31

La signalisation

Le transport de l'information de commande, ou signalisation, est un aspect capital de l'infrastructure des réseaux. On peut même dire que l'avenir des réseaux réside dans la capacité de les piloter et d'automatiser leur configuration. L'objectif est de signaler une information, par exemple celle de déclencher les processus de mise en place de l'infrastructure, afin qu'une application puisse se dérouler.

La signalisation est depuis longtemps étudiée par les organismes de normalisation, et plus particulièrement par l'UIT-T. Elle a beaucoup évolué au cours des dix dernières années et va continuer à devoir s'adapter aux bouleversements du monde IP.

Ce chapitre ne vise pas à dresser un panorama exhaustif des protocoles de signalisation. Nous en abordons un grand nombre dans l'ensemble de l'ouvrage, en les introduisant directement dans les chapitres traitant des architectures ou des protocoles de service concernés. Le présent chapitre se contente d'introduire quelques notions de base et de les accompagner d'un certain nombre d'exemples éloquents, qui, même s'ils sont introduits par ailleurs dans l'ouvrage, représentent des cas importants.

Nous commençons par donner les raisons d'être de la signalisation et présenter les principales fonctions dont elle s'occupe. Nous introduisons ensuite les exemples de signalisation les plus importants, avec les protocoles RSVP, COPS et SIP, ainsi que les protocoles associés au concept de middle box. Nous terminons avec le CCITT n° 7, qui a longtemps été la norme de signalisation la plus répandue, du fait de son adoption ancienne par les réseaux de type circuit, notamment le réseau téléphonique.

Caractéristiques de la signalisation

La signalisation désigne les moyens à mettre en œuvre pour transmettre une information telle que l'ouverture ou la fermeture d'un chemin. Elle existe dans tous les réseaux, y compris ceux qui, comme IP, souhaitent la réduire au minimum afin de préserver la simplicité du système. La signalisation doit donc être capable de fonctionner selon toutes les techniques du monde des réseaux, en particulier des réseaux IP.

La signalisation demande généralement un mode routé. En effet, il faut indiquer à qui la signalisation est adressée et, pour cela, exhiber l'adresse complète du récepteur dans le paquet de signalisation. Tous les réseaux commutés ont donc besoin d'un réseau routé pour mettre en œuvre la signalisation.

La signalisation est capable de prendre en charge des services à différents niveaux de l'architecture. Par exemple, elle doit être capable d'effectuer une négociation de SLA, de demander l'authentification d'un utilisateur, de collecter des informations sur les ressources disponibles, etc. Le protocole de signalisation doit être extensible de façon à permettre l'arrivée de nouveaux services de façon simple. Le protocole de signalisation doit de plus être modulaire et flexible, afin de répondre aux besoins de chaque application particulière. La modularité facilite l'ajout de nouveaux modules lors des phases de développement.

Fonctionnement de la signalisation

Un protocole de signalisation comporte deux modes de fonctionnement : *dans la bande* (inband) et *hors bande* (outband). Dans le premier cas, les messages de signalisation sont transportés dans le chemin de données, tandis que dans le second ils sont indépendants du chemin suivi par les données.

Une autre caractéristique de la signalisation est la possibilité de couplage (path-coupled) ou au contraire de découplage du chemin (path-decoupled). Dans le premier cas, la signalisation suit les données dans la bande ou hors bande en empruntant la même succession de nœuds. Par exemple, le protocole RSVP est path-coupled et le protocole COPS path-decoupled.

La signalisation doit être capable de fonctionner à la fois dans les modes interdomaines et intradomaines. La signalisation doit également pouvoir fonctionner en modes bout-en-bout, bordure à bordure et end-to-edge (signalisation entre un end-host et un edge-node).

Dans l'environnement Internet hétérogène actuel, il existe un grand nombre de protocoles de signalisation, plus ou moins adaptés aux différentes applications. Cette grande diversité a poussé l'IETF à créer le groupe de travail NSIS (Next Step in Networking) afin de proposer une nouvelle norme unique destinée à rassembler toutes les précédentes.

