM (flags: D...).

**Listing 8.17. Sortie écran de la commande `show ip mroute` sur le routeur R4.**

```
R4#show ip mroute
IP Multicast Routing Table
Flags: D-Dense, S-Sparse, C-Connected, L-Local, P-Pruned
     R-RP-bit set, F-Register flag, T-SPT-bit set, J-Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:33:02/00:00:00, RP 0.0.0.0, flags: DJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:33:02/00:00:00

(*, 234.9.9.9), 00:25:30/00:02:58, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    TokenRing0, Forward/Sparse-Dense, 00:22:49/00:00:00
    Ethernet0, Forward/Sparse-Dense, 00:25:30/00:00:00

(10.2.0.111, 234.9.9.9), 00:15:11/00:02:59, flags: CT
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse-Dense, 00:15:11/00:00:00

(10.6.0.15, 234.9.9.9), 00:22:22/00:03:29, flags: CT
  Incoming interface: TokenRing0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:22:23/00:00:00

(*, 234.5.6.7), 00:34:10/00:02:59, RP 10.0.0.2, flags: SJC
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse-Dense, 00:34:10/00:02:07

(10.2.0.111, 234.5.6.7), 00:12:19/00:02:59, flags: CJT
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse-Dense, 00:12:19/00:02:07

(10.3.0.120, 234.5.6.7), 00:34:10/00:02:59, flags: CT
  Incoming interface: Ethernet0, RPF nbr 10.1.0.1
  Outgoing interface list:
    TokenRing0, Forward/Sparse-Dense, 00:34:10/00:02:06

(10.6.0.10, 234.5.6.7), 00:01:32/00:01:57, flags: CT
  Incoming interface: TokenRing0, RPF nbr 0.0.0.0
  Outgoing interface list:
```

© copyright Éditions Eyrolles

```
    Ethernet0, Forward/Sparse-Dense, 00:01:32/00:03:23

(10.6.0.15, 234.5.6.7), 00:21:40/00:03:29, flags: CT
  Incoming interface: TokenRing0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0, Forward/Sparse-Dense, 00:21:40/00:03:22
```

Si nous affichons la table de routage du routeur R2 (cf. listing 8.18), nous remarquons qu'elle contient une entrée pour chacune des interfaces de sortie vers chacun des routeurs voisins.

**Listing 8.18. Sortie écran de la commande `show ip mroute` sur le routeur R2.**

```
R2#show ip mroute
...
(*, 234.5.6.7), 00:01:11/00:02:59, RP 10.0.0.2, flags: S
 Incoming interface: Null, RPF nbr 0.0.0.0
 Outgoing interface list:
  Serial0, 10.255.0.11,Forward/Sparse-Dense,00:00:35/00:02:54
  Serial0, 10.255.0.10,Forward/Sparse-Dense,00:00:21/00:03:08
```

En observant les différentes instances de MCASTER sur les trois machines interconnectées, il est clair que PIM-DM n'est pas fonctionnel sur un réseau Frame Relay non intégralement maillé, en dépit de la présence de la commande **ip pim nbma-mode** sur la configuration du routeur R2.

Les figures 8.6 à 8.11 montrent le contenu de la fenêtre principale de MCASTER sur les trois machines. Les figures sont regroupées par paires. La première paire (figures 8.6 et 8.7) montre la fenêtre principale de MCASTER pour les groupes 234.5.6.7 et 234.9.9.9 sur la machine H1 (MANHATTAN). Notez bien que chaque groupe a trois membres, MCASTER en montre trois pour le groupe 234.5.6.7 mais seulement deux pour le groupe 234.9.9.9. La deuxième fenêtre ne montre pas la machine H3 (THUNDER) qui est derrière le nuage Frame Relay. Le groupe 234.9.9.9 est sous PIM-DM, il apparaît clairement que PIM-DM n'est pas complètement fonctionnel sur un réseau NBMA non intégralement maillé.

**Figure 8.6**

*Fenêtre principale de MCASTER sur la machine H1 (MANHATTAN) pour le groupe 234.5.6.7.*

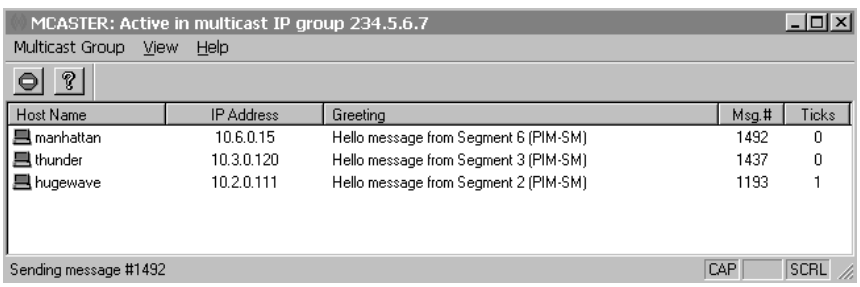
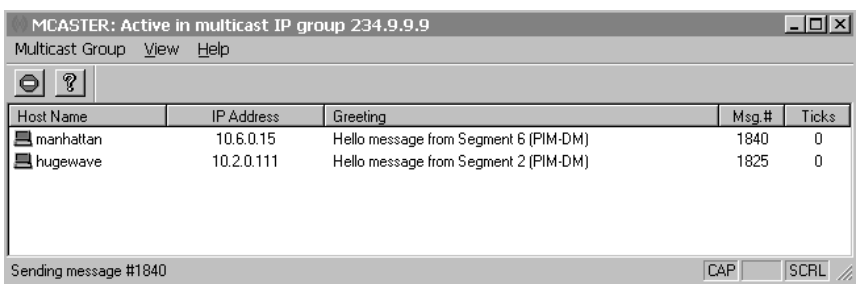



**Figure 8.7**

*Fenêtre principale de MCASTER sur la machine H1 (MANHATTAN) pour le groupe 234.9.9.9.*




© copyright Éditions Eyrolles

La situation est différente dans le cas de H2 (HUGEWAVE). La machine H2 voit les trois membres de chacun des deux groupes. La raison en est que H2 est derrière le routeur R2 qui est le seul routeur qui peut accepter du trafic de la part des deux autres parties du réseau, bien qu'il soit en PIM-DM.

