Configuration et administration de réseaux Oracle® Solaris 11.1

Référence : E36455–01
Octobre 2012

ORACLE®

130327@25097

Table des matières

Préface

Bienvenue dans *Configuration et administration de réseaux Oracle Solaris 11.1*. Ce manuel fait partie de la série *Establishing An Oracle Solaris 11.1 Network* qui couvre les thèmes et procédures nécessaires pour configurer les réseaux Oracle Solaris. Ce manuel part du principe que vous avez déjà installé Oracle Solaris. Vous devez être prêt à configurer votre réseau ou tout logiciel de gestion de réseau requis.

Utilisateurs de ce manuel

Ce document s'adresse aux administrateurs de systèmes réseau exécutant Oracle Solaris. Pour utiliser ce manuel, vous devez avoir au moins deux ans d'expérience en administration de systèmes UNIX. Une formation en administration de systèmes UNIX peut se révéler utile.

Accès à Oracle Support

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Conventions typographiques

Le tableau ci-dessous décrit les conventions typographiques utilisées dans ce manuel.

TABLEAU P–1 Conventions typographiques

Type de caractères	Description	Exemple
`AaBbCc123`	Noms des commandes, fichiers et répertoires, ainsi que messages système.	Modifiez votre fichier `.login`. Utilisez `ls -a` pour afficher la liste de tous les fichiers. `nom_machine% Vous avez reçu du courrier.`

TABLEAU P–1 Conventions typographiques *(Suite)*

Type de caractères	Description	Exemple
AaBbCc123	Ce que vous entrez, par opposition à ce qui s'affiche à l'écran.	`nom_machine%` **`su`** `Mot de passe :`
aabbcc123	Paramètre fictif : à remplacer par un nom ou une valeur réel(le).	La commande permettant de supprimer un fichier est `rm` *nom_fichier*.
AaBbCc123	Titres de manuel, nouveaux termes et termes importants.	Reportez-vous au chapitre 6 du *Guide de l'utilisateur*. Un *cache* est une copie des éléments stockés localement. *N'enregistrez pas* le fichier. **Remarque :** en ligne, certains éléments mis en valeur s'affichent en gras.

Invites de shell dans les exemples de commandes

Le tableau suivant présente l'invite système UNIX par défaut et l'invite superutilisateur pour les shells faisant partie du SE Oracle Solaris. L'invite système par défaut qui s'affiche dans les exemples de commandes dépend de la version Oracle Solaris.

TABLEAU P–2 Invites de shell

Shell	Invite
Bash shell, korn shell et bourne shell	`$`
Bash shell, korn shell et bourne shell pour superutilisateur	`#`
C shell	`nom_machine%`
C shell pour superutilisateur	`nom_machine#`

◆ ◆ ◆ CHAPITRE 1

Planification du développement du réseau

Ce chapitre présente brièvement les différents éléments à prendre en considération lors de la configuration du réseau. Ces problématiques vous aideront à déployer votre réseau de manière organisée et économique. Notez que cet ouvrage ne détaille pas la planification du réseau. Seules des indications d'ordre général sont fournies.

Cet ouvrage est destiné à des lecteurs qui connaissent la terminologie et les concepts de base de la mise en réseau. Pour une description de l'implémentation de la suite de protocoles TCP/IP dans Oracle Solaris 11, reportez-vous à la section "Pile réseau dans Oracle Solaris" du manuel *Introduction à la mise en réseau d'Oracle Solaris 11*.

Planification réseau (liste des tâches)

Le tableau suivant répertorie les différentes tâches de planification de la configuration réseau.

Tâche	Description	Référence
Identification des conditions matérielles requises pour la topologie réseau planifiée	Déterminez les types d'équipement dont vous avez besoin pour votre site réseau.	"Détermination du matériel réseau" à la page 12 Pour obtenir des informations sur un type d'équipement spécifique, reportez-vous à la documentation du constructeur de l'équipement.
Identification du type d'adresses IP à utiliser pour obtenir des adresses IP enregistrées	Choisissez entre le déploiement d'un réseau IPv4 et un réseau IPv6, ou un réseau qui utilise ces deux types d'adresses IP. Obtenez des adresses IP uniques pour communiquer sur les réseaux publics du réseau Internet.	"Choix du format d'adressage IP du réseau" à la page 13 "Obtention du numéro IP du réseau" à la page 15.

Tâche	Description	Référence
Identification d'un schéma de noms qui identifie les hôtes du réseau et du service de noms à utiliser	Créez une liste de noms à affecter aux systèmes du réseau et décidez des bases de données à utiliser (NIS, LDAP, DNS ou les bases de données réseau du répertoire local `/etc`).	"Administration des noms d'hôtes" à la page 16 "Sélection d'un service de noms et d'un service d'annuaire" à la page 16
(Facultatif) Création de sous-divisions administratives et élaboration d'une stratégie pour les sous-réseaux	Décidez si le site requiert une division du réseau en sous-réseaux pour servir les sous-divisions administratives.	"Utilisation de sous-réseaux" à la page 17
Détermination de l'emplacement auquel positionner les routeurs dans le réseau	Si le réseau est étendu et, par conséquent, requiert des routeurs, créez une topologie réseau prenant en charge ces derniers.	"Planification des routeurs du réseau" à la page 18
Création ou non de réseaux virtuels dans le schéma de configuration réseau complet	Vous devrez peut-être créer des réseaux virtuels dans un système afin de réduire l'empreinte matérielle sur le réseau.	*Utilisation de réseaux virtuels dans Oracle Solaris 11.1*

Détermination du matériel réseau

Le nombre de systèmes à prendre en charge modifie la manière de configurer le réseau. Votre organisation peut avoir besoin d'un petit réseau de plusieurs douzaines de systèmes autonomes résidant dans un même bâtiment et au même étage. Vous pouvez aussi configurer un réseau comprenant plus de 1 000 systèmes situés dans différents bâtiments. Cette configuration requiert une division supplémentaire du réseau en sous-divisions appelées *sous-réseaux*.

Dans le cadre de la planification, vous devez prendre les décisions suivantes concernant le matériel :

- Définir la topologie réseau, la disposition et les connexions du matériel réseau
- Définir le type et le nombre de systèmes hôtes que votre réseau peut prendre en charge, y compris les serveurs qui peuvent être requis
- Définir les périphériques réseau à installer sur ces systèmes
- Définir le type de média réseau à utiliser (Ethernet, etc.)
- Définir si des ponts ou routeurs doivent étendre ce média ou connecter le réseau local à des réseaux externes

Remarque – Pour une description du fonctionnement des routeurs, reportez-vous à la section "Planification des routeurs du réseau" à la page 18. Pour une présentation des ponts, reportez-vous à la section "Présentation du pontage" du manuel *Gestion des performances du réseau Oracle Solaris 11.1*

Choix du format d'adressage IP du réseau

Lors de la planification du schéma d'adressage du réseau, tenez compte des facteurs suivants :

- Le type d'adresse IP à employer : IPv4 ou IPv6
- Le nombre de systèmes potentiels sur le réseau
- Le nombre de systèmes multiréseau ou routeurs, qui requièrent plusieurs cartes d'interface réseau avec leur adresse IP
- Si des adresses privées doivent être utilisées sur le réseau
- Si les pools d'adresses IPv4 doivent être gérés par un serveur DHCP

En résumé, le type d'adresse IP inclut les éléments suivants :

Adresses IPv4

Ces adresses 32 bits correspondent au format d'adressage IP conçu pour TCP/IP. L'IETF a ensuite développé des adresses *CIDR* (Classless Inter-Domain Routing, routage inter-domaine sans classe) dans le but de résoudre à court ou moyen terme le problème d'épuisement des adresses IPv4 et de remédier au manque de capacité des tables de routage Internet.

Pour plus d'informations, reportez-vous aux ressources suivantes :

- Internet Protocol DARPA Internet Program Protocol Specification (`http://tools.ietf.org/html/rfc791`)
- Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan (`http://tools.ietf.org/html/rfc4632`)

Le tableau suivant fournit les sous-réseaux au format décimal avec points ainsi que sous la forme d'une notation CIDR.

TABLEAU 1–1 Préfixes CIDR et leurs équivalents décimaux

Préfixe de réseau CIDR	Equivalent en numérotation décimale avec points	Adresses IP disponibles
/19	255.255.224.0	8,192
/20	255.255.240.0	4,096

TABLEAU 1–1 Préfixes CIDR et leurs équivalents décimaux *(Suite)*

Préfixe de réseau CIDR	Equivalent en numérotation décimale avec points	Adresses IP disponibles
/21	255.255.248.0	2,048
/22	255.255.252.0	1,024
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32

Adresses DHCP

Le protocole DHCP (Dynamic Host Configuration Protocol, protocole de configuration dynamique d'hôte) permet à un système de recevoir à l'initialisation les informations de configuration d'un serveur DHCP, notamment une adresse IP. Les serveurs DHCP tiennent à jour des pools d'adresses IP à partir desquels attribuer des adresses aux clients DHCP. Cela permet à un site DHCP d'utiliser un pool d'adresses IP plus petit que celui qui serait nécessaire si tous les clients possédaient une adresse IP permanente. Vous pouvez configurer le service DHCP afin de gérer les adresses IP de votre site ou une partie des adresses. Pour plus d'informations, reportez-vous au Chapitre 1, “A propos de DHCP (présentation)” du manuel *Utilisation de DHCP dans Oracle Solaris 11.1*.

Adresses IPv6

Les adresses IPv6 128 bits fournissent un espace d'adressage plus étendu que IPv4. Comme les adresses IPv4 au format CIDR, les adresses IPv6 n'ont pas de classe et utilisent des préfixes pour désigner la partie de l'adresse définissant le réseau du site. Pour plus d'informations sur l'adressage IPv6, reportez-vous à la section Internet Protocol, Version 6 (IPv6) Specification (`http://tools.ietf.org/html/rfc2460`).

Adresses privées et préfixes de documentation

L'IANA a réservé un bloc d'adresses IPv4 et un préfixe de site IPv6 à utiliser sur les réseaux privés. Ces adresses privées sont utilisées pour le trafic réseau au sein d'un réseau privé. Ces adresses sont également utilisées dans la documentation

Le tableau suivant répertorie les plages d'adresses IPv4 privées et des masques de réseau respectifs.

Plage d'adresses IPv4	Masque de réseau
10.0.0.0 - 10.255.255.255	10.0.0.0
172.16.0.0 - 172.31.255.255	172.16.0.0
192.168.0.0 - 192.168.255.255	192.168.0.0

Pour les adresses IPv6, le préfixe `2001:db8::/32` est un préfixe IPv6 spécial utilisé spécifiquement dans les exemples de documentation. Les exemples de ce manuel utilisent des adresses IPv4 privées et le préfixe de documentation IPv6 réservé.

Obtention du numéro IP du réseau

Un réseau IPv4 se définit à l'aide d'un numéro de réseau IPv4 et d'un *masque de réseau*. Un réseau IPv6 est défini par son *préfixe de site* et s'il dispose d'un sous-réseau, par son *préfixe de sous-réseau*.

Pour que le réseau privé communique avec des réseaux externes du réseau Internet, vous devez demander un numéro d'IP enregistré pour votre réseau auprès de l'organisation adéquate. Cette adresse devient le numéro de réseau de votre schéma d'adressage IPv4 ou le préfixe de site de votre schéma d'adressage IPv6.

Les fournisseurs d'accès Internet (FAI) procurent des adresses IP pour les réseaux à un coût dépendant du niveau de service assuré. Comparez les offres de divers FAI afin de déterminer celui qui fournit le service le plus adéquat pour votre réseau. Les FAI offrent généralement des adresses allouées dynamiquement ou des adresses IP statiques aux entreprises. Certains FAI proposent à la fois des adresses IPv4 et IPv6.

Si le site est un FAI, vous pouvez obtenir les blocs d'adresses IP pour vos clients auprès de l'IR (Internet Registry, registre Internet) correspondant à votre environnement linguistique. L'IANA (Internet Assigned Numbers Authority, autorité de numéros assignés sur Internet) est actuellement responsable de la délégation des adresses IP enregistrées aux IR dans le monde entier. Chaque IR possède des modèles et des informations d'enregistrement dédiés à l'environnement linguistique assuré par l'IR. Pour plus d'informations sur l'IANA et les IR, reportez-vous à la page des services d'adresse IP de l'IANA (`http://www.iana.org/ipaddress/ip-addresses.htm`).

Attribution de noms aux entités du réseau

Les protocoles TCP/IP détectent un système sur le réseau à l'aide de son adresse IP. Cependant, un nom d'hôte permet d'identifier plus facilement les systèmes que les adresses IP. Par conséquent, les protocoles TCP/IP (et Oracle Solaris) nécessitent à la fois l'adresse IP et le nom d'hôte pour identifier de manière unique le système.

Dans le cadre de TCP/IP, un réseau correspond à un ensemble d'entités nommées. Un hôte correspond à une entité possédant un nom. Un routeur correspond à une entité possédant un nom. Le réseau correspond à une entité possédant un nom. Vous pouvez également attribuer un nom à un groupe ou service dans lequel le réseau est installé, ainsi qu'à une division, une région ou une société. Théoriquement, la hiérarchie de noms utilisée pour identifier un réseau est illimitée. Le nom de domaine identifie un *domaine*.

Administration des noms d'hôtes

Planifiez un schéma de nommage pour les systèmes inclus dans le réseau. Pour les systèmes faisant office de serveurs et possédant plusieurs NIC, au moins un nom d'hôte associé à l'adresse IP de son interface réseau principale doit être fourni.

Vous ne pouvez pas attribuer le même nom d'hôte à deux ordinateurs du réseau. Par conséquent, le nom d'hôte doit être unique à chaque système. Cependant, un hôte ou un système avec son nom unique assigné peut posséder plusieurs adresses IP.

Lors de la planification du réseau, dressez la liste des adresses IP et des noms d'hôtes associés afin d'en faciliter l'accès lors des processus de configuration. Cette liste permet de vérifier que chaque nom d'hôte est unique.

Sélection d'un service de noms et d'un service d'annuaire

Dans Oracle Solaris, vous pouvez sélectionner parmi trois types de services de noms : fichiers locaux, NIS et DNS. Les services de noms mettent à jour d'importantes informations sur les machines du réseau, par exemple les noms d'hôtes, les adresses IP, les adresses Ethernet, etc. Vous pouvez également utiliser le service d'annuaire LDAP en plus ou à la place d'un service de noms. Pour une présentation des services de noms sous Oracle Solaris, reportez-vous à la Partie I, “A propos des services d’annuaire et de noms” du manuel *Utilisation des services de noms et d’annuaire dans Oracle Solaris 11.1*.

Pendant l'installation du SE, vous devez fournir le nom d'hôte et l'adresse IP de votre serveur, de vos clients ou de votre système autonome. Le programme d'installation ajoute ces informations à la base de données `hosts` utilisée par le service réseau.

La configuration des bases de données réseau est d'une importance capitale. Par conséquent, vous devez choisir le service de noms à utiliser au cours du processus de planification réseau. En outre, l'utilisation des services de noms affecte également l'organisation du réseau en un domaine administratif.

Vous pouvez choisir parmi les services de noms suivants :

- NIS ou DNS : les services de noms NIS et DNS maintiennent des bases de données réseau sur divers serveurs du réseau. *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1* décrit ces services de noms et la configuration des bases de données. En outre, ce manuel explique en détail les concepts d'espace de noms et de domaine administratif.
- Fichiers locaux : si vous n'implémentez pas NIS, LDAP ou DNS, le réseau utilise des *fichiers locaux* pour fournir le service de noms. Le terme "fichiers locaux" fait référence à la série de fichiers du répertoire `/etc` utilisé par les bases de données réseau. Sauf indication contraire, les procédures de ce manuel partent du principe que vous utilisez des fichiers locaux comme service de noms.

Remarque – Si vous décidez d'utiliser des fichiers locaux en tant que service de noms pour le réseau, vous pouvez configurer plus tard un autre service de noms.

Noms de domaine

De nombreux réseaux organisent leurs hôtes et routeurs selon une hiérarchie de domaines administratifs. Si vous utilisez le service de noms NIS ou DNS, vous devez sélectionner pour l'organisation un nom de domaine unique au monde. Pour vérifier que le nom de domaine est unique, enregistrez-le auprès de l'InterNIC. Si vous souhaitez utiliser DNS, vous devez également enregistrer votre nom de domaine auprès de l'InterNIC.

La structure des noms de domaine est hiérarchique. En général, tout nouveau domaine se place sous un domaine existant associé. Par exemple, le nom de domaine d'une filiale peut se placer sous le nom de domaine de la maison mère. Si le nom de domaine n'a pas d'autre relation, une organisation peut placer son nom de domaine directement sous l'un des domaines supérieurs existants, tels que `.com`, `.org`, `.edu`, `.gov` etc.

Utilisation de sous-réseaux

L'utilisation de sous-réseaux est liée au fait que les sous-divisions administratives doivent faire face à des problèmes de taille et de contrôle. A mesure que les nombres d'hôtes et de serveurs augmentent, la gestion du réseau devient de plus en plus complexe. La création de divisions administratives et l'utilisation de sous-réseaux simplifient la gestion d'un réseau complexe. La configuration de sous-divisions administratives pour le réseau dépend des facteurs ci-dessous :

- **Taille du réseau**

Les sous-réseaux sont également utiles dans le cas d'un petit réseau, dont les sous-divisions s'étendent sur une large zone géographique.

- **Besoins courants des groupes d'utilisateurs**

  Par exemple, un réseau peut résider entièrement dans un bâtiment et prendre en charge des machines relativement peut nombreuses. Ces machines sont réparties en plusieurs sous-réseaux. Chaque sous-réseau prend en charge des groupes d'utilisateurs ayant des besoins différents. Dans cet exemple, il serait judicieux de créer une sous-division administrative par sous-réseau.

Planification des routeurs du réseau

Il existe deux types d'entités TCP/IP sur un réseau : les hôtes et les routeurs. Tout réseau doit contenir des hôtes, mais les routeurs ne sont pas toujours requis. La topologie physique du réseau détermine si des routeurs sont requis. Cette section présente les concepts de routage et de topologie réseau. Ces concepts sont importants pour l'ajout d'un réseau à l'environnement réseau existant.

Remarque – Pour plus d'informations et des tâches de configuration des routeurs sur les réseaux IPv4, reportez-vous à la section "Configuration des composants système sur le réseau" à la page 35. Pour obtenir les détails de la configuration des routeurs sur les réseaux IPv6, ainsi que la description des tâches associées, reportez-vous à la section "Configuration d'un routeur IPv6" à la page 66.

Présentation de la topologie réseau

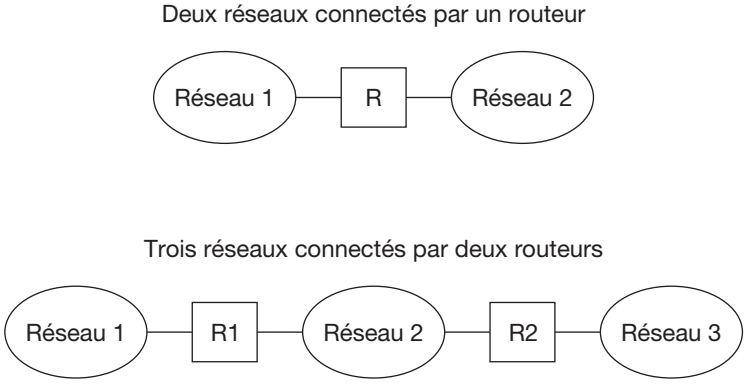
La topologie réseau décrit l'organisation des réseaux. Les routeurs constituent des entités connectant les réseaux les uns aux autres. Toute machine possédant plusieurs interfaces réseau et implémentant la transmission IP constitue un routeur. Toutefois, pour fonctionner en tant que routeur, le système doit être configuré, comme décrit à la section "Configuration d'un routeur IPv4" à la page 43.

Les routeurs connectent plusieurs réseaux pour former des interréseaux plus étendus. Les routeurs doivent être configurés de manière à transmettre des paquets entre deux réseaux adjacents. Les routeurs doivent également être à même de transmettre les paquets vers les réseaux résidant au-delà des réseaux adjacents.

La figure ci-dessous indique les composants de base d'une topologie réseau. La première illustration présente une configuration simple de deux réseaux connectés par un routeur. La deuxième illustration présente la configuration de trois réseaux interconnectés par deux routeurs. Dans le premier exemple, le routeur R joint le réseau 1 et le réseau 2 pour former un

interréseau plus étendu. Dans le deuxième exemple, le routeur R1 connecte les réseaux 1 et 2. Le routeur R2 connecte les réseaux 2 et 3. Les connexions forment un réseau comprenant les réseaux 1, 2 et 3.

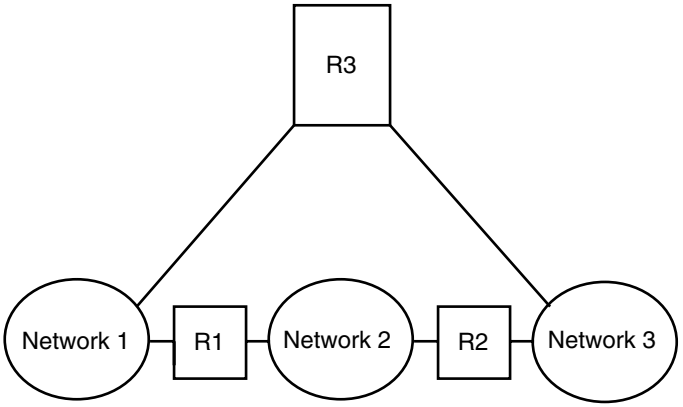
FIGURE 1–1 Topologie réseau de base



Outre la formation d'interréseaux par la jonction de réseaux, les routeurs assurent la transmission de paquets entre les réseaux en fonction des adresses du réseau de destination. A mesure que les interréseaux se complexifient, chaque routeur doit prendre de plus en plus de décisions relativement à la destination des paquets.

La figure ci-dessous présente un cas plus complexe. Le routeur R3 connecte directement les réseaux 1 et 3. La redondance améliore la fiabilité. Si le réseau 2 tombe en panne, le routeur R3 fournit encore une route entre les réseaux 1 et 3. Vous pouvez interconnecter plusieurs réseaux. Toutefois, les réseaux doivent employer les mêmes protocoles réseau.

FIGURE 1–2 Topologie réseau assurant un chemin supplémentaire entre des réseaux



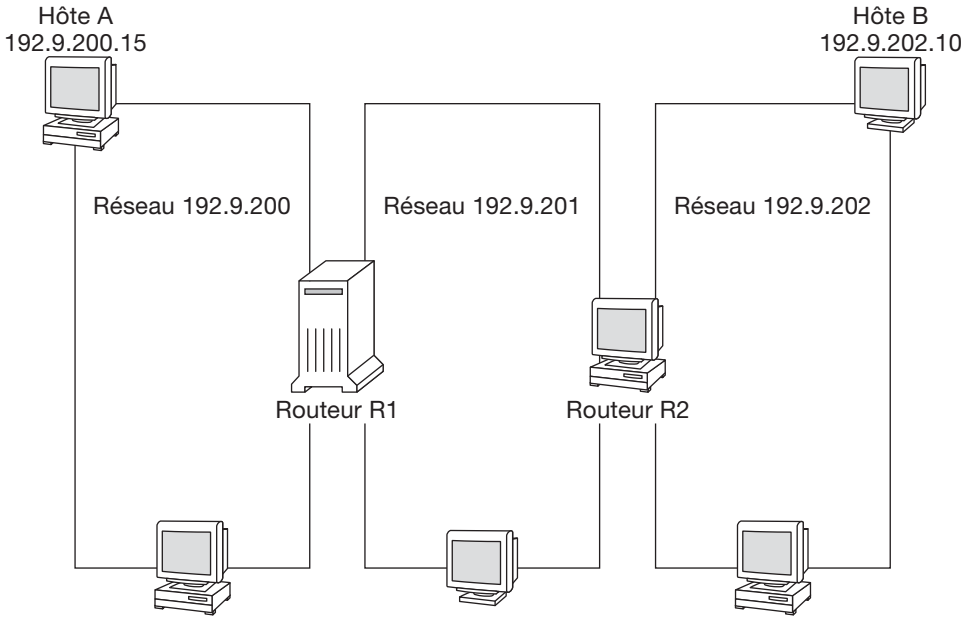
Transfert des paquets par les routeurs

L'adresse IP du destinataire, indiquée dans l'en-tête du paquet, détermine la méthode de routage du paquet. Si le numéro de réseau de l'adresse correspond au réseau local, le paquet se dirige directement vers l'hôte possédant cette adresse IP. Si le numéro de réseau ne correspond pas au réseau local, le paquet se dirige directement vers le routeur du réseau local.

Les routeurs conservent les informations de routage dans des *tables de routage*. Ces tables contiennent les adresses IP des hôtes et routeurs résidant sur les réseaux auxquels le routeur est connecté. Elles incluent également des pointeurs vers ces réseaux. A la réception d'un paquet, le routeur recherche dans sa table de routage l'adresse de destination indiquée dans l'en-tête du paquet. Si l'adresse de destination ne se trouve pas dans la table, le routeur transfère le paquet à un autre routeur répertorié dans sa table de routage. Pour plus d'informations sur les routeurs, reportez-vous à la section "Configuration d'un routeur IPv4" à la page 43.

La figure ci-dessous présente une topologie réseau constituée de trois réseaux connectés par deux routeurs.

FIGURE 1–3 Topologie réseau correspondant à trois réseaux interconnectés



Le routeur R1 connecte les réseaux `192.9.200` et `192.9.201`. Le routeur R2 connecte les réseaux `192.9.201` et `192.9.202`.

SI l'hôte A du réseau `192.9.200` envoie un message à l'hôte B du réseau `192.9.202`, les événements suivants se produisent :

1. L'hôte A envoie un paquet au réseau `192.9.200`. L'en-tête du paquet contient l'adresse IPv4 de l'hôte destinataire, soit l'hôte B, `192.9.202.10`.
2. Aucune machine du réseau `192.9.200` ne possède l'adresse IPv4 `192.9.202.10`. Par conséquent, le routeur R1 accepte le paquet.
3. Le routeur R1 examine ses tables de routage. Aucune machine du réseau `192.9.201` ne possède l'adresse `192.9.202.10`. Toutefois, le routeur R2 est répertorié dans les tables de routage.
4. R1 sélectionne ensuite R2 comme routeur du "saut suivant". R1 envoie le paquet à R2.
5. Comme il connecte le réseau `192.9.201` au réseau `192.9.202`, R2 possède des informations de routage pour l'hôte B. Il transfère ensuite le paquet vers le réseau `192.9.202`, au niveau duquel l'hôte B l'accepte.

Déploiement de réseaux virtuels

Cette version Oracle Solaris prend en charge la création de réseaux virtuels dans un seul réseau, en configurant des zones ainsi que des cartes réseau virtuelles (VNIC). Les VNIC sont des interfaces réseau créées sur des NIC physiques. La combinaison de zones et de VNIC est un moyen efficace de consolider un centre de données contenant un grand nombre de systèmes physiques dans des systèmes de plus petite taille. Pour plus d'informations sur la mise en réseau virtuelle, reportez-vous à la section *Utilisation de réseaux virtuels dans Oracle Solaris 11.1*.

◆ ◆ ◆ CHAPITRE 2

Eléments à prendre en compte lors de l'utilisation d'adresses IPv6

Ce chapitre est un complément du Chapitre 1, “Planification du développement du réseau” et décrit les éléments supplémentaires à prendre en compte en cas d'utilisation d'adresses IPv6 sur votre réseau.

Si vous prévoyez d'utiliser des adresses IPv6 en plus des adresses IPv4, assurez-vous que votre FAI actuel prend en charge les deux types d'adresses. Autrement, vous devrez faire appel à un autre FAI pour prendre en charge les adresses IPv6.

Pour une présentation des concepts IPv6, reportez-vous aux ressources Internet Protocol, Version 6 (IPv6) Specification (`http://www.ietf.org/rfc/rfc2460.txt`).

Planification IPv6 (liste des tâches)

Le tableau suivant répertorie différents éléments à prendre en compte lorsque vous planifiez l'implémentation d'IPv6 sur votre réseau.

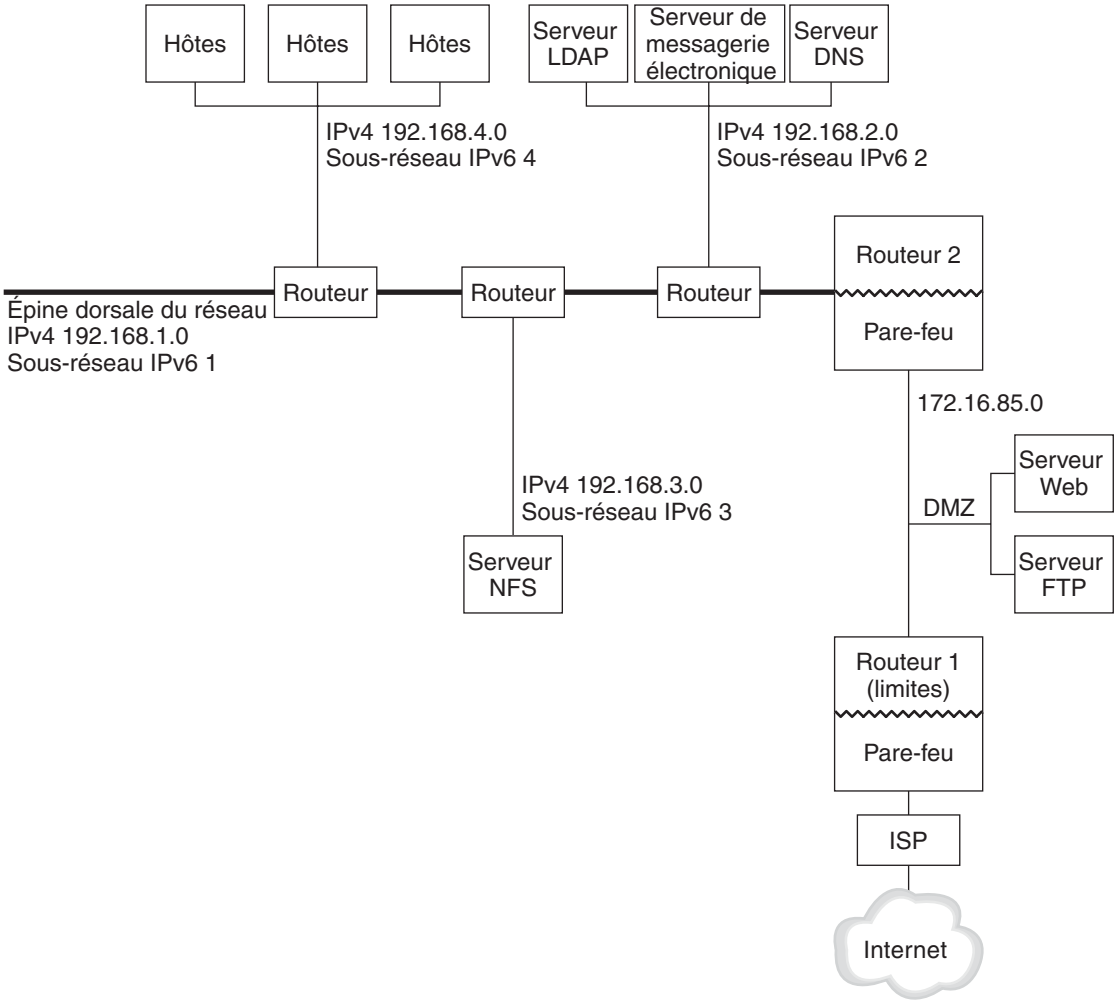
Tâche	Description	Voir
Préparation du matériel pour qu'il prenne en charge IPv6.	Vérifiez qu'il est possible de mettre le matériel à niveau vers IPv6.	“Vérification de la prise en charge d'IPv6” à la page 26
Vérification de la compatibilité des applications avec IPv6.	Assurez-vous que les applications peuvent s'exécuter dans un environnement IPv6.	“Configuration des services réseau pour la prise en charge d'IPv6” à la page 29
Conception d'un plan pour l'utilisation de tunnels.	Déterminez les routeurs qui vont exécuter les tunnels vers d'autres sous-réseaux ou des réseaux externes.	“Planification de l'utilisation de tunnels dans le réseau” à la page 31

Tâche	Description	Voir
Planifiez la sécurisation de vos réseaux et le développement d'une stratégie de sécurité IPv6.	Pour des raisons de sécurité, vous devez disposer d'un plan d'adressage pour la DMZ et ses entités avant de configurer IPv6. Décidez de la méthode d'implémentation de la sécurité que vous souhaitez utiliser (avec un filtre IP, l'architecture IPsec (IP security), le protocole IKE (Internet Key Exchange) et d'autres fonctionnalités de sécurité de cette version).	"Considérations de sécurité relatives à l'implémentation d'IPv6" à la page 31 *Sécurisation du réseau dans Oracle Solaris 11.1*
Création d'un plan d'adressage pour les entités du réseau.	Vous devez disposer au préalable d'un plan pour l'adressage des serveurs, des routeurs et des hôtes avant d'effectuer la configuration d'IPv6. Cette étape inclut l'obtention d'un préfixe de site pour votre réseau ainsi que la planification de sous-réseaux Pv6, le cas échéant.	"Création d'un plan d'adressage IPv6 pour les noeuds" à la page 27

Scénario de topologie de réseau IPv6

En règle générale, IPv6 est utilisé dans une topologie de réseau mixte qui utilise également IPv4, tel qu'illustré dans la figure suivante. Cette figure est utilisée en guise de référence dans la description des tâches de configuration d'IPv6 dans les sections suivantes.

FIGURE 2–1 Scénario de topologie de réseau IPv6



Le scénario de réseau d'entreprise se compose de cinq sous-réseaux disposant d'adresses IPv4. Les liaisons du réseau correspondent directement aux sous-réseaux administratifs. Les quatre réseaux internes sont affichés avec des adresses IPv4 privées de type RFC 1918, ce qui correspond à une solution courante pour le manque d'adresses IPv4. Le schéma d'adressage de ces réseaux internes est comme suit :

- Le sous-réseau 1 correspond à l'épine dorsale du réseau interne `192.168.1`.
- Le sous-réseau 2 correspond au réseau interne `192.168.2`, avec LDAP, `sendmail` et serveurs DNS.

- Le sous-réseau 3 correspond au réseau interne `192.168.3`, avec les serveurs NFS de l'entreprise.
- Le sous-réseau 4 correspond au réseau interne `192.168.4` qui contient les hôtes des employés de l'entreprise.

Le réseau public externe `172.16.85` fait office de DMZ pour l'entreprise. Ce réseau contient des serveurs Web, des serveurs FTP anonymes et d'autres ressources que l'entreprise propose au monde extérieur. Le routeur 2 exécute un pare-feu et sépare le réseau public `172.16.85` de l'épine dorsale interne. Sur l'autre extrémité de la DMZ, le routeur 1 exécute un pare-feu et fait office de serveur de limites de l'entreprise.

Sur la Figure 2–1, la DMZ publique possède l'adresse privée RFC 1918 `172.16.85`. Dans le monde réel, la DMZ publique doit disposer d'une adresse IPv4 enregistrée. La plupart des sites IPv4 utilisent une combinaison d'adresses publiques et d'adresses privées RFC 1918. Cependant, lors de l'introduction d'IPv6, le concept d'adresses publiques et privées est modifié. Dans la mesure où IPv6 dispose d'un espace d'adresse beaucoup plus important, les adresses publiques IPv6 s'utilisent à la fois sur les réseaux privés et publics.