D'une façon générale, un protocole de signalisation doit pouvoir coopérer avec d'autres protocoles. Pour ce faire, il doit être capable de transporter les messages d'autres protocoles de signalisation. Il est aussi possible de définir des interfaces permettant de transformer un message concernant un protocole en un message concernant un autre protocole.

La signalisation doit supporter la gestion de toutes les ressources du réseau. Elle prend en charge le transport des informations permettant d'exprimer les demandes des applications

en terme de réservation et d'allocation de ressources. Pour ce faire, la signalisation interagit avec des entités spécifiques, telles que les serveurs de gestion de ressource, comme les bandwidth brokers, ou serveurs de bande passante. Enfin, la signalisation doit supporter la négociation de SLA entre un utilisateur et un fournisseur ou entre fournisseurs et la configuration des entités dans le réseau selon le nouveau SLA.

La signalisation peut supporter le monitoring des services et les états des entités dans le réseau et prendre en charge la facturation des services.

Afin de valider les demandes de service d'un utilisateur, la signalisation est aussi utilisée pour réaliser une authentification. Elle permet en ce cas de transporter les informations nécessaires à cette interaction. Ce transport doit être suffisamment générique pour autoriser les mécanismes existants et à venir.

La sécurité

La signalisation joue un rôle très important dans la sécurisation d'un réseau. En premier lieu, elle doit être elle-même sécurisée. Les primitives doivent pouvoir s'authentifier pour garantir qu'elles ne proviennent pas d'attaquants. La signalisation doit aussi implémenter des moyens de protection des messages de signalisation contre leur modification malicieuse. Elle doit en outre permettre de détecter qu'un ancien message est réutilisé, afin d'éviter le rejeu, et de cacher les informations de topologie du réseau. Elle peut enfin supporter des mécanismes de confidentialité des informations, tels que le chiffrement.

Les protocoles de signalisation peuvent coopérer avec les protocoles d'authentification et les agréments de clés (Key Agreement) pour négocier les associations de sécurité.

La signalisation doit aussi posséder des moyens pour négocier des mécanismes de sécurité selon les besoins des applications et des utilisateurs.

La mobilité

La signalisation joue un rôle important dans la gestion de la mobilité. Elle intervient dans les diverses actions à effectuer quand le mobile change de cellule, quand il effectue un roaming, lorsqu'il négocie son SLA ou bien pour la mise en place d'une application.

Quand un handover a lieu, la signalisation doit être capable de rétablir la connexion et de reconstituer rapidement et efficacement les états installés dans la nouvelle station de base. Le processus de rétablissement peut être local ou de bout en bout. Si le réseau mobile est surchargé, la signalisation des handovers doit avoir une priorité plus élevée que celle d'une signalisation démarrant une nouvelle connexion.

La charge du réseau

Dans une situation normale, le trafic de signalisation occupe une part peu importante du trafic du réseau. Cependant, dans certaines situations de congestion, de panne ou de problème, le trafic de signalisation peut augmenter de façon significative et créer une sévère congestion de la signalisation dans le réseau. Par exemple, une erreur de routage d'un paquet de signalisation peut entraîner une explosion en chaîne des messages de

notification. Un protocole de signalisation doit être capable de maintenir la stabilité de la signalisation.

La signalisation doit être robuste, efficace et consommer le moins possible de ressource dans le réseau. Ce dernier doit être capable de fonctionner, même dans le cas d'une forte congestion.

Le réseau doit être capable d'assigner une priorité aux messages de signalisation. Cela permet de réduire les délais de transit des signalisations correspondant à des applications fortement prioritaires. Il faut également faire attention aux attaques par déni de service, qui peuvent saturer le réseau par des messages de signalisation de haute priorité.

Le protocole de signalisation doit permettre de regrouper des messages de signalisation. Cela peut concerner, par exemple, le regroupement des messages de rafraîchissement, comme RSVP, afin d'éviter de rafraîchir individuellement les états de réservation, ou soft-state.