**Figure 8.8**

*Fenêtre principale de MCASTER sur la machine H2 (HUGEWAVE) pour le groupe 234.5.6.7.*

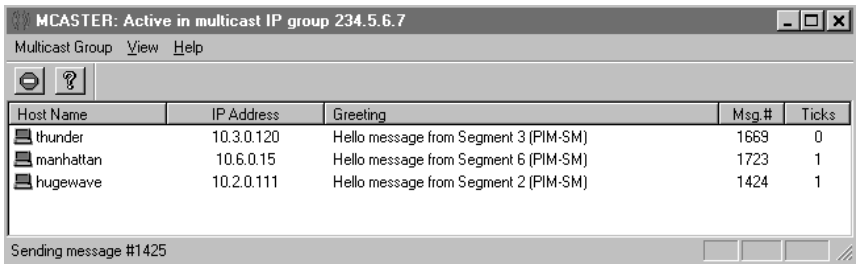
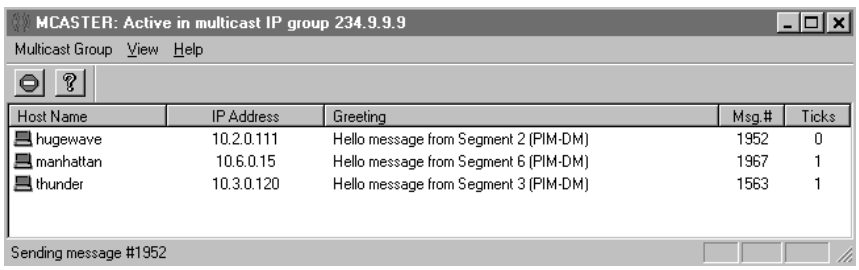



**Figure 8.9**

*Fenêtre principale de MCASTER sur la machine H2 (HUGEWAVE) pour le groupe 234.9.9.9.*




La situation sur la machine H3 est très similaire à celle de H1 car H3 ne peut pas voir H1 dans le groupe 234.9.9.9.

**Figure 8.10**

*Fenêtre principale de MCASTER sur la machine H3 (THUNDER) pour le groupe 234.5.6.7.*

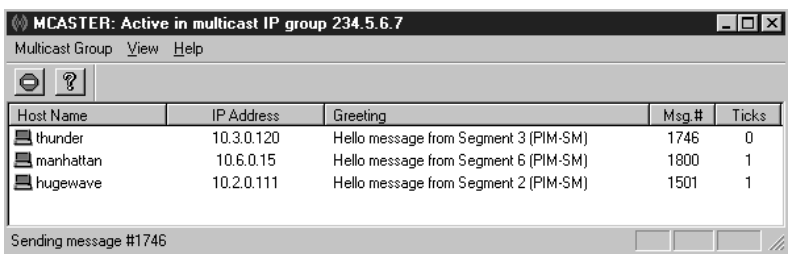
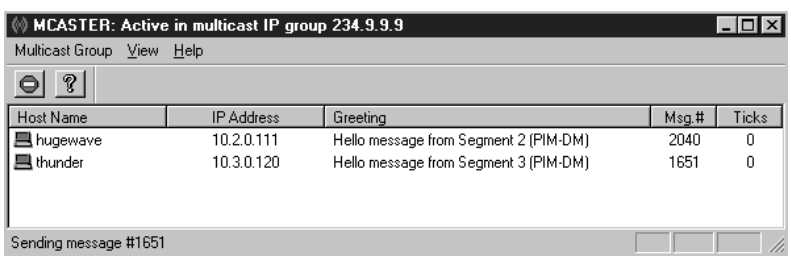



**Figure 8.11**

*Fenêtre principale de MCASTER sur la machine H3 (THUNDER) pour le groupe 234.9.9.9.*




© copyright Éditions Eyrolles

# A

# Connexion de deux routeurs Cisco dos à dos en utilisant deux câbles série

Il est parfois nécessaire de relier deux routeurs Cisco dos à dos pour faire des essais. Il nous faut dans ce cas nous procurer deux câbles dont l'un, de type CAB-V35MT, comprend un connecteur mâle ETTD (Équipement Terminal de Traitement de Données) ou DTE (*Data Terminal Equipment*), et l'autre, de type CAB-V35FC comporte un connecteur femelle ETCD (Équipement de Terminaison de Circuit de Données) ou DCE (*Data Communication Équipment*). Une fois ces câbles connectés aux interfaces série des routeurs, ceux-ci en détectent automatiquement le type.

Ces câbles doivent ensuite être reliés entre eux, et comme il s'agit d'une liaison synchrone, une source d'horloge est nécessaire. Normalement, celle-ci provient de l'équipement ETCD appelé CSU/DSU (*Channel Service Unit/Data Service Unit*) qui relie le routeur à la ligne synchrone. C'est l'interface série DCE de l'un des routeurs – sur laquelle nous devons définir la vitesse d'horloge par la commande **clock rate** <*valeur*> – qui va jouer le rôle de synchronisation. Le paramètre *valeur* est renseigné en bits par seconde (bit/s) suivant les valeurs du tableau A.1. qui peuvent être affichées (ou non) par la commande **clock rate ?** selon la version du système IOS de Cisco utilisée.

Une autre commande disponible, **clockrate** <*valeur*>, n'affiche pas, quant à elle, les valeurs autorisées du paramètre.

© copyright Éditions Eyrolles

**Tableau A.1. Valeurs disponibles pour la vitesse horloge.**

| Valeur basses | Valeurs hautes |
|---|---|
| 1200 | 125000 |
| 2400 | 148000 |
| 4800 | 250000 |
| 9600 | 500000 |
| 19200 | 800000 |
| 38400 | 1000000 |
| 56000 | 1300000 |
| 64000 | 2000000 |
| 72000 | 4000000 |

© copyright Éditions Eyrolles

# B

# Configuration d'un routeur Cisco en commutateur Frame Relay

Les routeurs Cisco peuvent être configurés en commutateurs Frame Relay en procédant selon les étapes décrites ci-dessous.

1. En mode de configuration globale, activer la fonction de commutation par la commande **frame-relay switching**.
2. En mode de configuration d'interface, pour toutes les interfaces série destinées à la commutation Frame Relay, préciser le type d'encapsulation par la commande **encapsulation frame-relay**.
3. Toutes les interfaces série avec encapsulation Frame Relay doivent aussi être configurées en tant qu'interfaces ETCD (ou DCE), par la commande **frame-relay intf-type dce**.