Le protocole double pile Oracle Solaris prend en charge à la fois les opérations IPv4 et IPv6. Vous pouvez exécuter des opérations relatives à IPv4 pendant et après le déploiement d'IPv6 sur votre réseau. Lorsque vous déployez IPv6 sur un réseau en cours de fonctionnement qui utilise déjà IPv4, assurez-vous de ne pas perturber les opérations en cours.

Les sections suivantes décrivent les domaines à prendre en compte lors de la préparation de l'implémentation d'IPv6.

Vérification de la prise en charge d'IPv6

Consultez la documentation du fabricant en matière de compatibilité IPv6 en ce qui concerne les classes de matériel suivantes :

- Routeurs
- Pare-feux
- Serveurs
- Commutateurs

Remarque – Toutes les procédures décrites dans ce manuel partent du principe qu'il est possible de mettre le matériel à niveau (en particulier les routeurs) vers IPv6.

Certains modèles de routeurs ne permettent pas une mise à niveau vers IPv6. Pour obtenir des informations supplémentaires et une solution au problème, reportez-vous à la section "IPv4 Router Cannot Be Upgraded to IPv6" du manuel *Troubleshooting Network Issues*.

Pour chaque NIC des serveurs IPv6, configurez manuellement la partie ID d'interface de l'adresse IPv6 plutôt que d'obtenir automatiquement l'ID à l'aide du protocole Neighbor Discovery. De cette façon, si une NIC est remplacée, le même ID d'interface peut être appliqué à la nouvelle NIC. Un ID différent généré automatiquement par le protocole Neighbor Discovery peut entraîner un comportement inattendu du serveur.

Préparation d'un plan d'adressage IPv6

Le développement d'un plan d'adressage constitue une des parties les plus importantes de la transition d'IPv4 à IPv6. Cette tâche nécessite les préparatifs suivants :

- "Obtention d'un préfixe de site" à la page 27
- "Création du schéma de numérotation IPv6" à la page 27

Obtention d'un préfixe de site

Vous devez disposer d'un préfixe de site préalablement à la configuration d'IPv6. Le préfixe de site permet de dériver les adresses IPv6 pour tous les noeuds de votre implémentation IPv6.

Tout FAI prenant en charge IPv6 devrait être en mesure de fournir un préfixe de site IPv6 de 48 octets. Si votre FAI ne prend en charge que IPv4, vous pouvez faire appel à un autre FAI pour la prise en charge d'IPv6 tout en conservant votre FAI actuel pour la prise en charge d'IPv4. Dans ce cas, il existe plusieurs solutions au problème. Pour plus d'informations, reportez-vous à la section "Current ISP Does Not Support IPv6" du manuel *Troubleshooting Network Issues*.

Si votre entreprise est un FAI, les préfixes de site pour vos clients s'obtiennent auprès du registre Internet adéquat. Pour plus d'informations, reportez-vous au site Internet Assigned Numbers Authority (IANA) (`http://www.iana.org`).

Création du schéma de numérotation IPv6

Si votre réseau IPv6 n'est pas entièrement nouveau, basez le schéma de numérotation IPv6 sur la topologie IPv4 existante.

Création d'un plan d'adressage IPv6 pour les noeuds

Pour la plupart des hôtes, la configuration automatique d'adresses IPv6 sans état pour leurs interfaces constitue une stratégie adéquate et rapide. Lorsque l'hôte reçoit le préfixe de site en provenance du routeur le plus proche, la détection de voisin génère automatiquement des adresses IPv6 pour chaque interface de l'hôte.

Les serveurs doivent disposer d'adresses IPv6 stables. Si vous ne configurez pas manuellement les adresses IPv6 d'un serveur, une nouvelle adresse IPv6 est configurée automatiquement à

chaque fois qu'une carte d'interface réseau est remplacée sur le serveur. Tenez compte des conseils suivants lors de la création d'adresses de serveurs :

- Attribuez aux serveurs des ID d'interface significatifs et stables. Vous pouvez par exemple utiliser un schéma de numérotation séquentiel pour les ID d'interface. Par exemple, l'interface interne du serveur LDAP dans la Figure 2–1 pourrait devenir `2001:db8:3c4d:2::2`.
- Si vous ne renommez pas régulièrement votre réseau IPv4, vous pouvez également utiliser les adresses IPv4 des routeurs et serveurs en tant qu'ID d'interface. Dans la Figure 2–1, on suppose que l'interface du routeur 1 vers la DMZ a pour adresse IPv4 `123.456.789.111`. Vous pouvez convertir l'adresse IPv4 vers le format hexadécimale et utiliser le résultat de la conversion en tant qu'ID d'interface. Le nouvel ID d'interface serait `::7bc8:156F`.

  Cette approche est applicable uniquement si vous êtes propriétaire de l'adresse IPv4 enregistrée, non pas si vous l'avez obtenue auprès d'un FAI. Si vous utilisez une adresse IPv4 qui vous a été fournie par un FAI, vous créez une dépendance qui risque d'entraîner des problèmes en cas de changement de FAI.

En raison du nombre limité d'adresses IPv4, un concepteur de réseau devait auparavant se demander s'il devait utiliser des adresses globales enregistrées ou des adresses privées RFC 1918. Cependant, la notion d'adresses IPv4 privées et publiques ne s'applique pas aux adresses IPv6. Vous pouvez utiliser des adresses globales unicast incluant le préfixe de site, sur toutes les liaisons du réseau, DMZ publique incluse.

Création d'un schéma de numérotation pour les sous-réseaux

Commencez par mapper les sous-réseaux IPv4 existants vers les sous-réseaux IPv6 équivalents. Par exemple, utilisez les sous-réseaux illustrés sur la Figure 2–1. Les sous-réseaux 1 à 4 utilisent l'identification d'adresse privée IPv4 RFC 1918 pour les 16 premiers octets de leurs adresses, en plus des chiffres 1 à 4 qui identifient le sous-réseau. Par exemple, supposons que le préfixe IPv6 `2001:db8:3c4d/48` a été assigné au site.

Le tableau suivant illustre le mappage des préfixes IPv4 privés vers les préfixes IPv6.

Préfixe de sous-réseau IPv4	Préfixe de sous-réseau IPv6 équivalent
`192.168.1.0/24`	`2001:db8:3c4d:1::/64`
`192.168.2.0/24`	`2001:db8:3c4d:2::/64`
`192.168.3.0/24`	`2001:db8:3c4d:3::/64`
`192.168.4.0/24`	`2001:db8:3c4d:4::/64`

Configuration des services réseau pour la prise en charge d'IPv6

Les services réseau IPv4 suivants de la version active d'Oracle Solaris sont compatibles avec le protocole IPv6 :

- `sendmail`
- NFS
- HTTP (versions Apache 2 ou Orion)
- DNS
- LDAP

Le service de messagerie IMAP est compatible uniquement avec IPv4.

Les noeuds configurés pour IPv6 peuvent exécuter des services IPv4. Lors de l'activation d'IPv6, tous les services n'acceptent pas les connexions IPv6. Les services préparés pour IPv6 acceptent les connexions. Les services qui ne le sont pas continuent de fonctionner avec la partie IPv4 de la pile de protocole.

Certains problèmes peuvent survenir après une mise à niveau des services vers IPv6. Pour plus de détails, reportez-vous à la section "Problems After Upgrading Services to IPv6" du manuel *Troubleshooting Network Issues*.

▼ Préparation de services réseau pour la prise en charge d'IPv6

1 **Mettez les services réseau suivants à jour afin qu'ils prennent en charge IPv6 :**

- Serveurs de courrier
- Serveurs NIS
- NFS

Remarque – LDAP prend en charge IPv6 sans aucune configuration supplémentaire nécessaire.

---

2 **Assurez-vous que le matériel de votre pare-feu est compatible avec le protocole IPv6.**

Reportez-vous à la documentation adéquate pour obtenir des instructions.

3 **Assurez-vous que les autres services de votre réseau ont été préparés pour prendre en charge le protocole IPv6.**

Pour plus d'informations, reportez-vous à la documentation technique et marketing du logiciel.

4 **Si votre site déploie les services suivante, assurez-vous d'avoir pris les mesures adéquates pour ces services :**

- Pare-feux

  Pensez à renforcer les stratégies en place pour le protocole IPv4 afin qu'elles prennent en charge le protocole IPv6. Pour prendre connaissance de problèmes de sécurité supplémentaires, reportez-vous à la section "Considérations de sécurité relatives à l'implémentation d'IPv6" à la page 31.

- Messagerie

  Vous pouvez envisager d'ajouter les adresses IPv6 de votre serveur de courrier aux enregistrements MX pour DNS.

- DNS

  Pour prendre connaissance des considérations spécifiques à DNS, reportez-vous à la section "Préparation de DNS pour la prise en charge d'IPv6" à la page 30.

- IPQoS

  Utilisez les mêmes stratégies Diffserv que celles utilisées pour le protocole IPv4 sur l'hôte. Pour plus d'informations, reportez-vous à la section "Module de classification" du manuel *Gestion d'IPQoS (IP Quality of Service) dans Oracle Solaris 11.1*.

5 **Auditez tout service réseau offert par un noeud avant de convertir ce dernier vers IPv6.**

▼ Préparation de DNS pour la prise en charge d'IPv6

La version d'Oracle Solaris actuelle prend en charge la résolution de DNS côté client et côté serveur. Procédez comme suit pour préparer les services DNS à IPv6.

Pour plus d'informations sur la prise en charge de DNS pour IPv6, reportez-vous au document *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.

1 **Assurez-vous que le serveur DNS effectuant la résolution récursive de nom est double pile (IPv4 et IPv6) ou uniquement compatible avec IPv4.**

2 **Dans le serveur DNS, renseignez la base de données DNS avec les enregistrements AAAA de base de données IPv6 dans la zone de transfert.**

Remarque – Les serveurs exécutant plusieurs services critiques requièrent une attention particulière. Assurez-vous du bon fonctionnement du réseau. En outre, tous les services critiques doivent avoir été préparés pour IPv6. Ensuite, ajoutez l'adresse IPv6 du serveur à la base de données DNS.

---

3 **Ajoutez les enregistrements PTR associés aux enregistrements AAAA dans la zone d'inversion.**

4. **Ajoutez des données exclusivement IPv4 ou des données IPv6 et IPv4 à l'enregistrement NS décrivant les zones.**

Planification de l'utilisation de tunnels dans le réseau

L'implémentation d'IPv6 prend en charge un certain nombre de configurations de tunnel faisant office de mécanismes de transition lors de la migration de votre réseau vers un mélange d'IPv4 et d'IPv6. Les tunnels permettent aux réseaux IPv6 isolés de communiquer. Dans la mesure où Internet exécute essentiellement IPv4, les paquets IPv6 de votre site doivent circuler dans Internet via des tunnels ayant pour destination des réseaux IPv6.

Vous trouverez ici les scénarios les plus courants d'utilisation de tunnels dans la topologie de réseau IPv6 :

- Le FAI qui vous fournit des services IPv6 vous permet de créer un tunnel à partir du routeur de bordure du site vers le réseau du FAI. La Figure 2–1 représente un de ces tunnels. Dans ce cas, vous devez exécuter un tunnel manuel IPv6 sur IPv4.
- Vous gérez un réseau distribué de grande taille avec connectivité IPv4. Pour connecter les sites distribués utilisant IPv6, vous pouvez exécuter un tunnel automatique 6to4 à partir du routeur de périphérie de chaque sous-réseau.
- Il est parfois impossible de mettre un routeur à niveau vers IPv6 dans l'infrastructure de l'entreprise. Dans ce cas, vous pouvez créer un tunnel manuel à travers le routeur IPv4, avec deux routeurs IPv6 en guise d'extrémités.

Pour connaître les procédures de configuration des tunnels, reportez-vous à la section "Configuration des tunnels (liste des tâches)" à la page 112. Pour obtenir des informations conceptuelles à propos des tunnels, reportez-vous à la section "Présentation des tunnels IP" à la page 103.

Considérations de sécurité relatives à l'implémentation d'IPv6

En cas d'introduction d'IPv6 dans un réseau existant, veillez à ne pas compromettre la sécurité du site. Tenez compte des problèmes de sécurité suivants lors de l'implémentation progressive d'IPv6 :

- La même quantité de filtrage est requise pour les paquets IPv6 et IPv4.
- Les paquets IPv6 sont souvent mis en tunnel via un pare-feu. Par conséquent, implémentez l'un des deux scénarios suivants :
  - Paramétrez le pare-feu de sorte qu'il inspecte le contenu du tunnel.
  - Placez un pare-feu IPv6 avec des règles similaires à l'extrémité opposée du tunnel.
- Certains mécanismes de transition utilisent des tunnels IPv6 sur UDP sur IPv4. Ces mécanismes peuvent s'avérer dangereux et court-circuiter le pare-feu.

- Globalement, il est possible d'atteindre les noeuds IPv6 à partir de l'extérieur du réseau de l'entreprise. Si votre stratégie de sécurité interdit tout accès public, vous devez établir des règles de pare-feu plus strictes. Vous pourriez par exemple configurer un pare-feu avec état.

Ce manuel inclut des fonctionnalités de sécurité qu'il est possible d'utiliser dans une implémentation IPv6.

- La fonction d'architecture IPsec (sécurité IP) permet d'obtenir une protection cryptographique des paquets IPv6. Pour plus d'informations, reportez-vous au Chapitre 6, "Architecture IPsec (présentation)" du manuel *Sécurisation du réseau dans Oracle Solaris 11.1*.
- La fonctionnalité IKE (Internet Key Exchange, échange de clé Internet) permet d'utiliser l'authentification de clé publique pour les paquets IPv6. Pour plus d'informations, reportez-vous au Chapitre 9, "Protocole IKE (présentation)" du manuel *Sécurisation du réseau dans Oracle Solaris 11.1*.

◆ ◆ ◆ CHAPITRE 3

Configuration d'un réseau IPv4

La configuration réseau s'effectue en deux étapes : l'assemblage du matériel, puis la configuration des démons, fichiers et services d'implémentation du protocole TCP/IP.

Le présent chapitre décrit la configuration d'un réseau implémentant les services et l'adressage IPv4.

De nombreuses tâches abordées dans ce chapitre s'appliquent aussi bien aux réseaux IPv4 uniquement qu'aux réseaux IPv6. Les tâches spécifiques aux réseaux IPv6 figurent dans le Chapitre 4, “Activation d'IPv6 sur le réseau”.

Remarque – Avant de configurer TCP/IP, passez en revue les différentes taches de planification répertoriées dans le Chapitre 1, “Planification du développement du réseau”. Si vous prévoyez d'utiliser des adresses IPv6, consultez également le Chapitre 2, “Eléments à prendre en compte lors de l'utilisation d'adresses IPv6”.

Le présent chapitre contient les informations suivantes :

Configuration réseau (liste des tâches)

Le tableau suivant répertorie les tâches supplémentaires à effectuer une fois que vous êtes passé d'une configuration réseau sans sous-réseaux à un réseau utilisant des sous-réseaux. Le tableau comprend la description des actions de chaque tâche et la section de la documentation actuelle dans laquelle les étapes permettant d'effectuer ces tâches sont décrites en détails.

Tâche	Description	Voir
Configuration des interfaces IP du système	Attribue des adresses IP aux interfaces IP du système.	"Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*
Configuration d'un système en mode Fichiers locaux	Modifie des fichiers de configuration spécifiques dans le répertoire `/etc` du système et configure le service SMF `nis/domain`.	"Configuration d'un système en mode fichiers locaux" à la page 40
Configuration d'un serveur de configuration réseau	Active le démon `in.tftp` et modifie les autres fichiers de configuration dans le répertoire `/etc` du système.	"Configuration d'un serveur de configuration réseau" à la page 42
Configuration d'un système en mode Client réseau	Modifie les fichiers de configuration dans le répertoire `/etc` du système.	"Configuration d'un système en mode Client réseau" à la page 41
Spécification de la stratégie de routage du client réseau	Configure les systèmes pour une utilisation du routage statique ou du routage dynamique.	"Activation du routage statique sur un hôte à interface unique" à la page 52 et "Activation du routage dynamique sur un système à interface unique" à la page 53

Avant de commencer la configuration réseau

Dans cette version d'Oracle Solaris, la configuration réseau d'un système est gérée par un *NCP (network configuration profile, profil de configuration réseau)* actif. La configuration réseau du système est automatique si le NCP actuel est réactif (NCP `automatic`, par exemple). Si le NCP actif est `DefaultFixed`, le mode de configuration réseau du système est fixe. Le système a un comportement différent selon que sa configuration réseau est réactive ou fixe.

Toute configuration effectuée s'applique au NCP actif. Par conséquent, avant de suivre une procédure de configuration, il faut commencer par déterminer le NCP actif. Le système a donc un comportement normal une fois que vous avez terminé les procédures de configuration. Pour déterminer le NCP actif sur un système, entrez la commande suivante :

```
# netadm list
TYPE        PROFILE        STATE
ncp         DefaultFixed   online
ncp         Automatic      disabled
loc         Automatic      offline
loc         NoNet          offline
loc         User           offline
loc         DefaultFixed   online
```

Le profil dont le statut est répertorié comme en ligne correspond au NCP actif sur le système.

Pour plus d'informations sur les profils de configuration réseau sur le système, ajoutez `-x` à la commande `netadm`.

```
netadm list -x
TYPE        PROFILE        STATE          AUXILIARY STATE
ncp         DefaultFixed   online         active
ncp         Automatic      disabled       disabled by administrator
loc         Automatic      offline        conditions for activation are unmet
loc         NoNet          offline        conditions for activation are unmet
loc         User           offline        conditions for activation are unmet
loc         DefaultFixed   online         active
```

Pour basculer d'un type de profil à un autre (d'un profil réactif à un profil fixe, par exemple), entrez la commande suivante :

```
# netadm enable -p ncp NCP-name
```

Remplacez *NCP-name* par le nom d'un type de NCP.

Pour une présentation de la configuration réseau gérée par profil, reportez-vous à la section "Profils de configuration réseau" du manuel *Introduction à la mise en réseau d'Oracle Solaris 11*. Pour plus d'informations sur les NCP, reportez-vous au manuel *Connexion de systèmes à l'aide d'une configuration réseau réactive dans Oracle Solaris 11.1*.

Configuration des composants système sur le réseau

Lorsque vous configurez des systèmes réseau, les informations de configuration suivantes sont nécessaires :

- Nom d'hôte de chaque système.
- Adresse IP et masque réseau de chaque système. Si le réseau est divisé en sous-réseaux, vous devez disposer des numéros de sous-réseau et du schéma d'adresse IP à appliquer aux systèmes dans chaque sous-réseau, incluant leurs masques réseau respectifs.
- Nom de domaine auquel chaque système appartient.
- Adresse de routeur par défaut.

  Vous devez fournir cette information lorsqu'un routeur unique est connecté à chaque réseau de la topologie. Vous devez également la fournir lorsque les routeurs n'utilisent pas de protocoles de routage tels que RDISC (Router Discovery Server Protocol) ou RIP (Router

Information Protocol). Pour plus d'informations sur les routeurs et pour consulter la liste des protocoles de routage qu'Oracle Solaris prend en charge, reportez-vous à la section "Protocoles de routage dans Oracle Solaris" à la page 129.

Remarque – Vous pouvez configurer le réseau tout en installant Oracle Solaris. Pour obtenir des instructions, reportez-vous au document *Installation des systèmes Oracle Solaris 11.1*.

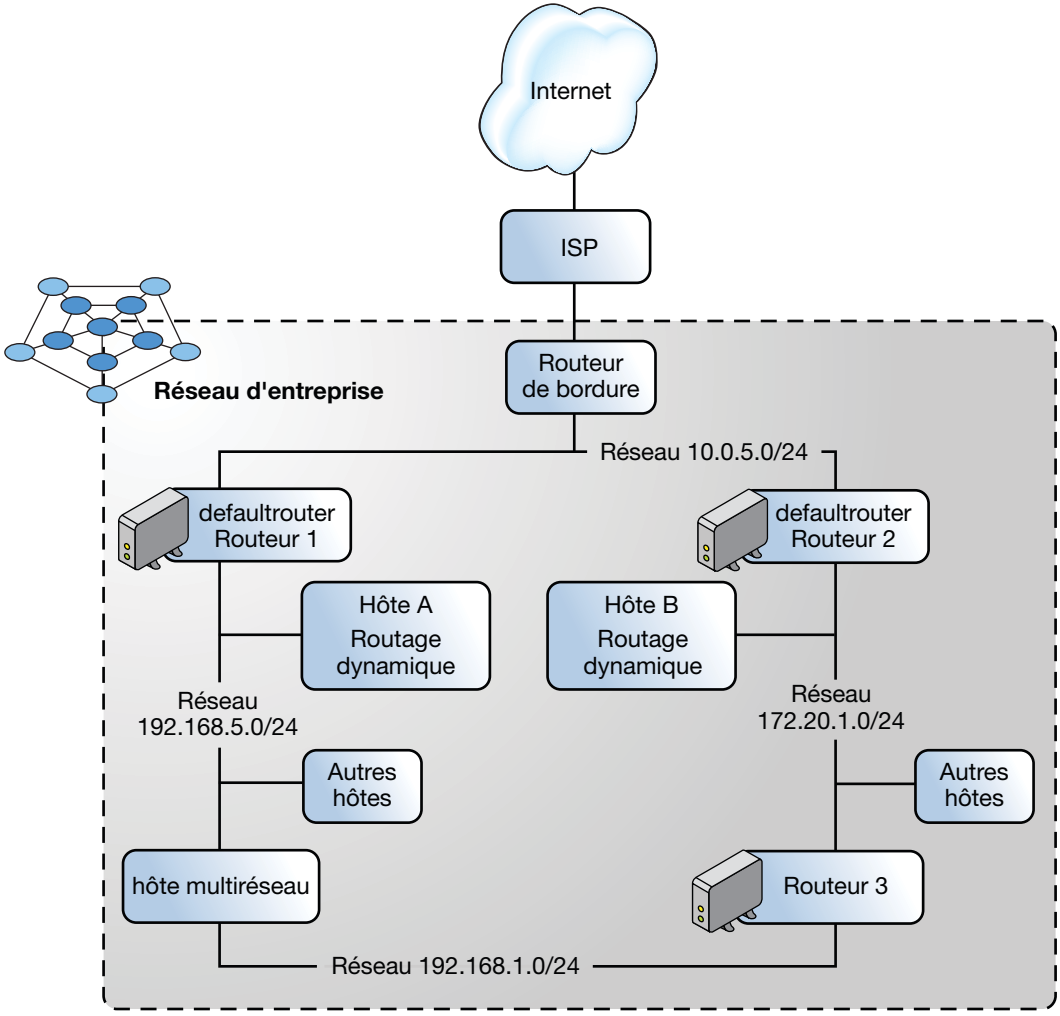
Dans cette documentation, les procédures supposent que vous configurez le réseau après avoir installé le SE.

Utilisez la Figure 3–1 dans la section suivante en tant que référence pour configurer les systèmes de composant du réseau.

Topologie du système autonome IPv4

Les sites comportant plusieurs routeurs et réseaux gèrent généralement leur topologie réseau comme un domaine de routage unique, également appelé *système autonome AS (Autonomous System)* .

FIGURE 3–1 Système autonome comportant plusieurs routeurs IPv4



La Figure 3–1représente un AS divisé en trois réseaux locaux, `10.0.5.0`, `172.16.1.0` et `192.168.5.0`. Le réseau comporte les trois types de systèmes suivants :

- Les routeurs utilisent des protocoles de routage pour gérer comment les paquets de réseau sont dirigés ou acheminés de leur source vers leurs destinations au sein du réseau local ou vers des réseaux externes. Pour obtenir des informations sur les protocoles de routage pris en charge dans Oracle Solaris, reportez-vous à la section "Tableaux des protocoles de routage dans Oracle Solaris" à la page 130.

  Les routeurs sont saisis comme suit :

- Le *routeur de bordure* connecte le réseau local tel que `10.0.5.0` de façon externe à un fournisseur de service.
- Les *routeurs par défaut* gèrent le routage de paquets dans le réseau local, lequel peut inclure plusieurs réseaux locaux. Par exemple, dans la Figure 3–1, le routeur 1 fait office de routeur par défaut pour `192.168.5`. En même temps, Router 1 est également connecté au réseau interne `10.0.5.0`. Les interfaces de Router 2 se connectent aux réseaux internes `10.0.5.0` et `172.16.1.0`.
- Les *routeurs de transfert de paquet* transfèrent les paquets entre les réseaux internes, mais n'exécutent pas de protocoles de routage. Dans la Figure 3–1, Router 3 est un routeur de transfert de paquets avec des connexions aux réseaux `172.16.1` et `192.168.5`.

- Systèmes client
  - Systèmes multiréseau ou systèmes dotés de plusieurs NIC. Dans Oracle Solaris, ces systèmes par défaut peuvent transférer des paquets à d'autres systèmes dans le même segment de réseau.
  - Les systèmes à interface unique reposent sur les routeurs locaux pour le transfert de paquets et la réception des informations de configuration.

Modes de configuration système

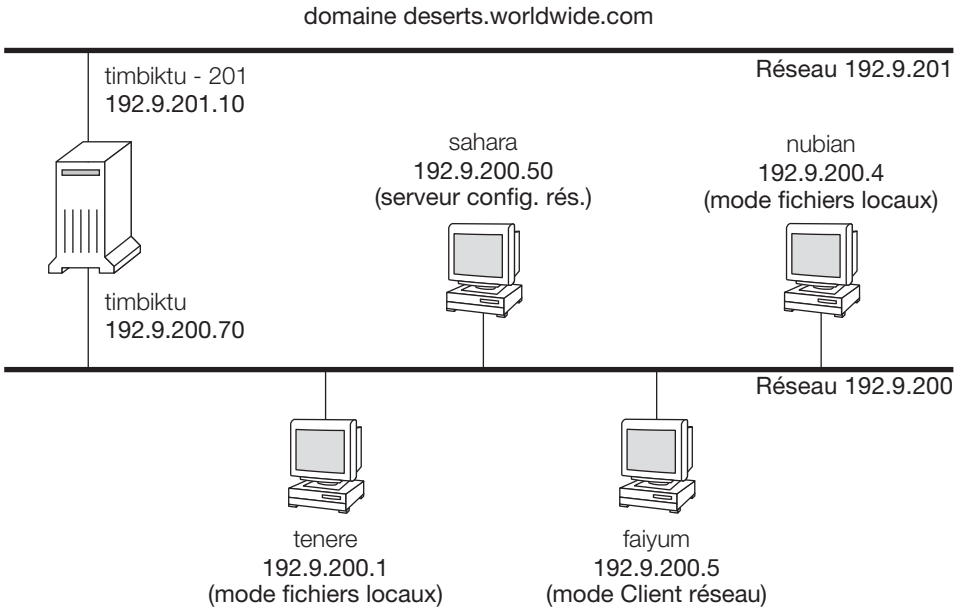
Cette section décrit les procédures de configuration d'un système en vue d'une exécution en *mode fichiers locaux* ou en *mode client réseau*. En cas d'exécution en mode fichiers locaux, un système obtient toutes les informations de configuration TCP/IP auprès de fichiers qui se trouvent dans le répertoire local. En mode Client réseau, les informations de configuration sont fournies à tous les systèmes dans le réseau par un serveur de configuration réseau.

En règle générale, les serveurs dans le réseau s'exécutent en mode fichiers locaux, comme suit :

- Serveurs de configuration réseau
- Serveurs NFS
- Serveurs de noms fournissant les services NIS, LDAP ou DNS
- Serveurs de courrier
- Routeurs

Les clients peuvent s'exécuter dans les deux modes. Par conséquent, dans le réseau, vous pouvez avoir une combinaison de ces deux modes en fonction de la configuration des différents systèmes, comme illustré dans la figure suivante :

FIGURE 3–2 Systèmes dans un scénario de topologie de réseau IPv4



La Figure 3–2 représente les systèmes dans un réseau `192.9.200`.

- Tous les systèmes appartiennent au domaine d'organisation `deserts.worldwide.com`.
- `sahara` est un serveur de configuration. En tant que serveur, il s'exécute en mode fichiers locaux, où les informations de configuration TCP/IP sont obtenues auprès du disque local du système.

Remarque – Si vous configurez des clients pour qu'ils s'exécutent en mode Client réseau, vous devez alors configurer au moins un serveur de configuration réseau qui fournira les informations de configuration à ces clients.

- `tenere`, `nubian` et `faiyum` sont des clients dans le réseau. `tenere` et `nubian` s'exécutent en mode fichiers locaux. Quel que soit le disque local de `faiyum`, le système est configuré pour fonctionner en mode Client réseau.
- `timbuktu` est configuré en tant que routeur et fonctionne par conséquent en mode fichiers locaux. Le système inclut deux NIC, chacun ayant ses propres interfaces IP configurées. La première interface IP est nommée `timbuktu` et se connecte au réseau `192.9.200`. La deuxième interface IP est nommée `timbuktu-201` et se connecte au réseau `192.9.201`.

▼ Configuration d'un système en mode fichiers locaux

Suivez cette procédure pour configurer un système pour s'exécuter en mode fichiers locaux.

1. **Configurez les interfaces IP du système avec les adresses IP attribuées.**

   Pour connaître la procédure à suivre, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

2. **Vérifiez le nom d'hôte défini dans le fichier `/etc/nodename`.**

3. **Vérifiez que les entrées du fichier `/etc/inet/hosts` sont à jour.**

   Le programme d'installation Oracle Solaris crée des entrées pour l'interface réseau principale, l'adresse loopback et toute interface supplémentaire configurée lors de l'installation, le cas échéant.

   Ce fichier doit inclure le nom du routeur par défaut et l'adresse IP du routeur.

   a. **(Facultatif) Ajoutez les adresses IP et les noms correspondants des interfaces réseau ajoutées au système après l'installation.**

   b. **(Facultatif) Si le système de fichiers `/usr` est monté sur un système NFS, ajoutez la ou les adresses IP du serveur de fichiers.**

4. **Spécifiez le domaine complet du système en tant que propriété du service SMF `nis/domain`.**

   Par exemple, vous pourriez spécifier `deserts.worldwide.com` en tant que valeur de la propriété `domainname` du service SMF `nis/domain`, comme suit :

   ```
   # domainname domainname
   ```

   Cette étape entraîne une modification persistante.

5. **Entrez le nom du routeur dans le fichier `/etc/defaultrouter`.**

6. **Ajoutez les informations de masque de réseau, le cas échéant.**

   Remarque – Si vous utilisez des services DHCP, passez cette étape.

   a. **Tapez le numéro et le masque de réseau dans le fichier `/etc/inet/netmasks`.**

      Pour créer des entrées, utilisez le format *réseau-numéro de masque de réseau*. Par exemple, pour le numéro de réseau de Classe C `192.168.83`, vous devez taper :

      ```
      192.168.83.0    255.255.255.0
      ```

Pour les adresses CIDR, remplacez le préfixe réseau par la représentation décimale avec points équivalente. Les préfixes de réseau et leurs équivalents décimaux à points sont répertoriés dans le Tableau 1–1. Par exemple, pour exprimer le préfixe réseau CIDR `192.168.3.0/22`, tapez ce qui suit :

```
192.168.3.0     255.255.252.0
```

b. Modifiez l'ordre de recherche des masques de réseau dans la propriété SMF du commutateur de sorte que la recherche s'effectue d'abord dans les fichiers locaux, puis actualisez l'instance.

```
# svccfg -s name-service/switch setprop config/host = astring: '"files nis"'
# svccfg -s name-service/switch:default refresh
```

7 Redémarrez le système.

▼ Configuration d'un système en mode Client réseau

Effectuez la procédure suivante sur chaque hôte à configurer en mode Client réseau.

Avant de commencer

Les clients réseau reçoivent leurs informations de configuration des serveurs de configuration réseau. Par conséquent, avant de configurer un système en tant que client réseau, assurez-vous de configurer au moins un serveur de configuration pour le réseau.

1 Connectez-vous en tant qu'administrateur.

Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2 Configurez les interfaces IP du système avec les adresses IP attribuées.

Pour connaître la procédure à suivre, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

3 Assurez-vous que le fichier `/etc/inet/hosts` contient uniquement le nom `localhost` et l'adresse IP de l'interface réseau loopback.

```
# cat /etc/inet/hosts
# Internet host table
#
127.0.0.1       localhost
```

4 Supprimez toute valeur attribuée à la propriété `domainname` du service SMF `nis/domain`.

```
# domainname ''
```

Cette étape entraîne une modification persistante.

5 Assurez-vous que les chemins de recherche dans le service `name-service/switch` du client reflètent les mêmes exigences service de votre réseau.

▼ Configuration d'un serveur de configuration réseau

Vous trouverez des informations sur la configuration de serveurs d'installation et d'initialisation dans le document *Installation des systèmes Oracle Solaris 11.1*.

1. **Connectez-vous en tant qu'administrateur.**

   Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2. **Activez le démon `in.tftpd` comme suit :**

   a. **Accédez au répertoire root (`/`) du serveur de configuration réseau désigné.**

   b. **Créez le répertoire `/tftpboot` :**

      ```
      # mkdir /tftpboot
      ```

      Cette commande configure le système en tant que serveur RARP, bootparams et TFTP.

   c. **Créez un lien symbolique vers le répertoire.**

      ```
      # ln -s /tftpboot/. /tftpboot/tftpboot
      ```

3. **Ajoutez la ligne `tftp` dans le fichier `/etc/inetd.conf`.**

   La ligne devrait être comme suit :

   ```
   tftp dgram udp6 wait root /usr/sbin/in.tftpd in.tftpd -s /tftpboot
   ```

   Cette ligne empêche `in.tftpd` d'extraire un fichier autre que ceux figurant dans `/tftpboot`.

4. **Dans la base de données `/etc/hosts`, ajoutez les noms d'hôte et les adresses IP de tous les clients sur le réseau.**

5. **Dans la base de données `/etc/ethers`, créez des entrées pour chaque système sur le réseau qui s'exécute en mode client réseau.**

   Les entrées de cette base de données sont au format suivant :

   MAC Address *host name* *#comment*

   Pour plus d'informations, reportez-vous à la page de manuel `ethers(4)`.

6. **Dans la base de données `/etc/bootparams`, créez une entrée pour chaque système sur le réseau qui s'exécute en mode client réseau.**

   Pour obtenir des informations sur la modification de cette base de données, consultez la page de manuel `bootparams(4)`.

7. **Convertissez l'entrée `/etc/inetd.conf` en un fichier manifeste de service SMF (Service Management Facility) et activez le service obtenu.**

   ```
   # /usr/sbin/inetconv
   ```

8 **Assurez-vous que `in.tftpd` fonctionne correctement.**

```
# svcs network/tftp/udp6
```

La sortie que vous devez recevoir ressemble à ce qui suit :

```
STATE          STIME    FMRI
online         18:22:21 svc:/network/tftp/udp6:default
```

Informations supplémentaires

Gestion du démon `in.tftpd`

Le démon `in.tftpd` est géré par SMF (Service Management Facility). La commande `svcadm` permet d'effectuer les opérations de gestion sur `in.tftpd` (par exemple, l'activation, la désactivation ou le redémarrage). L'initiation et la réinitialisation du service s'effectue par l'intermédiaire de la commande `inetd` . Utilisez la commande `inetadm` pour modifier la configuration et afficher les informations de configuration pour `in.tftpd`. La commande `svcs` permet d'interroger l'état du service. Pour une présentation de l'utilitaire de gestion des service (SMF), reportez-vous au Chapitre 1, “Gestion des services (présentation)” du manuel *Gestion des services et pannes dans Oracle Solaris 11.1*.