La signalisation doit pouvoir passer l'échelle (scalabilité), c'est-à-dire s'appliquer à un petit réseau aussi bien qu'à un immense réseau de plusieurs millions de nœuds. Elle doit aussi être capable de prendre en charge et de modifier les différents mécanismes de sécurité en fonction des besoins en performance des applications.

Le protocole RSVP

Le protocole RSVP (Resource reSerVation Protocol) est conçu pour supporter la réservation unicast et multicast de ressources de bout en bout. Repris dans l'environnement IntServ (Integrated Services), RSVP est un protocole de signalisation modulaire, qui offre la possibilité de définir de nouveaux objets pour des extensions à venir. Il supporte le soft-state, fonctionne en mode path-coupled et fait la réservation en mode receiver-initiator, c'est-à-dire depuis le récepteur vers l'émetteur.

Des extensions de RSVP ont été proposées pour permettre la livraison fiable de messages de signalisation (message ACK/NACK) et le regroupement des messages de rafraîchissement dans un seul message SREFRESH. Une autre proposition d'extension permet l'agrégation des demandes de réservation de ressources pour augmenter la performance et la scalabilité du protocole.

Caractéristiques de RSVP

Une caractéristique importante de RSVP est de permettre la réservation de ressources dans un mode multicast. Il est difficile d'utiliser le mode émetteur vers récepteur pour prendre en charge la réservation de ressources en multicast puisqu'on ne connaît pas les caractéristiques du récepteur au démarrage de la demande de réservation. C'est la raison pour laquelle RSVP fonctionne en mode récepteur vers émetteur.

Dans ce mode, c'est le récepteur des données qui déclenche et maintient la réservation des ressources dans le réseau. En d'autres termes, l'émetteur envoie une demande au récepteur, lequel déclenche la primitive de réservation de ressources et l'envoie vers l'émetteur. Le paquet portant cette primitive effectue la réservation dans les nœuds traversés en allant du récepteur vers l'émetteur. La raison de ce choix tient à la nécessité d'effectuer une réservation adéquate puisque la primitive de réservation connaît les caractéristiques de l'émetteur et du récepteur, et pas seulement celles de l'émetteur. Par exemple, si l'émetteur demande un débit de 1 Mbit/s mais que le récepteur ne puisse

recevoir qu'un débit de 128 Kbit/s, il est inutile de réserver une bande passante de 1 Mbit/s, 128 Kbit/s suffisant.

Le maintien d'une réservation repose sur la notion de soft-state au niveau des nœuds intermédiaires et d'extrémité. Les soft-states sont des états périodiquement réactivés, permettant de mettre à jour dynamiquement le chemin à réserver en fonction des arrivées et des départs de participants et des changements de route. L'état de réservation peut donc être périssable, grâce à l'usage d'un temporisateur.

Les autres caractéristiques importantes de RSVP sont les suivantes :

- RSVP transporte et maintient des paramètres de contrôle de trafic (QoS) et de contrôle de politique (Policy Control) qui lui sont opaques. En fait, RSVP véhicule des structures d'objets définissables en dehors de lui.
- RSVP est pris en charge aussi bien par IPv4 qu'IPv6. Les composants du protocole sont définis pour cela pour IPv4 comme pour IPv6. Par exemple, les objets FILTER_SPEC sont définis différemment en IPv4 et en IPv6.
- RSVP est unidirectionnel. Il n'établit donc de réservation pour les flux de données que dans un seul sens. La réservation de ressources pour des transferts bidirectionnels requiert deux sessions RSVP indépendantes.
- RSVP n'est pas un protocole de routage. La signalisation RSVP utilise des protocoles de routage qui déterminent le chemin vers la destination. Précisons que RSVP fournit un mode opératoire transparent aux routeurs qui ne le supportent pas. Les routeurs non RSVP-aware routent les paquets IP qui transportent les messages RSVP comme des paquets normaux.