**REMARQUE** Il n'est pas obligatoire que l'interface Frame Relay configurée en DCE soit physiquement un tel équipement.

4. En mode de configuration d'interface, établir la commutation de chaque CVP par la commande **frame-relay route** <*DLCI en entrée*> **interface serial** <*numéro d'interface*> <*DLCI en sortie*> sur toutes les interfaces impliquées dans la commutation Frame Relay. Le premier paramètre désigne le numéro de DLCI du CVP de l'interface en entrée, le deuxième paramètre désigne le numéro de l'interface de sortie sur laquelle ce CVP doit être routé sous le numéro de CVP du DLCI renseigné en dernier paramètre.

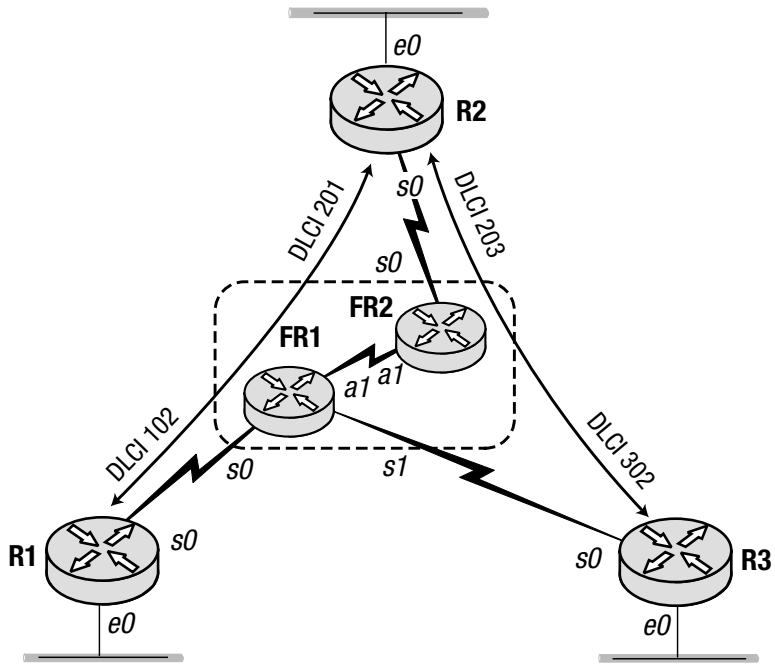
**ASTUCE** Pour router les CVP, on peut aussi utiliser la commande **interface tunnel** à la place de **interface serial**. Par cette méthode qui est employée dans l'exemple, le trafic Frame Relay est encapsulé en IP pour être routé à travers un réseau TCP/IP vers la destination désirée.

© copyright Éditions Eyrolles

5. En option, vous pouvez configurer le type de LMI en utilisant la commande **frame-relay lmi-type** {**ansi**|**cisco**|**q933a**} sur les interfaces de commutation Frame Relay. Par défaut, le type de LMI est **cisco**.

Étudions l'exemple du schéma de réseau sur la figure B.1. Il illustre un moyen très économique de montage d'une maquette de commutation Frame Relay sur un banc d'essai, si les routeurs utilisés n'ont pas plus de deux interfaces série. Cette solution met à contribution l'interface auxiliaire asynchrone dans chaque routeur, qui est souvent négligée. On peut ainsi réserver les interfaces série du routeur pour d'autres usages comme dans ce cas.

**Figure B.1**
*Routeurs FR1 et FR2 configurés en commutateurs Frame Relay.*




Les listings B.1 et B.2 montrent la configuration de chacun des routeurs FR1 et FR2 qui jouent le rôle de commutateurs Frame Relay. Les listings B.3 à B.5 concernent les autres routeurs.

**Listing B.1. Configuration du routeur FR1.**

```
username FR2 password 0 cisco
frame-relay switching

interface Tunnel0
 tunnel source 1.0.0.1
 tunnel destination 1.0.0.2

interface Ethernet0
 ip address 169.124.84.34 255.255.255.0

interface Serial0
 encapsulation frame-relay
 clockrate 64000
```

© copyright Éditions Eyrolles

```
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 102 interface Tunnel0 421

interface Serial1
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 302 interface Tunnel0 423

interface Async1
 ip address 1.0.0.1 255.255.255.0
 encapsulation ppp
 async default routing
 async mode dedicated
 ppp authentication chap

line aux 0
 rxspeed 38400
 txspeed 38400
```

**Listing B.2. Configuration du routeur FR2.**

```
username FR1 password 0 cisco
frame-relay switching

interface Tunnel0
 tunnel source 1.0.0.2
 tunnel destination 1.0.0.1

interface Ethernet0
 ip address 169.124.84.36 255.255.255.0

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 201 interface Tunnel0 421
 frame-relay route 203 interface Tunnel0 423

interface Async1
 ip address 1.0.0.2 255.255.255.0
 encapsulation ppp
 async default routing
 async mode dedicated
 ppp authentication chap

line aux 0
 rxspeed 38400
 txspeed 38400
```

© copyright Éditions Eyrolles

**Listing B.3. Configuration du routeur R1.**

```
interface Ethernet0
 ip address 10.1.0.1 255.255.255.0

interface Serial0
 encapsulation frame-relay
 frame-relay lmi-type ansi

interface Serial0.1 point-to-point
 ip address 10.255.0.10 255.255.255.248
 frame-relay interface-dlci 102
```

**Listing B.4. Configuration du routeur R2.**

```
interface Ethernet0
 ip address 10.2.0.1 255.255.255.0

interface Serial0
 ip address 10.255.0.9 255.255.255.248
 encapsulation frame-relay
 clockrate 64000
 frame-relay map ip 10.255.0.10 201 broadcast
 frame-relay map ip 10.255.0.11 203 broadcast
 frame-relay lmi-type ansi
```

**Listing B.5. Configuration du routeur R3.**

```
interface Ethernet0
 ip address 10.3.0.1 255.255.255.0

interface Serial0
 encapsulation frame-relay
 clockrate 64000
 frame-relay lmi-type ansi

interface Serial0.1 multipoint
 ip address 10.255.0.11 255.255.255.248
 frame-relay map ip 10.255.0.9 302 broadcast
```

Utiliser l'interface auxiliaire asynchrone pour relier les routeurs en commutateurs Frame Relay, demande une certaine patience car pour certaines versions de l'IOS de Cisco, la connexion peut s'avérer instable. Si le protocole PPP utilisé dans cet exemple pose des problèmes, on peut le remplacer par le protocole SLIP qui est disponible pour les interfaces asynchrones.

La commande pour vérifier l'état des commutateurs Frame Relay, ainsi que ses sorties se trouvent sur les listings B.6 et B.7.