Configuration d'un routeur IPv4

Un routeur fournit l'interface entre deux réseaux ou plus. Par conséquent, vous devez attribuer un nom unique et une adresse IP à chacune des interfaces de réseau physique du routeur. Par conséquent, chaque routeur possède un nom d'hôte et une adresse IP associés à son interface réseau principale ainsi qu'un nom et une adresse IP uniques pour chaque interface réseau supplémentaire.

Vous pouvez également effectuer la procédure suivante pour configurer un système doté d'une seule interface physique (un hôte, par défaut) en tant que routeur. Vous pouvez configurer un système d'interface unique en tant que routeur si le système fait office de point d'extrémité sur un lien PPP, comme décrit à la section “Planification d’une liaison PPP commutée” du manuel *Gestion de réseaux série à l’aide d’UUCP et de PPP dans Oracle Solaris 11.1*.

▼ Configuration d'un routeur IPv4

La procédure suivante suppose que vous configurez les interfaces du routeur après l'installation.

Avant de commencer

Une fois le routeur installé physiquement sur le réseau, configurez le routeur de sorte qu'il fonctionne en mode fichiers locaux, tel que décrit à la section “Configuration d'un système en mode fichiers locaux” à la page 40. Cette configuration garantit l'initialisation du routeur en cas de panne du serveur de configuration.

1 **Connectez-vous en tant qu'administrateur.**

Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2 **Configurez les interfaces IP sur les cartes réseau du système.**

Pour obtenir des instructions de configuration détaillées, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

Assurez-vous que chaque interface IP est configurée avec l'adresse IP du réseau pour lequel le système acheminera les paquets. Par conséquent, si le système sert les réseaux `192.168.5.0` et `10.0.5.0`, il faut configurer une NIC pour chaque réseau.

Attention – Si vous souhaitez configurer des routeurs IPv4 pour utiliser DHCP, vous devez maîtriser l'administration DHCP.

3 **Ajoutez le nom d'hôte et l'adresse IP de chaque interface au fichier `/etc/inet/hosts`.**

Par exemple, supposons que les noms assignés aux deux interfaces du routeur 1 sont `krakatoa` et `krakatoa-1`, respectivement. Les entrées dans le fichier `/etc/inet/hosts` seraient comme suit :

```
192.168.5.1      krakatoa        #interface for network 192.168.5.0
10.0.5.1         krakatoa-1      #interface for network 10.0.5.0
```

4 **Effectuez le reste des étapes pour configurer ce routeur pour qu'il s'exécute en mode fichiers locaux.**

Reportez-vous à la section "Configuration d'un système en mode fichiers locaux" à la page 40.

5 **Si le routeur est connecté à un réseau comportant des sous-réseaux, ajoutez le numéro de réseau et le masque de réseau au fichier `/etc/inet/netmasks`.**

Par exemple, pour une adresse IPv4 de numérotation classique, telle que `192.168.5.0`, vous devez taper :

```
192.168.5.0    255.255.255.0
```

6 **Activez le transfert de paquets IPv4 sur le routeur.**

```
# ipadm set-prop -p forwarding=on ipv4
```

7 **(Facultatif) Lancez le protocole de routage.**

Utilisez l'une des syntaxes de commande suivantes :

- `# routeadm -e ipv4-routing -u`
- `# svcadm enable route:default`

  Le FMRI SMF associé au démon `in.routed` est `svc:/network/routing/route`.

Lorsque vous démarrez un protocole de routage, le démon de routage `/usr/sbin/in.routed` met automatiquement à jour la table de routage ; il s'agit d'un processus appelé *routage dynamique*. Pour plus d'informations sur les types de routage, reportez-vous à la section "Tables et types de routage" à la page 46. Pour plus d'informations sur la commande `routeadm`, reportez-vous à la page de manuel `routeadm(1M)`.

Exemple 3–1 Configuration du routeur par défaut d'un réseau

Cet exemple est basé sur la Figure 3–1. Le routeur 2 contient deux connexions réseau câblées, une connexion au réseau `172.16.1.0` et une connexion au réseau `10.0.5.0`. L'exemple indique comment configurer Router 2 pour qu'il soit le routeur par défaut du réseau `172.16.1.0`. L'exemple suppose également que le routeur 2 a été configuré afin de fonctionner en mode fichiers locaux, tel que décrit dans la section "Configuration d'un système en mode fichiers locaux" à la page 40.

Prenez le rôle de superutilisateur ou un rôle équivalent, puis déterminez l'état des interfaces du système.

```
# dladm show-link
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --
net2     phys      1500    up      --       --
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
```

Seule `net0` a été configurée avec une adresse IP. Pour que le routeur 2 soit le routeur par défaut, vous devez connecter l'interface `net1` au réseau `10.0.5.0`.

```
# ipadm create-ip net1
# ipadm create-addr -a 10.0.5.10/24 net1
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           static   ok           172.16.1.10/24
net1/v4           static   ok           10.0.5.10/24
```

Ensuite, mettez à jour les bases de données réseau suivantes à l'aide des informations sur l'interface que vous venez de configurer et le réseau auquel elle est connectée.

```
# vi /etc/inet/hosts
127.0.0.1       localhost
172.16.1.10        router2        #interface for network 172.16.1
10.0.5.10          router2-out    #interface for network 10.0.5
# vi /etc/inet/netmasks
172.16.1.0    255.255.255.0
10.0.5.0      255.255.255.0
```

Enfin, activez le transfert de paquets ainsi que le démon de routage `in.routed`.

```
# ipadm set-prop -p forwarding=on ipv4
# svcadm enable route:default
```

La transmission de paquets IPv4 et le routage dynamique via RIP sont maintenant activés sur le Routeur 2. La configuration du routeur par défaut pour le réseau `172.16.1.0` n'est cependant pas terminée. Procédez comme suit :

- Modifiez les hôtes du réseau `172.16.1.0` pour qu'ils reçoivent leurs informations de routage du nouveau routeur par défaut. Pour plus d'informations, reportez-vous à la section "Activation du routage statique sur un hôte à interface unique" à la page 52.
- Définissez une route statique menant au routeur de bordure dans la table de routage du Routeur 2. Pour plus d'informations, reportez-vous à la section "Tables et types de routage" à la page 46.

Tables et types de routage

Les routeurs et les hôtes maintiennent une *table de routage*. La table de routage dresse la liste des adresses IP des réseaux connus du système, notamment le réseau local par défaut. Elle répertorie également la liste des adresses IP d'un système de passerelle pour chaque réseau connu. La *passerelle* est un système qui peut recevoir des paquets sortants et les transférer au saut au-delà du réseau local.

Ce qui suit est une table de routage simple pour un système sur un réseau IPv4 uniquement :

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    532   net0
224.0.0.0            10.0.5.100           U         1      0   net1
10.0.0.0             10.0.5.100           U         1      0   net1
127.0.0.1            127.0.0.1            UH        1     57   lo0
```

Vous pouvez configurer deux types de routage sur un système Oracle Solaris : statique et dynamique. Vous pouvez configurer l'un ou l'autre, ou les deux sur un même système. Un système qui implémente le *routage dynamique* repose sur les protocoles de routage, notamment RIP pour les réseaux IPv4 et RIPng pour les réseaux IPv6, pour acheminer le trafic réseau et pour mettre à jour les informations de routage dans la table. Avec le *routage statique*, les informations de routage sont conservées manuellement par l'intermédiaire de l'utilisation de la commande `route`. Pour plus d'informations, reportez-vous à la page de manuel `route(1M)`.

Lors de la configuration du routage du réseau local ou d'un système autonome, réfléchissez au type de routage à prendre en charge sur des hôtes et des routeurs particuliers.

Le tableau suivant présente les différents types de routage et les scénarios de mise en réseau auquel chaque type de routage convient le mieux.

Type de routage	Utilisation privilégiée
Statique	Réseaux de petite taille, hôtes qui obtiennent leurs routes d'un routeur par défaut et routeurs par défaut qui n'ont besoin de connaître qu'un ou deux routeurs sur les quelques sauts suivants.
Dynamique	Interréseaux volumineux, routeurs sur des réseaux locaux comportant de nombreux hôtes et hôtes sur des systèmes autonomes d'envergure. Le routage dynamique représente le meilleur choix pour les systèmes résidant sur la plupart des réseaux.
Combinaison statique-dynamique	Routeurs effectuant la connexion entre un réseau au routage statique et un réseau au routage dynamique, et routeurs de bordure reliant un système interne autonome aux réseaux externes. La combinaison routage statique et routage dynamique est pratique courante.

Sur la Figure 3–1, l'AS allie le routage statique au routage dynamique.

Remarque – Lorsque deux routes présentent la même destination, le système ne procède pas automatiquement à un basculement ou à un équilibrage des charges. Pour bénéficier de ces fonctions, utilisez IPMP, conformément à la description donnée au Chapitre 5, "Présentation du multipathing sur réseau IP (IPMP)" du manuel *Gestion des performances du réseau Oracle Solaris 11.1*.

▼ Ajout d'une route statique à la table de routage

1 Affichez l'état actuel de la table de routage.

Pour exécuter la forme suivante de la commande `netstat`, utilisez votre compte utilisateur standard :

```
% netstat -rn
```

La sortie doit ressembler à ceci :

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
192.168.5.125        192.168.5.10          U      1   5879   net0
224.0.0.0            198.168.5.10          U      1  0       net0
default              192.168.5.10          UG     1  91908
127.0.0.1            127.0.0.1             UH     1  811302   lo0
```

2 Connectez-vous en tant qu'administrateur.

Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

3 (Facultatif) Supprimez les entrées existantes de la table de routage.

```
# route flush
```

4 Ajoutez une route qui persiste aux réinitialisations du système.

```
# route -p add -net network-address -gateway gateway-address
```

`-p`	Crée une route qui doit être conservée après les réinitialisations du système. Si vous souhaitez configurer la route pour la session en cours uniquement, n'utilisez pas l'option `-p`.
`-net` *network-address*	Indique que la route intègre le réseau avec l'adresse *network-address*.
`-gateway` *gateway-address*	Indique que le système de passerelle pour la route spécifiée possède l'adresse IP *gateway-address*.

Exemple 3–2 Ajout d'une route statique à la table de routage

L'exemple suivant indique comment ajouter une route statique au routeur 2 de la Figure 3–1. La route statique est nécessaire pour le routeur de bordure de l'AS, `10.0.5.150`.

Pour afficher la table de routage sur Router 2, effectuez l'opération suivante :

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          U         1      0 ce0
10.0.5.0             10.0.5.20            U         1     78 bge0
127.0.0.1            127.0.0.1            UH        1     57 lo0
```

D'après la table de routage, Router 2 a connaissance de deux routes. La route par défaut utilise l'interface `172.16.1.10` de Router 2 comme passerelle. La deuxième route, `10.0.5.0`, a été détectée par le démon `in.routed` exécuté sur Router 2. La passerelle de cette route est Routeur 1, avec l'adresse IP `10.0.5.20` .

Pour ajouter une seconde route au réseau `10.0.5.0`, dont la passerelle est le routeur de bordure, procédez comme suit :

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

La table de routage contient désormais une route destinée au routeur de bordure dont l'adresse IP est `10.0.5.150/24`.

```
# netstat -rn
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
default              172.16.1.10          UG        1    249 ce0
224.0.0.0            172.16.1.10          U         1      0 ce0
10.0.5.0             10.0.5.20            U         1     78 bge0
```

```
10.0.5.0           10.0.5.150         U         1    375 bge0
127.0.0.1          127.0.0.1          UH        1     57 lo0
```

Configuration des hôtes multiréseau

Dans Oracle Solaris, un système doté de plus d'une interface est considéré comme un *hôte multiréseau*. Les interfaces d'un hôte multiréseau se connectent à différents sous-réseaux, soit sur des réseaux physiques différents, soit sur le même réseau physique.

Sur un système dont les interfaces se connectent à un même sous-réseau, vous devez d'abord configurer les interfaces en tant que groupe IPMP. Dans le cas contraire, le système ne pourra pas être un hôte multiréseau. Pour plus d'informations sur IPMP, reportez-vous au Chapitre 5, "Présentation du multipathing sur réseau IP (IPMP)" du manuel *Gestion des performances du réseau Oracle Solaris 11.1*.

Un hôte multiréseau ne transfère pas les paquets IP, mais il peut être configuré afin d'exécuter des protocoles de routage. Les systèmes habituellement configurés en tant qu'hôtes multiréseau sont les suivants :

- Les serveurs NFS (en particulier ceux qui fonctionnent en tant que vastes centres de données) peuvent être reliés à plusieurs réseaux et permettre ainsi à un grand nombre d'utilisateurs de partager des fichiers. Ils ne doivent pas forcément gérer des tables de routage.
- Tout comme les serveurs NFS, les serveurs de bases de données peuvent posséder plusieurs interfaces réseau en vue de mettre des ressources à la disposition d'un grand nombre d'utilisateurs.
- Les passerelles pare-feu connectent un réseau d'entreprise avec des réseaux publics, tels qu'Internet. Un pare-feu constitue une mesure de sécurité mise en oeuvre par les administrateurs. Configuré en tant que pare-feu, l'hôte ne transmet pas de paquets entre les réseaux qui sont reliés à ses interfaces. Toutefois, l'hôte peut toujours fournir des services TCP/IP standard, tels que `ssh`, aux utilisateurs autorisés.

Remarque – Lorsque les pare-feux sur les interfaces d'un hôte multiréseau sont différents, évitez au maximum toute perturbation accidentelle des paquets de l'hôte. Ce problème se produit particulièrement avec les pare-feux avec état. Une des solutions consiste à configurer des pare-feux sans état. Pour plus d'informations sur les pare-feux, reportez-vous à la section "Systèmes pare-feu" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité* ou à la documentation de votre pare-feu tiers.

▼ Création d'un hôte multiréseau

1. **Connectez-vous en tant qu'administrateur.**

   Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2. **Configurez chaque interface réseau supplémentaire qui n'a pas été configurée lors de l'installation d'Oracle Solaris.**

   Reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

3. **Si le transfert de paquets est activé, désactivez ce service.**

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on           --           off          on,off

ipadm set-prop -p forwarding=off ipv4
```

4. **(Facultatif) Activez le routage dynamique pour l'hôte multiréseau.**

   Utilisez l'une des syntaxes de commande suivantes :

   - # **routeadm -e ipv4-routing -u**
   - # **svcadm enable route:default**

     Le FMRI SMF associé au démon `in.routed` est `svc:/network/routing/route`.

Exemple 3–3 Configuration d'un hôte multiréseau

L'exemple suivant illustre comment configurer l'hôte multiréseau représenté dans la Figure 3–1. Dans cet exemple, le nom d'hôte du système est `hostc`. Cet hôte présente deux interfaces connectées au réseau `192.168.5.0`.

Commencez par afficher l'état des interfaces du système.

```
# dladm show-link
LINK     CLASS     MTU     STATE   BRIDGE   OVER
net0     phys      1500    up      --       --
net1     phys      1500    up      --       --

# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
```

La commande `dladm show-link` rapporte que `hostc` dispose de deux liaisons de données. Cependant, seule `net0` a été configurée avec une adresse IP. Pour configurer `hostc` en tant

qu'hôte multiréseau, configurez `net1` avec une adresse IP dans le même réseau `192.168.5.0`. Assurez-vous que la NIC sous-jacente physique de `net1` est connectée physiquement au réseau.

```
# ipadm create-ip net1
# ipadm create-addr static -a 192.168.5.85/24 net1
# ipadm show-addr
ADDROBJ        TYPE     STATE        ADDR
lo0/v4         static   ok           127.0.0.1/8
net0/v4        static   ok           192.168.5.82/24
net1/v4        static   ok           192.168.5.85/24
```

Ensuite, ajoutez l'interface `net1` à la base de données `/etc/hosts` :

```
# vi /etc/inet/hosts
127.0.0.1           localhost
192.168.5.82        hostc    #primary network interface for host3
192.168.5.85        hostc-2  #second interface
```

Désactivez ensuite le transfert de paquets si ce service s'exécute sur `hostc` :

```
# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

# routeadm
              Configuration   Current              Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   enabled              enabled
               IPv6 routing   disabled             disabled

           Routing services   "route:default ripng:default"
```

La commande `routeadm` rapporte que le routage dynamique via le démon `in.routed` est actuellement activé.

Configuration du routage de systèmes à interface unique

Les systèmes à interface unique peuvent être configurés avec un routage dynamique ou statique. Avec le routage statique, l'hôte doit utiliser les services d'un routeur par défaut pour les informations de routage. Les sections suivantes décrivent les procédures d'activation des deux types de routage.

▼ Activation du routage statique sur un hôte à interface unique

Vous pouvez également suivre la procédure ci-dessous pour configurer le routage statique sur un hôte multiréseau.

1. **Connectez-vous en tant qu'administrateur.**

   Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2. **Configurez l'interface IP du système avec une adresse IP pour le réseau auquel appartient le système.**

   Pour obtenir des instructions, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

3. **Avec un éditeur de texte, créez ou modifiez le fichier `/etc/defaultrouter` en ajoutant l'adresse IP du routeur que le système utilisera.**

4. **Ajoutez une entrée pour le routeur par défaut dans le fichier local `/etc/inet/hosts` .**

5. **Assurez-vous que le routage est désactivé.**

```
# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   enabled             disabled
               IPv6 routing   disabled             disabled

          Routing services   "route:default ripng:default"

# svcadm disable route:default
```

6. **Assurez-vous que le transfert de paquets est désactivé.**

```
# # ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on        --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

Exemple 3–4 Configuration du routage statique sur un système à interface unique

L'exemple suivant explique comment configurer le routage statique pour `hostb`, un système à interface unique sur le réseau `172.16.1.0`, comme illustré dans la Figure 3–1. `hostb` doit utiliser Router 2 en tant que routeur par défaut. L'exemple suppose que vous avez déjà configuré l'interface IP du système.

Tout d'abord, connectez-vous à `hostb` en tant qu'administrateur. Vérifiez ensuite la présence du fichier `/etc/defaultrouter` sur le système :

```
# cd /etc
# ls | grep defaultrouter

# vi /etc/defaultrouter
172.16.1.10
```

L'adresse IP `172.16.1.10` appartient à Router 2.

```
# vi /etc/inet/hosts
127.0.0.1          localhost
172.16.1.18        host2    #primary network interface for host2
172.16.1.10        router2  #default router for host2

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4

# routeadm
   Configuration   Current             Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   enabled             disabled
               IPv6 routing   disabled             disabled

          Routing services   "route:default ripng:default"

# svcadm disable route:default
```

Activation du routage dynamique sur un système à interface unique

Le routage dynamique qui utilise un protocole de routage constitue le moyen le plus simple de gérer le routage dans un système.

1. **Connectez-vous en tant qu'administrateur.**

   Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2. **Configurez l'interface IP du système avec une adresse IP pour le réseau auquel appartient le système.**

   Pour obtenir des instructions, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

3. **Supprimez toute entrée dans le fichier `/etc/defaultrouter`.**

   Un fichier `/etc/defaultrouter` vide oblige le système à utiliser le routage dynamique.

4. **Assurez-vous que le transfert de paquets est désactivé.**

   ```
   # ipadm set-prop -p forwarding=off ipv4
   ```

5 **Activez les protocoles de routage sur le système.**

Exécutez l'une des commandes suivantes :

- `# routeadm -e ipv4-routing -u`
- `# svcadm enable route:default`

Exemple 3–5 Exécution du routage dynamique sur un système à interface unique

L'exemple suivant montre comment configurer le routage dynamique pour `hosta`, un système à interface unique sur le réseau `192.168.5.0` illustré dans la Figure 3–1. Le système utilise le routeur 1 en tant que routeur par défaut. L'exemple suppose que vous avez déjà configuré l'interface IP du système.

Tout d'abord, connectez-vous à `hosta` en tant qu'administrateur. Il faudrait ensuite supprimer le fichier `/etc/defaultrouter` s'il figure sur le système :

```
# cd /etc
# ls | grep defaultrouter
defaultrouter

# rm defaultrouter

# routeadm   Configuration   Current              Current
                     Option   Configuration        System State
---------------------------------------------------------------
               IPv4 routing   disabled             disabled
               IPv6 routing   disabled             disabled

           Routing services   "route:default ripng:default"

# svcadm enable route:default

# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY     PERM CURRENT   PERSISTENT   DEFAULT      POSSIBLE
ipv4  forwarding   rw   on          --           off          on,off

# ipadm set-prop -p forwarding=off ipv4
```

Ajout d'un sous-réseau à un réseau

Si vous passez d'un réseau n'utilisant pas de sous-réseau à un réseau qui en utilise un, effectuez les tâches figurant dans la liste suivante. La liste suppose que vous avez déjà préparé un schéma de sous-réseau.

- Assignez les adresses IP avec le nouveau numéro de sous-réseau aux systèmes qui appartiennent au sous-réseau.

Pour des références, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

- Ajoutez l'adresse IP et le masque de réseau corrects à chaque fichier `/etc/netmasks` de chaque système.
- Révisez chaque fichier `/et/inet/hosts` de chaque système avec l'adresse IP correcte de sorte qu'elle corresponde aux noms d'hôtes.
- Réinitialisez tous les systèmes dans le sous-réseau.

La procédure est étroitement liée aux sous-réseaux. Si vous implémentez la création de sous-réseaux bien après avoir effectué la configuration initiale du réseau sans cette création de sous-réseaux, effectuez la procédure suivante pour implémenter les modifications.

▼ Modification de l'adresse IPv4 et des autres paramètres de configuration réseau

Cette section décrit la procédure de modification de l'adresse IPv4, du nom d'hôte et des autres paramètres réseau d'un système déjà installé. Cette procédure permet de modifier l'adresse IP d'un serveur ou d'un système autonome en réseau. Elle ne s'applique pas aux appareils ou clients réseau. Cette procédure entraîne la création d'une configuration qui sera conservée après les réinitialisations du système.

Remarque – Les instructions s'appliquent explicitement à la modification de l'adresse IPv4 de l'interface réseau principale. Pour ajouter une autre interface au système, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

Dans la plupart des cas, les étapes suivantes font appel à la numérotation décimale avec points IPv4 classique afin de spécifier l'adresse IPv4 et le masque de sous-réseau. Vous pouvez aussi indiquer l'adresse IPv4 à l'aide de la numérotation CIDR dans tous les fichiers pertinents.

1 Connectez-vous en tant qu'administrateur.

Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2 Modifiez l'adresse IP en utilisant la commande `ipadm`.

La commande `ipadm` ne permet pas de modifier une adresse IP directement. Vous devez d'abord supprimer l'objet d'adressage qui représente l'adresse IP que vous souhaitez modifier. Vous pouvez ensuite affecter une nouvelle adresse à l'aide du même nom d'objet d'adressage.

```
# ipadm delete-addr addrobj
# ipadm create-addr -a IP-address interface
```

3. **Le cas échéant, modifiez l'entrée de nom d'hôte dans le service SMF `system/identity:node` :**

   `# hostname` *newhostname*

   Cette étape entraîne une modification persistante.

4. **En cas de changement du masque de sous-réseau, modifiez les entrées de sous-réseau dans le fichiers `/etc/netmasks`.**

5. **En cas de changement de l'adresse de sous-réseau, remplacez l'adresse IP du routeur par défaut dans `/etc/defaultrouter` par celle du routeur par défaut du nouveau sous-réseau.**

6. **Redémarrez le système.**

   ```
   # reboot -- -r
   ```

Exemple 3–6 Modification de l'adresse IP et du nom d'hôte

Cet exemple illustre la modification du nom d'hôte, de l'adresse IP de l'interface réseau principale et du masque de sous-réseau. L'adresse IP de l'interface réseau principale `net0` passe de `10.0.0.14` à `192.168.34.100`.

```
# ipadm show-addr
ADDROBJ      TYPE     STATE   ADDR
lo0/v4       static   ok      127.0.0.1/8
net0/v4      static   ok      10.0.0.14/24

# ipadm delete-addr net0/v4
# ipadm create-addr -a 192.168.34.100/24 net0
# hostname mynewhostname

# ipadm show-addr
ADDROBJ         TYPE     STATE   ADDR
lo0/v4          static   ok      127.0.0.1/8
net0/v4         static   ok      192.168.34.100/24

# hostname
mynewhostname
```

Voir aussi Pour modifier l'adresse IP d'une autre interface que l'interface réseau principale, reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

Contrôle et modification des services de couche transport

Les protocoles de couche de transport TCP, SCTP et UDP font partie du package Oracle Solaris standard. Généralement, ces protocoles fonctionnent correctement sans que l'utilisateur ait à intervenir. Toutefois, dans certaines conditions, vous serez peut-être amené à consigner ou modifier des services exécutés via les protocoles de couche transport. Vous devez ensuite modifier les profils de ces services dans l'utilitaire de gestion des services (SMF) décrit au Chapitre 1, "Gestion des services (présentation)" du manuel *Gestion des services et pannes dans Oracle Solaris 11.1*.

Le démon `inetd` est chargé de lancer les services Internet standard lors de l'initialisation d'un système. Ces services incluent les applications utilisant les protocoles de couche transport TCP, SCTP ou UDP. Vous pouvez modifier les services Internet existants ou ajouter de nouveaux services à l'aide des commandes SMF. Pour plus d'informations sur `inetd`, reportez-vous à la section "Démon de services Internet `inetd`" à la page 127.

Opérations impliquant les protocoles de couche transport :

- Journalisation de toutes les connexions TCP entrantes
- Ajout de services faisant appel à un protocole de couche transport, utilisant SCTP comme exemple
- Configuration des wrappers TCP dans le cadre du contrôle d'accès

Pour plus d'informations sur le démon `inetd`, reportez-vous à la page de manuel `inetd`(1M).

▼ Journalisation des adresses IP de toutes les connexions TCP entrantes

1 Connectez-vous en tant qu'administrateur.

Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2 Activez le suivi TCP pour tous les services gérés par `inetd`.

```
# inetadm -M tcp_trace=TRUE
```

▼ Ajout de services utilisant le protocole SCTP

Le protocole de transport SCTP fournit des services aux protocoles de couche d'application de façon similaire à TCP. Toutefois, SCTP permet la communication entre deux systèmes multiréseau ou deux systèmes dont l'un est multiréseau. La connexion SCTP s'appelle une

association. Dans une association, une application divise les données à transmettre en *plusieurs flux de messages*. Une connexion SCTP peut atteindre les extrémités à l'aide de plusieurs adresses IP, ce qui s'avère particulièrement important dans le cadre d'applications de téléphonie. Les capacités multiréseau de SCTP améliorent la sécurité des sites ayant recours à IP Filter ou IPsec. La page de manuel `sctp(7P)` répertorie les points à prendre en considération au niveau de la sécurité.

Par défaut, le protocole SCTP fait partie d'Oracle Solaris et ne nécessite aucune configuration supplémentaire. Toutefois, vous devrez peut-être configurer explicitement certains services de couche d'application pour utiliser SCTP. `echo` et `discard` sont des exemples d'applications. La procédure suivante illustre l'ajout d'un service d'écho qui utilise un socket de type SCTP bi-univoque.

Remarque – La procédure suivante permet également d'ajouter des services pour les protocoles de couche transport TCP et UDP.

La tâche suivante illustre l'ajout dans le référentiel SMF d'un service `inet` SCTP géré par le démon `inetd`. La tâche décrit ensuite la procédure d'ajout du service à l'aide des commandes SMF (Service Management Facility).

- Pour plus d'informations sur les commandes SMF, reportez-vous à la section "Utilitaires d'administration en ligne de commande SMF" du manuel *Gestion des services et pannes dans Oracle Solaris 11.1*.
- Pour plus d'informations sur la syntaxe, consultez les pages de manuel sur les commandes SMF citées dans la procédure.
- Pour plus d'informations sur SMF, reportez-vous à la page de manuel `smf(5)`.

Avant de commencer Avant d'effectuer la procédure suivante, créez un fichier manifeste pour le service. En exemple, la procédure fait référence à un fichier manifeste du service `echo` intitulé `echo.sctp.xml`.

1 Connectez-vous au système local avec un compte utilisateur disposant de privilèges d'écriture sur les fichiers système.

2 Modifiez le fichier `/etc/services` et ajoutez la définition du nouveau service.

Définissez le service à l'aide de la syntaxe suivante.

service-name |*port/protocol* | *aliases*

3 Ajoutez le nouveau service.

Accédez au répertoire de stockage du manifeste de service et tapez ce qui suit :

```
# cd dir-name
# svccfg import service-manifest-name
```

La page de manuel svccfg(1M) contient la syntaxe complète de `svccfg`.

Admettons que vous voulez ajouter un service `echo` SCTP à l'aide du manifeste `echo.sctp.xml` résidant dans le répertoire `service.dir`. Vous devez taper ce qui suit :

```
# cd service.dir
# svccfg import echo.sctp.xml
```

4 Assurez-vous que le manifeste de service a été ajouté :

```
# svcs FMRI
```

Pour l'argument *FMRI*, utilisez le FMRI (Fault Managed Resource Identifier, identificateur de ressources gérées erronées) du manifeste de service. Par exemple, pour le service SCTP `echo`, vous devez utiliser la commande suivante :

```
# svcs svc:/network/echo:sctp_stream
```

La sortie doit ressembler à ceci :

```
    STATE          STIME    FMRI
disabled       16:17:00 svc:/network/echo:sctp_stream
```

Pour plus d'informations sur la commande `svcs`, reportez-vous à la page de manuel svcs(1).

D'après la sortie, le nouveau manifeste de service est désactivé.

5 Dressez la liste des propriétés du service afin d'identifier les modifications à apporter.

```
# inetadm -l FMRI
```

Pour plus d'informations sur la commande `inetadm`, reportez-vous à la page de manuel inetadm(1M).

Par exemple, pour le service SCTP `echo`, vous devez saisir les informations suivantes :

```
# inetadm -l svc:/network/echo:sctp_stream
SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
         .
         .
         default  tcp_trace=FALSE
           default  tcp_wrappers=FALSE
```

6 Activez le nouveau service :

```
# inetadm -e FMRI
```

7 Assurez-vous que le service est activé.

Par exemple, pour le nouveau service echo, vous devez taper :

```
# inetadm | grep sctp_stream
.
.
    enabled   online         svc:/network/echo:sctp_stream
```

Exemple 3–7 Ajout d'un service utilisant le protocole de transport SCTP

L'exemple suivant indique les commandes à utiliser et les entrées de fichier requises pour que le service d'écho utilise le protocole de couche transport SCTP.

```
$ cat /etc/services
.
.
echo            7/tcp
echo            7/udp
echo            7/sctp

# cd service.dir

    # svccfg import echo.sctp.xml

# svcs network/echo*
    STATE          STIME    FMRI
    disabled       15:46:44 svc:/network/echo:dgram
    disabled       15:46:44 svc:/network/echo:stream
    disabled       16:17:00 svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE
             name="echo"
             endpoint_type="stream"
             proto="sctp"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
    disabled  disabled       svc:/network/echo:stream
```

```
disabled  disabled       svc:/network/echo:dgram
enabled   online         svc:/network/echo:sctp_stream
```

Contrôle d'accès aux services TCP à l'aide des wrappers TCP

Le programme `tcpd` met en oeuvre les *wrappers TCP*. Les wrappers TCP représentent une mesure de sécurité supplémentaire pour les démons de services, notamment pour `ftpd`. En effet, ils s'interposent entre le démon et les demandes de service entrantes. Les wrappers TCP consignent les réussites et les échecs des tentatives de connexion. En outre, ils offrent un contrôle d'accès en autorisant ou en refusant la connexion en fonction de l'origine de la demande. Enfin, ils permettent de protéger les démons, notamment SSH, Telnet et FTP. L'application `sendmail` peut également utiliser des wrappers TCP, comme décrit à la section "Prise en charge des wrappers TCP à partir de la version 8.12 de sendmail" du manuel *Gestion des services sendmail dans Oracle Solaris 11.1*.

1. **Connectez-vous en tant qu'administrateur.**

   Pour plus d'informations, reportez-vous à la section "Utilisation de vos droits d'administration" du manuel *Administration d'Oracle Solaris 11.1 : Services de sécurité*.

2. **Activez les wrappers TCP.**

   ```
   # inetadm -M tcp_wrappers=TRUE
   ```

3. **Configurez la stratégie de contrôle d'accès des wrappers TCP, telle que décrite à la page de manuel `hosts_access`(3).**

   Cette page de manuel se trouve dans le répertoire `/usr/sfw/man`.

◆ ◆ ◆ CHAPITRE 4

Activation d'IPv6 sur le réseau

Ce chapitre contient les informations d'activation du protocole IPv6 sur un réseau. Il aborde les principaux thèmes suivants :

Configuration d'une interface IPv6

L'étape initiale de l'utilisation d'IPv6 sur un réseau consiste à configurer IPv6 sur l'interface IP du système.

Lors de l'installation d'Oracle Solaris, vous pouvez activer le protocole IPv6 sur une ou plusieurs interfaces d'un système. Si vous activez la prise en charge d'IPv6 pendant l'installation, une fois celle-ci terminée, les fichiers et tables IPv6 suivants sont en place :

- Le service SMF `name-service/switch` a été modifié pour prendre en charge les recherches utilisant les adresses IPv6.
- La table des règles de sélection des adresses IPv6 est créée. Cette table définit l'ordre de priorité des formats d'adresse IP à utiliser pour la transmission des données sur une interface IPv6.

Cette section décrit comment activer IPv6 sur les interfaces une fois l'installation d'Oracle Solaris terminée.

▼ Configuration d'un système pour IPv6

La première étape du processus de configuration IPv6 consiste à activer le protocole sur les interfaces des systèmes à définir en tant que noeuds IPv6. En principe, l'adresse IPv6 de l'interface est définie via le processus de configuration automatique décrit à la section "Processus de configuration automatique" à la page 146. Vous pouvez alors personnaliser la configuration du noeud selon sa fonction au sein du réseau IPv6 (hôte, serveur ou routeur).

Remarque – Si l'interface est définie sur un lien sur lequel un routeur publie un préfixe IPv6, ce préfixe de site figure dans les adresses configurées automatiquement. Pour plus d'informations, reportez-vous à la section "Configuration d'un routeur compatible IPv6" à la page 66.

La procédure suivante explique comment activer le protocole IPv6 sur une interface ajoutée après l'installation d'Oracle Solaris.

1 Configurez l'interface IP en utilisant les commandes appropriées.

Reportez-vous à la section "Configuration d'une interface IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

Remarque – Lorsque vous attribuez l'adresse IP, veillez à utiliser l'option correcte pour attribuer une adresse IPv6 :

```
# ipadm create-addr -T addrconf interface
```

Pour ajouter davantage d'adresses, utilisez la syntaxe suivante :

```
# ipadm create-addr -a ipv6-address interface
```

---

2 Démarrez le démon IPv6 `in.ndpd`.

```
# /usr/lib/inet/in.ndpd
```

3 (Facultatif) Créez une route IPv6 statique par défaut.