Fonctionnement de RSVP

Le protocole de signalisation RSVP met en place une connexion logique appelée session. Une session RSVP est définie par les trois éléments suivants :

- adresse IP destination (unicast ou multicast) ;
- identifiant du protocole sur IP ;
- port destination (optionnel pour IPv4), TCP ou UDP.

Le modèle RSVP se fonde sur l'échange de deux messages fondamentaux, les messages Path et Resv *(voir figure 31.1) :*

- Le message Path (à l'initiative de l'émetteur) permet à l'émetteur du flux de données de spécifier les caractéristiques du trafic qu'il va générer.
- Le message Resv (à l'initiative du récepteur) permet à un récepteur ayant préalablement reçu un message Path de spécifier la QoS requise et de déclencher la réservation sur le chemin. Ce message suit le même chemin que celui du message Path mais dans le sens inverse.

Les principaux messages qui ont été définis dans RSVP sont les suivants :

- Message d'erreur :
  - PathErr, envoyé à l'émetteur qui a créé l'erreur.
  - ResvErr, envoyé vers le récepteur pour notifier une erreur en fusionnant les demandes de réservation.

- Message de confirmation de réservation :
  - ResvConf, utilisé pour indiquer le succès de la réservation si le récepteur ne reçoit pas le message ResvErr.
- Message de suppression :
  - PathTear, envoyé vers le récepteur pour supprimer l'état Path qui a été établi par le message Path.
  - ResvTear, envoyé vers l'émetteur pour supprimer l'état Resv qui a été établi par le message Resv.

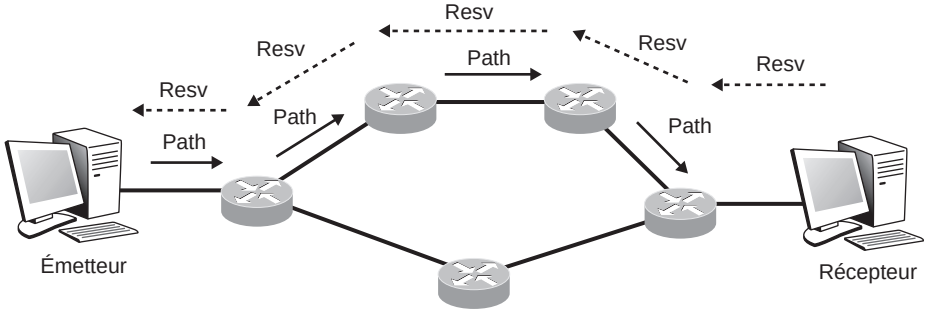


Figure 31.1
Fonctionnement du protocole RSVP

Format des messages de RSVP

Un message RSVP est constitué d'un en-tête commun et d'un nombre variable d'objets en fonction du type du message :

<RSVP message>::=<En-tête commun><ensemble des objets RSVP>

Comme illustré à la figure 31.2, l'en-tête du message RSVP est structuré de la façon suivante :

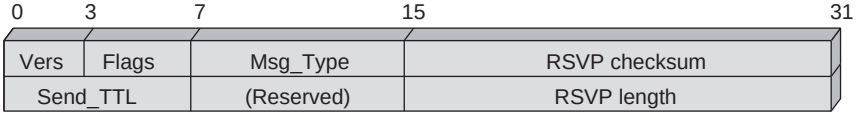


Figure 31.2
En-tête du message RSVP

- Vers (4 bits) : version du protocole.
- Flags (4 bits) : drapeau.
- Msg_Type :
  - Path
  - Resv
  - PathErr

  - ResvErr
  - PathTear
  - ResvTear
  - ResvConf
- RSVP Checksum (16 bits).
- Send_TTL : contient au départ la valeur du champ TTL du datagramme IP qui transporte le message RSVP. Grâce à ce champ, il est possible de détecter la traversée de routeurs non-RSVP. En effet, ceux-ci décrémentent le champ TTL IP et pas celui de RSVP, alors que les routeurs RSVP modifient les deux valeurs.
- RSVP Length (16 bits) : indique la longueur du message RSVP en octets.