© copyright Éditions Eyrolles

**Listing B.6. Sortie de la commande `show frame-relay route` sur le routeur FR1.**

```
FR1#show frame-relay route
Input Intf   Input Dlci   Output Intf   Output Dlci   Status
Serial0      102          Tunnel0       421           active
Serial1      302          Tunnel0       423           active
Tunnel0      421          Serial0       102           active
Tunnel0      423          Serial1       302           active
```

**Listing B.6. Sortie de la commande `show frame-relay route` sur le routeur FR2.**

```
FR2#show frame-relay route
Input Intf   Input Dlci   Output Intf   Output Dlci   Status
Serial0      201          Tunnel0       421           active
Serial0      203          Tunnel0       423           active
Tunnel0      421          Serial0       201           active
Tunnel0      423          Serial0       203           active
```

© copyright Éditions Eyrolles

# C

# Commandes RSH et RCP sur les routeurs Cisco

Dans cette annexe, nous allons décrire deux commandes très utiles et néanmoins assez ésotériques, disponibles dans l'IOS de Cisco. Il s'agit de RSH et RCP – dérivées de la série des utilitaires de commande à distance ou ***R***(*emote*)-*utilities* développés à l'université de Berkeley. De par de leur origine, ces commandes sont compatibles avec les tâches de fond correspondantes (*daemons*) d'un serveur Unix. Le routeur Cisco peut lui-même exécuter ces mêmes tâches de fond et servir un client RSH ou RCP implémenté dans Unix ou Windows NT.

Pour configurer RSH et RCP sur un routeur Cisco, procédez comme suit :

- Démarrer les *daemons* RSH et RCP par les commandes **ip rcmd rsh-enable** et **ip rcmd rcp-enable**.
- Les deux *daemons* requièrent un certain niveau d'authentification qui permet d'enregistrer aussi bien l'hôte distant que le nom d'utilisateur local sous lequel les commandes RSH et RCP en provenance de cet hôte doivent s'exécuter sur le routeur. Cette authentification se configure par la commande **ip rcmd remote-host** <*nom local utilisateur*>{<*adresse IP distante*>|<*nom hôte distant*>}<*nom utilisateur distant*> [**enable**]. Le deuxième paramètre est, soit l'adresse IP de la machine distante, soit son nom DNS. Le dernier paramètre est le nom sous lequel l'utilisateur distant se connecte à sa machine ; il est passé, avec les autres paramètres, dans les PDU des protocoles RSH et RCP au routeur correspondant. L'option **enable** permet à l'utilisateur distant, une fois authentifié, d'exécuter des commandes en mode exec privilégié Cisco.
- La commande **ip rcmd remote-username** <*nom utilisateur*> enregistre dans le routeur local le nom d'utilisateur sous lequel les commandes RSH et RCP seront envoyées vers la machine distante.

© copyright Éditions Eyrolles

**REMARQUE** Si le nom DNS est utilisé, le routeur doit en posséder une configuration valide, sinon il est préférable de désactiver la consultation DNS par la commande **no ip domain-lookup**, ce qui peut diligenter l'exécution des commandes dans le routeur, surtout dans un environnement d'essai où, normalement, il n'est pas besoin d'un serveur DNS.

Le listing C.1 montre un exemple de configuration des *daemons* RSH et RCP sur un routeur Cisco.

**Listing C.1. Configuration du routeur R1.**

```
username Admin1

ip rcmd rcp-enable
ip rcmd rsh-enable
ip rcmd remote-host Admin1 10.6.0.15 Administrator enable

interface TokenRing0
 ip address 10.6.0.1 255.255.255.0
 ring-speed 16
```

Le listing C.2 montre comment la commande RSH est introduite sur une machine Windows NT pour l'exécuter sur un routeur Cisco distant.

**Listing C.2. Commande `rsh` sur machine Windows NT pour exécution sur le routeur distant R1.**

```
C:\>rsh 10.6.0.1 -l Admin1 -n show ip route

Codes: C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
  D-EIGRP, EX-EIGRP external, O-OSPF, IA-OSPF inter area
  N1-OSPF NSSA external type 1, N2-OSPF NSSA external type 2
  E1-OSPF external type 1, E2-OSPF external type 2, E-EGP
  i-IS-IS, L1-IS-IS level-1, L2-IS-IS level-2, *-candidate
 default
  U-per-user static route, o-ODR
  T-traffic engineered route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.6.0.0 is directly connected, TokenRing0
```

La commande **rsh** sous Windows NT avec l'extension **-l** permet de renseigner le paramètre du nom d'utilisateur sous lequel le routeur distant doit exécuter cette commande. Ce nom (Admin1) est configuré sur le routeur R1 par les commandes **username** et **ip rcmd remote-host** (premier paramètre).

Si ce paramètre est omis dans la commande RSH, Windows NT va le renseigner d'office avec le nom de connexion locale, c'est-à-dire « Administrator », dans ce cas.

© copyright Éditions Eyrolles

Nous pouvons voir sur le listing C.3 que la commande RSH sans nom d'utilisateur valide n'aboutit pas.

**Listing C.3. Commande `rsh` sans nom d'utilisateur.**

```
C:\>rsh 10.6.0.1 -n show ip route
10.6.0.1: Permission denied.
rsh: can't establish connection
```

Nous pouvons utiliser la commande **debug ip tcp rcmd** pour pouvoir suivre le déroulement du protocole RSH, quand une commande échoue. Par exemple, si nous entrons cette commande sur le routeur R1 avant que celui-ci ne reçoive la commande RSH de l'hôte Windows NT, nous verrons ce qui est affiché sur le listing C.4. lors de l'exécution de cette dernière.

**Listing C.4. Sortie de la commande `debug ip tcp rcmd` sur le routeur R1.**

```
R1#debug ip tcp rcmd
RCMD transactions debugging is on
R1#
RCMD: [514 <- 10.6.0.15:1018] recv 1017\0
RCMD: [514 <- 10.6.0.15:1018] recv
 Administrator\0Administrator\0show ip route\0
RCMD: [514 <- 10.6.0.15:1018] recv --
 Administrator 10.6.0.15 Administrator not in trusted hosts
 database
RCMD: [514 -> 10.6.0.15:1018] send <BAD,Permission denied.>\n
```

Nous pouvons constater sur le listing C.4, sans même savoir pourquoi la commande a échoué, que le nom « Administrator » de connexion locale dans l'hôte Window NT n'est pas enregistré dans la base d'authentification du routeur R1 comme utilisateur. Celui-ci doit être configuré selon les directives mentionnées plus haut, par les commandes **username** et **ip rcmd remote-host**.