```
# /usr/sbin/route -p add -inet6 default ipv6-address
```

4 (Facultatif) Créez un fichier `/etc/inet/ndpd.conf` définissant les paramètres des variables d'interface du noeud.

Si vous devez créer des adresses temporaires pour l'interface de l'hôte, reportez-vous à la section "Utilisation d'adresses temporaires pour une interface" à la page 69. Pour de plus amples informations sur `/etc/inet/ndpd.conf`, reportez-vous à la page de manuel `ndpd.conf(4)`, ainsi qu'à la section "Fichier de configuration `ndpd.conf`" à la page 133.

5 **(Facultatif) Pour afficher le statut des interfaces IP avec leurs configurations IPv6, saisissez la commande suivante :**

```
# ipadm show-addr
```

Exemple 4–1 Activation d'une interface IPv6 après l'installation

Cet exemple illustre l'activation du protocole IPv6 sur l'interface `net0`. Avant de commencer, vérifiez l'état de toutes les interfaces configurées sur le système.

```
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1/8
net0/v4   static   ok      172.16.27.74/24
```

L'interface `net0` est la seule interface actuellement configurée sur le système. Pour activer le protocole IPv6 sur cette interface, effectuez la procédure suivante :

```
# ipadm create-addr -T addrconf net0
# ipadm create-addr -a 2001:db8:3c4d:15:203/64 net0
# /usr/lib/inet/in.ndpd

# ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      172.16.27.74/24
net0/v6       addrconf   ok      fe80::203:baff:fe13:14e1/10
lo0/v6        static     ok      ::1/128
net0/v6a      static     ok      2001:db8:3c4d:15:203/64

# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

Étapes suivantes

- Pour configurer le noeud IPv6 en tant que routeur, reportez-vous à la section "Configuration d'un routeur IPv6" à la page 66.
- Pour désactiver la configuration automatique sur le noeud, reportez-vous à la section "Désactivation de la configuration automatique des adresses IPv6" à la page 65.
- Pour personnaliser un noeud et le définir en tant que serveur, reportez-vous aux suggestions de la section "Administration d'interfaces compatibles IPv6 sur des serveurs" à la page 74.

▼ Désactivation de la configuration automatique des adresses IPv6

En règle générale, la configuration automatique d'adresse permet de générer les adresses IPv6 pour les interfaces des hôtes et des serveurs. Cependant, la désactivation de la configuration automatique peut s'avérer nécessaire, en particulier pour configurer un jeton manuellement, suivant les explications de la section "Configuration d'un jeton IPv6" à la page 72.

1. **Créez un fichier `/etc/inet/ndpd.conf` pour le noeud.**

   Le fichier `/etc/inet/ndpd.conf` définit les variables d'interface pour le noeud. Pour désactiver la configuration automatique d'adresse d'une interface du serveur, le fichier doit contenir les éléments suivants :

   interface `StatelessAddrConf false`

   Pour désactiver la configuration automatique d'adresse de toutes les interfaces, utilisez l'entrée suivante :

   ```
   ifdefault StatelessAddrConf false
   ```

   Pour plus d'informations sur `/etc/inet/ndpd.conf`, reportez-vous à la page de manuel `ndpd.conf`(4), ainsi qu'à la section "Fichier de configuration `ndpd.conf`" à la page 133.

2. **Mettez le démon IPv6 à jour avec vos modifications.**

   ```
   # pkill -HUP in.ndpd
   ```

Configuration d'un routeur IPv6

Cette section décrit les tâches de configuration d'un routeur IPv6. En fonction des exigences de votre site, il se peut que vous ne deviez effectuer que certaines tâches.

▼ Configuration d'un routeur compatible IPv6

La procédure suivante part du principe que vous avez déjà configuré le système pour IPv6. Pour connaître les procédures, reportez-vous à la section "Configuration d'une interface IPv6" à la page 63.

1. **Configurez le transfert de paquets IPv6 sur toutes les interfaces du routeur.**

   ```
   # ipadm set-prop -p forwarding=on ipv6
   ```

2. **Démarrez le démon de routage.**

   Le démon `in.ripngd` gère le routage IPv6. Activez le routage IPv6 à l'aide de l'une des méthodes suivantes :

   - Utilisez la commande `routeadm` :

     ```
     # routeadm -e ipv6-routing -u
     ```

   - Utilisez la commande SMF adéquate :

     ```
     # svcadm enable ripng:default
     ```

   Pour obtenir des informations sur la syntaxe de la commande `routeadm`, reportez-vous à la page de manuel `routeadm`(1M).

3. **Créez le fichier `/etc/inet/ndpd.conf`.**

   Spécifiez le préfixe de site que doit publier le routeur et les autres informations de configuration dans `/etc/inet/ndpd.conf`. Ce fichier est lu par le démon `in.ndpd`, qui implémente le protocole de détection de voisins IPv6.

   Pour obtenir une liste des variables et des valeurs autorisables, reportez-vous à la section "Fichier de configuration `ndpd.conf`" à la page 133 et à la page de manuel `ndpd.conf`(4).

4. **Saisissez le texte suivant dans le fichier `/etc/inet/ndpd.conf` :**

   ```
   ifdefault AdvSendAdvertisements true
   prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on
   ```

   Ce texte indique au démon `in.ndpd` qu'il doit envoyer les publications de routeur à toutes les interfaces du routeur qui sont configurées pour IPv6.

5. **Ajoutez du texte supplémentaire au fichier `/etc/inet/ndpd.conf` pour configurer le préfixe de site sur les différentes interfaces du routeur.**

   Le texte doit posséder le format suivant :

   `prefix` *global-routing-prefix*`:`*subnet ID*`/64` *interface*

   Le fichier d'exemple `/etc/inet/ndpd.conf` suivant configure le routeur de sorte qu'il publie le préfixe de site `2001:0db8:3c4d::/48` sur les interfaces `net0` et `net1`.

   ```
   ifdefault AdvSendAdvertisements true
   prefixdefault AdvOnLinkFlag on AdvAutonomousFlag on

   if net0 AdvSendAdvertisements 1
   prefix 2001:0db8:3c4d:15::0/64 net0

   if net1 AdvSendAdvertisements 1
   prefix 2001:0db8:3c4d:16::0/64 net1
   ```

6. **Redémarrez le système.**

   Le routeur IPv6 commence la publication sur la liaison locale de tout préfixe de site dans le fichier `ndpd.conf`.

Exemple 4–2 Sortie `ipadm show-addr` indiquant les interfaces IPv6

L'exemple suivant illustre la sortie de la commande `ipadm show-addr` telle que vous la recevriez une fois la procédure "Configuration d'un routeur IPv6" à la page 66 terminée.

```
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
net0/v4      static     ok      172.16.15.232/24
net1/v4      static     ok      172.16.16.220/24
net0/v6      addrconf   ok      fe80::203:baff:fe11:b115/10
lo0/v6       static     ok      ::1/128
net0/v6a     static     ok      2001:db8:3c4d:15:203:baff:fe11:b115/64
net1/v6      addrconf   ok      fe80::203:baff:fe11:b116/10
net1/v6a     static     ok      2001:db8:3c4d:16:203:baff:fe11:b116/64
```

Dans cet exemple, chaque interface configurée pour IPv6 possède maintenant deux adresses. L'entrée avec le nom d'objet d'adresse comme *interface*`/v6` indique l'adresse lien-local de l'interface. L'entrée avec le nom d'objet d'adresse comme *interface* `/v6add` indique une adresse globale IPv6. Cette adresse inclut le préfixe de site que vous avez configuré dans le fichier `/etc/ndpd.conf`, en plus de l'ID d'interface. Notez que la désignation `v6add` est une chaîne définie de façon aléatoire. Vous pouvez définir d'autres chaînes pour constituer la seconde partie du nom d'objet d'adresse, à condition que l'*interface* reflète l'interface sur laquelle vous créez les adresses IPv6, par exemple `net0/mystring`, `net0/ipv6addr` et ainsi de suite.

Voir aussi

- Pour configurer des tunnels à partir des routeurs identifiés dans la topologie de réseau IPv6, reportez-vous à la section "Configuration et administration du tunnel avec la commande `dladm`" à la page 112.
- Pour obtenir des informations sur la configuration de commutateurs et de hubs sur votre réseau, reportez-vous à la documentation du fabricant.
- Pour configurer les hôtes IPv6, reportez-vous à la section "Modification de la configuration d'interface IPv6 pour les hôtes et les serveurs" à la page 68.
- Pour améliorer la prise en charge d'IPv6 sur les serveurs, reportez-vous à la section "Administration d'interfaces compatibles IPv6 sur des serveurs" à la page 74.
- Pour plus d'informations sur les commandes, fichiers et démons IPv6, reportez-vous à la section "Implémentation IPv6 sous Oracle Solaris" à la page 133.

Modification de la configuration d'interface IPv6 pour les hôtes et les serveurs

Cette section explique comment modifier la configuration d'interfaces compatibles IPv6 sur les noeuds qui sont des hôtes ou des serveurs. Dans la plupart des cas, il est conseillé d'utiliser la configuration automatique d'adresse des interfaces IPv6. Vous pouvez cependant, le cas échéant, modifier l'adresse IPv6 d'une interface comme expliqué dans les tâches décrites dans cette section.

Vous devez effectuer trois tâches générales dans la séquence suivante :

1. Désactivation de la configuration automatique de l'adresse IPv6. Reportez-vous à la section "Désactivation de la configuration automatique des adresses IPv6" à la page 65.
2. Créez une adresse temporaire pour un hôte. Reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.
3. Configurez un jeton IPv6 pour l'ID d'interface. Reportez-vous à la section "Configuration d'un jeton IPv6 spécifié par l'utilisateur" à la page 72.

Utilisation d'adresses temporaires pour une interface

Une *adresse temporaire* IPv6 contient un numéro de 64 bits généré de manière aléatoire en tant qu'ID d'interface, plutôt que l'adresse MAC d'une interface. Vous pouvez utiliser des adresses temporaires pour toute interface d'un noeud IPv6 dont vous souhaitez préserver l'anonymat. Par exemple, il peut s'avérer utile d'employer des adresses temporaires pour les interfaces d'un hôte devant accéder à des serveurs Web publics. Les adresses temporaires implémentent des améliorations de confidentialité pour IPv6. Ces améliorations sont décrites dans le document RFC 3041, disponible à l'adresse "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" (`http://www.ietf.org/rfc/rfc3041.txt?number=3041`).

L'activation d'une adresse temporaire s'effectue dans le fichier `/etc/inet/ndpd.conf`, pour une ou plusieurs interfaces, le cas échéant. Cependant, à la différence des adresses IPv6 standard configurées automatiquement, une adresse temporaire se compose d'un préfixe de sous-réseau de 64 bits et d'un numéro de 64 bits généré de façon aléatoire. Ce numéro devient le segment correspondant à l'ID d'interface de l'adresse IPv6. Une adresse lien-local n'est pas générée avec l'adresse temporaire en tant qu'ID d'interface.

Notez que la *durée de vie préférée* par défaut des adresses temporaires est d'un jour. Lors de l'activation de la génération d'adresses temporaires, il est également possible de configurer les variables suivantes dans le fichier `/etc/inet/ndpd.conf` :

Durée de vie valide TmpValidLifetime	Durée d'existence de l'adresse temporaire ; une fois la durée écoulée, l'adresse est supprimée de l'hôte.
Durée de vie préférée TmpPreferredLifetime	Temps écoulé avant que l'adresse temporaire soit désapprouvée. Cette durée doit être inférieure à la durée de vie valide.
Régénération d'adresse	Durée avant l'expiration de la durée de vie préférée, pendant laquelle l'hôte devrait générer une nouvelle adresse temporaire.

La durée des adresses temporaires s'exprime comme suit :

n	*n* nombre de secondes, valeur par défaut
n h	*n* nombre d'heures (h)
n d	*n* nombre de jours (d)

▼ Configuration d'une adresse temporaire

1 Si nécessaire, activez IPv6 sur les interfaces de l'hôte.

Reportez-vous à la section "Configuration d'un système pour IPv6" à la page 64.

2 Modifiez le fichier `/etc/inet/ndpd.conf` afin d'activer la génération d'adresses temporaires.

- Pour configurer des adresses temporaires sur les interfaces d'un hôte, ajoutez la ligne suivante au fichier `/etc/inet/ndpd.conf` :

  `ifdefault TmpAddrsEnabled true`
- Pour configurer une adresse temporaire pour une interface spécifique, ajoutez la ligne suivante au fichier `/etc/inet/ndpd.conf` :

  `if` *interface* **`TmpAddrsEnabled true`**

3 (Facultatif) Spécifiez la durée de vie valide de l'adresse temporaire.

`ifdefault TmpValidLifetime` *duration*

Cette syntaxe spécifie la durée de vie valide de toutes les interfaces d'un hôte. La durée *duration* s'exprime en secondes, en heures ou en jours. La durée de vie valide par défaut est de 7 jours. Vous pouvez également utiliser `TmpValidLifetime` avec des mots-clés d'*interface* `if` afin de spécifier la durée de vie valide de l'adresse temporaire d'une interface en particulier.

4 (Facultatif) Spécifiez une durée de vie préférée pour l'adresse temporaire après laquelle celle-ci est désapprouvée.

`if` *interface* **`TmpPreferredLifetime`** *duration*

Cette syntaxe spécifie la durée de vie préférée de l'adresse temporaire d'une interface donnée. La durée de vie préférée par défaut est d'un jour. Vous pouvez également utiliser `TmpPreferredLifetime` avec le mot-clé `ifdefault` afin de spécifier la durée de vie préférée des adresses temporaires de toutes les interfaces d'un hôte.

Remarque – La sélection d'adresse par défaut attribue une priorité moindre aux adresses IPv6 désapprouvées. Si une adresse temporaire IPv6 est désapprouvée, la sélection d'adresses par défaut choisit une adresse qui n'a pas été désapprouvées en tant qu'adresse source d'un paquet. Une adresse non désapprouvée peut être l'adresse IPv6 générée automatiquement ou, éventuellement, l'adresse IPv4 de l'interface. Pour plus d'informations sur la sélection d'adresses par défaut, reportez-vous à la section "Administration de la sélection des adresses par défaut" à la page 100.

---

5 (Facultatif) Spécifiez la durée de production en avance de la désapprobation d'adresse, pendant laquelle l'hôte devrait générer une nouvelle adresse temporaire.

`ifdefault TmpRegenAdvance duration`

Cette syntaxe spécifie le délai qui doit s'écouler avant la désapprobation d'adresse pour les adresses temporaires de toutes les interfaces d'un hôte. La valeur par défaut est 5 secondes.

6 Modifiez la configuration du démon `in.ndpd`.

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

7 **Vérifiez que des adresses temporaires ont bien été créées en exécutant la commande `ipadm show-addr`, comme indiqué dans l'Exemple 4–4.**

La sortie de la commande affiche l'indicateur `t` dans le champ `CURRENT` des adresses temporaires.

Exemple 4–3 Variables d'adresses temporaires dans le fichier `/etc/inet/ndpd.conf`

L'exemple suivant comporte un segment d'un fichier `/etc/inet/ndpd.conf` avec les adresses temporaires activées pour l'interface du réseau principal.

```
ifdefault TmpAddrsEnabled true

ifdefault TmpValidLifetime 14d

ifdefault TmpPreferredLifetime 7d

ifdefault TmpRegenAdvance 6s
```

Exemple 4–4 Sortie de commande `ipadm show-addr` avec adresses temporaires activées

Cet exemple indique la sortie de la commande `ipadm show-addr` une fois les adresses temporaires créées. Notez que seules les informations relatives à IPv6 sont incluses dans l'exemple de sortie.

```
# ipadm show-addr -o all
ADDROBJ   TYPE     STATE CURRENT PERSISTENT ADDR
lo0/v6    static   ok    U----   ---        ::1/128
net0/v6   addrconf ok    U----   ---        fe80::a00:20ff:feb9:4c54/10
net0/v6a  static   ok    U----   ---        2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?    addrconf ok    U--t-   ---        2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

Notez que pour l'objet d'adresse `net0/?`, l'indicateur `t` est défini sous le champ `CURRENT`. L'indicateur informe que l'adresse correspondante est dotée d'un ID d'interface temporaire.

Voir aussi

- Pour définir la prise en charge du service de noms pour les adresses IPv6, reportez-vous à la section "Configuration de prise en charge de services de noms pour IPv6" à la page 75.
- Pour configurer des adresses IPv6 pour un serveur, reportez-vous à la section "Configuration d'un jeton IPv6 spécifié par l'utilisateur" à la page 72.
- Pour contrôler les activités sur les nœuds IPv6, reportez-vous au Chapitre 5, "Administration d'un réseau TCP/IP".

Configuration d'un jeton IPv6

L'ID d'interface 64 bits d'une adresse IPv6 est également appelée *jeton*. Lors de la configuration automatique d'adresses, le jeton est associé à l'adresse MAC de l'interface. Dans la plupart des cas, les noeuds qui n'effectuent pas de routage, c'est-à-dire les hôtes et les serveurs IPv6, doivent utiliser leurs jetons configurés automatiquement.

Cependant, l'utilisation de jetons configurés automatiquement peut être problématique pour les serveurs dont les interfaces sont régulièrement dans le cadre de la maintenance système. Lorsque la carte de l'interface est modifiée, l'adresse MAC l'est également. Cela peut entraîner des problèmes pour les serveurs qui dépendent d'adresses IP. Différentes parties de l'infrastructure de réseau, comme le DNS ou le NIS, peuvent disposer d'adresses IPv6 stockées pour les interfaces du serveur.

Pour les problèmes liés aux modifications d'adresses, vous pouvez configurer un jeton manuellement pour l'utiliser en tant qu'ID d'interface dans une adresse IPv6. Pour créer le jeton, spécifiez un numéro hexadécimal de 64 bits maximum afin d'occuper la portion d'ID d'interface de l'adresse IPv6. Par la suite, lors de la configuration automatique d'adresses, le protocole de détection de voisins ne crée pas d'ID d'interface basé sur l'adresse MAC de l'interface. Le jeton créé manuellement devient l'ID d'interface. Ce jeton reste assigné à l'interface, même en cas de remplacement d'une carte.

Remarque – La différence entre les jetons spécifiés par les utilisateurs et les adresses temporaires réside dans le fait que ces dernières sont générées de façon aléatoire et non pas créées explicitement par un utilisateur.

▼ Configuration d'un jeton IPv6 spécifié par l'utilisateur

Les instructions suivantes sont particulièrement utiles pour les serveurs dont les interfaces sont régulièrement remplacées. Elles sont également valides pour la configuration de jetons spécifiés par l'utilisateur sur tout noeud IPv6.

1 Vérifiez que l'interface que vous souhaitez configurer avec un jeton existe et qu'aucune adresse IPv6 n'est configurée sur l'interface.

Remarque – Assurez-vous que l'interface n'est dotée d'aucune adresse IPv6 configurée.

```
# ipadm show-if
IFNAME   CLASS      STATE   ACTIVE   OVER
lo0      loopback   ok       yes     ---
net0     ip         ok       yes     ---

# ipadm show-addr
```

```
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
```

Cette sortie indique que l'interface réseau `net0` existe sans adresse IPv6 configurée.

2 Créez un ou plusieurs numéros hexadécimaux de 64 bits à utiliser en tant que jetons pour les interfaces du noeud en respectant le format *xxxx*:*xxxx*:*xxxx*:*xxxx*.

3 Configurez chaque interface avec un jeton.

Utilisez le format suivant de la commande `ipadm` pour chaque interface afin de disposer d'un ID d'interface spécifiée par l'utilisateur (jeton) :

```
# ipadm create-addr -T addrconf -i interface-ID interface
```

Par exemple, exécutez la commande suivante afin de configurer l'interface `net0` avec un jeton :

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
```

Remarque – Une fois l'objet d'adresse créé avec le jeton, ce dernier ne peut plus être modifié.

4 Mettez le démon IPv6 à jour avec vos modifications.

```
# pkill -HUP in.ndpd
```

Exemple 4–5 Configuration d'un jeton spécifié par l'utilisateur sur une interface IPv6

L'exemple suivant représente `net0` en cours de configuration avec une adresse IPv6 et un jeton.

```
# ipadm show-if
IFNAME   CLASS      STATE    ACTIVE     OVER
lo0      loopback   ok       yes        ---
net0     ip         ok       yes        ---

# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8

# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
# pkill -HUP in.ndpd
# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v6       static     ok      ::1/128
net0/v6      addrconf   ok      fe80::1a:2b:3c:4d/10
net0/v6a     addrconf   ok      2002:a08:39f0:1:1a:2b:3c:4d/64
```

Une fois le jeton configuré, l'objet d'adresse `net0/v6` dispose à la fois de l'adresse locale du lien ainsi que d'une adresse avec `1a:2b:3c:4d` configurée pour son ID d'interface. Notez qu'après la création de `net0/v6`, ce jeton ne peut plus être modifié pour cette interface.

Voir aussi

- Pour la mise à jour des services de noms pour les adresses IPv6 du serveur, reportez-vous à la section "Configuration de prise en charge de services de noms pour IPv6" à la page 75.
- Pour contrôler les performances du serveur, reportez-vous au Chapitre 5, "Administration d'un réseau TCP/IP".

Administration d'interfaces compatibles IPv6 sur des serveurs

Lors de la planification d'IPv6 sur un serveur, vous devez prendre un certain nombre de décisions relatives à l'activation d'IPv6 sur les interfaces du serveur. Vos décisions affectent la stratégie à utiliser pour la configuration des ID d'interface, également appelés *jetons*, de l'adresse IPv6 d'une interface.

▼ Activation d'IPv6 sur les interfaces d'un serveur

Cette procédure indique les étapes générales permettant d'activer IPv6 sur les serveurs de votre réseau. Certaines étapes peuvent varier en fonction de la manière dont vous souhaitez implémenter IPv6.

1. **Activez IPv6 sur les interfaces IP du serveur.**

   Pour connaître les procédures, reportez-vous à la section "Configuration d'une interface IPv6" à la page 63.

2. **Assurez-vous qu'un préfixe de sous-réseau IPv6 est configuré sur un routeur situé sur la même liaison que le serveur.**

   Pour plus d'informations, reportez-vous à la section "Configuration d'un routeur IPv6" à la page 66.

3. **Utilisez la stratégie adéquate pour l'ID des interfaces compatibles IPv6 du serveur.**

   Par défaut, la configuration automatique d'adresses IPv6 utilise l'adresse MAC d'une interface lors de la création de la partie ID d'interface de l'adresse IPv6. Si l'adresse IPv6 de l'interface est bien connue, remplacer une interface par une autre peut entraîner des problèmes. L'adresse MAC de la nouvelle interface sera différente. Un nouvel ID d'interface est généré lors de la configuration automatique d'adresses.

   - Dans le cas d'une interface compatible IPv6 que vous ne souhaitez pas remplacer, utilisez l'adresse IPv6 configurée automatiquement, comme indiqué à la section "Processus de configuration automatique" à la page 146.

- Dans le cas d'interfaces compatibles IPv6 devant apparaître anonymes hors du réseau local, vous pouvez utiliser un jeton généré de façon aléatoire comme ID d'interface. Pour obtenir des instructions et un exemple, reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.
- Dans le cas d'interfaces compatibles IPv6 que vous pensez échanger régulièrement, créez des jetons pour les ID d'interface. Pour obtenir des instructions et un exemple, reportez-vous à la section "Configuration d'un jeton IPv6 spécifié par l'utilisateur" à la page 72.

Configuration de prise en charge de services de noms pour IPv6

Cette section décrit la procédure de configuration des services de noms DNS et NIS pour la prise en charge de services IPv6.

Remarque – LDAP prend en charge IPv6 sans aucune configuration supplémentaire nécessaire.

Pour obtenir des informations détaillées sur l'administration de DNS, NIS et LDAP, reportez-vous à la section *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.

▼ Ajout d'adresses IPv6 à DNS

1 **Modifiez le fichier de zone DNS adéquat en ajoutant les enregistrements AAAA pour chaque noeud compatible IPv6 :**

hostname `IN AAAA` *host-address*

2 **Modifiez les fichiers de zone inversée DNS et ajoutez des enregistrements PTR :**

hostaddress `IN PTR` *hostname*

Pour obtenir des informations détaillées sur l'administration de DNS, reportez-vous à la section *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.

Exemple 4–6 Fichier de zone inversée DNS

Cet exemple représente une adresse IPv6 dans le fichier de zone inversée.

```
$ORIGIN    ip6.int.
8.2.5.0.2.1.e.f.f.f.9.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0 \
    IN        PTR        vallejo.Eng.apex.COM.
```

▼ Affichage des informations relatives au service de noms IPv6

La commande `nslookup` permet d'afficher des informations relatives au service de noms IPv6.

1. **Après vous être connecté à l'aide de votre compte utilisateur, exécutez la commande `nslookup`.**

   ```
   % /usr/sbin/nslookup
   ```

   Le nom et l'adresse par défaut du serveur s'affichent, suivis du crochet d'invite de la commande `nslookup`.

2. **Pour obtenir des informations sur un hôte en particulier, saisissez les commandes suivantes à partir du crochet d'invite :**

   ```
   >set q=any
   >hostname
   ```

3. **Saisissez la commande suivante afin d'afficher les enregistrements AAAA :**

   ```
   >set q=AAAA
   hostname
   ```

4. **Quittez la commande `nslookup` en saisissant `exit`.**

Exemple 4–7 Utilisation de `nslookup` pour l'affichage d'informations IPv6

Cet exemple illustre les résultats de l'exécution de `nslookup` dans un environnement de réseau IPv6.

```
%  /usr/sbin/nslookup
Default Server:  dnsserve.local.com
Address:  10.10.50.85
> set q=AAAA
> host85
Server:  dnsserve.local.com
Address:  10.10.50.85

host85.local.com      IPv6 address = 2::9256:a00:fe12:528
> exit
```

▼ Vérification de la mise à jour correcte des enregistrements PTR DNS IPv6

Dans cette procédure, utilisez la commande `nslookup` afin d'afficher les enregistrements PTR pour le service DNS IPv6.

1. **Une fois connecté à votre compte utilisateur, exécutez la commande `nslookup`.**

   ```
   % /usr/sbin/nslookup
   ```

   Le nom et l'adresse par défaut du serveur s'affichent, suivis du crochet d'invite de la commande `nslookup`.

2. **Saisissez ce qui suit devant le crochet d'invite afin de visualiser les enregistrements PTR :**

   ```
   >set q=PTR
   ```

3. **Quittez la commande en saisissant `exit`.**

Exemple 4–8 Utilisation de `nslookup` pour l'affichage d'enregistrements PTR

L'exemple suivant illustre l'affichage d'enregistrements PTR à l'aide de la commande `nslookup`.

```
%  /usr/sbin/nslookup
Default Server:  space1999.Eng.apex.COM
Address:  192.168.15.78
> set q=PTR
> 8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int

8.2.5.0.2.1.e.f.f.f.0.2.0.0.a.0.6.5.2.9.0.0.0.0.0.0.0.0.2.0.0.0.ip6.int name =
vallejo.ipv6.Eng.apex.COM
ip6.int nameserver = space1999.Eng.apex.COM
> exit
```

▼ Affichage d'informations IPv6 à l'aide de NIS

Dans cette procédure, la commande `ypmatch` permet d'afficher des informations IPv6 par le biais de NIS :

- **Une fois connecté à votre compte utilisateur, saisissez ce qui suit afin d'afficher les adresses IPv6 dans NIS :**

  ```
  % ypmatch hostname hosts .byname
  ```

  Les informations sur l'hôte *hostname* spécifié s'affichent.

◆ ◆ ◆ CHAPITRE 5

Administration d'un réseau TCP/IP

Ce chapitre présente les tâches permettant d'administrer un réseau TCP/IP. Cette partie comprend les rubriques suivantes :

Remarque – Pour surveiller les interfaces réseau, reportez-vous à la section "Contrôle d'interfaces et d'adresses IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

L'exécution des tâches présentées dans ce chapitre nécessite l'installation d'un réseau TCP/IP opérationnel sur votre site (IPv4 uniquement ou IPv4/IPv6 double pile). Pour plus d'informations sur l'implémentation d'un réseau IPv6, reportez-vous aux chapitres suivants :

- Pour planifier une implémentation IPv6, reportez-vous au Chapitre 2, "Eléments à prendre en compte lors de l'utilisation d'adresses IPv6".
- Pour configurer un réseau IPv6 et créer un environnement double pile, reportez-vous au Chapitre 4, "Activation d'IPv6 sur le réseau".

Principales tâches d'administration TCP/IP (liste des tâches)

Le tableau suivant répertorie les autres tâches permettant de gérer le réseau après la configuration initiale, notamment l'affichage des informations réseau. Le tableau comprend la description des actions de chaque tâche et la section de la documentation actuelle dans laquelle les étapes permettant d'effectuer ces tâches sont décrites en détails.

Tâche	Description	Référence
Affichage des statistiques par protocole.	Contrôlez les performances des protocoles réseau sur un système donné.	"Affichage des statistiques par protocole" à la page 81
Affichage du statut du réseau.	Contrôlez le système en affichant tous les sockets et toutes les entrées de table de routage. La sortie inclut la famille d'adresses inet pour les réseaux IPv4 et la famille d'adresses inet6 pour les réseaux IPv6.	"Affichage du statut des sockets" à la page 84
Affichage du statut des interfaces réseau.	Contrôlez les performances des interfaces réseau, notamment afin de dépanner les transmissions de données.	"Affichage du statut de l'interface réseau" à la page 84
Affichage du statut de transmission des paquets.	Contrôlez le statut des paquets lors de leur transmission sur le réseau câblé.	"Affichage du statut des transmissions de paquets associés à un type d'adresse spécifique" à la page 86
Contrôle de l'affichage des sorties de commandes IPv6.	Contrôle la sortie des commandes `ping`, `netstat` et `traceroute`. Crée un fichier nommé `inet_type`. Définit la variable `DEFAULT_IP` de ce fichier.	"Contrôle de la sortie d'affichage des commandes IP" à la page 89
Contrôle du trafic réseau.	Affichez tous les paquets IP à l'aide de la commande `snoop`.	"Contrôle du trafic réseau IPv6" à la page 96
Affichage de toutes les routes connues par les routeurs du réseau.	Affichez toutes les routes à l'aide de la commande `traceroute`.	"Affichage du suivi de toutes les routes" à la page 93

Remarque – Pour surveiller les interfaces réseau, reportez-vous à la section "Contrôle d'interfaces et d'adresses IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*

Contrôle du statut du réseau à l'aide de la commande netstat

La commande netstat génère des affichages illustrant le statut du réseau ainsi que les statistiques des protocoles. Vous pouvez afficher le statut des points d'extrémité TCP, SCTP et UDP sous forme de table. Vous pouvez également afficher les informations de table de routage ainsi que les informations d'interface.

La commande netstat permet d'afficher différents types d'informations sur le réseau, suivant l'option de ligne de commande sélectionnée. Les affichages obtenus constituent la principale référence pour l'administration du système. L'exemple ci-dessous illustre la syntaxe de base de la commande netstat :

netstat [-m] [-n] [-s] [-i | -r] [-f *famille-adresses*]

Cette section décrit les options fréquemment utilisées avec la commande netstat. Pour obtenir une description détaillée des toutes les options netstat, reportez-vous à la page de manuel netstat(1M).

▼ Affichage des statistiques par protocole

L'option -s de la commande netstat permet d'afficher les statistiques des protocoles UDP, TCP, SCTP, ICMP et IP.

Remarque – Vous pouvez obtenir la sortie de la commande netstat à l'aide du compte utilisateur Oracle Solaris.

- **Affichez le statut du protocole.**

```
$ netstat -s
```

Exemple 5–1 Statistiques des protocoles réseau

L'exemple suivant illustre la sortie de la commande netstat -s. Certaines parties de la sortie ont été tronquées. La sortie peut signaler les opérations ayant généré des problèmes pour les différents protocoles. Par exemple, les statistiques affichées pour ICMPv4 et ICMPv6 peuvent signaler les opérations ayant généré des erreurs pour le protocole ICMP.

```
RAWIP
        rawipInDatagrams    =  4701     rawipInErrors       =     0
        rawipInCksumErrs    =     0     rawipOutDatagrams   =     4
        rawipOutErrors      =     0

UDP
        udpInDatagrams      = 10091     udpInErrors         =     0
        udpOutDatagrams     = 15772     udpOutErrors        =     0
```

```
TCP     tcpRtoAlgorithm     =     4     tcpRtoMin           =   400
        tcpRtoMax           = 60000     tcpMaxConn          =    -1
        .
        .
        tcpListenDrop       =     0     tcpListenDropQ0     =     0
        tcpHalfOpenDrop     =     0     tcpOutSackRetrans   =     0

IPv4    ipForwarding        =     2     ipDefaultTTL        =   255
        ipInReceives        =300182     ipInHdrErrors       =     0
        ipInAddrErrors      =     0     ipInCksumErrs       =     0
        .
        .
        ipsecInFailed       =     0     ipInIPv6            =     0
        ipOutIPv6           =     3     ipOutSwitchIPv6     =     0

IPv6    ipv6Forwarding      =     2     ipv6DefaultHopLimit =   255
        ipv6InReceives      = 13986     ipv6InHdrErrors     =     0
        ipv6InTooBigErrors  =     0     ipv6InNoRoutes      =     0
        .
        .
        rawipInOverflows    =     0     ipv6InIPv4          =     0

       ipv6OutIPv4         =     0     ipv6OutSwitchIPv4   =     0

ICMPv4  icmpInMsgs          = 43593     icmpInErrors        =     0
        icmpInCksumErrs     =     0     icmpInUnknowns      =     0
        .
        .
        icmpInOverflows     =     0

ICMPv6  icmp6InMsgs         = 13612     icmp6InErrors       =     0
        icmp6InDestUnreachs =     0     icmp6InAdminProhibs =     0
        .
        .
        icmp6OutGroupQueries=     0     icmp6OutGroupResps  =     2
        icmp6OutGroupReds   =     0

IGMP:
      12287 messages received
          0 messages received with too few bytes
          0 messages received with bad checksum
      12287 membership queries received
SCTP  sctpRtoAlgorithm     =  vanj
      sctpRtoMin           =  1000
      sctpRtoMax           = 60000
      sctpRtoInitial       =  3000
      sctpTimHearBeatProbe =     2
      sctpTimHearBeatDrop  =     0
      sctpListenDrop       =     0
      sctpInClosed         =     0
```

▼ Affichage du statut des protocoles de transport

La commande netstat permet d'afficher le statut des protocoles de transport. Pour plus d'informations, reportez-vous à la page de manuel netstat(1M).

1. **Affichez le statut des protocoles de transport TCP et SCTP sur un système.**

   ```
   $ netstat
   ```

2. **Affichez le statut d'un protocole de transport donné sur un système.**

   ```
   $ netstat -P transport-protocol
   ```

   La variable *transport-protocol* peut être définie sur les valeurs suivantes : `tcp`, `sctp` ou `udp`.

Exemple 5–2 Affichage du statut des protocoles de transport TCP et SCTP

L'exemple ci-dessous illustre la sortie de base de la commande `netstat`. Les informations contenues dans la sortie se rapportent uniquement à IPv4.

```
$ netstat

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost-1.login      ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost-1.1014     mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT
SCTP:
Local Address    Remote Address  Swind  Send-Q  Rwind  Recv-Q StrsI/O  State
---------------- --------------  -----  ------ ------ ------  ------   -------
 *.echo            0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.discard         0.0.0.0            0       0 102400      0   128/1   LISTEN
 *.9001            0.0.0.0            0       0 102400      0   128/1   LISTEN
```

Exemple 5–3 Affichage du statut d'un protocole de transport donné

L'exemple ci-dessous illustre le résultat obtenu suite à l'exécution de la commande `netstat` avec l'option `-P`.

```
$ netstat -P tcp

TCP: IPv4
   Local Address      Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- -------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost.login        ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost.1014       mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT

TCP: IPv6
 Local Address    Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------- ---------------------- ------ ----- ------ ------------ -----
localhost.38983   localhost.32777       49152      0 49152      0 ESTABLISHED
localhost.32777   localhost.38983       49152      0 49152      0 ESTABLISHED
localhost.38986   localhost.38980       49152      0 49152      0 ESTABLISHED
```

▼ Affichage du statut de l'interface réseau

L'option i de la commande netstat illustre le statut des interfaces réseau configurées sur le système local. Cette option permet de déterminer le nombre de paquets transmis et reçus sur un système sur les différents réseaux.