La structure d'un objet RSVP *(voir figure 31.3)* suit celle de TLV (Type/Length/Value) :

- Longueur (16 bits) : indique la longueur de l'objet RSVP en octets.
- Class-Num : identifie la classe de l'objet.

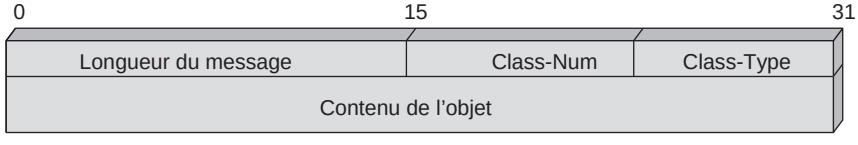


Figure 31.3
Structure d'un objet RSVP

Les classes qui sont définies sont récapitulées au tableau 31.1.

Numéro de la classe d'objet	Description
0 NULL	Ignoré par le récepteur
1 SESSION	Contient l'adresse IP destination, le protocole ID sur IP et le port de destination (obligatoire dans tous les messages).
3 RSVP_HOP	Transporte l'adresse IP du nœud RSVP qui a émis le message.
4 INTEGRITY	Données d'authentification
5 TIME_VALUES	Fréquence de rafraîchissement utilisée par le créateur du message
6 ERROR_SPEC	Spécifie une erreur dans un message PathErr, ResvErr ou ResvConf.
7 SCOPE	Transporte une liste des émetteurs de données vers lesquels les informations dans le message doivent être transmises.
8 STYLE	Style de réservation
9 FLOW_SPEC	Définit la QoS demandée.
10 FILTER_SPEC	Définit un sous-ensemble de paquets d'une session qui doit recevoir la QoS demandée (spécifié dans FLOW_SPEC) dans un message RESV.
11 SENDER_TEMPLATE	Contient l'adresse IP de l'émetteur et autre information pour identifier l'émetteur.
12 SENDER_TSPEC	Définit la caractéristique de trafic de l'émetteur de données.
13 ADSPEC	Transporte des données de OPWA.
14 POLICY_DATA	Transporte l'information de politique.
15 RESV_CONFIRM	Transporte l'adresse IP du récepteur qui a demandé la confirmation.

TABLEAU 31.1 • Classes d'objets RSVP

COPS (Common Open Policy Service)

COPS est un protocole d'échange de politiques. Il a été introduit au chapitre précédent en même temps que l'architecture globale du contrôle par politique. Cependant, nous ne sommes pas entrés dans le détail du protocole COPS en tant que signalisation, et c'est ce que nous allons faire ici.

Le protocole COPS est issu de travaux démarrés en 1996 dans le contexte de la réservation de ressources. COPS a été étendu en 1999 dans un contexte plus large dans le groupe de travail RAP (Resource Allocation Protocol) de l'IETF et normalisé par la RFC 2748 de janvier 2000.

Dans sa version actuelle, COPS a pour objectif l'échange d'informations de politiques réseau entre un PDP (Policy Decision Point) et un PEP (Policy Enforcement Point). Le PDP et le PEP font partie de l'architecture de gestion de réseau à base de politique définie par les groupes PFWG (Policy Framework Working Group) et DMTF (Distributed Management Task Force) de l'IETF. Le rôle du PDP est de prendre des décisions sur les politiques réseau, tandis que celui du PEP est d'appliquer les décisions que lui a communiquées le PEP.

Deux modes de signalisation sont actuellement standardisés au sein de l'IETF :

- COPS-Outsourcing, issu des premiers travaux, intègre COPS dans un réseau où existe un protocole de signalisation tel que RSVP. Les événements déclencheurs d'échanges COPS sont les messages de signalisation arrivant au PEP. Le PDP est alors sollicité pour prendre la décision sur la politique à appliquer. La première RFC qui se réfère à ce mode est COPS-RSVP (COPS usage for Resource ReserVation Protocol), que nous examinons un peu plus loin.
- COPS-Configuration, aussi appelé COPS-Provisioning, permet l'intégration de COPS dans un réseau où les politiques sont transmises au préalable par le PDP au PEP et engendrent la configuration du PEP. La RFC qui se réfère à ce mode est COPS-PR (COPS usage for Policy Provisioning), que nous étudions également plus loin.