Une version en mode client est également disponible pour RSH sur les routeurs Cisco. S'il est besoin d'exécuter une commande RSH d'un routeur vers un autre, nous devons entrer le format de commande **rsh** {<*adresse IP distante*|<*nom système distant*>} [/**user** <*nom local utilisateur*>] <*commande routeur*>.

**REMARQUE** Si l'option /user avec un nom d'utilisateur (configuré dans le routeur distant par **username** ou tout autre moyen d'authentification comme RADIUS, TACACS+, etc.) est omise lors de la connexion, le routeur local va se connecter au routeur distant avec son propre nom d'hôte qui lui a été défini par la commande **hostname**.

Essayons de nous connecter par la commande RSH (cf. listing C.5) d'un routeur à un autre ; dans notre exemple, du routeur R2 vers R1, sur le même Token Ring, sans préciser le nom d'utilisateur.

© copyright Éditions Eyrolles

**Listing C.5. Commande `rsh` du routeur R2 vers routeur R1 en échec.**

```
R2#rsh 10.6.0.1 /user Admin1 show ip route
%Permission denied.
```

En examinant la sortie de la commande **debug ip tcp rcmd** sur le listing C.6, nous nous apercevons que le routeur R2 envoie son propre nom d'hôte en tant que nom d'utilisateur sous lequel la commande doit s'exécuter sur le routeur R1 ; ce nom n'étant pas enregistré sur ce dernier, la commande RSH échoue.

**Listing C.6. Sortie de la commande `debug ip tcp rcmd` sur le routeur R1.**

```
R1#debug ip tcp rcmd
RCMD transactions debugging is on
R1#
01:48:07: %SYS-5-CONFIG_I: Configured from console by console
01:48:37: RCMD: [514 <- 10.6.0.2:1016] recv \0
01:48:37: RCMD: [514 <- 10.6.0.2:1016] recv R2\0Admin1\0show
 ip route\0
01:48:37: RCMD: [514 <- 10.6.0.2:1016] recv --
 Admin1 10.6.0.2 R2 not in trusted hosts database
01:48:37: RCMD: [514 -> 10.6.0.2:1016] send
 <BAD,Permission denied.>\n
```

La commande RCP est très utile pour télécharger une image IOS d'un serveur RCP vers le routeur. Cette méthode est plus sûre et plus rapide que de passer par TFTP.

Dans la commande **copy** habituelle il faut remplacer le mot clef **tftp** par **rcp**. Par exemple, si on veut transférer une image de l'IOS du serveur RCP vers la mémoire flash du routeur à mettre à niveau, on peut utiliser la commande **copy rcp flash** du listing C.7.

**Listing C.7. Commande `copy rcp flash` sur le routeur R2.**

```
R2#
R2#copy rcp flash
                         ****  NOTICE  ****
Flash load helper v1.0
This process will accept the copy options and then terminate
the current system image to use the ROM based image for the
copy. Routing functionality will not be available during
that time. If you are logged in via telnet, this connection
will terminate. Users with console access can see the results
of the copy operation.
                         ---- ******** ----
Proceed? [confirm]
Address or name of remote host []?10.6.0.1
Source username [R2]? Admin1
Source filename []? c2500-d-l.120-2a.bin
Destination filename [c2500-d-l.120-2a.bin]?
The returned username is R2(8CF68)
```

© copyright Éditions Eyrolles

**ASTUCE** Il est conseillé de basculer vers la version de IOS qui se trouve sur la ROM (*Read Only Memory*) avant de télécharger en local la nouvelle image de IOS. La commande de basculement dépend de la version du système IOS et du modèle de routeur, mais elle commence toujours par **boot system**. Seuls les paramètres varient d'une version à l'autre ou d'un modèle à l'autre. En version 11.1 du Cisco IOS, sur un modèle de routeur de la série 2500, cette commande, dont la syntaxe est **boot system rom**, doit être introduite en mode de configuration globale. Il faut ensuite sauvegarder la configuration actuelle en mémoire non volatile ou NVRAM par la commande **copy running-config startup-config** ou celle plus ancienne, **write memory**, et redémarrer le routeur.

Si le système IOS de Cisco tourne déjà en version ROM, il n'est pas nécessaire de redémarrer le routeur avant le téléchargement en local de la nouvelle image IOS. Cependant il faut se rappeler que la version ROM de IOS ne possède qu'une fonction de routage rudimentaire. À moins d'avoir le serveur RCP contenant le fichier à télécharger sur le même segment que le routeur à mettre à niveau, celui-ci doit être configuré par la commande **ip default-gateway** [<*adresse IP*>] avec l'adresse IP d'un autre routeur du segment qui servira de relais.

L'utilisation conjointe de la version ROM de Cisco IOS et de la commande RCP s'avère fort utile pour mettre à jour l'image de l'IOS sur un routeur distant *via* un réseau WAN de type Frame Relay, par exemple.

© copyright Éditions Eyrolles

# D

# Horodatage de Ping

La commande **ping** est disponible sur la plupart des systèmes, avec cependant l'inconvénient de ne pas horodater les résultats. L'horodatage peut être très utile pour mesurer des durées – délai de reprise du routeur de secours suite à la défaillance du routeur principal ou délai de commutation vers la ligne de secours, suite à la panne de la ligne principale, par exemple.

Ces deux cas ont été vus dans le chapitre 7 traitant de HSRP (*Hot Standby Router Protocol*) et de la commutation vers la ligne de secours. Nous proposons ci-après un script en langage Perl qui lance la commande **ping** toutes les secondes avec horodatage du résultat. Le listing D.1 contient le programme avec l'option « **-n 1** » qui n'envoie qu'un seul paquet ICMP (cette option varie suivant le système d'exploitation et doit être modifiée en conséquence). La commande **ping** a comme seul argument l'adresse IP destinataire (aucune vérification syntaxique n'étant faite sur l'adresse, celle-ci doit être entrée correctement).