- **Affichez le statut des interfaces sur le réseau.**

```
$ netstat -i
```

Exemple 5–4 Affichage du statut d'interface réseau

L'exemple suivant illustre le statut du flux de paquets IPv4 et IPv6 sur les interfaces de l'hôte.

Par exemple, le nombre de paquets entrants (Ipkts) affiché pour un serveur peut augmenter à chaque tentative de démarrage d'un client alors que le nombre de paquets sortants (Opkts) reste inchangé. Ce résultat suggère que le serveur détecte les paquets de demande de démarrage envoyés par le client, mais qu'il ne parvient pas à formuler la réponse appropriée. Cette confusion peut être causée par une adresse incorrecte dans la base de données hosts ou ethers.

En revanche, si le nombre de paquets entrants reste inchangé sur la durée, l'ordinateur ne détecte même pas l'envoi des paquets. Ce résultat suggère un autre type d'erreur, vraisemblablement lié à un problème d'ordre matériel.

```
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      142    0     142    0     0      0
net0  1500 host58        host58        1106302 0     52419  0     0      0

Name  Mtu  Net/Dest      Address                   Ipkts  Ierrs Opkts  Oerrs Collis
lo0   8252 localhost     localhost                  142    0     142    0     0
net0  1500 fe80::a00:20ff:feb9:4c54/10 fe80::a00:20ff:feb9:4c54 1106305 0 52422 0  0
```

▼ Affichage du statut des sockets

L'option -a de la commande netstat permet d'afficher le statut des sockets sur l'hôte local.

- **Pour afficher le statut des sockets et des entrées de table de routage, saisissez la commande suivante :**

  L'exécution de cette option de la commande netstat peut s'effectuer à l'aide du compte utilisateur.

```
% netstat -a
```

Exemple 5–5 Affichage de l'ensemble des sockets et des entrées de table de routage

La sortie de la commande netstat -a contient de nombreuses statistiques. L'exemple ci-dessous illustre certaines parties d'une sortie classique de la commande netstat -a.

```
UDP: IPv4
   Local Address         Remote Address     State
-------------------- -------------------- -------
      *.bootpc                            Idle
host85.bootpc                             Idle
      *.*                                 Unbound
      *.*                                 Unbound
      *.sunrpc                            Idle
      *.*                                 Unbound
      *.32771                             Idle
      *.sunrpc                            Idle
      *.*                                 Unbound
      *.32775                             Idle
      *.time                              Idle
       .
       .
      *.daytime                           Idle
      *.echo                              Idle
      *.discard                           Idle

UDP: IPv6
   Local Address                      Remote Address                    State      If
--------------------------------- --------------------------------- ---------- -----
      *.*                                                          Unbound
      *.*                                                          Unbound
      *.sunrpc                                                     Idle
      *.*                                                          Unbound
      *.32771                                                      Idle
      *.32778                                                      Idle
      *.syslog                                                     Idle
      .
      .
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
      *.*                  *.*                0      0 49152      0 IDLE
localhost.4999             *.*                0      0 49152      0 LISTEN
      *.sunrpc             *.*                0      0 49152      0 LISTEN
      *.*                  *.*                0      0 49152      0 IDLE
      *.sunrpc             *.*                0      0 49152      0 LISTEN
      .
      .
      *.printer            *.*                0      0 49152      0 LISTEN
      *.time               *.*                0      0 49152      0 LISTEN
      *.daytime            *.*                0      0 49152      0 LISTEN
      *.echo               *.*                0      0 49152      0 LISTEN
      *.discard            *.*                0      0 49152      0 LISTEN
      *.chargen            *.*                0      0 49152      0 LISTEN
      *.shell              *.*                0      0 49152      0 LISTEN
      *.shell              *.*                0      0 49152      0 LISTEN
      *.kshell             *.*                0      0 49152      0 LISTEN
      *.login
       .
       .
           *.*                0      0 49152      0 LISTEN
   *TCP: IPv6
 Local Address            Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------------- ---------------------- ----- ------ ----- ------    ----
   *.*                        *.*                0      0 49152      0      IDLE
```

```
   *.sunrpc                *.*                0      0 49152      0      LISTEN
   *.*                     *.*                0      0 49152      0      IDLE
   *.32774                 *.*                0      0 49152
```

Affichage du statut des transmissions de paquets associés à un type d'adresse spécifique

L'option `-f` de la commande `netstat` permet d'afficher les statistiques relatives aux transmissions de paquets associées à une famille d'adresses donnée.

- **Affichez les statistiques relatives aux transmissions de paquets IPv4 ou IPv6.**

  `$ netstat -f` *inet* | *inet6*

  Pour afficher les informations relatives aux transmissions IPv4, définissez l'argument `inet` pour la commande `netstat -f`. Pour afficher les informations relatives aux transmissions IPv6, définissez l'argument `inet6` pour la commande `netstat -f`.

Exemple 5–6 Statut de transmission de paquets IPv4

L'exemple suivant illustre la sortie de la commande `netstat -f inet`.

```
TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q  State
-------------------- -------------------- ----- ------ ----- ------ -------
host58.734         host19.nfsd       49640      0 49640      0 ESTABLISHED
host58.38063       host19.32782      49640      0 49640      0 CLOSE_WAIT
host58.38146       host41.43601      49640      0 49640      0 ESTABLISHED
host58.996         remote-host.login 49640      0 49206      0 ESTABLISHED
```

Exemple 5–7 Statut de transmission de paquets IPv6

L'exemple suivant illustre la sortie de la commande `netstat -f inet6`.

```
TCP: IPv6
 Local Address          Remote Address        Swind Send-Q Rwind Recv-Q   State    If
------------------ ------------------------- ----- ------ ----- ------ --------- -----
localhost.38065         localhost.32792       49152   0 49152      0    ESTABLISHED
localhost.32792         localhost.38065       49152   0 49152      0    ESTABLISHED
localhost.38089         localhost.38057       49152   0 49152      0    ESTABLISHED
```

Affichage du statut des routes connues

L'option `-r` de la commande `netstat` permet d'afficher la table de routage de l'hôte local. Cette table représente le statut de toutes les routes connues de l'hôte. L'exécution de cette option de la commande `netstat` peut s'effectuer à l'aide du compte utilisateur.

- **Affichez la table de routage IP.**

  ```
  $ netstat -r
  ```

Exemple 5–8 Sortie de table de routage obtenue à l'aide de la commande `netstat`

L'exemple suivant illustre la sortie de la commande `netstat -r`.

```
Routing Table: IPv4
  Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
host15               myhost               U         1  31059  net0
10.0.0.14            myhost               U         1      0  net0
default              distantrouter        UG        1      2  net0
localhost            localhost            UH        42019361  lo0

Routing Table: IPv6
  Destination/Mask      Gateway                           Flags Ref  Use   If
--------------------- --------------------------          ----- --- ------ -----
2002:0a00:3010:2::/64 2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd U     1   0      net0:1
fe80::/10             fe80::1a2b:3c4d:5e6f:12a2           U     1   23     net0
ff00::/8              fe80::1a2b:3c4d:5e6f:12a2           U     1   0      net0
default               fe80::1a2b:3c4d:5e6f:12a2           UG    1   0      net0
localhost             localhost                           UH    9   21832  lo0
```

Le tableau suivant décrit les différents paramètres de la sortie à l'écran de la commande `netstat -r`.

Paramètre	Description
`Destination` `Destination/Mask`	Spécifie l'hôte correspondant au point d'extrémité de destination de la route. Dans la table de routage IPv6, le point d'extrémité de destination est représenté par un préfixe de point d'extrémité de tunnel 6to4 (`2002:0a00:3010:2::/64`).
`Gateway`	Spécifie la passerelle de transmission des paquets.
`Flags`	Indique le statut actuel de la route. L'indicateur `U` signifie que la route fonctionne. L'indicateur `G` signifie que la route mène à une passerelle.
`Use`	Affiche le nombre de paquets envoyés.
`Interface`	Indique l'interface de l'hôte local correspondant au point d'extrémité source de la transmission.

Test des hôtes distants à l'aide de la commande ping

La commande ping permet de déterminer le statut d'un hôte distant. Lors de l'exécution de la commande ping, le protocole ICMP envoie un datagramme à l'hôte spécifié et attend la réponse. Le protocole ICMP permet de gérer les erreurs se produisant sur les réseaux TCP/IP. L'exécution de la commande ping permet de déterminer l'existence d'une connexion IP pour l'hôte distant spécifié.

L'exemple suivant illustre la syntaxe de base de la commande ping :

`/usr/sbin/ping` *hôte [délai]*

Dans cette syntaxe, la variable *hôte* correspond au nom de l'hôte distant. L'argument *délai* indique la durée en secondes pendant laquelle la commande ping tente de contacter l'hôte distant. La valeur par défaut est de 20 secondes. Pour plus d'informations sur la syntaxe et les options de la commande, reportez-vous à la page de manuel ping(1M)

▼ Vérification de l'exécution d'un hôte distant

- **Tapez la commande ping suivante :**

  `$ ping` *hostname*

  Si l'hôte *hostname* accepte les transmissions ICMP, le message suivant s'affiche :

  hostname `is alive`

  Ce message indique que *hostname* a répondu à la demande ICMP. En revanche, si *hostname* ne fonctionne pas ou ne reçoit pas les paquets ICMP, la commande ping génère la réponse suivante :

  `no answer from` *hostname*

▼ Détection de l'abandon de paquets sur un hôte

L'option -s de la commande ping permet de vérifier qu'un hôte distant est en cours d'exécution et de détecter toute perte de paquet sur cet hôte.

- **Tapez la commande ping suivante :**

  `$ ping -s` *hostname*

Exemple 5–9 Sortie de la commande ping permettant la détection de l'abandon de paquet

La commande ping -s *hostname* envoie des paquets en continu à l'hôte spécifié pendant un laps de temps donné ou jusqu'à l'envoi d'un caractère d'interruption. Les réponses affichées sont comparables à celles de l'écran suivant :

```
& ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms

^C

----host1.domain8  PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

Les statistiques de perte de paquets indiquent si l'hôte a perdu des paquets. En cas d'échec de la commande `ping`, vérifiez le statut du réseau signalé par les commandes `ipadm` et `netstat`. Reportez-vous aux sections "Contrôle d'interfaces et d'adresses IP" du manuel *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1* et "Contrôle du statut du réseau à l'aide de la commande `netstat`" à la page 81.

Administration et journalisation des affichages de statut du réseau

Les tâches suivantes illustrent les procédures de vérification du statut du réseau à l'aide de commandes de réseau standard.

▼ Contrôle de la sortie d'affichage des commandes IP

Vous pouvez contrôler la sortie de la commande `netstat` pour afficher uniquement les informations IPv4, ou les informations IPv4 et IPv6.

1. **Créez le fichier `/etc/default/inet_type`.**

2. **Ajoutez l'une des entrées suivantes au fichier `/etc/default/inet_type` :**

   - Pour afficher uniquement les informations IPv4 :

     ```
     DEFAULT_IP=IP_VERSION4
     ```

   - Pour afficher les informations IPv4 et IPv6 :

     ```
     DEFAULT_IP=BOTH
     ```

     Ou

     ```
     DEFAULT_IP=IP_VERSION6
     ```

     Pour plus d'informations sur le fichier `inet_type`, reportez-vous à la page de manuel `inet_type(4)`.

Remarque – L'indicateur `-f` dans la commande `netstat` remplace les valeurs dans le fichier `inet_type`.

Exemple 5–10 Contrôle de la sortie pour la sélection des informations IPv4 et IPv6

- Si vous spécifiez la variable `DEFAULT_IP=BOTH` ou la variable `DEFAULT_IP=IP_VERSION6` dans le fichier `inet_type`, la sortie suivante s'affiche :

```
% ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      10.46.86.54/24
lo0/v6        static     ok      ::1/128
net0/v6       addrconf   ok      fe80::a00:fe73:56a8/10
net0/v6add    static     ok      2001:db8:3c4d:5:a00:fe73:56a8/64
```

- Si vous spécifiez la variable `DEFAULT_IP=IP_VERSION4` dans le fichier `inet_type`, la sortie suivante s'affiche :

```
% ipadm show-addr
ADDROBJ       TYPE       STATE   ADDR
lo0/v4        static     ok      127.0.0.1/8
net0/v4       static     ok      10.46.86.54/24
```

▼ Journalisation des actions du démon de routage IPv4

Si vous pensez que le démon de routage IPv4 `routed` ne fonctionne pas correctement, vous pouvez créer un journal permettant d'effectuer le suivi de l'activité correspondante. Le journal inclut tous les transferts de paquets à compter du démarrage du démon `routed`.

- **Créez un fichier journal permettant d'effectuer le suivi des opérations du démon :**

```
# /usr/sbin/in.routed /var/log-file-name
```

Attention – Sur les réseaux à forte activité, la sortie de cette commande peut être générée sur une base quasi continue.

Exemple 5–11 Journal réseau du démon `in.routed`

L'exemple suivant illustre le début du journal créé à l'aide de la procédure "Journalisation des actions du démon de routage IPv4" à la page 90.

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0  #1   127.0.0.1      -->127.0.0.1/32
   <UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
```

```
Add interface net0 #2   10.10.48.112    -->10.10.48.0/25
    <UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add    10.0.0.0        -->10.10.48.112      metric=0  net0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  net0  <IF|NOPROP>
```

Suivi des activités du démon de détection des voisins IPv6

Si vous pensez que le démon IPv6 `in.ndpd` ne fonctionne pas correctement, vous pouvez générer le suivi de l'activité correspondante. Le suivi s'affiche sur la sortie standard jusqu'à l'arrêt du processus. Il inclut tous les transferts de paquets à compter du démarrage du démon `in.ndpd`.

1. **Générez le suivi du démon `in.ndpd`.**

   ```
   # /usr/lib/inet/in.ndpd -t
   ```

2. **Pour arrêter le processus de suivi, appuyez sur les touches Ctrl-C.**

Exemple 5–12 Suivi de l'activité du démon `in.ndpd`

La sortie suivante illustre le début du suivi du démon `in.ndpd`.

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to  ff02::2 (16 bytes) on net0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:b9:4c:54>
Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on net0
Nov 18 17:27:28         Max hop limit: 0
Nov 18 17:27:28         Managed address configuration: Not set
Nov 18 17:27:28         Other configuration flag: Not set
Nov 18 17:27:28         Router lifetime: 1800
Nov 18 17:27:28         Reachable timer: 0
Nov 18 17:27:28         Reachable retrans timer: 0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28         Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
Nov 18 17:27:28         Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
```

Affichage des informations de routage à l'aide de la commande traceroute

La commande traceroute permet d'obtenir le suivi de la route empruntée par un paquet IP pour accéder à un système distant. Pour plus d'informations sur la commande traceroute, reportez-vous à la page de manuel traceroute(1M).

La commande traceroute permet de détecter les erreurs de configuration de routage et les échecs de chemin de routage. Si un hôte est inaccessible, la commande traceroute permet d'afficher le chemin suivi par les paquets afin de détecter les emplacements susceptibles d'être à l'origine de l'échec.

La commande traceroute affiche également le délai d'aller-retour de chaque passerelle sur le chemin d'accès à l'hôte cible. Ces informations permettent notamment de déterminer l'emplacement des ralentissements de trafic entre les deux hôtes.

▼ Détermination de la route menant à un hôte distant

- **Pour déterminer la route menant à un hôte distant, exécutez la commande suivante :**

  % **traceroute** *destination-hostname*

  L'exécution de cette forme de la commande traceroute peut s'effectuer à l'aide du compte utilisateur.

Exemple 5–13 Affichage de la route menant à un hôte distant à l'aide de la commande traceroute

La sortie suivante de la commande traceroute affiche le chemin à sept sauts suivi par les paquets pour circuler du système local nearhost vers le système distant farhost. La sortie illustre également le temps nécessaire à un paquet pour traverser les différents sauts.

```
istanbul% traceroute farhost.faraway.com
    traceroute to farhost.faraway.com (172.16.64.39), 30 hops max, 40 byte packets
     1  frbldg7c-86 (172.16.86.1)  1.516 ms  1.283 ms  1.362 ms
     2  bldg1a-001 (172.16.1.211)  2.277 ms  1.773 ms  2.186 ms
     3  bldg4-bldg1 (172.16.4.42)  1.978 ms  1.986 ms  13.996 ms
     4  bldg6-bldg4 (172.16.4.49)  2.655 ms  3.042 ms  2.344 ms
     5  ferbldg11a-001 (172.16.1.236)  2.636 ms  3.432 ms  3.830 ms
     6  frbldg12b-153 (172.16.153.72)  3.452 ms  3.146 ms  2.962 ms
     7  sanfrancisco (172.16.64.39)  3.430 ms  3.312 ms  3.451 ms
```

▼ Affichage du suivi de toutes les routes

Cette procédure permet d'afficher le suivi de toutes les routes à l'aide de l'option `-a` de la commande `traceroute`.

- **Exécutez la commande suivante sur le système local :**

  `% traceroute -a`*host-name*

  L'exécution de cette forme de la commande `traceroute` peut s'effectuer à l'aide du compte utilisateur.

Exemple 5–14 Affichage du suivi de toutes les routes menant à un hôte double pile

L'exemple ci-dessous illustre toutes les routes possibles pour accéder à un hôte double pile.

```
% traceroute -a v6host.remote.com
traceroute: Warning: Multiple interfaces found; using 2::56:a0:a8 @ eri0:2
traceroute to v6host (2001:db8:4a3b::102:a00:fe79:19b0),30 hops max, 60 byte packets
 1  v6-rout86 (2001:db8:4a3b:56:a00:fe1f:59a1)  35.534 ms  56.998 ms *
 2  2001:db8::255:0:c0a8:717  32.659 ms  39.444 ms *
 3  farhost.faraway.COM (2001:db8:4a3b::103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
 4  distant.remote.com (2001:db8:4a3b::100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
 5  v6host (2001:db8:4a3b::102:a00:fe79:19b0)  66.111 ms *  36.965 ms

traceroute to v6host.remote.com  (192.168.10.75),30 hops max,40 byte packets
 1  v6-rout86 (172.16.86.1)  4.360 ms  3.452 ms  3.479 ms
 2  flrmpj17u.here.COM (172.16.17.131)  4.062 ms  3.848 ms  3.505 ms
 3  farhost.farway.com (10.0.0.23)  4.773 ms *  4.294 ms
 4  distant.remote.com (192.168.10.104)  5.128 ms  5.362 ms *
 5  v6host  (192.168.15.85)  7.298 ms  5.444 ms *
```

Contrôle du transfert des paquets à l'aide de la commande snoop

La commande `snoop` permet de contrôler le statut des transferts de données. La commande `snoop` permet de capturer les paquets réseau et d'afficher leur contenu au format spécifié. Les paquets peuvent être affichés dès leur réception ou dès l'enregistrement dans un fichier. L'écriture des données dans un fichier intermédiaire par la commande `snoop` permet de réduire la probabilité de perte de paquet liée à l'activité de suivi. Le fichier est alors également interprété par la commande `snoop`.

Pour capturer des paquets en provenance et à destination de l'interface par défaut en mode promiscuité, vous devez prendre le rôle d'administrateur réseau ou de superutilisateur. Dans sa forme contractée, la commande `snoop` affiche uniquement les données en rapport avec le protocole principal. Par exemple, un paquet NFS affiche uniquement les informations NFS. Les informations RPC, UDP, IP et Ethernet sont supprimées, mais vous pouvez y accéder en sélectionnant l'une des options détaillées de la commande.

L'exécution répétée à intervalles fréquents de la commande snoop permet d'identifier les comportements normaux du système. Pour obtenir de l'aide sur l'analyse des paquets, consultez les livres blancs et documents RFC récents et demandez conseil aux experts dans les domaines concernés (par exemple, NFS ou NIS). Pour plus d'informations sur l'utilisation de la commande snoop et des options associées, reportez-vous à la page de manuel snoop(1M)

▼ Vérification des paquets en provenance de toutes les interfaces

1 **Imprimez les informations sur les interfaces connectées au système.**

```
# ipadm show-if
```

La commande snoop utilise normalement le premier périphérique non-loopback (en principe, l'interface réseau principale).

2 **Commencez la capture des paquets en exécutant la commande snoop sans argument, comme illustré dans l'Exemple 5–15.**

3 **Pour arrêter le processus, appuyez sur les touches Ctrl-C.**

Exemple 5–15 Sortie de la commande snoop

La commande snoop standard renvoie une sortie comparable à l'écran suivant (pour un hôte double pile).

```
% snoop
Using device /dev/net (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
    0:10:7b:31:37:80
router5.local.com -> BROADCAST     TFTP Read "network-confg" (octet)
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com  myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
   niserve2.
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

Les paquets capturés dans cette sortie comprennent une section de connexion à distance, qui contient des demandes vers les serveurs NIS et DNS pour la résolution d'adresse. Ils comprennent également des paquets ARP périodiques en provenance du routeur local et des publications de l'adresse IPv6 lien-local sur `in.ripngd`.

▼ Capture de la sortie de la commande snoop dans un fichier

1 Capturez une session de commande snoop dans un fichier.

```
# snoop -o filename
```

Exemple :

```
# snoop -o /tmp/cap
Using device /dev/eri (promiscuous mode)
30 snoop: 30 packets captured
```

Dans cet exemple, 30 paquets sont capturés dans le fichier `/tmp/cap`. Ce fichier peut se trouver dans tout répertoire contenant suffisamment d'espace disque. Le nombre de paquets capturés s'affiche sur la ligne de commande. Vous pouvez dès lors appuyez sur les touches Ctrl-C à tout moment pour arrêter le processus.

La commande snoop génère une charge réseau conséquente, ce qui risque de fausser légèrement les résultats. Pour garantir la précision des résultats, exécutez la commande snoop à partir d'un système tiers.

2 Consultez le fichier de capture de sortie de la commande snoop.

```
# snoop -i filename
```

Exemple 5–16 Contenu du fichier de capture de sortie de la commande snoop

La sortie suivante illustre diverses captures susceptibles d'être obtenues suite à l'exécution de la commande `snoop -i`.

```
# snoop -i /tmp/cap
1   0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375
    ICMPv6 Neighbor advertisement
...
10  0.91493    10.0.0.40 -> (broadcast)  ARP C Who is 10.0.0.40, 10.0.0.40 ?
34  0.43690 nearserver.here.com  -> 224.0.1.1  IP  D=224.0.1.1 S=10.0.0.40 LEN=28,
      ID=47453, TO =0x0, TTL=1
35  0.00034  10.0.0.40 -> 224.0.1.1    IP  D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
     TOS=0x0, TTL=47
```

▼ Vérification des paquets transmis entre un client et un serveur IPv4

1 Définissez un système snoop à partir d'un hub connecté soit au serveur soit au client.

Le système tiers (système snoop) vérifie tous les types de trafic entre les deux ordinateurs. Le suivi obtenu grâce à la commande snoop reflète donc le transfert réel de données.

2. **Exécutez la commande snoop associée aux options appropriées, puis enregistrez la sortie dans un fichier.**

3. **Consultez et interprétez la sortie.**

   Reportez-vous au document RFC 1761, Snoop Version 2 Packet Capture File Format (`http://www.ietf.org/rfc/rfc1761.txt?number=1761`) pour plus d'informations sur le fichier de capture snoop.

▼ Contrôle du trafic réseau IPv6

La commande snoop permet d'afficher les paquets IPv6 uniquement.

- **Capturez les paquets IPv6.**

  ```
  # snoop ip6
  ```

  Pour plus d'informations sur la commande snoop, reportez-vous à la page de manuel snoop(1M).

Exemple 5–17 Affichage du trafic réseau IPv6 uniquement

L'exemple suivant illustre la sortie standard susceptible d'être obtenue suite à l'exécution de la commande snoop ip6 sur un noeud.

```
# snoop ip6
fe80::a00:20ff:fecd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
      solicitation
fe80::a00:20ff:febb:e09 -> ff02::9      RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

Contrôle des paquets à l'aide de périphériques de couche IP

Les périphériques de couche IP ont été introduits dans Oracle Solaris pour améliorer l'observabilité. Ces périphériques donnent accès à tous les paquets avec les adresses associées à l'interface réseau du système. Ces adresses incluent des adresses locales ainsi que des adresses hébergées sur des interfaces sans loopback ou des interfaces logiques. Le trafic observable peut correspondre aux adresses IPv4 et IPv6. Par conséquent, vous pouvez surveiller l'ensemble du trafic destiné au système. Le trafic peut être du trafic d'IP avec loopback, des paquets provenant de machines distantes, des paquets envoyés à partir du système ou la totalité du trafic transféré.

Les périphériques de couche IP permettent à l'administrateur d'une zone globale de surveiller le trafic entre les zones ainsi qu'au sein d'une zone. L'administrateur d'une zone non globale peut également observer le trafic envoyé et reçu par cette zone.

Pour surveiller le trafic sur la couche IP, une nouvelle option, `-I`, est ajoutée à la commande `snoop`. Cette option indique à la commande d'utiliser les nouveaux périphériques de couche IP plutôt que le périphérique sous-jacent de couche liaison pour afficher les données de trafic.

▼ Vérification des paquets sur la couche IP

1 Si nécessaire, imprimez les informations sur les interfaces connectées au système.

```
# ipadm show-if
```

2 Capturez le trafic IP sur une interface spécifique.

```
# snoop -I interface [-V | -v]
```

Exemples de vérification des paquets

Tous les exemples sont basés sur la configuration système suivante :

```
# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
net0/v4      static     ok      192.68.25.5/24
lo0/?        static     ok      127.0.0.1/8
net0/?       static     ok      172.0.0.3/24
net0/?       static     ok      172.0.0.1/24
lo0/?        static     ok      127.0.0.1/8
```

Supposons que deux zones, `sandbox` et `toybox`, utilisent les adresses IP suivantes :

- `sandbox` – `172.0.0.3`
- `toybox` – `172.0.0.1`

Vous pouvez exécuter la commande `snoop -I` sur les différentes interfaces du système. L'affichage des informations du paquet dépend de si vous êtes administrateur de la zone globale ou de la zone non globale.

EXEMPLE 5–18 Trafic sur l'interface loopback

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
   localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
   localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

Pour générer une sortie détaillée, utilisez l'option `-v`.

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 1 arrived at 10:40:33.68506
IPNET:  Packet size = 108 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 4
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 0
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
...
```

La prise en charge de l'observation des paquets sur la couche IP introduit un nouvel en-tête ipnet qui précède les paquets observés. Les ID de source et de destination sont tous deux indiqués. L'ID '0' indique que le trafic est généré à partir de la zone globale.

EXEMPLE 5–19 Flux de paquets du périphérique net0 dans les zones locales

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491
```

La sortie présente le trafic des différentes zones au sein du système. Vous pouvez voir tous les paquets associés aux adresses IP `net0`, y compris les paquets livrés localement aux autres zones. Si vous générez une sortie détaillée, vous pouvez voir les zones impliquées dans le flux de paquets.

```
# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:         .... ..0. = not ECN capable transport
IP:         .... ...0 = no ECN congestion experienced
```

EXEMPLE 5–19 Flux de paquets du périphérique net0 dans les zones locales *(Suite)*

```
IP:   Total length = 40 bytes
IP:   Identification = 22629
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
IP:   Header checksum = 0000
IP:   Source address = 172.0.0.1, 172.0.0.1
IP:   Destination address = 172.0.0.3, 172.0.0.3
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        0... .... = No ECN congestion window reduced
TCP:        .0.. .... = No ECN echo
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP         .... 0... = No push
TCP         .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 49152
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
```

L'en-tête ipnet indique que le paquet provient de la zone globale (ID 0) et se dirige vers Sandbox (ID 1).

EXEMPLE 5–20 Observation du trafic par identification de la zone

```
# snoop -I hme0 sandboxsnoop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
#
```

La capacité d'observer des paquets par identification de la zone est utile dans les systèmes dotés de plusieurs zones. Actuellement, la zone s'identifie uniquement par l'intermédiaire de l'ID de zone. L'utilisation de snoop avec les noms de zone n'est pas prise en charge.

Administration de la sélection des adresses par défaut

Oracle Solaris permet à une interface unique de disposer de plusieurs adresses IP. Par exemple, certaines fonctionnalités telles que la fonctionnalité IPMP (multipathing sur réseau IP) permettent la connexion de plusieurs cartes d'interface réseau (NIC, network interface card) sur la même couche de liaison IP. Cette liaison peut être associée à une ou plusieurs adresses IP. Les interfaces des systèmes IPv6 possèdent également une adresse IPv6 lien-local, au moins une adresse de routage IPv6 ainsi qu'une adresse IPv4 pour au moins une interface.

Lorsque le système génère une transaction, une application envoie un appel vers le socket `getaddrinfo`. `getaddrinfo` détecte les adresses susceptibles d'être utilisées sur le système de destination. Le noyau établit alors l'ordre de priorité de cette liste afin de déterminer la destination appropriée pour le paquet. Ce processus est appelé *classement des adresses de destination*. Le noyau Oracle Solaris sélectionne le format approprié pour l'adresse source en fonction de l'adresse de destination déterminée pour le paquet. Ce processus est appelé *sélection des adresses*. Pour plus d'informations sur le classement des adresses de destination, reportez-vous à la page de manuel `getaddrinfo(3SOCKET)`.

Le processus de sélection des adresses par défaut doit s'effectuer sur les systèmes IPv4 uniquement ainsi que sur les systèmes double pile IPv4/IPv6. Dans la plupart des cas, il n'est pas nécessaire de modifier les mécanismes de sélection des adresses par défaut. Toutefois, vous devrez peut-être modifier l'ordre de priorité des formats d'adresse de manière à prendre en charge la fonctionnalité IPMP ou à préférer les formats d'adresse 6to4, par exemple.

▼ Administration de la table des règles de sélection d'adresses IPv6

La section ci-dessous décrit la procédure de modification de la table des règles de sélection d'adresses. Pour plus d'informations concernant la sélection des adresses IPv6 par défaut, reportez-vous à la section Commande ipaddrsel.

Attention – La table des règles de sélection d'adresses IPv6 doit uniquement être modifiée sur la base des motifs décrits dans la tâche suivante. Les erreurs de définition de la table des règles risquent d'entraîner des problèmes de fonctionnement du réseau. Veillez à enregistrer une copie de sauvegarde de la table des règles, comme indiqué à la procédure suivante.

1 Consultez la table de stratégie de sélection d'adresse IPv6 actuelle.

```
# ipaddrsel
# Prefix                  Precedence Label
::1/128                           50 Loopback
::/0                              40 Default
2002::/16                         30 6to4
::/96                             20 IPv4_Compatible
::ffff:0.0.0.0/96                 10 IPv4
```

2 Effectuez une copie de la table des règles de sélection d'adresses par défaut.

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

3 Apportez les modifications souhaitées au fichier `/etc/inet/ipaddrsel.conf` dans un éditeur de texte.

Utilisez la syntaxe suivante pour les entrées de fichier `/etc/inet/ipaddrsel` :

prefix/prefix-length precedence label [# comment]

Les exemples ci-dessous illustrent les modifications susceptibles d'être apportées le plus souvent à la table des règles :

- Définition des adresses 6to4 sur la priorité la plus élevée :

```
2002::/16                       50 6to4
::1/128                         45 Loopback
```

  Le format d'adresse 6to4 dispose dorénavant de la plus haute priorité (50). Loopback, qui disposait auparavant d'une priorité de 50, dispose dorénavant d'une priorité de 45. Les autres formats d'adresse restent inchangés.

- Définition d'une adresse source spécifique pour les communications avec une adresse de destination donnée :

```
::1/128                         50 Loopback
2001:1111:1111::1/128           40 ClientNet
2001:2222:2222::/48             40 ClientNet
::/0                            40 Default
```

  Ce type de configuration s'utilise notamment pour les hôtes associés à une seule interface physique. Dans cet exemple, l'adresse source `2001:1111:1111::1/128` est définie en tant qu'adresse prioritaire pour les paquets adressés aux destinations du réseau `2001:2222:2222::/48`. L'adresse source `2001:1111:1111::1/128` est associée à la priorité 40, priorité supérieure à celle des autres formats d'adresse configurés pour l'interface.

- Préférence des adresses IPv4 par rapport aux adresses IPv6 :

```
::ffff:0.0.0.0/96               60 IPv4
::1/128                         50 Loopback
.
.
```

  La priorité par défaut du format IPv4 `::ffff:0.0.0.0/96` passe de 10 à 60, soit la priorité la plus élevée de la table.

4 Chargez la table de règles modifiée dans le noyau.

```
ipaddrsel -f /etc/inet/ipaddrsel.conf
```

5 Si la table des règles modifiée génère des erreurs, restaurez la table des règles de sélection des adresses IPv6 par défaut.