Le mode d'échange de COPS est de type client-serveur, avec une relation maître à esclave. Le PDP est le maître et le PEP l'esclave. Il n'y a pas de classification de message de type requête/réponse. COPS ne peut fonctionner qu'au-dessus de TCP. Une connexion persistante TCP est établie entre le PEP et le PDP. La fiabilité est donc assurée par TCP.

COPS définit un fonctionnement général, qui peut être étendu pour générer des fonctionnements plus spécifiques, propres à la politique ou au mode de gestion de la politique. Ces fonctionnements spécifiques sont définis hors de COPS dans des extensions que nous verrons ultérieurement.

Dans le protocole COPS, le mode de communication est unique et direct entre le PEP et le PDP, et il n'y a pas d'entités intermédiaires. Dans COPS et ses deux extensions COPS-RSVP et COPS-PR, le PEP est une entité logique qui représente un équipement actif du réseau. Le PDP est une entité logique qui représente un équipement de management du réseau. En retour des requêtes du client, il envoie des décisions. C'est le fonctionnement naturel dans le mode outsourcing. Le PEP et le PDP conservent l'état des requêtes/décisions échangées, selon un fonctionnement « stateful ». L'architecture complète dans laquelle s'insère COPS est illustrée à la figure 31.4.

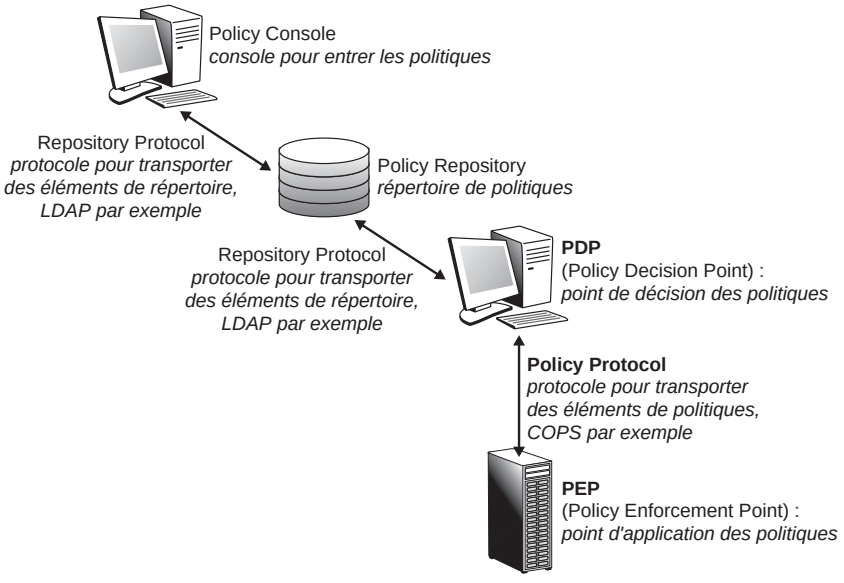


Figure 31.4
Architecture de gestion de réseau par politique

Les messages COPS

Les messages COPS ont tous la même structure générale : un en-tête commun donnant les informations sur le type du message et un corps transportant les objets spécifiques. Cette structure est représentée à la figure 31.5.