**Listing D.1. Script Perl de la commande `ping` avec horodatage.**

```
# This procedure creates a stamp on the moment it's invoked.
sub tstamp
{
    local($sec,$min,$hour) = localtime;
    return sprintf( "%02i:%02i", $min, $sec );
}

# This variable will be used to invoke the NT ping command
# to ping the remote destination only once. In other
# operating systems the option "-n 1" must be replaced
# with the option that makes ping send only a single
# ICMP packet.
$OS_SPING = 'ping -n 1 ';

# The destination IP address is passed as the first and the
```

© copyright Éditions Eyrolles

```
# only command line argument. I do not perform any command
# line syntax checking, so it's important to pass the
# correct IP address.
$IP_ADDR = $ARGV[0];
$T_PREV = '';

while (1)
{
    # Wait until one second expires
    while( $T_PREV eq &tstamp ) {}

    # Parse ping output
    ( $tmp1,$tmp2, $tmp3, $P_RES ) =
        split( "\n", `$OS_SPING $IP_ADDR` );

    # Print time stamp
    print "[". &tstamp ."] ";

    # Print the fourth line of ping output
    print $P_RES . "\n";

    # Refresh the time stamp
    $T_PREV = &tstamp;
}
```

Comme vous avez pu le constater, le programme tourne sur une boucle infinie. Pour l'arrêter, utilisez la combinaison de touches Ctrl+Echap.

© copyright Éditions Eyrolles

# E

# Utilisation de Windows NT en tant que machine hôte

L'utilisation d'une machine Windows NT version 4.x pour tester les différentes configurations de routeurs peut s'avérer très pratique. Ce système d'exploitation permet d'avoir plusieurs cartes d'interface réseau et peut dérouler des protocoles comme RIP, OSPF, etc. Cependant, il faut savoir que Windows NT va installer autant de routes pointant vers la passerelle par défaut qu'il a de cartes réseau. Celles-ci – qui peuvent être visualisées par la commande **netstat rn** – peuvent semer une certaine confusion quant à la route à choisir. Il est donc utile d'installer des routes spécifiques compatibles avec vos procédures d'essai. Par exemple si la machine comporte deux cartes réseau, avec les adresses IP respectives 10.1.1.10/24 et 172.16.1.10/24, il est bon d'ajouter deux routes par les commandes suivantes :

```
route add 10.0.0.0 mask 255.0.0.0 10.1.1.1 metric 1
route add 172.16.0.0 mask 255.255.0.0 172.16.1.1 metric 1
```

© copyright Éditions Eyrolles

# F

# Aide-mémoire pour les routeurs Cisco

## Interface de ligne de commande (CLI)

### Mouvements du curseur

Les combinaisons de touches sont les suivantes pour pointer le curseur vers :

- le début de ligne par Ctrl+A ;
- la fin de ligne par Ctrl+E ;
- le caractère précédent par Ctrl+B (ou flèche gauche) ;
- le caractère suivant par Ctrl+F (ou flèche droite) ;
- le mot précédent par Echap+B ;
- le mot suivant par Echap+F.

Toutes ces touches sont actives par défaut ; sinon la commande **terminal editing** permet de les activer.

### Fonction historique

La CLI (*Commande Line Interface*) permet de mémoriser les commandes entrées auparavant pour pouvoir les réutiliser après modification éventuelle. Pour accéder à la commande précédente et celle la plus récente, la flèche haute ou Ctrl+P et la flèche basse ou Ctrl+N sont à utiliser, respectivement. La mémoire des commandes contient par défaut 10 entrées au maximum. On peut changer sa taille par session ou de manière définitive. Pour un changement par session, entrer la commande **terminal history size** suivie du nombre d'entrées à mémoriser. Pour un changement définitif, en mode de configuration d'interface, il faut introduire la commande **history size** suivie du nombre voulu.

© copyright Éditions Eyrolles

Par exemple :

```
router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
router(config)#line vty 0 4
router(config-line)#history size 50
```

### Fonctions d'aide

La CLI possède des fonctions d'aide à utiliser comme suit :

Accoler le caractère « ? » à un début de commande tronquée permet d'avoir la liste de tous les mots complets qui commencent par ce début. Ce début de commande est ensuite réaffiché avec le curseur pointant sur la position ou se trouvait « ? ». Par exemple :

```
router#show co?
compress  configuration  context  controllers

router#show co_
```

Le caractère « ? » précédé d'une espace après une partie de la commande, affiche tous les arguments possibles qui s'y rapportent. La partie incomplète est ensuite réaffichée avec le curseur pointant sur la position où se trouvait « ? ». Par exemple :

```
router#show ip eigrp ?
  interfaces  IP-EIGRP interfaces
  neighbors   IP-EIGRP neighbors
  topology    IP-EIGRP Topology Table
  traffic     IP-EIGRP Traffic Statistics

router#show ip eigrp _
```

Appuyer sur la touche de tabulation au milieu d'un début de mot permet de le compléter. Si ce début possède plusieurs terminaisons possibles, il sera complété jusqu'au tronc commun. Par exemple :

```
router#show ip ac<TAB>
router#show ip acc_
```

ou

```
router#conf<TAB>
router#configure_
```

### Touches et commandes de remontée vers les modes supérieurs

Pour remonter vers les modes supérieurs de la CLI, on peut se servir des mots ou des touches suivantes:

En mode de configuration autre que celui de configuration globale, la combinaison de touches Ctrl+Z ou l'entrée de la commande **end** provoque le passage du mode courant à celui d'EXEC privilégié.

En mode de configuration autre que celui de configuration globale, l'entrée du mot **exit** fait passer du mode courant à celui du niveau supérieur qui, dans la plupart des cas, est le mode de configuration globale.

© copyright Éditions Eyrolles

La combinaison de touches Ctrl+Z après une commande, lance l'exécution de celle-ci et bascule en mode EXEC privilégié. Par exemple, en mode de configuration d'interface et après avoir tapé **ip address 10.0.0.1 255.0.0.0**, la frappe de Ctrl+Z exécute cette commande et bascule en mode EXEC privilégié. Pour passer à ce mode sans exécuter la commande, appuyer sur Ctrl+C.

## Fonctions du terminal

### Sortie longue

Pour inhiber la sortie en mode paginé (qui se termine par la marque « ***more*** » pour indiquer que d'autres lignes sont en attente d'affichage, et qu'on peut visualiser soit en appuyant sur la touche entrée pour voir la suite ligne par ligne, soit en appuyant sur la barre espace pour une visualisation page par page), la commande **terminal length 0** permet d'afficher toute la sortie dans la foulée. Cette commande n'est effective que par session. On peut retourner en mode paginé par la même commande, en précisant cette fois-ci le nombre de lignes à afficher par page. L'autre moyen, c'est de quitter la session et de se reconnecter.

### Diriger la sortie de la commande `debug`

Pour visualiser les résultats d'une commande **debug** à partir d'une connexion réseau telnet (ou de même type tels que rlogin, PAD, etc.), il faut introduire la commande **terminal monitor**. Par défaut, la sortie de la commande **debug** n'est affichée que sur la console directement connectée au routeur.