```
# ipaddrsel -d
```

▼ Modification de la table des règles de sélection des adresses IPv6 pour la session en cours uniquement

Les modifications apportées au fichier `/etc/inet/ipaddrsel.conf` sont conservées lors des sessions suivantes. Si vous souhaitez modifier la table des règles uniquement pour la session en cours, effectuez la procédure suivante.

1. **Copiez le contenu du fichier `/etc/inet/ipaddrsel` dans le fichier *filename*, où *filename* désigne le nom de votre choix.**

   ```
   # cp /etc/inet/ipaddrsel filename
   ```

2. **Apportez les modifications souhaitées à la table des règles dans le fichier *filename* .**

3. **Chargez la table de règles modifiée dans le noyau.**

   ```
   # ipaddrsel -f filename
   ```

   Le noyau utilise la nouvelle table des règles jusqu'au prochain redémarrage du système.

◆ ◆ ◆ CHAPITRE 6

Configuration de tunnels IP

Ce chapitre contient des descriptions des tunnels IP ainsi que les procédures de configuration et de maintenance des tunnels dans Oracle Solaris.

Présentation des tunnels IP

Les tunnels IP fournissent un moyen de transporter des paquets de données entre différents domaines lorsque le protocole de ces domaines n'est pas pris en charge par les réseaux intermédiaires. Par exemple, avec l'introduction du protocole IPv6, les réseaux IPv6 nécessitent un moyen de communiquer en dehors de leurs frontières dans un environnement où la plupart des réseaux utilisent le protocole IPv4. La communication devient possible avec l'utilisation des tunnels. Le tunnel IP fournit une liaison virtuelle entre deux noeuds atteignables en utilisant IP. La liaison peut donc être utilisée pour le transport de paquets IPv6 au sein des réseaux IPv4 afin de permettre la communication IPv6 entre les deux sites IPv6.

Administration de tunnels IP dans Oracle Solaris 11

Dans cette version d'Oracle Solaris, l'administration de tunnel a été révisée afin d'être en cohérence avec le nouveau modèle d'administration de liaison de données de réseau. Les tunnels sont maintenant créés et configurés à l'aide des nouvelles sous-commandes `dladm`. Les tunnels peuvent également utiliser d'autres fonctionnalités de liaison de données du nouveau modèle d'administration. Par exemple, la prise en charge des noms choisis par l'administrateur permet d'attribuer des noms significatifs aux tunnels. Pour plus d'informations sur les sous-commandes `dladm`, reportez-vous à la page de manuel `dladm(1M)`.

Types de tunnels

La mise sous tunnel implique d'encapsuler un paquet IP dans un autre paquet. Cette encapsulation permet au paquet d'atteindre sa destination par le biais de réseaux intermédiaires qui ne prennent pas en charge le protocole du paquet.

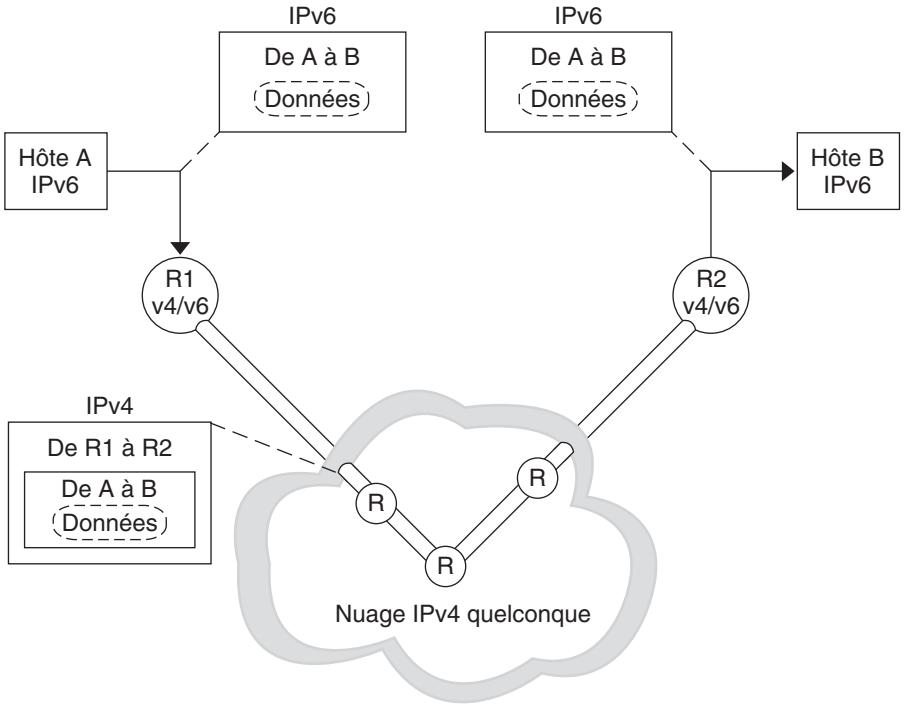
Les tunnels diffèrent en fonction du type d'encapsulation de paquet. Les types de tunnels suivants sont pris en charge dans Oracle Solaris :

- *Tunnels IPv4* : les paquets IPv4 ou IPv6 sont encapsulés dans un en-tête IPv4 et envoyés à une destination IPv4 unicast préconfigurée. Pour indiquer de façon plus spécifique les paquets acheminés dans le tunnel, les tunnels IPv4 sont également appelés *tunnels IPv4 sur IPv4* ou *tunnels IPv6 sur IPv4*.
- *Tunnels IPv6* : les paquets IPv4 ou IPv6 sont encapsulés dans un en-tête IPv6 et envoyés à une destination IPv6 unicast préconfigurée. Pour indiquer de façon plus spécifique les paquets acheminés dans le tunnel, les tunnels IPv6 sont également appelés *tunnels IPv4 sur IPv6* ou *tunnels IPv6 sur IPv6*.
- *Tunnels 6to4* : les paquets IPv6 sont encapsulés dans un en-tête IPv4, puis envoyés à une destination IPv4 déterminée automatiquement en fonction du nombre de paquets. La détermination est basée sur un algorithme défini dans le protocole 6to4.

Tunnels dans les environnements réseau combinant IPv6 et IPv4

La plupart des sites dotés de domaines IPv6 communiquent avec les autres domaines IPv6 en traversant des réseaux IPv4, lesquels sont plus répandus que les réseaux exclusivement IPv6. La figure suivante illustre le mécanisme de mise en tunnel entre deux hôtes IPv6 via des routeurs IPv4, signalés dans la figure par la lettre “R.”

FIGURE 6–1 Mécanisme de mise en tunnel IPv6



Dans la figure, le tunnel se compose de deux routeurs configurés afin de disposer d'une liaison virtuelle point à point entre les deux routeurs sur le réseau IPv4.

Un paquet IPv6 est encapsulé dans un paquet IPv4. Le routeur de bordure du réseau IPv6 configure un tunnel point à point sur plusieurs réseaux IPv4 jusqu'au routeur de bordure du réseau IPv6 de destination. Le paquet est transporté dans le tunnel au routeur de bordure de destination, où le paquet est décapsulé. Le routeur transmet ensuite le paquet IPv6 distinct au noeud de destination.

Tunnels 6to4

Dans Oracle Solaris, les tunnels 6to4 constituent la méthode temporaire recommandée pour effectuer la transition entre les adressages IPv4 et IPv6. Les tunnels 6to4 permettent aux sites IPv6 isolés de communiquer, par le biais d'un tunnel automatique, avec un réseau IPv4 ne prenant pas en charge le protocole IPv6. Pour utiliser des tunnels 6to4, vous devez configurer un routeur de bordure sur le réseau IPv6 en tant que point d'extrémité du tunnel 6to4 automatique. Par la suite, le routeur 6to4 peut participer à un tunnel vers un autre site 6to4 ou vers un site IPv6 natif et non-6to4, le cas échéant.

Cette section fournit des références sur les rubriques concernant les tunnels 6to4 :

- Topologie d'un tunnel 6to4
- Description du flux de paquets dans un tunnel 6to4
- Topologie d'un tunnel reliant un routeur 6to4 et un routeur relais 6to4
- Informations importantes pour la configuration de la prise en charge d'un routeur relais 6to4

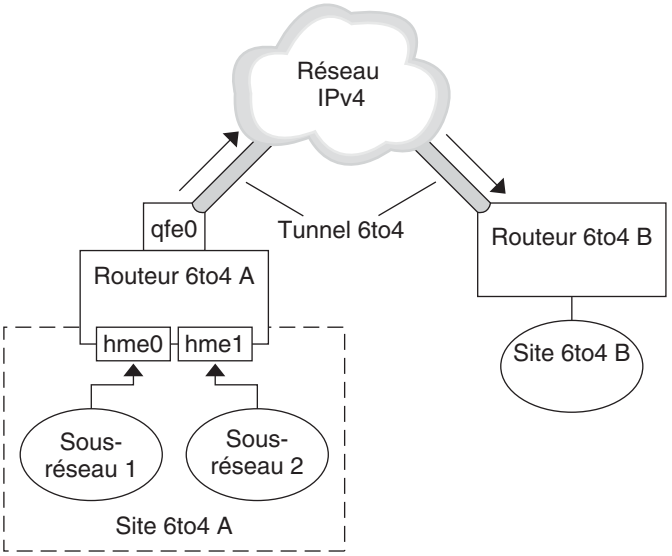
Le tableau suivant décrit les autres tâches permettant de configurer des tunnels 6to4 et les ressources permettant obtenir d'autres informations utiles.

Tâche ou détail	Référence
Configuration d'un tunnel 6to4	“Configuration d'un tunnel 6to4” à la page 117
RFC lié aux 6to4	RFC 3056, "Connection of IPv6 Domains via IPv4 Clouds" (`http://www.ietf.org/rfc/rfc3056.txt`)
Informations détaillées sur la commande `6to4relay` (prise en charge des tunnels vers un routeur relais 6to4)	`6to4relay`(1M)
Problèmes de sécurité avec 6to4	Security Considerations for 6to4 (`http://www.ietf.org/rfc/rfc3964.txt`)

Topologie d'un tunnel 6to4

Un tunnel 6to4 offre la connexion IPv6 à tous les sites 6to4, quel que soit leur emplacement. De même, le tunnel offre un lien à l'ensemble des sites IPv6, notamment l'Internet IPv6 natif, à condition d'être configuré pour la transmission vers un routeur relais. La figure suivante illustre un tunnel 6to4 connectant des sites 6to4.

FIGURE 6–2 Tunnel entre deux sites 6to4



La figure représente deux réseaux 6to4 isolés, le site A et le site B. Chaque site a configuré un routeur avec une connexion externe à un réseau IPv4. Un tunnel 6to4 à l'échelle du réseau IPv4 offre une connexion entre sites 6to4.

Pour convertir un site IPv6 en site 6to4, vous devez configurer au moins une interface de routeur prenant en charge 6to4. Cette interface doit assurer la connexion externe au réseau IPv4. L'adresse que vous configurez sur `qfe0` doit être globale et unique. Sur cette figure, l'interface du routeur A (`qfe0`) connecte le site A au réseau IPv4. L'interface `qfe0` doit déjà être configurée avec une adresse IPv4 pour que vous puissiez définir `qfe0` en tant que pseudointerface 6to4.

Dans la figure, le site 6to4 A est composé de deux sous-réseaux qui sont connectés aux interfaces `hme0` et `hme1` du routeur A. Tous les hôtes IPv6 sur l'un des sous-réseaux du site A sont reconfigurés automatiquement avec des adresses dérivées de 6to4 une fois la publication du routeur A reçue.

Le site B est un autre site 6to4 isolé. Pour recevoir correctement le trafic du site A, un routeur de bordure sur le site B doit être configuré pour prendre en charge 6to4. Dans le cas contraire, le routeur ne reconnaît pas les paquets reçus du site A et les abandonne.

Description du flux de paquets dans un tunnel 6to4

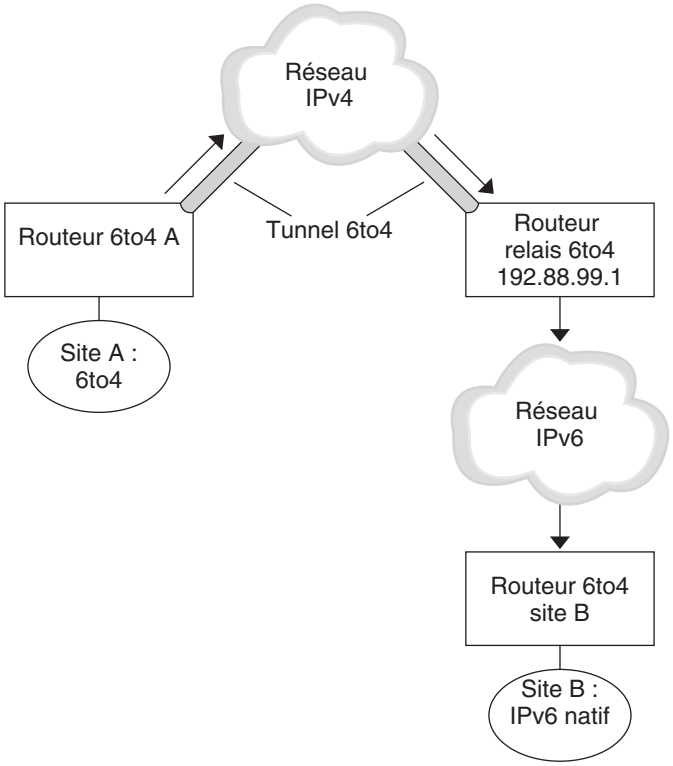
Cette section décrit le flux de paquets allant d'un hôte sur un site 6to4 à un autre hôte sur un site 6to4 distant. Ce scénario nécessite la topologie illustrée sur la Figure 6–2. Cela suppose également de configurer au préalable les routeurs et les hôtes 6to4.

1. Un hôte du sous-réseau 1 appartenant au site 6to4 A envoie une transmission à un hôte du site 6to4 B. Chaque en-tête de paquet possède des adresses 6to4 dérivées source et cible.
2. Le routeur du site A encapsule chaque paquet 6to4 dans un en-tête IPv4. Dans ce processus, le routeur définit l'adresse cible IPv4 de l'en-tête d'encapsulation sur l'adresse du routeur du site B. L'adresse cible IPv6 de chaque paquet IPv6 transmis via l'interface du tunnel contient également l'adresse cible IPv4. Ainsi, le routeur est en mesure de déterminer l'adresse cible IPv4 définie sur l'en-tête d'encapsulation. Ensuite, il utilise la procédure de routage IPv4 standard pour transmettre le paquet sur le réseau IPv4.
3. Tout routeur IPv4 rencontré par les paquets utilise l'adresse IPv4 cible de ces derniers pour la transmission. Cette adresse constitue l'adresse IPv4 globale et unique de l'interface du routeur B, qui sert également de pseudointerface 6to4.
4. Les paquets du site A arrivent sur le routeur B qui les décapsule en paquets IPv6 à partir de l'en-tête IPv4.
5. Le routeur B se sert alors de l'adresse cible des paquets IPv6 pour transmettre ces derniers à l'hôte destinataire sur le site B.

Informations importantes pour la création de tunnels vers un routeur relais 6to4

Les routeurs relais 6to4 fonctionnent en tant que points d'extrémité des tunnels reliant des routeurs 6to4 à des réseaux IPv6 natifs, non 6to4. Les routeurs relais constituent essentiellement des ponts entre le site 6to4 et les sites IPv6 natifs. Ce type de routeur risque de ne pas garantir la sécurité du réseau ; c'est pourquoi il n'est pas pris en charge par Oracle Solaris. Cependant, si votre site nécessite un tel tunnel, vous pouvez exécuter la commande `6to4relay` pour créer le type de tunnel suivant.

FIGURE 6–3 Tunnel entre un site 6to4 et un routeur relais 6to4



Dans la Figure 6–3, le site 6to4 A doit communiquer avec un noeud du site natif IPv6 B. La figure indique le chemin du trafic en provenance du site A dans un tunnel 6to4 sur un réseau IPv4. Le tunnel dispose d'un routeur A 6to4 et d'un routeur relais 6to4 à chaque extrémité. Au-delà du routeur 6to4 se trouve le réseau IPv6 auquel le site B IPv6 est connecté.

Flux de paquets entre un site 6to4 et un site IPv6 natif

Cette section décrit le flux de paquets se déplaçant d'un site 6to4 vers un site IPv6 natif. Ce scénario nécessite la topologie illustrée sur la Figure 6–3.

1. Un hôte sur le site 6to4 A envoie une transmission spécifiant un hôte sur le site natif IPv6 B en tant que destination. Chaque en-tête de paquet dispose d'une adresse dérivée de 6to4 en tant qu'adresse source. L'adresse de destination correspond à une adresse IPv6 standard.
2. Le routeur 6to4 du site A encapsule chaque paquet dans un en-tête IPv4, dont la destination correspond à l'adresse IPv4 du routeur relais 6to4. Ensuite, il utilise la procédure de routage IPv4 standard pour transmettre le paquet sur le réseau IPv4. Tout routeur IPv4 rencontré par les paquets envoie ceux-ci vers le routeur relais 6to4.

3. Le routeur relais 6to4 anycast le plus proche (physiquement) du site A récupère les paquets destinés au groupe anycast `192.88.99.1`.

Remarque – Les routeurs relais 6to4 faisant partie du groupe anycast de routeurs relais 6to4 possèdent l'adresse IP `192.88.99.1`. Cette adresse anycast constitue l'adresse par défaut des routeurs relais 6to4. Si vous avez besoin d'un routeur relais 6to4 spécifique, vous pouvez supprimer celui par défaut et spécifier l'adresse IPv4 du routeur en question.

---

4. Ce routeur relais décapsule ensuite l'en-tête IPv4 des paquets 6to4, dévoilant l'adresse de destination sur le réseau IPv6.
5. Le routeur relais envoie ensuite les paquets qui sont à présent IPv6 uniquement sur le réseau IPv6, où ils seront ensuite récupérés par un routeur sur le site B. Le routeur transmet ensuite les paquets au noeud de destination IPv6.

Déploiement des tunnels

Pour déployer les tunnels IP correctement, vous devez effectuer deux tâches principales. Commencez par créer la liaison de tunnel. Ensuite, configurez une interface IP sur le tunnel. Cette section offre une brève description des exigences en matière de création des tunnels et de leurs interfaces IP correspondantes.

Exigences en matière de création de tunnels

Pour créer des tunnels correctement, vous devez remplir les exigences suivantes :

- Si vous utilisez des noms d'hôte plutôt que des adresses IP littérales, ces noms doivent être résolus en adresses IP valides compatibles avec le type de tunnel.
- Le tunnel IPv4 ou IPv6 que vous créez ne doit pas partager les mêmes adresses source et de destination de tunnel avec un autre tunnel configuré.
- Le tunnel IPv4 ou IPv6 que vous créez ne doit pas partager la même adresses source avec un tunnel 6to4 existant.
- Si vous créez un tunnel 6to4, celui-ci ne doit pas partager la même adresse source avec un autre tunnel configuré.

Pour obtenir des informations sur la configuration de tunnels sur votre réseau, reportez-vous à la section "Planification de l'utilisation de tunnels dans le réseau" à la page 31.

Exigences relatives aux tunnels et aux interfaces IP

Chaque type de tunnel est doté d'exigences spécifiques en matière d'adresses IP sur l'interface IP que vous configurez sur le tunnel. Les exigences sont résumées dans le tableau ci-dessous.

TABLEAU 6–1 Exigences en matière de tunnels et d'interface IP

Type de tunnel	Interface IP autorisée sur le tunnel	Exigence d'interface IP
Tunnel IPv4	Interface IPv4	Les adresses locales et distantes sont spécifiées manuellement.
	Interface IPv6	Les adresses locales et distantes de liaison locale sont définies automatiquement lors de l'exécution de la commande `ipadm create-addr -T addrconf`. Pour plus d'informations, reportez-vous à la page de manuel `ipadm`(1M).
Tunnel IPv6	Interface IPv4	Les adresses locales et distantes sont spécifiées manuellement.
	Interface IPv6	Les adresses locales et distantes de liaison locale sont définies automatiquement lors de l'exécution de la commande `ipadm create-addr -T addrconf`. Pour plus d'informations, reportez-vous à la page de manuel `ipadm`(1M).
Tunnel 6to4	Interface IPv6 uniquement	L'adresse IPv6 par défaut est sélectionnée automatiquement lors de l'exécution de la commande `ipadm create-ip`. Pour plus d'informations, reportez-vous à la page de manuel `ipadm`(1M).

Vous pouvez remplacer l'adresse d'interface IPv6 par défaut par des tunnels 6to4 en spécifiant une adresse IPv6 différente à l'aide de la commande `ipadm`.

De même, pour remplacer les adresses de liaison locale définies automatiquement pour les interfaces IPv6 sur les tunnels IPv4 ou IPv6, vous pouvez spécifier différentes adresses source et de destination dans le fichier hôte du tunnel.

Configuration et administration du tunnel avec la commande dladm

Cette section décrit les procédures utilisant la commande dladm pour configurer les tunnels.

Sous-commandes dladm

A partir de cette version d'Oracle Solaris, l'administration de tunnel est maintenant séparée de la configuration de l'interface IP. L'aspect données-liaison des tunnels IP est maintenant administré à l'aide de la commande dladm. En outre, la configuration d'interface IP, incluant l'interface de tunnel IP, s'effectue à l'aide de la commande ipadm.

Les sous-commandes dladm suivantes permettent de configurer les tunnels IP :

- create-iptun
- modify-iptun
- show-iptun
- delete-iptun
- set-linkprop

Pour plus d'informations sur la commande dladm, reportez-vous à la page de manuel dladm(1M).

Remarque – L'administration de tunnels IP est étroitement liée à la configuration d'IPsec. Par exemple, les VPN IPsec sont l'une des utilisations principales de la mise sous tunnel IP. Pour plus d'informations sur la sécurité dans Oracle Solaris, reportez-vous au Chapitre 6, "Architecture IPsec (présentation)" du manuel *Sécurisation du réseau dans Oracle Solaris 11.1*. Pour plus d'informations sur la configuration des protocoles IPsec, reportez-vous au Chapitre 7, "Configuration d'IPsec (tâches)" du manuel *Sécurisation du réseau dans Oracle Solaris 11.1*.

Configuration des tunnels (liste des tâches)

Tâche	Description	Voir
Création d'un tunnel IP.	Configuration du tunnel à utiliser pour communiquer sur les réseaux.	"Création et configuration d'un tunnel IP" à la page 113
Modification de la configuration d'un tunnel.	Modification des paramètres d'origine du tunnel, comme l'adresse source ou de destination du tunnel.	"Modification d'une configuration de tunnel IP" à la page 121

Tâche	Description	Voir
Affichage d'une configuration de tunnel.	Affichage des informations de configuration pour un tunnel spécifique ou pour tous les tunnels IP du système.	"Affichage d'une configuration de tunnel IP" à la page 122
Suppression d'un tunnel.	Suppression d'une configuration de tunnel.	"Suppression d'un tunnel IP" à la page 123

▼ Création et configuration d'un tunnel IP

1 Créez le tunnel.

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

Les options ou arguments suivants sont disponibles pour cette commande :

`-t` — Crée un tunnel temporaire. Par défaut, la commande crée un tunnel persistant.

Remarque – Si vous souhaitez configurer une interface IP sur le tunnel, vous devez créer un tunnel persistant et ne pas utiliser l'option `-t`.

`-T` *type* — Spécifie le type de tunnel à créer. Cet argument est requis pour créer tous les types de tunnel.

`-a [local|remote]=`*address,...* — Spécifie les adresses IP littérales ou les noms d'hôte correspondant aux adresses locales et à l'adresse de tunnel distant. Les adresses doivent être valides et déjà créées dans le système. Suivant le type de tunnel, spécifiez soit une seule adresse, soit les adresses locales et distantes. Si vous spécifiez les adresses locales et distantes, vous devez les séparer à l'aide d'une virgule.

- Les tunnels `IPv4` nécessitent des adresses IPv4 locales et distantes pour fonctionner.
- Les tunnels `IPv6` nécessitent des adresses IPv6 locales et distantes pour fonctionner.
- Les tunnels `6to4` nécessitent une adresse IPv4 locale pour fonctionner.

Remarque – Pour les configurations de liaison de données de tunnel IP, si vous utilisez des noms d'hôte en guise d'adresses, ces noms d'hôte sont enregistrés dans le stockage de configuration. Lors d'une initialisation ultérieure du système, si la résolution de noms donne des adresses IP différentes de celles utilisées lors de la création du tunnel, ce dernier acquiert une nouvelle configuration.

tunnel-link

Spécifie la liaison de tunnel IP. Avec la prise en charge des noms significatifs dans une administration réseau-liaison, les noms de tunnel ne sont plus limités au type de tunnel que vous créez. En revanche, vous pouvez attribuer au tunnel tout nom choisi par l'administrateur. Les noms de tunnel se composent d'une chaîne et du numéro de point de connexion physique, par exemple, *montunnel0*. Pour les règles qui régissent l'affectation de noms explicites, reportez-vous à la section "Règles d'affectation de noms de liaison valides" du manuel *Introduction à la mise en réseau d'Oracle Solaris 11*.

Si vous ne spécifiez pas la liaison de tunnel, le nom est fourni automatiquement, conformément aux conventions d'attribution de nom suivantes :

- Pour les tunnels IPv4 : `ip.tun#`
- Pour les tunnels IPv6 : `ip6.tun#`
- Pour les tunnels 6to4 : `ip.6to4tun#`

Le symbole # correspond au numéro de point de connexion physique le plus bas disponible pour le type de tunnel que vous êtes en train de créer.

2 (Facultatif) Définissez les valeurs de la limite de saut ou de la limite d'encapsulation.

```
# dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link
```

hoplimit — Spécifie la limite de saut de l'interface de tunnel pour la mise en tunnel sur IPv6. *hoplimit* est l'équivalent du champ IPv4 de durée de vie pour la mise en tunnel sur IPv4.

encaplimit — Spécifie le nombre de niveaux de tunnels imbriqués autorisés pour un paquet. Cette option s'applique uniquement aux tunnels IPv6.

Spécifie le nombre de niveaux de tunnels imbriqués autorisés pour un paquet. Cette option s'applique uniquement aux tunnels IPv6.

Remarque – Les valeurs définies pour `hoplimit` et `encaplimit` doivent être comprises dans une plage acceptable. `hoplimit` et `encaplimit` sont des propriétés de liaison de tunnel. Par conséquent, ces propriétés sont administrées par les mêmes sous-commandes `dladm` que pour les autres propriétés de liaison. Les sous-commandes sont `dladm set-linkprop`, `dladm reset-linkprop` et `dladm show-linkprop`. Reportez-vous à la page de manuel `dladm`(1M) pour connaître les différentes sous-commandes utilisées avec la commande `dladm` pour l'administration de liens.

3 Créez une interface IP sur le tunnel.

```
# ipadm create-ip tunnel-interface
```

où *tunnel-interface* utilise le même nom que la liaison de tunnel.

4 Assignez des adresses IP locales et distantes à l'interface de tunnel.

```
# ipadm create-addr [-t] -a local=address,remote=address  interface
```

`-t`	Indique une configuration d'IP temporaire plutôt qu'une configuration d'IP persistante sur le tunnel. Si vous n'utilisez pas cette option, la configuration d'interface IP est persistante.
`-a local=`*address*`,remote=`*address*	Spécifie l'adresse IP de l'interface de tunnel. Les adresses IP source et de destination sont requises, comme représenté par `local` et `remote`. Les adresses locales et distantes peuvent être des adresses IPv4 ou IPv6.
interface	Spécifie l'interface du tunnel.

Pour plus d'informations sur la commande `ipadm` et les diverses options permettant de configurer les interfaces IP, y compris les interfaces de tunnel, reportez-vous à la page de manuel `ipadm`(1M) et à la section *Connexion de systèmes à l'aide d'une configuration réseau fixe dans Oracle Solaris 11.1*.

5 Ajoutez les informations de configuration de tunnel au fichier `/etc/hosts`.

6 (Facultatif) Vérifiez le statut de la configuration d'interface de l'IP du tunnel.

```
# ipadm show-addr interface
```

Exemple 6–1 Création d'une interface IPv6 sur un tunnel IPv4

Cet exemple illustre la création d'un tunnel IPv6 persistant sur IPv4.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
```

```
# ipadm create-addr -T addrconf private0
# ipadm show-addr private/
ADDROBJ       TYPE     STATE   ADDR
private0/v6   static   ok      fe80::a08:392e/10 --> fe80::8191:9a56
```

Pour ajouter d'autres adresses, respectez la même syntaxe. Par exemple, vous pouvez ajouter une adresse globale comme suit :

```
# ipadm create-addr -a local=2001:db8:4728::1, \
remote=2001:db8:4728::2 private0
# ipadm show-addr private0/
ADDROBJ          TYPE       STATE   ADDR
private0/v6      addrconf   ok      fe80::a08:392e/10 --> fe80::8191:9a56
private0/v6a     static     ok      2001:db8:4728::1 --> 2001:db8:4728::2
```

Notez que le préfixe `2001:db8` de l'adresse IPv6 est un préfixe IPv6 spécial utilisé spécifiquement pour les exemples de documentation.

Exemple 6–2 Création d'une interface IPv4 sur un tunnel IPv4

Cet exemple illustre la création d'un tunnel IPv4 persistant sur IPv4.

```
# dladm create-iptun -T ipv4 -a local=63.1.2.3,remote=192.4.5.6 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -a local=10.0.0.1,remote=10.0.0.2 vpn0
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADDR
lo0/v4    static   ok      127.0.0.1
vpn0/v4   static   ok      10.0.0.1-->10.0.0.2
```

Vous pouvez configurer la stratégie IPsec davantage pour fournir des connexions sécurisées aux paquets circulant dans ce tunnel. Pour plus d'informations sur la configuration des protocoles IPsec, reportez-vous au Chapitre 7, “Configuration d’IPsec (tâches)” du manuel *Sécurisation du réseau dans Oracle Solaris 11.1*.

Exemple 6–3 Création d'une interface IPv6 sur un tunnel IPv6

Cet exemple illustre la création d'un tunnel IPv6 persistant sur IPv6.

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 \
tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0
# ipadm show-addr
ADDROBJ   TYPE       STATE   ADDR
lo0/v6    static     ok      ::1/128
tun0/v6   addrconf   ok      2001:db8:feed::1234 --> 2001:db8:beef::4321
```

Pour ajouter des adresses comme une adresse globale ou d'autres adresses locales et distantes, utilisez la commande `ipadm` comme suit :

```
# ipadm create-addr \
-a local=2001:db8::4728:56bc,remote=2001:db8::1428:57ab tun0
# ipadm show-addr tun0
ADDROBJ    TYPE      STATE ADDR
tun0/v6    addrconf  ok    2001:db8:feed::1234 --> 2001:db8:beef::4321
tun0/v6a   static    ok    2001:db8::4728:56bc --> 2001:db8::1428:57ab
```

Configuration d'un tunnel 6to4

Dans les tunnels 6to4, un routeur 6to4 doit agir en tant que routeur IPv6 pour les noeuds des sites 6to4 du réseau. Par conséquent, lors de la configuration d'un routeur 6to4, ce routeur doit également être configuré en tant que routeur IPv6 sur ses interfaces physiques. Pour plus d'informations sur le routage IPv6, reportez-vous à la section "Routage IPv6" à la page 151.

1 Créez un tunnel 6to4.

 # dladm create-iptun -T 6to4 -a local=*address tunnel-link*

Les options ou arguments suivants sont disponibles pour cette commande :

-a local=*address* Spécifie l'adresse locale de tunnel qui doit déjà exister dans le système pour être valide.

tunnel-link Spécifie la liaison de tunnel IP. Avec la prise en charge des noms significatifs dans une administration réseau-liaison, les noms de tunnel ne sont plus limités au type de tunnel que vous créez. En revanche, vous pouvez attribuer au tunnel tout nom choisi par l'administrateur. Les noms de tunnel se composent d'une chaîne et du numéro de point de connexion physique, par exemple, *montunnel0*. Pour les règles qui régissent l'affectation de noms explicites, reportez-vous à la section "Règles d'affectation de noms de liaison valides" du manuel *Introduction à la mise en réseau d'Oracle Solaris 11*.

2 Créez l'interface IP de tunnel.

 # ipadm create-ip *tunnel-interface*

où *tunnel-interface* utilise le même nom que la liaison de tunnel.

3 (Facultatif) Ajoutez d'autres adresses IPv6 pour une utilisation par le tunnel.

4 Modifiez le fichier `/etc/inet/ndpd.conf` pour publier le routage 6to4 en ajoutant les deux lignes suivantes :

```
if subnet-interface AdvSendAdvertisements 1
IPv6-address subnet-interface
```

La première ligne spécifie le sous-réseau qui reçoit cette publication. *subnet-interface* fait référence à la liaison à laquelle est connecté le sous-réseau. Les adresses IPv6 sur la seconde ligne doivent avoir le préfixe 6to4 `2000` qui est utilisé pour les adresses IPv6 dans les tunnels 6to4.

Pour obtenir des informations détaillées sur le fichier `ndpd.conf`, reportez-vous à la page de manuel `ndpd.conf(4)`.

5 Activez le transfert IPv6.

```
# ipadm set-prop -p forwarding=on ipv6
```

6 Réinitialisez le routeur.

Vous pouvez également exécuter la commande `sighup` sur le démon de `/etc/inet/in.ndpd` pour commencer la publication du routeur. Les noeuds IPv6 de chaque sous-réseau devant recevoir le préfixe 6to4 sont alors automatiquement définis sur les nouvelles adresses 6to4 dérivées.

7 Ajoutez ces nouvelles adresses au service de noms utilisé par le site 6to4.

Vous trouverez les instructions correspondantes dans la section "Configuration de prise en charge de services de noms pour IPv6" à la page 75.

Exemple 6–4 Création de tunnel 6to4

Dans cet exemple, l'interface de sous-réseau est `bge0`, à laquelle `/etc/inet/ndpd.conf` fait référence dans l'étape adéquate.

Cet exemple illustre la création d'un tunnel 6to4. Notez que seules les interfaces IPv6 peuvent être configurées sur les tunnels 6to4.

```
# dladm create-iptun -T 6to4 -a local=192.168.35.10 tun0
# ipadm create-ip tun0
# ipadm show-addr
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      192.168.35.10/24
lo0/v6        static   ok      ::1/128
tun0/_a       static   ok      2002:c0a8:57bc::1/64

# ipadm create-addr -a 2002:c0a8:230a::2/16 tun0
# ipadm create-addr -a 2002:c0a8:230a::3/16 tun0
# ipadm show-addr tun0
ADDROBJ       TYPE     STATE   ADDR
lo0/v4        static   ok      127.0.0.1/8
net0/v4       static   ok      192.168.35.10/24
lo0/v6        static   ok      ::1/128
tun0/_a       static   ok      2002:c0a8:57bc::1/64
tun0/v6       static   ok      2002:c0a8:230a::2/16
tun0/v6a      static   ok      2002:c0a8:230a::3/16
```

```
# vi /etc/inet/ndpd.conf
if bge0 AdvSendAdvertisements 1
2002:c0a8:57bc::1/64 bge0

# ipadm set-prop -p forwarding=on ipv6
```

Notez que pour les tunnels 6to4, le préfixe de l'adresse IPv6 est `2002`.

▼ Configuration d'un tunnel 6to4 relié à un routeur relais 6to4

Attention – Pour des raisons de sécurité, la prise en charge des routeurs relais 6to4 est désactivée par défaut dans Oracle Solaris. Reportez-vous à la section "Security Issues When Tunneling to a 6to4 Relay Router" du manuel *Troubleshooting Network Issues*.

Avant de commencer

Avant de configurer un tunnel relié à un routeur relais 6to4, vous devez avoir effectué les tâches suivantes :

- Configuration d'un routeur 6to4 sur votre site, comme décrit dans la section "Création et configuration d'un tunnel IP" à la page 113
- Vérification des problèmes de sécurité susceptibles de se produire avec un tunnel relié à un routeur relais 6to4

1 Vous pouvez relier un tunnel à un routeur relais 6to4 de deux façons :

- Liaison a un routeur relais 6to4 de type anycast.

  ```
  # /usr/sbin/6to4relay -e
  ```

  L'option `-e` configure un tunnel entre le routeur 6to4 et un routeur relais 6to4 anycast. Les routeurs relais 6to4 anycast possèdent l'adresse IPv4 courante `192.88.99.1`. Le routeur relais anycast le plus proche (physiquement) de votre site devient le point d'extrémité du tunnel 6to4. Ce routeur relais gère ensuite l'envoi des paquets entre votre site 6to4 et un site IPv6 natif.

  Pour plus d'informations sur les routeurs relais 6to4 Anycast, reportez-vous à la page RFC 3068, "An Anycast Prefix for 6to4 Relay Routers" (`ftp://ftp.rfc-editor.org/in-notes/rfc3068.txt`).

- Liaison a un routeur relais 6to4 de type spécifique.