### Afficher les sessions et leur coupures éventuelles

Utilisez la commande **show users all** pour afficher les sessions actives :

```
router#show users all
    Line     User      Host(s)              Idle Location
   0 con 0                                00:00:00
   1 aux 0                                00:00:00
*  2 vty 0   you       idle                 00:00:00 10.0.1.111
   3 vty 1   badguy    idle                 00:00:02 10.1.2.120
   4 vty 2                                00:00:00
   5 vty 3                                00:00:00
   6 vty 4                                00:00:00
```

Pour déconnecter un utilisateur indésirable, la commande à utiliser est **clear line** suivie du nom de la ligne et de son numéro (en italique, ci-dessus). Elle s'exécute comme suit :

```
router#clear line vty 1
[confirm]<ENTER>
 [OK]
router#
```

### Activation et désactivation de la fonction DNS

Si l'on commet une erreur de saisie lorsque l'on introduit une commande, le routeur suppose qu'il s'agit d'un nom de connexion pour une session par terminal, de type telnet. Si le routeur n'est pas configuré avec un serveur DNS, il va vainement chercher à consulter un serveur DNS pour traduire la mauvaise commande en adresse IP, et il peut s'écouler un certain temps avant

© copyright Éditions Eyrolles

que l'échéance de temporisation de cette consultation inopportune arrive à terme, surtout s'il n'est pas possible d'annuler la commande par les touches Ctrl+Maj+6 ou Ctrl+Ctrl. Le temps d'affichage des commandes **ping** et **traceroute** peut aussi être allongé par la consultation DNS (pour la conversion d'adresses IP en noms), si cette fonction n'est pas configurée dans le routeur. Pour indiquer à celui-ci que la fonction DNS est désactivée, il faut entrer la commande **no ip domain-lookup** (en mode de configuration globale).

Pour configurer la fonction DNS, saisissez la commande inverse **ip domain-lookup** en spécifiant le nom du domaine avec la commande **ip domain-name** <*nom de domaine*>, et un (ou des) serveur(s) DNS avec **ip name-server** <*adresse IP du serveur*>.

Malheureusement, la commande **terminal no ip domain-lookup**, valable pour une session donnée**,** ne désactive la fonction DNS que pour les commandes **show**.

## Commandes `show` utiles

Le tableau A donne la liste des commandes **show** qui sont particulièrement utiles pour avoir rapidement des informations liées à IP dans un routeur.

**Tableau A. Commandes `show` utiles.**

| Commande | Action |
|---|---|
| **show interfaces** | Affiche l'état de toutes les interfaces. |
| **show interfaces <*interface*>** | Affiche l'état de l'interface dont le paramètre inclut son nom et son numéro. |
| **show ip interface** | Affiche les informations relatives à IP pour toutes les interfaces. |
| **show ip interface <*interface*>** | Affiche les informations relatives à IP pour l'interface donnée en paramètre avec son nom et son numéro. |
| **show ip route** | Affiche la table de routage en entier. |
| **show ip route <préfixe réseau>** | Affiche les informations de routage concernant le préfixe donné en paramètre qui peut être soit un sous-réseau, soit un réseau à classe, auquel cas les informations sur tous les sous-réseaux de ce réseau, s'ils existent, sont affichées. Sinon, seules les informations sur le réseau, sont affichées. Si aucune information n'est disponible pour le paramètre, la sortie de la commande affiche % Network not in table. |
| **show ip route <*source*>** | Affiche les informations de routage en provenance de la source qui correspond à l'un des mots clefs du tableau B. |
| **show ip route summary** | Affiche le sommaire des informations de la table de routage. |
| **show ip protocols** | Affiche les informations sur les protocoles de routage IP configurés. |
| **show startup-config ou show config (ancienne version)** | Affiche la configuration enregistrée en mémoire non volatile (NVRAM). |
| **show running-config ou write terminal (ancienne version)** | Affiche la configuration courante en mémoire vive (RAM) qui peut être différente de celle de la NVRAM, si certaines commandes entrées récemment n'y ont pas encore été enregistrées. |

© copyright Éditions Eyrolles

**Tableau B. Mots clefs disponibles pour le paramètre** <*source*> **de la commande `show ip route`.**

| Mot clef | Source de l'information de routage |
|---|---|
| **bgp** | Protocole de routage inter-domaines ou BGP (Border Gateway Protocol) |
| **connected** | Connecté |
| **egp** | Protocole de routage inter-domaine ou EGP (Exterior Gateway Protocol) |
| **eigrp** | Protocole de routage intra-domaine amélioré ou EIGRP (Enhanced IGRP) |
| **igrp** | Protocole de routage intra-domaine ou IGRP (Interior Gateway Routing Protocol) |
| **isis** | Protocole de routage intra-domaine ISO ou IS-IS (Intermediate System to Intermediate System) |
| **odr** | Routes d'aire confinée (stub) sur demande |
| **ospf** | Protocole ouvert au chemin le plus court ou OSPF (Open Shortest Path First) |
| **rip** | Protocole d'information de routage ou RIP (Routing Information Protocol) |
| **static** | Routes statiques |

## Outils de dépannage de réseau

### Ping

La commande **ping** lancée sans paramètres ou suivie du mot clef **ip**, permet de préciser des options très utiles. La plus importante étant l'adresse source pour les paquets **ping**, comme dans l'exemple ci-dessous :

```
router#ping ip
Target IP address: 10.1.0.111
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.23.34.101
...
Sending 5, 100-byte ICMP Echos to 10.1.0.111,
 timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip
 min/avg/max = 4/8/12 ms
```

**REMARQUE** Certains des paramètres optionnels ont été remplacés par des points de suspension (…).

Deux autres options de la commande **ping** sont la taille du datagramme (*datagram size*) et le nombre d'envois (*repeat count*) qui permettent de générer un trafic intense à des fins de test.

### Traceroute

La commande **traceroute** sans paramètres ou suivie du mot clef **ip**, propose le même dialogue d'options à renseigner que celui de la commande **ping**.

© copyright Éditions Eyrolles

### Telnet

La commande **telnet** peut parfois servir d'outil de dépannage si on lui ajoute le mot clef **/source-interface** avec le paramètre <*interface*> pour spécifier l'adresse IP source de la connexion.

## Adressage IP

Le tableau C fournit la relation entre longueur de préfixe (ou masque) de sous-réseau et le nombre d'hôtes qu'elle peut contenir.

**Tableau C. Relation entre longueur de préfixe sous-réseau et le nombre d'hôtes qu'elle peut contenir.**

| Nombre d'hôtes | Longueur masque sous-réseau | Masque sous-réseau |
|---|---|---|
| **1** | /32 | 255.255.255.255 |
| **Jusqu'à 2** | /30 | 255.255.255.252 |
| **Jusqu'à 6** | /29 | 255.255.255.248 |
| **Jusqu'à 14** | /28 | 255.255.255.240 |
| **Jusqu'à 30** | /27 | 255.255.255.224 |
| **Jusqu'à 62** | /26 | 255.255.255.192 |
| **Jusqu'à 126** | /25 | 255.255.255.128 |
| **Jusqu'à 254** | /24 | 255.255.255.0 |
| **Jusqu'à 510** | /23 | 255.255.254.0 |
| **Jusqu'à 1022** | /22 | 255.255.252.0 |
| **Jusqu'à 2046** | /21 | 255.255.248.0 |

## Routage IP

### Distance administrative des sources d'informations de routage

On peut assigner une nouvelle valeur de distance administrative pour les sources de mises à jour par la commande **distance** <*valeur de distance*> [<*adresse IP source/masque générique*>], en mode de configuration routeur (**router rip**). La nouvelle distance administrative s'appliquera à toutes les routes pointant vers les préfixes réseau annoncés par la source dont l'adresse IP correspond au deuxième paramètre de la commande. Si ce paramètre (optionnel) est omis, la nouvelle distance s'appliquera à toutes les mises à jour reçues, quelle que soit leur origine.

Si on renseigne le paramètre <*valeur de distance*> par 255, les routes diffusées par la source correspondante, ne seront jamais inscrites dans la table de routage, mais tout simplement ignorées.

La distance administrative des routes statiques peut aussi être changée par la commande **ip route** en renseignant son dernier paramètre optionnel [<*distance*>] avec la valeur voulue.

La distance administrative des routes d'interfaces connectées et celle des routes agrégées de EIGRP ne peuvent pas être modifiées.

© copyright Éditions Eyrolles

Le tableau D donne la valeur de distance administrative de toutes les sources d'informations de routage disponibles dans le système IOS de Cisco.

**Tableau D. Valeur par défaut de la distance administrative.**

| Origine de la route | Distance |
|---|---|
| **Interface connectée** | 0 |
| **Route statique** | 1 |
| **IGRP amélioré (route agrégée)** | 5 |
| **BGP externe** | 20 |
| **IGRP amélioré (interne)** | 90 |
| **IGRP amélioré (externe)** | 170 |
| **IGRP** | 100 |
| **OSPF** | 110 |
| **IS-IS** | 115 |
| **RIP** | 120 |
| **EGP** | 140 |
| **BGP interne** | 200 |
| **Inconnu** | 255 |

### Les métriques de IGRP et EIGRP

La formule que IGRP utilise pour calculer la métrique de chaque route est la suivante :

$$M_{IGRP} = k1*B_{IGRP} + \frac{k2*B_{IGRP}}{256 - L} + k3*D_{IGRP}\frac{k5}{R + k4}$$

$B_{IGRP}$ est le débit IGRP du chemin, calculé selon la formule ci-dessous :

$$B_{IGRP} = \frac{10^7}{B_{MIN}}$$

$B_{MIN}$ est le débit logique minimal du chemin exprimé en Kbit/s (Kilobits par seconde). $D_{IGRP}$ est le délai du chemin qui est égal à la somme des délais de tous les segments qui le constituent, exprimé en unités de 10 µs. *L* est la charge de l'interface correspondante, exprimée par un chiffre de 1 (minimum) à 255 (100 %). R est le degré de fiabilité du segment auquel est reliée l'interface, dont la valeur est exprimée dans la même fourchette que *L*. Les facteurs de pondération, k1, k2, k3, k4 et k5 peuvent être configurés administrativement ; leur valeur par défaut se trouve dans le tableau E.

Si on applique la valeur par défaut pour les facteurs k, la formule de calcul de métrique de IGRP est réduite à la suivante :

$$M_{IGRP} = B_{IGRP} + D_{IGRP}$$

La métrique calculée par EIGRP est basée sur la même formule que celle de IGRP, avec une multiplication du résultat par 256, ce qui donne :

$$M_{EIGRP} = M_{IGRP} * 256$$

© copyright Éditions Eyrolles

La redistribution de routes entre IGRP et EIGRP se fait sans que leurs métriques soient revalorisées, mais multipliées par un facteur de 256 sauvegardant ainsi leurs valeurs accumulées dans leurs domaines de routage respectifs.

**Tableau E. Valeur par défaut des facteurs de pondération.**

| Facteur | Valeur par défaut |
|---|---|
| K1 | 1 |
| K2 | 0 |
| K3 | 1 |
| K4 | 0 |
| K5 | 0 |

### La métrique de RIP

Dans RIP la métrique d'une route se calcule en nombre de sauts. Une fois celle-ci apprise *via* ses voisins, le routeur l'annonce à son tour avec une métrique augmentée de 1.

### La métrique de OSPF

Le protocole OSPF calcule la métrique d'une route en cumulant le coût de tous les segments qui la constituent, selon la formule suivante.

$$C = \frac{10^8}{B}$$

*B* désigne le débit logique de l'interface mesuré en bits par seconde (bit/s).

### Règles de routage

- Le routeur n'utilise les métriques que dans le cadre d'un protocole de routage particulier. Si un protocole à vecteur de distance reçoit plusieurs messages de mise à jour pour le même préfixe, la route à la meilleure métrique prévaudra sur les autres pour être inscrite dans la table de routage.
- Si l'information de routage provient de plusieurs origines (par exemple, les protocoles de routage dynamique, les routes statiques et les routes d'interfaces connectées) pour la même destination, la route installée dans la table sera celle dont l'origine possède la plus petite valeur de distance administrative au détriment des autres.
- Le routeur utilise l'algorithme de recherche par la correspondance la plus longue pour trouver la meilleure route vers une destination, dans sa table de routage.
- Un protocole à vecteur de distance n'annonce pas une route qu'il n'a pas installée dans la table de routage.
- Dans un protocole à état des liens, le déroulement de l'algorithme de Dijkstra pour le calcul du plus court chemin permet de remplir la table de routage.

© copyright Éditions Eyrolles

# Index



© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

## S



© copyright Éditions Eyrolles

© copyright Éditions Eyrolles

Retrouvez nos eBooks sur:

www.ebooks.eyrolles.com

Également disponibles :




Distribution numérique par
www.GiantChair.com