  ```
  # /usr/sbin/6to4relay -e -a relay-router-address
  ```

L'option `-a` est toujours suivie d'une adresse de routeur spécifique. Remplacez *relay-router-address* par l'adresse IPv4 du routeur relais 6to4 spécifique que vous souhaitez relier au tunnel.

Le tunnel relié au routeur relais 6to4 reste actif pendant la suppression de la pseudointerface du tunnel 6to4.

2 Supprimez le tunnel relié au routeur relais 6to4 lorsqu'il n'est plus nécessaire :

```
# /usr/sbin/6to4relay -d
```

3 (Facultatif) Configurez un tunnel au routeur relais 6to4 qui conserve ses paramètres après chaque redémarrage.

Si votre site requiert, pour quelque raison qu'il soit, que les paramètres du tunnel relié au routeur relais 6to4 soient redéclarés à chaque redémarrage du routeur, effectuez la procédure suivante :

a. Modifiez le fichier `/etc/default/inetinit`.

La ligne à modifier se trouve à la fin du fichier.

b. Remplacez la valeur "NO" de la ligne `ACCEPT6TO4RELAY=NO` par "YES".

c. (Facultatif) Créez un tunnel relié à un routeur relais 6to4 spécifique dont les paramètres sont conservés après chaque redémarrage.

Pour le paramètre `RELAY6TO4ADDR`, remplacez l'adresse `192.88.99.1` par l'adresse IPv4 du routeur relais 6to4 à utiliser.

Exemple 6–5 Obtention d'informations sur le statut de la prise en charge des routeurs relais 6to4

La commande `/usr/bin/6to4relay` vous permet de savoir si les routeurs relais 6to4 sont pris en charge ou non par votre site. L'exemple suivant présente la sortie obtenue lorsque les routeurs relais 6to4 ne sont pas pris en charge (sortie par défaut d'Oracle Solaris) :

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is disabled.
```

Lorsque les routeurs relais 6to4 sont pris en charge, la sortie suivante s'affiche :

```
# /usr/sbin/6to4relay
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 remote address of Relay Router=192.88.99.1
```

▼ Modification d'une configuration de tunnel IP

- **Modifiez la configuration du tunnel.**

  # dladm modify-iptun -a [local|remote]=*addr,...* *tunnel-link*

  Vous ne pouvez pas modifier le type d'un tunnel existant. Par conséquent, l'option `-T` *type* n'est pas autorisée pour cette commande. Seuls les paramètres de tunnel suivants peuvent être modifiés :

  `-a [local|remote]=`*address,...* Spécifie les adresses IP littérales ou les noms d'hôte correspondant aux adresses locales et à l'adresse de tunnel distant. Suivant le type de tunnel, spécifiez soit une seule adresse, soit les adresses locales et distantes. Si vous spécifiez les adresses locales et distantes, vous devez les séparer à l'aide d'une virgule.

  - Les tunnels `IPv4` nécessitent des adresses IPv4 locales et distantes pour fonctionner.
  - Les tunnels `IPv6` nécessitent des adresses IPv6 locales et distantes pour fonctionner.
  - Les tunnels `6to4` nécessitent une adresse IPv4 locale pour fonctionner.

  Pour les configurations de liaison de données de tunnel IP, si vous utilisez des noms d'hôte en guise d'adresses, ces noms d'hôte sont enregistrés dans le stockage de configuration. Lors d'une initialisation ultérieure du système, si la résolution de noms donne des adresses IP différentes de celles utilisées lors de la création du tunnel, ce dernier acquiert une nouvelle configuration.

  Si vous modifiez les adresses locales et distantes du tunnel, assurez-vous que ces adresses sont cohérentes par rapport au type de tunnel que vous modifiez.

Remarque – Si vous souhaitez modifier le nom de la liaison de tunnel, n'utilisez pas la sous-commande `modify-iptun`. Utilisez plutôt `dladm rename-link`.

 # dladm rename-link *old-tunnel-link new-tunnel-link*

De même, n'utilisez pas la commande `modify-iptun` pour modifier les propriétés de tunnel telles que `hoplimit` ou `encaplimit` . Préférez la commande `dladm set-linkprop` pour définir les valeurs de ces propriétés.

Exemple 6–6 Modification de l'adresse et des propriétés d'un tunnel

Cet exemple comporte deux procédures. Tout d'abord, les adresses locales et distantes du tunnel IPv4 vpn0 sont modifiées temporairement. Une fois le système réinitialisé, le tunnel revient à l'utilisation des adresses d'origine. Une seconde procédure modifie la valeur hoplimit de vpn0 à 60.

```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.1.2.3 vpn0

# dladm set-linkprop -p hoplimit=60 vpn0
```

▼ Affichage d'une configuration de tunnel IP

- **Affichez la configuration du tunnel IP.**

  # dladm show-iptun [-p] -o *fields* **[***tunnel-link***]**

  Vous pouvez utiliser les options avec la commande :

-p	Affiche les informations dans une format analysable. Cet argument est facultatif.
-o *fields*	Affiche les champs sélectionnés qui fournissent des informations spécifiques au tunnel.
tunnel-link	Spécifie le tunnel dont vous souhaitez afficher les informations de configuration. Cet argument est facultatif. Si vous omettez le nom de tunnel, la commande affiche les informations à propos de tous les tunnels du système.

Exemple 6–7 Affichage des informations à propos de tous les tunnels

Dans cet exemple, un seul tunnel existe sur le système.

```
# dladm show-iptun
LINK     TYPE     FLAGS     LOCAL          REMOTE
tun0     6to4     --        192.168.35.10  --
vpn0     ipv4     --        10.8.48.149    192.1.2.3
```

Exemple 6–8 Affichage de champs sélectionnés dans un format analysable

Dans cet exemple, seuls les champs spécifiques comportant des informations sur les tunnels sont affichés.

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

▼ Affichage des propriétés d'un tunnel IP

- **Affichez les propriétés de la liaison du tunnel.**

  # dladm show-linkprop [-c] [-o *fields***] [***tunnel-link***]**

  Vous pouvez utiliser les options avec la commande :

`-c`	Affiche les informations dans une format analysable. Cet argument est facultatif.
`-o` *fields*	Affiche les champs sélectionnés fournissant des informations spécifiques à propos des propriétés du lien.
tunnel-link	Spécifie le tunnel dont vous souhaitez afficher les informations sur les propriétés. Cet argument est facultatif. Si vous omettez le nom de tunnel, la commande affiche les informations à propos de tous les tunnels du système.

Exemple 6–9 Affichage des propriétés d'un tunnel

Cet exemple indique comment afficher toutes les propriétés d'une liaison de tunnel.

```
# dladm show-linkprop tun0
LINK     PROPERTY     PERM     VALUE     DEFAULT     POSSIBLE
tun0     autopush     --       --        --          --
tun0     zone         rw       --        --          --
tun0     state        r-       up        up          up,down
tun0     mtu          r-       65515     --          576-65495
tun0     maxbw        rw       --        --          --
tun0     cpus         rw       --        --          --
tun0     priority     rw       high      high        low,medium,high
tun0     hoplimit     rw       64        64          1-255
```

▼ Suppression d'un tunnel IP

1. **Utilisez la syntaxe adéquate pour déconnecter l'interface IP configurée sur le tunnel en fonction du type de l'interface.**

   # ipadm delete-ip *tunnel-link*

   Remarque – Pour supprimer un tunnel correctement, aucune interface IP existante ne peut être montée sur le tunnel.

2. **Supprimez le tunnel IP.**

   # dladm delete-iptun *tunnel-link*

La seule option pour cette commande est -t, laquelle entraîne une suppression temporaire du tunnel. Le tunnel est restauré lors de la réinitialisation du système.

Exemple 6–10 Suppression d'un tunnel IPv6 configuré avec une interface IPv6

Dans cet exemple, un tunnel persistant est supprimé définitivement.

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

◆ ◆ ◆ CHAPITRE 7

Référence IPv4

Ce chapitre fournit des informations de référence sur les fichiers de configuration réseau pour les réseaux TCP/IP, notamment les types de réseau, leur objectif et le format d'entrée des fichiers.

Ce chapitre contient les informations suivantes :

Fichiers de configuration TCP/IP

Dans un réseau, les informations de configuration sont stockées dans différents fichiers et bases de données qui régulent le fonctionnement du réseau. Cette section fournit une brève description de ces fichiers. Certains fichiers nécessitent une mise à jour et de la maintenance lors de l'implémentation de modifications sur le réseau. D'autres fichiers ne nécessitent que peu, voire pas d'administration.

`/etc/defaultrouter` Ce fichier contient les noms d'interface IP des routeurs directement connectés au réseau. L'existence de ce fichier dans le système est facultative. Si le fichier existe, le système est alors configuré pour prendre en charge le routage statique.

`/etc/inet/hosts` Ce fichier contient les adresses IPv4 dans le réseau ainsi que les noms d'interface correspondantes sur lesquelles les adresses sont configurées. Si vous utilisez un service de noms NIS ou DNS, ou le service de répertoire LDAP, les informations de l'hôte sont alors stockées dans une base de données différente, comme `hosts.byname`, qui existe dans les serveurs. Pour plus

	d'informations, reportez-vous à la section *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.
`/etc/inet/netmasks`	Ce fichier contient le numéro de réseau, par exemple `192.168.0.0`, ainsi que les informations de masque de réseau pour ce numéro de réseau, par exemple `255.255.255.0`. Dans un réseau utilisant NIS ou LDAP, ces informations sont stockées dans une base de données de masque de réseau, dans les serveurs. Reportez-vous à la page de manuel `netmasks(4)` pour plus d'informations.
`/etc/bootparams`	Ce fichier contient des paramètres qui déterminent les processus d'initialisation pour les systèmes configurés pour s'initialiser en mode client réseau. Pour plus d'informations reportez-vous à la section "Modes de configuration système" à la page 38. Ce fichier est la base pour la création de la base de données `bootparams` utilisée par le service de noms si vous n'utilisez pas le mode de fichiers locaux. Pour obtenir des informations spécifiques sur le contenu et le format de ce fichier, reportez-vous à la page de manuel `bootparams(4)`.
`/etc/ethers`	Le fichier associe les noms d'hôtes à leurs adresses MAC. Le fichier est la base de la création d'une base de données `ethers` en vue d'une utilisation dans le réseau où les systèmes sont configurés en tant que clients réseau. Pour plus d'informations, reportez-vous à la page de manuel `ethers(4)`.
`/etc/inet/networks`	Ce fichier associe les noms de réseau et les numéros de réseau. Il est possible d'ajouter des commentaires pour clarifier davantage chaque entrée dans la base de données. Ce fichier permet aux applications d'utiliser et d'afficher les noms plutôt que les numéros de réseau. Par exemple, le programme `netstat` utilise les informations de cette base de données pour générer les tables d'état. Tous les sous-réseaux qui se connectent au réseau local par l'intermédiaire de routeurs doivent être inclus dans ce fichier. Pour plus d'informations, reportez-vous à la page de manuel `networks(4)`.
`/etc/inet/protocols`	Ce fichier répertorie les protocoles TCP/IP installés sur votre système ainsi que leurs numéros de protocole. Ce fichier requiert rarement des tâches d'administration. Pour plus d'informations, reportez-vous à la page de manuel `protocols(4)`.
`/etc/inet/services`	Ce fichier répertorie les noms des services TCP et UDP ainsi que leurs numéros de ports connus. Cette liste est employée par les programmes faisant appel aux services réseau. En général, ce fichier ne requiert aucune tâche d'administration. Pour plus

d'informations, reportez-vous à la page de manuel `services(4)`.

Démon de services Internet `inetd`

Le démon `inetd` lance les services Internet standard à l'initialisation du système et peut redémarrer un service lorsque le système est en cours d'exécution. Le SMF (Service Management Facility, utilitaire de gestion de service) permet de modifier les services Internet standard et d'indiquer au démon `inetd` de démarrer d'autres services, le cas échéant.

Exécutez les commandes SMF suivantes pour gérer les services démarrés par `inetd` :

`svcadm` Permet d'effectuer des tâches administratives sur un service, telle que l'activation, la désactivation et le redémarrage. Pour de plus amples informations, reportez-vous à la page de manuel `svcadm(1M)`.

`svcs` Permet d'effectuer des requêtes relatives au statut d'un service. Pour de plus amples informations, reportez-vous à la page de manuel `svcs(1)`.

`inetadm` Permet d'afficher et modifier les propriétés d'un service. Pour plus d'informations, reportez-vous à la page de manuel `inetadm(1M)`.

La valeur du champ `proto` dans le profil `inetadm` d'un service particulier indique le protocole de couche de transport sur lequel le service s'exécute. Si le service gère exclusivement des demandes IPv4, le champ `proto` doit être défini sur `tcp`, `udp` ou `sctp`.

- Pour obtenir des instructions sur l'utilisation des commandes SMF, reportez-vous à la section "Utilitaires d'administration en ligne de commande SMF" du manuel *Gestion des services et pannes dans Oracle Solaris 11.1*.
- Pour une tâche utilisant les commandes SMF afin d'ajouter un service s'exécutant sur SCTP, reportez-vous à la section "Ajout de services utilisant le protocole SCTP" à la page 57.
- Pour obtenir des informations sur l'ajout de services gérant à la fois des demandes IPv4 et des demandes IPv6, reportez-vous à la section "Démon de services Internet `inetd`" à la page 127

Service SMF name-service/switch

Le service SMF `name-service/switch` définit l'ordre de recherche des bases de données réseau pour les informations de configuration. Certaines informations de configuration réseau qui étaient auparavant stockées dans des fichiers de configuration, comme le domaine par défaut, ont été converties pour devenir des propriétés de ce service SMF. Les propriétés de ce service SMF déterminent l'implémentation des services de noms sur le système. Les propriétés sont les suivantes :

```
% svccfg -s name-service/switch listprop config
config                      application
config/value_authorization  astring             solaris.smf.value.name-service.switch
config/default              astring             files
config/password             astring             "files nis"
config/group                astring             "files nis"
config/host                 astring             "files dns nis"
config/network              astring             "nis [NOTFOUND=return] files"
config/protocol             astring             "nis [NOTFOUND=return] files"
config/rpc                  astring             "nis [NOTFOUND=return] files"
config/ether                astring             "nis [NOTFOUND=return] files"
config/netmask              astring             "files nis"
config/bootparam            astring             "nis [NOTFOUND=return] files"
config/publickey            astring             "nis [NOTFOUND=return] files"
config/netgroup             astring             nis
config/automount            astring             "files nis"
config/alias                astring             "files nis"
config/service              astring             "files nis"
config/printer              astring             "user nis"
config/auth_attr            astring             "files nis"
config/prof_attr            astring             "files nis"
config/project              astring             "files nis"
```

Les valeurs définies pour chacune des propriétés déterminent dans quel service de noms rechercher les informations qui auraient une incidence sur les utilisateurs réseau, par exemple les mots de passe, les alias ou encore les masques réseau. Dans cet exemple, les propriétés de montage automatique et de mot de passe sont définies sur `files` et `nis`. Par conséquent, les informations de montage automatique et de mot de passe s'obtiennent à partir des fichiers et du service NIS.

Si vous souhaitez passer d'un service de noms à un autre, vous devez définir les propriétés adéquates du service SMF `name-service/switch` pour activer le service de noms sélectionné.

Par exemple, supposons que vous souhaitez utiliser le service de noms LDAP sur votre réseau. Les propriétés suivantes du service SMF doivent être configurées.

- `config/default` doit être défini afin d'utiliser les fichiers et LDAP.
- `config/host` doit être défini afin d'utiliser les fichiers et DNS.
- `config/netgroup` doit être défini afin d'utiliser LDAP.
- `config/printer` doit être défini afin d'utiliser l'utilisateur, les fichiers et LDAP.

Par conséquent, vous devez saisir les commandes suivantes pour définir ces propriétés correctement.

```
# svccfg -s name-service/switch setprop config/default = astring: '"files ldap"'
# svccfg -s name-service/switch setprop config/host = astring: '"files dns"'
# svccfg -s name-service/switch setprop config/netgroup = astring: '"ldap"'
# svccfg -s name-service/switch setprop config/printer = astring: '"user files ldap"'
# svccfg -s name-service/switch:default refresh
```

Pour plus d'informations sur le commutateur du service de noms, reportez-vous au manuel *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.

Impact des services de noms sur les bases de données réseau

Le format de la base de données réseau dépend du type de service de noms sélectionné pour votre réseau. Par exemple, la base de données `hosts` contient au moins le nom d'hôte et l'adresse IPv4 du système local, ainsi que toute interface réseau directement connectée au système local. Cependant, la base de données `hosts` peut contenir d'autres adresses IPv4 et noms d'hôtes, selon le type de service de noms utilisé sur le réseau.

Les bases de données réseau sont utilisées comme suit :

- Les réseaux qui utilisent les fichiers locaux pour leurs services de noms se basent sur les fichiers dans les répertoires `/etc/inet` et `/etc`.
- NIS utilise des bases de données appelées cartes NIS.
- DNS utilise des enregistrements dotés d'informations d'hôte.

Remarque – Les fichiers d'initialisation et de données DNS ne correspondent pas directement aux bases de données réseau.

Pour obtenir des informations sur les correspondances de bases de données réseau dans NIS, DNS et LDAP, reportez-vous au document *Utilisation des services de noms et d'annuaire dans Oracle Solaris 11.1*.

Protocoles de routage dans Oracle Solaris

Cette section décrit les protocoles de routage pris en charge par Oracle Solaris: RIP (Routing Information Protocol, protocole d'informations de routage) et RDISC (ICMP Router Discovery, détection de routeur ICMP). RIP et RDISC constituent des protocoles TCP/IP standard. Pour obtenir la liste complète des protocoles de routage disponibles dans Oracle Solaris, reportez-vous au Tableau 7–1 et au Tableau 7–2.

RIP (Routing Information Protocol)

Le protocole RIP est implémenté par le démon de routage `in.routed` qui démarre à l'initialisation du système. Exécuté sur un routeur avec l'option `s`, le démon `in.routed`

renseigne la table de routage du noyau en indiquant une route pour chaque réseau accessible et publie l'accessibilité via toutes les interfaces réseau.

Exécuté sur un hôte avec l'option `q`, le démon `in.routed` extrait les informations de routage mais ne publie pas l'accessibilité. Sur les hôtes, vous pouvez extraire les informations de routage de deux façons :

- Ne spécifiez *pas* l'indicateur `S` (S majuscule : mode d'économie d'espace). `in.routed` construit une table de routage complète exactement de la même manière que sur un routeur.
- Spécifiez l'indicateur `S`. `in.routed` crée une table de routage minimale pour le noyau, contenant une seule route par défaut pour chaque routeur disponible.

Protocole RDISC (ICMP Router Discovery)

Les hôtes utilisent RDISC pour obtenir les informations de routage des autres routeurs. Par conséquent, lorsque les hôtes exécutent RDISC, les routeurs doivent également exécuter un autre protocole, par exemple RIP, afin d'échanger les informations de routeur.

RDISC est implémenté par le démon `in.routed`, qui doit s'exécuter à la fois sur les routeurs et sur les hôtes. Sur les hôtes, `in.routed` utilise RDISC pour détecter les routes par défaut des routeurs qui se publient eux-mêmes via RDISC. Sur les routeurs, `in.routed` utilise RDISC pour publier les routes par défaut des hôtes sur les réseaux directement connectés. Reportez-vous aux pages de manuel `in.routed(1M)` et `gateways(4)`.

Tableaux des protocoles de routage dans Oracle Solaris

Le tableau suivant répertorie tous les protocoles de routage pris en charge dans Oracle Solaris.

TABLEAU 7–1 Protocoles de routage d'Oracle Solaris

Protocole	Démon associé	Description	Voir
RIP (Routing Information Protocol)	`in.routed`	IGP acheminant les paquets IPv4 et gérant une table de routage	"Configuration d'un routeur IPv4" à la page 43
Détection de routeur ICMP (Internet Control Message Protocol)	`in.routed`	Permet aux hôtes de détecter la présence d'un routeur sur le réseau	"Activation du routage statique sur un hôte à interface unique" à la page 52 et "Activation du routage dynamique sur un système à interface unique" à la page 53

TABLEAU 7–1 Protocoles de routage d'Oracle Solaris *(Suite)*

Protocole	Démon associé	Description	Voir
Protocole RIPng (Routing Information Protocol, next generation, protocole d'informations de routage, nouvelle génération)	`in.ripngd`	IGP acheminant les paquets IPv6 et gérant une table de routage	"Configuration d'un routeur compatible IPv6" à la page 66
Protocole ND (Neighbor Discovery)	`in.ndpd`	Signale la présence d'un routeur IPv6 et détecte les hôtes IPv6 sur un réseau	"Configuration d'une interface IPv6" à la page 63

Le tableau suivant répertorie la suite de protocoles de routage Quagga Open Source qu'Oracle Solaris prend également en charge.

TABLEAU 7–2 Protocoles Quagga Open Source

Protocole	Démon	Description
Protocole RIP	`ripd`	Protocole IGP à vecteur de distance IPv4 qui achemine les paquets IPv4 et signale sa table de routage aux routeurs adjacents.
RIPng	`ripngd`	Protocole IGP à vecteur de distance IPv6 qui achemine les paquets IPv6 et gère une table de routage.
Protocole OSPF (Open Shortest Path First)	`ospfd`	Protocole IGP d'état des liens IPv4 pour le routage des paquets et la mise en réseau à haute disponibilité.
BGP (Border Gateway Protocol)	`bgpd`	Protocole EGP IPv4 et IPv6 pour le routage d'un domaine administratif à l'autre.

◆ ◆ ◆ CHAPITRE 8

Référence IPv6

Ce chapitre contient des informations de référence concernant l'implémentation du protocole IPv6 sous Oracle Solaris.

Les tâches de configuration d'un réseau compatible IPv6 sont décrites dans le Chapitre 4, "Activation d'IPv6 sur le réseau". Pour obtenir des informations sur les tunnels IP, reportez-vous au Chapitre 6, "Configuration de tunnels IP".

Implémentation IPv6 sous Oracle Solaris

Cette section décrit les fichiers, commandes et démons nécessaires au protocole IPv6 sous Oracle Solaris.

Fichiers de configuration IPv6

Cette section décrit les fichiers de configuration faisant partie de l'implémentation IPv6 :

Fichier de configuration `ndpd.conf`

Le fichier de configuration `/etc/inet/ndpd.conf` sert à configurer les options utilisées par le démon Neighbor Discovery `in.ndpd`. Pour un routeur, `ndpd.conf` sert principalement à

configurer le préfixe du site à publier vers le lien. Pour un hôte, `ndpd.conf` sert à désactiver la configuration automatique des adresses ou à configurer des adresses temporaires.

Le tableau suivant présente les mots-clés utilisés dans le fichier `ndpd.conf`.

TABLEAU 8–1 Mots-clés de `/etc/inet/ndpd.conf`

Variable	Description
`ifdefault`	Spécifie le comportement du routeur pour toutes les interfaces. Utilisez la syntaxe suivante pour définir les paramètres du routeur et les valeurs correspondantes : `ifdefault` [*valeur variable*]
`prefixdefault`	Spécifie le comportement par défaut pour la publication du préfixe. Utilisez la syntaxe suivante pour définir les paramètres du routeur et les valeurs correspondantes : `prefixdefault` [*valeur variable*]
`if`	Définit les paramètres de l'interface. Utilisez la syntaxe suivante : `if` *interface* [*valeur variable*]
`prefix`	Publie les informations du préfixe par interface. Utilisez la syntaxe suivante : `prefix` *préfixe/longueur interface* [*valeur variable*]

Dans le fichier `ndpd.conf`, vous utilisez des mots-clés du tableau avec jeu de variables de configuration du routeur. Ces variables sont définies en détail dans le document RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (`http://www.ietf.org/rfc/rfc2461.txt?number=2461`).

Le tableau suivant répertorie les variables de configuration d'une interface et fournit une brève définition de chacune.

TABLEAU 8–2 Variables de configuration d'interface du fichier `/etc/inet/ndpd.conf`

Variable	Par défaut	Définition
`AdvRetransTimer`	0	Spécifie la valeur du champ Retrans Timer pour la publication de messages envoyés par le routeur.
`AdvCurHopLimit`	Diamètre actuel du réseau Internet	Spécifie la valeur à entrer dans le champ Hop Limit pour la publication de messages envoyés par le routeur.
`AdvDefaultLifetime`	3 + `MaxRtrAdvInterval`	Spécifie la durée de vie par défaut des publications du routeur.
`AdvLinkMTU`	0	Spécifie une valeur d'unité de transmission maximale (MTU) que le routeur doit envoyer. Une valeur nulle indique que ne routeur ne spécifie pas d'options MTU.

TABLEAU 8–2 Variables de configuration d'interface du fichier `/etc/inet/ndpd.conf` *(Suite)*

Variable	Par défaut	Définition
`AdvManaged Flag`	False	Spécifie la valeur à entrer dans l'indicateur de configuration de la gestion des adresses pour la publication du routeur.
`AdvOtherConfigFlag`	False	Spécifie la valeur à entrer dans l'indicateur de configuration des autres paquets avec état pour la publication du routeur.
`AdvReachableTime`	0	Spécifie la valeur du champ Reachable Time pour la publication de messages envoyés par le routeur.
`AdvSendAdvertisements`	False	Indique si le noeud doit envoyer des publications et répondre aux requêtes du routeur. Vous devez définir explicitement la variable sur TRUE dans le fichier `ndpd.conf` afin d'activer les fonctions de publication du routeur. Pour plus d'informations, reportez-vous à la section "Configuration d'un routeur compatible IPv6" à la page 66.
`DupAddrDetect Transmits`	1	Définit le nombre de messages de requête voisine consécutifs que le protocole Neighbor Discovery doit envoyer lors de la détection d'adresses du noeud local dupliquées.
`MaxRtrAdvInterval`	600 secondes	Spécifie le temps d'attente maximal lors de l'envoi de publications de multidiffusion non requises.
`MinRtrAdvInterval`	200 secondes	Spécifie le temps d'attente minimal lors de l'envoi de publications de multidiffusion non requises.
`StatelessAddrConf`	True	Détermine si le noeud configure son adresse IPv6 par le biais de la configuration automatique des adresses sans état. Si la valeur False est déclarée dans le fichier `ndpd.conf`, l'adresse doit être configurée manuellement. Pour plus d'informations, reportez-vous à la section "Configuration d'un jeton IPv6 spécifié par l'utilisateur" à la page 72.
`TmpAddrsEnabled`	False	Indique si une adresse temporaire doit être créée pour toutes les interfaces ou pour une interface particulière d'un noeud. Pour plus d'informations, reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.
`TmpMaxDesyncFactor`	600 secondes	Spécifie une valeur aléatoire à soustraire de la variable de durée de vie préférée `TmpPreferredLifetime` au démarrage de la commande `in.ndpd`. L'objectif de la variable `TmpMaxDesyncFactor` est d'éviter que tous les systèmes de votre réseau ne régénèrent leurs adresses temporaires en même temps. `TmpMaxDesyncFactor` permet de remplacer la limite supérieure par cette valeur.
`TmpPreferredLifetime`	False	Définit la durée de vie préférée d'une adresse temporaire. Pour plus d'informations, reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.
`TmpRegenAdvance`	False	Spécifie à l'avance la durée d'obtention d'une désapprobation pour une adresse temporaire. Pour plus d'informations, reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.

TABLEAU 8–2 Variables de configuration d'interface du fichier `/etc/inet/ndpd.conf` *(Suite)*

Variable	Par défaut	Définition
`TmpValidLifetime`	False	Définit la durée de vie correcte d'une adresse temporaire. Pour plus d'informations, reportez-vous à la section "Configuration d'une adresse temporaire" à la page 69.

Le tableau suivant répertorie les variables utilisées pour configurer les préfixes IPv6.

TABLEAU 8–3 Variables de configuration de préfixe du fichier `/etc/inet/ndpd.conf`

Variable	Par défaut	Définition
`AdvAutonomousFlag`	True	Spécifie la valeur à entrer dans le champ Autonomous Flag figurant dans les informations sur le préfixe.
`AdvOnLinkFlag`	True	Spécifie la valeur à entrer dans l'indicateur on-link "L-bit" figurant dans les informations sur le préfixe.
`AdvPreferredExpiration`	Non définie	Spécifie la date d'expiration préférée du préfixe.
`AdvPreferredLifetime`	604 800 secondes	Spécifie la valeur à entrer pour la durée de vie préférée dans les informations sur le préfixe.
`AdvValidExpiration`	Non définie	Spécifie la date d'expiration correcte du préfixe.
`AdvValidLifetime`	2 592 000 secondes	Spécifie la durée de vie correcte du préfixe qui est configurée.

EXEMPLE 8–1 Fichier `/etc/inet/ndpd.conf`

L'exemple suivant répertorie les mots-clés et les variables de configuration utilisés dans le fichier `ndpd.conf`. Supprimez le commentaire (#) pour activer la variable.

```
# ifdefault       [variable-value ]*
# prefixdefault [variable-value ]*
# if ifname     [variable-value ]*
# prefix prefix/length ifname
#
#  Per interface configuration variables
#
#DupAddrDetectTransmits
#AdvSendAdvertisements
#MaxRtrAdvInterval
#MinRtrAdvInterval
#AdvManagedFlag
#AdvOtherConfigFlag
#AdvLinkMTU
#AdvReachableTime
#AdvRetransTimer
#AdvCurHopLimit
#AdvDefaultLifetime
#
# Per Prefix:  AdvPrefixList configuration variables
```

EXEMPLE 8–1 Fichier `/etc/inet/ndpd.conf` *(Suite)*

```
#
#
#AdvValidLifetime
#AdvOnLinkFlag
#AdvPreferredLifetime
#AdvAutonomousFlag
#AdvValidExpiration
#AdvPreferredExpiration

ifdefault AdvReachableTime 30000 AdvRetransTimer 2000
prefixdefault AdvValidLifetime 240m AdvPreferredLifetime 120m

if qe0 AdvSendAdvertisements 1
prefix 2:0:0:56::/64 qe0
prefix fec0:0:0:56::/64 qe0

if qe1 AdvSendAdvertisements 1
prefix 2:0:0:55::/64 qe1
prefix fec0:0:0:56::/64 qe1

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:1::/64 qfe0

if hme1 AdvSendAdvertisements 1
prefix  2002:8192:56bb:2::/64 hme1
```

Fichier de configuration `/etc/inet/ipaddrsel.conf`

Le fichier `/etc/inet/ipaddrsel.conf` contient la table des règles de sélection d'adresse IPv6 par défaut. Si vous avez activé le protocole IPv6 lors de l'installation d'Oracle Solaris, ce fichier contient les éléments présentés dans le Tableau 8–4.

Vous pouvez modifier le contenu de `/etc/inet/ipaddrsel.conf`. Toutefois, cette opération n'est pas recommandée. Si cela s'avère nécessaire, reportez-vous à la procédure décrite à la section “Administration de la table des règles de sélection d'adresses IPv6” à la page 100. Pour plus d'informations sur le fichier `ippaddrsel.conf`, reportez-vous à la section “Raisons pour lesquelles le tableau des règles de sélection d'adresses IPv6 doit être modifié” à la page 138 ainsi qu'à la page de manuel `ipaddrsel.conf`(4).

Commandes associées à IPv6

Cette section décrit les commandes ajoutées lors de l'implémentation du protocole IPv6 sous Oracle Solaris. Les commandes existantes qui ont été modifiées pour prendre en charge IPv6 y sont également détaillées.

Commande `ipaddrsel`

La commande `ipaddrsel` permet de modifier le tableau des règles de sélection des adresses IPv6 par défaut.

Le noyau Oracle Solaris utilise la table des règles de sélection des adresses IPv6 par défaut pour le classement des adresses de destination et la sélection des adresses sources pour les en-têtes de paquet IPv6. Le fichier `/etc/inet/ipaddrsel.conf` contient ce tableau de règles.

Le tableau suivant répertorie les formats d'adresse par défaut ainsi que les priorités de chacune telles qu'elles doivent figurer dans le tableau de règles. Vous pouvez rechercher des informations techniques sur la sélection d'adresses IPv6 dans la page de manuel `inet6(7P)`.

TABLEAU 8–4 Tableau des règles de sélection des adresses IPv6 par défaut

Préfixe	Priorité	Définition
`::1/128`	50	Loopback
`::/0`	40	Par défaut
`2002::/16`	30	6to4
`::/96`	20	IPv4 Compatible
`::ffff:0:0/96`	10	IPv4

Dans ce tableau, les préfixes IPv6 (`::1/128` et `::/0`) ont la priorité sur les adresses 6to4 (`2002::/16`) et les adresses IPv4 (`::/96` et `::ffff:0:0/96`). Par conséquent, le noyau choisit par défaut l'adresse IPv6 globale de l'interface pour les paquets envoyés vers une autre destination IPv6. L'adresse IPv4 de l'interface est moins prioritaire, notamment pour les paquets envoyés vers une destination IPv6. Etant donné l'adresse IPv6 source sélectionnée, le noyau utilise également le format IPv6 pour l'adresse de destination.

Raisons pour lesquelles le tableau des règles de sélection d'adresses IPv6 doit être modifié

En règle générale, le tableau des règles de sélection d'adresses IPv6 par défaut n'a pas besoin d'être modifié. En cas de modification nécessaire, exécutez la commande `ipaddrsel`.

Les situations suivantes nécessitent une modification du tableau :

- Si le système possède une interface qui est utilisée pour un tunnel 6to4, vous pouvez définir une priorité plus élevée pour les adresses 6to4.
- Si vous souhaitez qu'une adresse source particulière communique avec une adresse de destination particulière, vous pouvez ajouter ces adresses au tableau de règles. Ensuite, vous pouvez les marquer comme vos adresses préférées à l'aide de la commande `ipadm`. Pour plus d'informations sur la commande `ipadm`, reportez-vous à la page de manuel `ipadm(1M)`.
- Si vous voulez que les adresses IPv4 aient la priorité sur les adresses IPv6, vous pouvez remplacer la priorité de `::ffff:0:0/96` par un chiffre plus élevé.

- Si vous devez assigner une priorité plus élevée à des adresses désapprouvées, vous pouvez ajouter ces adresses au tableau de règles. Prenons l'exemple des adresses de site locales, actuellement désapprouvées sur le réseau IPv6. Ces adresses possèdent le préfixe `fec0::/10`. Vous pouvez modifier le tableau de règles afin de définir une priorité plus élevée pour ces adresses.

Pour plus d'informations sur la commande `ipaddrsel`, reportez-vous à la page de manuel `ipaddrsel(1M)`.

Commande `6to4relay`

La *création de tunnel 6to4* permet à des sites 6to4 isolés de communiquer. Cependant, pour transférer des paquets vers un site IPv6 natif et non-6to4, le routeur 6to4 doit être relié au routeur relais 6to4 par un tunnel. Le *routeur relais 6to4* transfère ensuite les paquets 6to4 au réseau IPv6 et, finalement, au site IPv6 natif. Si un site 6to4 doit échanger des données avec un site IPv6, vous pouvez créer le tunnel approprié à l'aide de la commande `6to4relay`.

Sous Oracle Solaris, la liaison de tunnels à des routeurs relais est désactivée car l'utilisation des routeurs relais n'est pas sécurisée. Avant de relier un tunnel à un routeur relais 6to4, vous devez être conscient des problèmes qui peuvent survenir avec ce type de scénario. Pour plus d'informations sur les routeurs relais 6to4, reportez-vous à la section "Informations importantes pour la création de tunnels vers un routeur relais 6to4" à la page 108. Pour activer la prise en charge d'un routeur relais 6to4, vous pouvez suivre la procédure indiquée à la section "Création et configuration d'un tunnel IP" à la page 113.

Syntaxe de la commande 6to4relay

La commande `6to4relay` possède la syntaxe suivante :

```
6to4relay -e [-a IPv4-address] -d -h
```

`-e`	Assure la prise en charge de tunnels entre le routeur 6to4 et un routeur relais 6to4 anycast. Ainsi, l'adresse du point d'extrémité du tunnel est définie sur `192.88.99.1` , soit l'adresse du groupe anycast de routeurs relais 6to4.
`-a` *IPv4-address*	Assure la prise en charge de tunnels entre le routeur 6to4 et un routeur relais 6to4 possédant l'*IPv4-address* spécifiée.
`-d`	Désactive la prise en charge de tunnels vers un routeur relais 6to4 (paramètre par défaut d'Oracle Solaris).
`-h`	Affiche l'aide concernant la commande `6to4relay`.

Pour plus d'informations, reportez-vous à la page de manuel `6to4relay`(1M).

EXEMPLE 8–2 Affichage par défaut du statut de la prise en charge de routeurs relais 6to4

La commande `6to4relay`, sans argument, affiche le statut actuel de la prise en charge des routeurs relais 6to4. Cet exemple indique la sortie par défaut de l'implémentation du protocole IPv6 sous Oracle Solaris.

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is disabled
```

EXEMPLE 8–3 Affichage du statut avec prise en charge des routeurs relais 6to4 activée

Lorsque la prise ne charge des routeurs relais est activée, la commande `6to4relay` affiche la sortie suivante :

```
# /usr/sbin/6to4relay
6to4relay:6to4 Relay Router communication support is enabled
IPv4 destination address of Relay Router=192.88.99.1
```

EXEMPLE 8–4 Affichage du statut avec un routeur relais 6to4 spécifié

Si vous spécifiez l'option `-a` et une adresse IPv4 dans la commande `6to4relay`, l'adresse IPv4 fournie avec l'option `- a` remplace l'adresse `192.88.99.1`.

La commande `6to4relay` ne signale pas l'exécution des options `-d`, `-e` et `-a` *IPv4 address*. Cependant, `6to4relay` n'affiche aucun message d'erreur lié à l'exécution de ces options.

Modification de la commande `netstat` en vue de la prise en charge IPv6

La commande `netstat` affiche le statut des réseaux IPv4 et IPv6. Vous pouvez choisir les informations de protocole à afficher en définissant la valeur de `DEFAULT_IP` dans le fichier `/etc/default/inet_type` ou en utilisant l'option `-f` dans la ligne de commande. Avec une valeur de `DEFAULT_IP` permanente, vous vous assurez que la commande `netstat` affiche uniquement les informations IPv4. Vous pouvez ignorer ce paramètre et utiliser l'option `-f`. Pour plus d'informations sur le fichier `inet_type` , reportez-vous à la page de manuel `inet_type(4)`.

L'option `-p` de la commande `netstat` affiche la table des connexions réseau-média, c'est-à-dire la table des protocoles de résolution d'adresse pour l'IPv4 et le cache voisin pour l'IPv6. Pour plus d'informations, reportez-vous à la page de manuel `netstat(1M)` La section “Affichage du statut des sockets” à la page 84 décrit les procédures impliquant l'exécution de cette commande.

Modification de la commande snoop en vue de la prise en charge IPv6

La commande `snoop` permet de capturer des paquets IPv4 et IPv6. Cette commande peut s'afficher avec des en-têtes IPv6, des en-têtes d'extension IPv6, des en-têtes ICMPv6 et des données de protocole Neighbor Discovery. Par défaut, la commande `snoop` affiche les deux types de paquet (IPv4 et IPv6). Pour afficher soit l'un, soit l'autre, spécifiez le mot-clé de

protocole `ip` ou `ip6` avec la commande `snoop`. L'option de filtrage IPv6 vous permet de filtrer tous les paquets IPv4 et IPv6 et d'afficher uniquement les paquets IPv6. Pour plus d'informations, reportez-vous à la page de manuel `snoop`(1M) La section "Contrôle du trafic réseau IPv6" à la page 96 décrit les procédures impliquant l'exécution de la commande `snoop`.

Modification de la commande `route` en vue de la prise en charge IPv6

La commande `route` fonctionne sur les routes IPv4 (par défaut) et IPv6. Pour réaliser des opérations sur les routes IPv6, tapez l'option `-inet6` immédiatement à la suite de la commande `route` dans la ligne de commande. Pour plus d'informations, reportez-vous à la page de manuel `route`(1M).

Modification de la commande `ping` en vue de la prise en charge IPv6

La commande `ping` se sert des protocoles IPv4 et IPv6 pour sonder les hôtes cibles. Le choix du protocole dépend des adresses renvoyées par le serveur de noms pour l'hôte cible spécifique. Par défaut, si ce serveur renvoie une adresse IPv6 pour l'hôte cible, la commande `ping` utilise le protocole IPv6. S'il renvoie une adresse IPv4, la commande `ping` utilise le protocole IPv4. Pour ignorer cette action, vous pouvez taper l'option `-A` dans la ligne de commande et spécifier le protocole à utiliser.

Pour plus d'informations, reportez-vous à la page de manuel `ping`(1M) La section "Test des hôtes distants à l'aide de la commande `ping`" à la page 88 décrit les procédures impliquant l'exécution de la commande `ping` .

Modification de la commande `traceroute` en vue de la prise en charge IPv6

Vous pouvez exécuter la commande `traceroute` pour tracer les routes IPv4 et IPv6 vers un hôte spécifique. Du point de vue du protocole, `traceroute` utilise le même algorithme que la commande `ping`. Pour ignorer ce choix, tapez l'option `-A` dans la ligne de commande. Vous pouvez tracer chaque route vers chaque adresse d'un hôte multiréseau en tapant l'option `-a` dans la ligne de commande.

Pour plus d'informations, reportez-vous à la page de manuel `traceroute`(1M) La section "Affichage des informations de routage à l'aide de la commande `traceroute`" à la page 92 décrit les procédures qui impliquent l'exécution de la commande `traceroute`.

Démons liés à IPv6

Cette section présente les démons liés à IPv6.

Démon `in.ndpd` pour Neighbor Discovery

Le démon `in.ndpd` implémente le protocole IPv6 Neighbor Discovery ainsi que celui de découverte de routeur. Il implémente également la configuration automatique d'adresse IPv6. Les options suivantes sont prises en charge par `in.ndpd`.

`-a`	Désactive la configuration automatique d'adresse avec ou sans état.
`-d`	Active le débogage.
`-f` *config-file*	Spécifie un fichier de configuration spécifique à la place du fichier `/etc/inet/ndpd.conf`.
`-t`	Active le suivi de tous les paquets sortants et entrants.

Le démon `in.ndpd` est contrôlé par les paramètres définis dans le fichier de configuration `/etc/inet/ndpd.conf` et par ceux du fichier de démarrage `/var/inet/ndpd_state.` *interface* qui s'appliquent.

Lorsque le fichier `/etc/inet/ndpd.conf` existe, il est analysé et utilisé pour configurer un noeud en tant que routeur. Le Tableau 8–1 répertorie les mots-clés corrects susceptibles de figurer dans ce fichier. Lors de l'initialisation d'un hôte, les routeurs risquent de ne pas être disponibles immédiatement. Les paquets publiés par le routeur risquent d'être abandonnés. En outre, les paquets risquent de ne pas atteindre l'hôte.

Le fichier `/var/inet/ndpd_state.`*interface* est un fichier d'état. Ce fichier est régulièrement mis à jour par chaque noeud. En cas de défaillance et de redémarrage du noeud, ce dernier peut configurer ses interfaces en l'absence de routeurs. Ce fichier contient l'adresse de l'interface, l'heure de la dernière mise à jour du fichier et la durée de validité du fichier. Il contient également d'autres paramètres “hérités” de précédentes publications de routeur.

Remarque – Il est inutile de modifier le contenu des fichiers d'état. Le démon `in.ndpd` assure la maintenance automatique des fichiers d'état.

Reportez-vous aux pages de manuel `in.ndpd(1M)` et `ndpd.conf(4)` pour obtenir des listes des variables de configuration et des valeurs acceptables.

Démon `in.ripngd`, pour routage IPv6

Le démon `in.ripngd` implémente les informations de RIPng (Routing Information Protocol next-generation, protocole d'informations de routage nouvelle génération) pour les routeurs IPv6. Le RIPng définit l'équivalent IPv6 de RIP (Routing Information Protocol, protocole d'informations de routage). Lorsque vous configurez un routeur IPv6 avec la commande `routeadm` et activez le routage IPv6, le démon `in.ripngd` implémente RIPng sur le routeur.

Vous trouverez ci-dessous les options RIPng prises en charge.

`-p` *n* *n* spécifie le numéro de port UDP utilisé pour l'envoi ou la réception de paquets RIPng.

`-P` Supprime l'utilisation du mode poison reverse.

`-q` Supprime les informations de routage.

`-s` Force le routage d'informations même si le démon fait office de routeur.

`-t` Imprime tous les paquets envoyés et reçus vers une sortie standard.

`-v` Imprime toutes les modifications apportées à la table de routage vers la sortie standard, en incluant les horodatages.

Démon `inetd` et services IPv6

Une application de serveur compatible IPv6 peut gérer les demandes IPv4 et IPv6, ou les demandes IPv6 uniquement. Le serveur gère toujours les demandes par le biais d'un socket IPv6. En outre, le serveur utilise le même protocole qu'utilise le client correspondant.

Pour ajouter ou modifier un service pour IPv6, utilisez les commandes disponibles à partir du service SMF (Service Management Facility, utilitaire de gestion des services).

- Pour plus d'informations sur les commandes SMF, reportez-vous à la section "Utilitaires d'administration en ligne de commande SMF" du manuel *Gestion des services et pannes dans Oracle Solaris 11.1*.
- Pour obtenir une tâche d'exemple utilisant le service SMF pour configurer un manifeste de service IPv4 s'exécutant sur SCTP, reportez-vous à la section "Ajout de services utilisant le protocole SCTP" à la page 57.

Pour configurer un service IPv6, vous devez vous assurer que la valeur du champ `proto` dans le profil `inetadm` pour ce service répertorie la valeur adéquate :

- Pour un service assurant la gestion de demandes IPv4 et IPv6, sélectionnez `tcp6`, `udp6` ou `sctp`. Une valeur `proto` de `tcp6`, `udp6` ou `sctp6` a pour conséquence de faire passer `inetd` sur un socket IPv6 vers le serveur. Le serveur contient une adresse mappée IPv4 au cas où un client IPv4 recevrait une demande.
- Pour un service qui gère uniquement les demandes IPv6, sélectionnez `tcp6only` ou `udp6only`. Si `proto` a l'une de ces valeurs, `inetd` passe le serveur à un socket IPv6.

Si vous remplacez une command'Oracle Solaris par une autre implémentation, vous devez vous assurer que l'implémentation de ce service prend en charge le protocole IPv6. Si l'implémentation ne prend pas IPv6 en charge, vous devez spécifier la valeur `proto` en tant que `tcp`, `udp` ou `sctp`.

Voici un profil qui résulte de l'exécution de `inetadm` pour un manifeste de service `echo` prenant IPv4 et IPv6 en charge, et s'exécute sur SCTP :

```
# inetadm -l svc:/network/echo:sctp_stream
    SCOPE    NAME=VALUE      name="echo"
             endpoint_type="stream"
             proto="sctp6"
             isrpc=FALSE
             wait=FALSE
             exec="/usr/lib/inet/in.echod -s"
             user="root"
    default  bind_addr=""
    default  bind_fail_max=-1
    default  bind_fail_interval=-1
    default  max_con_rate=-1
    default  max_copies=-1
    default  con_rate_offline=-1
    default  failrate_cnt=40
    default  failrate_interval=60
    default  inherit_env=TRUE
    default  tcp_trace=FALSE
    default  tcp_wrappers=FALSE
```

La syntaxe suivante permet de modifier la valeur du champ `proto` :

```
# inetadm -m FMRI proto="transport-protocols"
```

Tous les serveurs fournis avec le logiciel Oracle Solaris ne nécessitent qu'une entrée de profil spécifiant `proto` en tant que `tcp6`, `udp6` ou `sctp6`. Cependant, le serveur shell distant (`shell`) et le serveur d'exécution distant (`exec`) sont à présent composés d'une instance de service unique, nécessitant une valeur `proto` contenant les valeurs `tcp` et `tcp6only`. Par exemple, pour définir la valeur `proto` pour `shell`, émettez la commande suivante :

```
# inetadm -m network/shell:default proto="tcp,tcp6only"
```

Consultez les extensions IPv6 de l'API Socket dans la section *Programming Interfaces Guide* pour obtenir des informations supplémentaires sur l'écriture de serveurs compatibles IPv6 qui utilisent des sockets.

Informations importantes relatives à la configuration d'un service pour IPv6

Gardez les éléments suivants à l'esprit lorsque vous ajoutez ou modifiez un service pour IPv6 :

- Vous devez spécifier la valeur `proto` en tant que `tcp6`, `sctp6` ou `udp6` afin d'activer les connexions IPv4 ou IPv6. Si vous spécifiez la valeur pour `proto` en tant que `tcp`, `sctp` ou `udp`, le service n'utilise qu'IPv4.
- Bien qu'il soit possible d'ajouter une instance de service utilisant des sockets SCTP de style un à plusieurs à `inetd`, il est déconseillé de le faire. `inetd` ne fonctionne pas avec les sockets SCTP de style un à plusieurs.
- Si un service nécessite deux entrées en raison de propriétés `wait-status` ou `exec` différentes, vous devez créer deux instances/services à partir du service d'origine.

Protocole ND IPv6

IPv6 présente le protocole Neighbor Discovery, comme décrit dans le document RFC 2461, Neighbor Discovery for IP Version 6 (IPv6) (`http://www.ietf.org/rfc/rfc2461.txt?number=2461`).

Cette section décrit les fonctionnalités suivantes du protocole ND :

Messages ICMP de la détection des voisins

La détection de voisins définit cinq nouveaux messages ICMP (Internet Control Message Protocol, protocole de messages de contrôle Internet). Les messages remplissent les fonctions suivantes :

- **Sollicitation de routeur** : lorsqu'une interface est activée, les hôtes peuvent demander des messages de sollicitation de routeur. Les sollicitations demandent aux routeurs de générer immédiatement des publications de routeurs, plutôt qu'à la prochaine heure prévue.
- **Publication de routeur** : les routeurs publient leur présence, divers liens de paramètres et divers liens de paramètres Internet. Les routeurs effectuent des publications régulières ou en réponse à un message de sollicitation de routeur. Les publications de routeur contiennent des préfixes utilisés pour la détermination sur lien ou la configuration d'adresse, une valeur de limite de saut recommandée, et ainsi de suite.
- **Sollicitation de voisin** : les noeuds envoient des messages de sollicitation de voisins afin de déterminer l'adresse de couche liaison du voisin. Les messages de sollicitation de voisin sont également envoyés afin de vérifier qu'un voisin est toujours accessible par une adresse de couche liaison mise en cache. Les sollicitations s'utilisent également pour la détection d'adresses dupliquées.
- **Publication de voisins** : un noeud envoie des messages de publication de voisinage en réponse à un message de sollicitation de voisinage Le noeud peut également envoyer des publications de voisinage non sollicitées pour signaler une modification de l'adresse de couche liaison.
- **Redirection** : les routeurs utilisent les messages de redirection afin d'informer les hôtes de l'existence d'un meilleur saut pour une destination ou que la destination se trouve sur la même liaison.

Processus de configuration automatique

Cette section comprend une présentation des étapes typiques effectuées par une interface lors d'une configuration automatique. La configuration automatique s'effectue uniquement sur des liaisons compatibles multicast.

1. Une interface compatible multicast est activée, par exemple, lors du démarrage système d'un noeud.
2. Le noeud démarre le processus de configuration automatique en générant une adresse lien-local pour l'interface.

   L'adresse lien-local est formée à partir de l'adresse MAC (Media Access Control) de l'interface.
3. Le noeud envoie un message de sollicitation de voisin contenant l'adresse lien-local provisoire en guise de cible.

   Le message a pour objectif de vérifier que l'adresse possible n'est pas déjà utilisée par un autre noeud sur la liaison. Une fois la vérification effectuée, l'adresse lien-local peut être assignée à l'interface.

   a. Si un autre noeud utilise déjà l'adresse proposée, celui-ci renvoie une publication de voisin indiquant que l'adresse est déjà en cours d'utilisation.

   b. Si un autre noeud tente également d'utiliser la même adresse, le noeud envoie également une sollicitation de voisinage pour la cible.

      Le nombre de transmissions ou de retransmissions de sollicitation de voisins, ainsi que le temps d'attente entre sollicitations, sont spécifiques aux liaisons. Au besoin, vous pouvez définir ces paramètres.
4. Si un noeud détermine que son adresse lien-local possible n'est pas unique, la configuration automatique est interrompue. Dans ce cas, vous devrez configurer manuellement l'adresse lien-local de l'interface.

   Pour simplifier la récupération, vous pouvez fournir un autre ID d'interface qui remplace l'identifiant par défaut. Ensuite, le mécanisme de configuration automatique peut reprendre, en utilisant le nouvel ID d'interface, qui est à priori unique.
5. Lorsqu'un noeud détermine l'unicité de sa future adresse lien-local, il assigne celle-ci à l'interface.

   Le noeud dispose alors d'une connectivité de niveau IP avec les noeuds voisins. Les étapes restantes de la configuration automatique sont effectuées exclusivement par les hôtes.

Obtention d'une publication de routeur

La phase suivante de la configuration automatique consiste à obtenir une publication de routeur ou à déterminer une absence totale de routeurs. Si les routeurs sont présents, ils envoient des publications de routeur qui spécifient le type de configuration automatique que doit effectuer un hôte.

Les routeurs envoient des publications de routeur à intervalles réguliers. Cependant, le temps d'attente entre publications successives est en règle générale plus long que le temps d'attente possible d'un hôte effectuant la configuration automatique. Afin d'obtenir une publication dans les plus brefs délais, un hôte envoie une ou plusieurs sollicitations de routeur au groupe multicast tous routeurs.

Variables de préfixes de configuration

La publication de routeur contient également des variables de préfixe avec des informations utilisées par la configuration automatique d'adresse sans état pour la génération de préfixes. Le champ de configuration automatique d'adresse sans état dans les publications de routeur sont traitées indépendamment. Un champ d'option contenant les informations de préfixe, l'indicateur de configuration automatique d'adresse, indique si l'option s'applique également à la configuration automatique sans état. Si le champ d'option s'y applique, des champs d'option supplémentaires contiennent un préfixe de sous-réseau avec des valeurs de durée de vie. Ces valeurs indiquent la durée de validité et de préférence des adresses créées à partir du préfixe.

Dans la mesure où les routeurs génèrent régulièrement des publications de routeur, les hôtes reçoivent de nouvelles publications en continu. Les hôtes compatibles IPv6 traitent les informations contenues dans chaque publication. Les hôtes ajoutent des informations. Ils actualisent également les informations reçues dans les publications précédentes.

Unicité des adresses

Pour des raisons de sécurité, l'unicité de toutes les adresses doit être vérifiée, préalablement à leur assignation à une interface. La situation est différente pour les adresses créées par configuration automatique sans état. L'unicité d'une adresse est déterminée principalement par la partie de l'adresse formée à partir d'un ID d'interface. Par conséquent, si un noeud a déjà vérifié l'unicité d'une adresse lien-local, il est inutile de tester les adresses supplémentaires individuellement. Les adresses doivent être créées à partir du même ID d'interface. Toutes les adresses obtenues manuellement doivent par contre être testées individuellement pour leur unicité. Les administrateurs système de certains sites pensent que les bénéfices de la détection d'adresses dupliquées ne vaut pas le temps système qu'elle utilise. Pour ces sites, l'utilisation de la détection des adresses dupliquées peut être désactivée en définissant un indicateur de configuration par interface.

Pour accélérer le processus de configuration automatique, un hôte peut générer son adresse lien-local et vérifier son unicité, pendant que l'hôte attend une publication de routeur. Un

routeur peut retarder une réponse à une sollicitation de routeur de quelques secondes. Par conséquent, le temps total nécessaire à la configuration automatique peut être bien plus long si les deux étapes sont effectuées en série.

Sollicitation de voisin et inaccessibilité

La détection de voisins utilise les messages de *sollicitation de voisin* pour déterminer si plusieurs noeuds sont assignés à la même adresse unicast. *La détection d'inaccessibilité de voisin* détecte la défaillance d'un voisin ou du chemin de transfert du voisin. Cette détection nécessite une confirmation de la réception des paquets par le voisin. La détection d'inaccessibilité de voisins détermine également que les paquets sont traités correctement par la couche IP du noeud.

La détection d'inaccessibilité de voisin utilise les confirmations en provenance de deux sources : les protocoles de couche supérieure et les messages de sollicitation de voisin. Lorsque c'est possible, les protocoles de couche supérieure fournissent une confirmation positive de la *progression* d'une connexion. Par exemple, à la réception d'accusés de réception TCP, il est confirmé que les données précédemment envoyées ont été livrées correctement.

Lorsqu'un noeud n'obtient pas de confirmation en provenance des protocoles de couche supérieure, le noeud envoie des messages de sollicitation de voisins. Ces messages sollicitent des publications de voisinage en tant que confirmation d'accessibilité à partir du prochain saut. Pour réduire le trafic réseau inutile, les messages de sonde sont envoyés uniquement au noeud envoyant des paquets activement.

Algorithme de détection d'adresse dupliquée

Pour garantir que toutes les adresses configurées sont susceptibles d'être uniques sur un lien donné, les noeuds exécutent un algorithme de *détection d'adresse dupliquée* sur les adresses. Les noeuds doivent exécuter l'algorithme avant d'assigner les adresses à une interface. L'algorithme de détection d'adresses dupliquées est exécuté sur toutes les adresses.

Le processus de configuration automatique décrit dans cette section s'applique uniquement aux hôtes et non aux routeurs. Dans la mesure où la configuration automatique utilise des informations publiées par les routeurs, ces derniers doivent être configurés différemment. Cependant, les routeurs génèrent des adresses lien-local à l'aide du mécanisme décrit dans ce chapitre. En outre, les routeurs doivent réussir l'algorithme de détection d'adresses dupliquées sur toutes les adresses préalablement à l'assignation d'une adresse à une interface.

Publications de proxy

Un routeur qui accepte les paquets à la place d'une adresse cible peut émettre des publications de voisin impossibles à ignorer. Le routeur peut accepter des paquets pour une adresse cible incapable de répondre aux sollicitations de voisins. L'utilisation de proxy n'est actuellement pas spécifiée. Cependant, la publication de proxy pourrait être utilisée pour la gestion de cas comme ceux de noeuds mobiles qui ont été déplacés hors liaison. Notez que l'utilisation de proxy n'est pas destinée à l'être en tant que mécanisme général de gestion des noeuds qui n'implémentent pas ce protocole.

Equilibrage de charge entrante

Les noeuds avec interfaces répliquées peuvent avoir besoin d'équilibrer la charge de la réception de paquets entrants sur plusieurs interfaces réseau situées sur la même liaison. Ces noeuds possèdent plusieurs adresses lien-local assignées à la même interface. Par exemple, un pilote de réseau unique peut représenter plusieurs cartes d'interface réseau en tant qu'interface logique unique possédant plusieurs adresses lien-local.

La gestion de l'équilibrage de charge s'effectue en autorisant les routeurs à omettre l'adresse lien-local source des paquets de publication de routeur. Par conséquent, les voisins doivent utiliser les messages de sollicitation de voisin afin de connaître les adresses lien-local des routeurs. Les messages renvoyés de publication des voisins peuvent contenir des adresses lien-local différentes, selon l'adresse qui a envoyé la demande.

Modification d'adresse lien-local

Un noeud qui sait que son adresse lien-local a été modifiée peut envoyer des paquets de publication de voisins multicast non sollicités. Le noeud peut envoyer des paquets multicast à tous les noeuds pour une mise à jour des adresses lien-local mises en cache qui ne sont plus valides. L'envoi de publications non sollicitées constitue uniquement une amélioration des performances. L'algorithme de détection d'inaccessibilité des voisins assure la fiabilité de la détection de la nouvelle adresse par le noeud, bien que le temps d'attente risque d'être légèrement plus long.

Comparaison du protocole ND et du protocole ARP et autres protocoles IPv4

La fonctionnalité du protocole ND (Neighbor Discovery, détection des voisins) IPv6 correspond à une combinaison des protocoles IPv4 : ARP (Address Resolution Protocol, protocole de résolution d'adresse), détection de routeur ICMP (Internet Control Message

Protocol, protocole de messages de contrôle Internet) et redirection ICMP. IPv4 ne possède pas de protocole ou de mécanisme accepté par tous pour la détection d'inaccessibilité. Cependant, les exigences de l'hôte spécifient les algorithmes possibles pour la détection de passerelles bloquées. La détection de passerelles bloquées est un sous-ensemble des problèmes résolus par la détection d'inaccessibilité de voisins.

La liste suivante compare le protocole de détection de voisins à la suite de protocoles IPv4 associés.

- La détection de routeur fait partie du jeu de protocoles IPv6 de base. Les hôtes IPv6 n'ont pas besoin d'émettre la commande `snoop` aux protocoles de routage pour rechercher un routeur. IPv4 utilise le protocole ARP, la détection de routeur ICMP et la redirection ICMP pour la détection de routeur.
- Les publications de routeur IPv6 gèrent les adresses lien-local. Aucun échange de paquet supplémentaire n'est nécessaire pour la résolution de l'adresse lien-local du routeur.
- Les publications de routeur gèrent les préfixes de site pour une liaison. Aucun mécanisme séparé n'est nécessaire pour la configuration du masque de réseau, contrairement à IPv4.
- Les publications de routeur sont compatibles avec la configuration automatique d'adresse. La configuration automatique n'est pas implémentée dans IPv4.
- La détection de voisins permet aux routeurs IPv6 de publier la MTU utilisable pour les hôtes sur la liaison. Par conséquent, tous les noeuds utilisent la même valeur de MTU sur des liaisons ne disposant pas d'une MTU correctement définie. Les hôtes IPv4 sur un même réseau peuvent avoir des MTU différentes.
- Contrairement aux adresses de diffusion IPv4, la multidiffusion de résolution d'adresse IPv6 est répartie sur 4 milliards (2^32) d'adresses multicast, ce qui réduit de façon significative les interruptions relatives à la résolution d'adresses sur des noeuds autres que la cible. En outre, les ordinateurs non IPv6 ne doivent pas être éteints.
- Les redirections IPv6 contiennent l'adresse lien-local du premier nouveau saut. La résolution d'adresse séparée n'est pas nécessaire lors de la réception d'une redirection.
- Il est possible d'associer plusieurs préfixes de site au même réseau IPv6. Par défaut, les hôtes sont informés de tous les préfixes de site locaux par les publications de routeur. Cependant, les routeurs peuvent être configurés afin d'omettre certains ou tous les préfixes des publications de routeur. Dans de tels cas, les hôtes partent du principe que les destinations se trouvent sur des réseaux distants. Par conséquent, les hôtes envoient le trafic aux routeurs. Un routeur peut alors émettre des redirections le cas échéant.
- Contrairement à IPv4, le destinataire d'un message IPv6 redirigé part du principe que le nouveau saut suivant se trouve sur le réseau local. Dans IPv4, un hôte ignore les messages de redirection qui spécifient un saut suivant qui ne se trouve pas sur le réseau local, selon le masque de réseau. Le mécanisme de redirection IPv6 est similaire à l'utilitaire XRedirect d'IPv4. Le mécanisme de redirection est utile sur des liens de non diffusion ou de médias partagés. Sur ces réseaux, les noeuds ne doivent pas effectuer de vérification sur tous les préfixes pour les destinations de liaison locale.

- La détection d'inaccessibilité de voisin IPv6 améliore la livraison de paquets en la présence de routeurs défaillants. Cette capacité améliore la livraison de paquets sur des liaisons partiellement défaillantes ou partitionnées. Cette capacité améliore également la livraison de paquet sur des noeuds qui modifient leurs adresses lien-local. Par exemple, les noeuds mobiles peuvent se déplacer hors du réseau local sans aucune perte de connectivité en raison d'anciens caches ARP. IPv4 ne possède pas de méthode correspondante de détection d'inaccessibilité de voisins.
- Contrairement au protocole ARP, la détection de voisins détecte les défaillances de demi liaison à l'aide de la détection d'inaccessibilité de voisins. La détection de voisins évite d'envoyer du trafic aux voisins en l'absence de connectivité bidirectionnelle.
- En utilisant les adresses lien-local pour identifier les routeurs de façon unique, les hôtes IPv6 peuvent conserver les associations de routeur. La capacité d'identification de routeurs est requise pour les publications de routeur et pour les messages de redirection. Les hôtes doivent conserver les associations de routeur si le site utilise de nouveaux préfixes globaux. IPv4 ne possède pas de méthode comparable d'identification des routeurs.
- Dans la mesure où les messages de détection de voisins ont une limite de saut de 255 après réception, le protocole n'est pas affecté par les attaques de mystification en provenance de noeuds hors liaison. Les noeuds IPv4 hors liaison sont eux capables d'envoyer des messages de redirection ICMP. Les noeuds IPv4 hors liaison peuvent également envoyer des messages de publication de routeur.
- En plaçant la résolution d'adresse à la couche ICMP, la détection de voisins est moins dépendante de médias que le protocole ARP. Par conséquent, les mécanismes standard d'authentification IP et de sécurité peuvent être utilisés.

Routage IPv6

Le routage IPv6 est quasiment identique au routage IPv4 sous CIDR (Classless Inter-Domain Routing, routage inter-domaine sans classe). La seule différence est la taille des adresses qui sont de 128 bits dans IPv6 au lieu de 32 bits dans IPv4. Avec des extensions simples, il est possible d'utiliser la totalité des algorithmes de routage d'IPv4 comme OSPF, RIP, IDRP et IS-IS.

IPv6 comprend également des extensions de routage simples qui prennent en charge de nouvelles capacités de routage puissantes. La liste suivante décrit les nouvelles capacités de routage :

- Sélection de fournisseur en fonction de la stratégie, des performances, des coûts, etc.
- Hébergement de mobilité, routage vers emplacement actuel
- Réadressage automatique, routage vers nouvelle adresse

Les nouvelles capacités de routage s'obtiennent par la création de séquences d'adresses IPv6 utilisant l'option de routage IPv6. Une source IPv6 utilise l'option de routage afin de répertorier

un ou plusieurs noeuds intermédiaires, ou groupes topologiques, à visiter en cours d'acheminement vers la destination du paquet. Cette fonction possède énormément de similitudes avec l'option IPv4 de source lâche et de route d'enregistrement.

Pour que les séquences d'adresses soient une fonction générale, les hôtes IPv6 doivent, dans la plupart des cas, inverser les routes d'un paquet reçu par un hôte. Le paquet doit être authentifié à l'aide de l'utilisation de l'en-tête d'authentification IPv6. Le paquet doit contenir des séquences d'adresse afin d'être renvoyé à son point d'origine. Cette technique force les implémentations d'hôtes IPv6 pour la prise en charge de la gestion et de l'inversion des routes source. La gestion et l'inversion des routes source est la clé permettant aux fournisseurs de travailler avec les hôtes qui implémentent les nouvelles capacités IPv6 comme la sélection de fournisseur et les adresses étendues.

Publication de routeur

Sur des liens compatibles multicast et des liens point à point, chaque routeur envoie régulièrement un paquet de publication de routeur au groupe multicast pour lui annoncer sa disponibilité. Un hôte reçoit des publications de routeur de la totalité des routeurs, constituant une liste des routeurs par défaut. Les routeurs génèrent des publications de routeur de façon suffisamment fréquente pour permettre aux hôtes d'être avertis de leur présence en quelques minutes. Cependant, les routeurs n'effectuent pas de publications à une fréquence suffisante pour se fier à une absence de publication permettant de détecter une défaillance de routeur. Un algorithme de détection séparé qui détermine l'inaccessibilité de voisin fournit la détection de défaillance.

Préfixes de publication de routeurs

Les publications de routeur contiennent une liste de préfixes de sous-réseau utilisés pour déterminer si un hôte se trouve sur le même lien que le routeur. La liste de préfixes est également utilisée pour la configuration d'adresses autonomes. Les indicateurs associés aux préfixes spécifient les utilisations spécifiques d'un préfixe particulier. Les hôtes utilisent les préfixes sur liaison publiés afin de constituer et de maintenir une liste utilisée pour décider lorsque la destination d'un paquet se trouve sur la liaison ou au-delà d'un routeur. Une destination peut se trouver sur une liaison même si celle-ci n'est couverte par aucun préfixe sur liaison publié. Dans de tels cas, un routeur peut envoyer une redirection. La redirection informe l'expéditeur que la destination est un voisin.

Les publications de routeur et les indicateurs par préfixe permettent aux routeurs d'informer des hôtes de la méthode qu'ils doivent utiliser pour effectuer une configuration automatique d'adresse sans état.

Messages de publication de routeurs

Les messages de publication de routeur contiennent également des paramètres Internet, comme la limite de saut, que les hôtes devraient utiliser dans des paquets entrants. Les messages de publication de routeur peuvent également (facultativement) contenir des paramètres de liens, comme le lien MTU. Cette fonctionnalité permet l'administration centralisée des paramètres critiques. Les paramètres peuvent être définis sur des routeurs et propagés automatiquement à tous les hôtes qui y sont connectés.

Les noeuds effectuent la résolution d'adresses par l'envoi de sollicitation de voisin à un groupe multicast, demandant au noeud cible de retourner son adresse de couche liaison. Les messages de sollicitation de voisin multicast sont envoyés à l'adresse de noeud multicast demandée de l'adresse cible. La cible retourne son adresse de couche liaison dans un message de publication d'un voisin unicast. Une paire de paquets de demande-réponse unique est suffisante pour permettre à l'initiateur et à la cible de résoudre les adresses de couche liaison de l'un et de l'autre. L'initiateur inclut son adresse de couche liaison dans la sollicitation de voisin.

Extensions IPv6 de services d'assignation de noms Oracle Solaris

Cette section décrit les modifications en matière d'attribution de noms introduites par l'implémentation d'IPv6. Vous pouvez stocker les adresses IPv6 dans les fichiers de services de noms, NIS, LDAP, DNS ou tout autre fichier Oracle Solaris de votre choix. Vous pouvez également utiliser le protocole NIS à travers les transports RPC IPv6 pour la récupération de données NIS.

Extensions DNS pour IPv6

Un enregistrement de ressources spécifique IPv6, l'enregistrement de ressource AAAA, a été spécifié dans le document RFC 1886 *DNS Extensions to Support IP Version 6*. Cet enregistrement AAAA mappe un nom d'hôte en une adresse IPv6 de 128 bits. L'enregistrement PTR est toujours utilisé avec IPv6 pour mapper les adresses IP en noms d'hôtes. Les 32 quartets de d'adresse 128 bits sont inversés pour une adresse IPv6. Chaque quartet est converti dans sa valeur hexadécimale ASCII correspondante. Ensuite, `ip6.int` est joint.

Modifications apportées aux commandes de services de noms

Pour la prise en charge d'IPv6, vous pouvez rechercher les adresses IPv6 avec les commandes de service de noms existantes. Par exemple, la commande `ypmatch` fonctionne avec les nouvelles cartes NIS. La commande `nslookup` peut rechercher les nouveaux enregistrements AAAA dans DNS.

Prise en charge IPv6 de NFS et RPC

Les logiciels NFS et RPC (Remote Procedure Call, appel de procédure distant) prennent IPv6 en charge de façon totalement fluide. Les commandes existantes relatives aux services NFS restent inchangées. Il est également possible d'exécuter la plupart des applications RPC sur IPv6 sans aucune modification. Certaines applications RPC avancées de reconnaissance d'acheminement peuvent nécessiter une mise à jour.

Prise en charge d'IPv6 sur ATM

Oracle Solaris prend en charge le protocole IPv6 sur des ATM, des PVC (Permanent Virtual Circuits, circuits virtuels permanents) et des SVC (Switched Virtual Circuits, circuits virtuels à commutation) statiques.

Index

D

E

F

H

I

L

M

N

P

Q

R

S

T

U

V

W