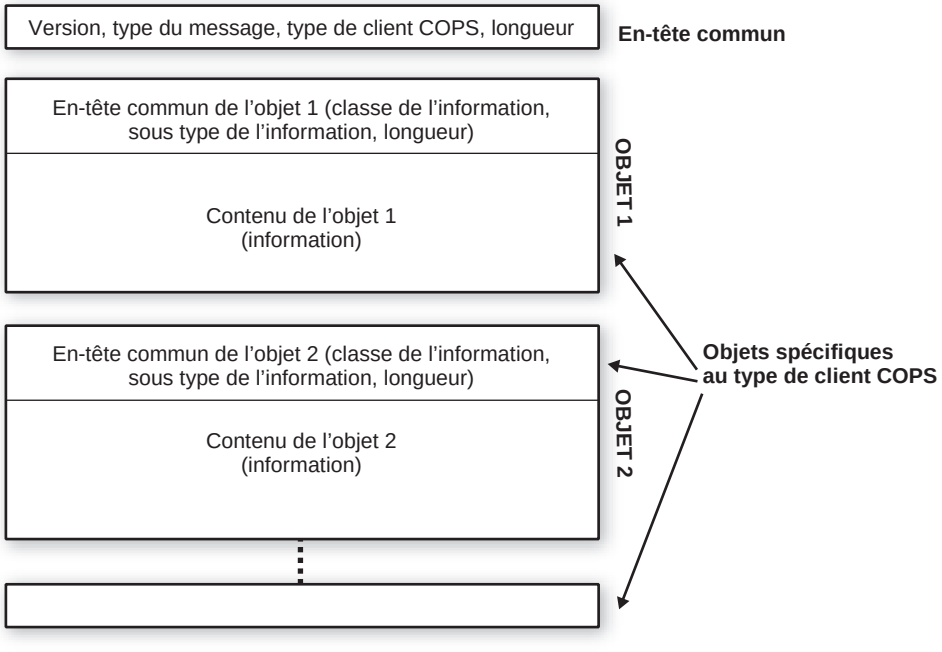


Figure 31.5
Format général des messages COPS

COPS comporte les 16 classes d'objets, ou plutôt classes d'information de contenu d'objets, récapitulées au tableau 31.2. La structure des objets est soit précisée dans la RFC, soit étendue et définie dans les extensions du protocole.

C-Num	Classe de l'objet	C-Type	Objet
1	Handle (Handle)	1	Client – Handle
2	Context (Context)	1	Context
3	In Interface (IN-Int)	1 2	IPv4 address + Interface IPv6 address + Interface
4	Out Interface (OUT-Int)	1 2	IPv4 address + Interface IPv6 address + Interface
5	Reason Code (Reason)	1	Reason Code
6	Decision (Decision)	1 2 3 4 5	Decision Flags Stateless Data Replacement Data Client Specific Decision Data Named Decision Data
7	LPDP Decision (LPDPDecision)	1 2 3 4 5	Decision Flags Stateless Data Replacement Data Client Specific Decision Data Named Decision Data
8	Error (Error)	1	Error
9	Client Specific Info (ClientSI)	1 2	Signaled ClientSI Named ClientSI
10	Keep-Alive Timer (KATimer)	1	Keep-alive Timer value
11	PEP Identification (PEPID)	1	PEP Identification
12	Report Type (Report-Type)	1	Report Type
13	PDP Redirect Address (PDPRedirAddr)	1 2	IPv4 + TCP port IPv6 + TCP port
14	Last PDP Address (LastPDPAddr)	1 2	IPv4 Address IPv6 Address
15	Accounting Timer	1	Accounting timer value
16	Message Integrity	1	HMAC digest

TABLEAU 31.2 • Classes d'objets de COPS

Dans COPS-RSVP, la classe d'objets Context object est utilisée pour transporter le type de message RSVP et la classe Client specific information pour transporter les objets RSVP.

Dans COPS-PR, de nouveaux objets sont encapsulés dans les sous-types Named Client-Specific Information object et Named Decision Data Object. Les objets spécifiques sont issus d'une base d'information de politiques, ou PIB (Policy Information Base), relative à chaque type de client COPS. L'ensemble de ces PIB réunies compose la PIB générale. Cette PIB suit la même convention que la MIB SNMP. Les formats d'encodage actuellement définis pour le stockage des informations et leur transport sont ASN.1 (Abstract Syntax Notation 1) et BER (Basic Encoding Rule).

Les messages sont définis d'une manière générale dans la RFC COPS, leur utilisation spécifique étant précisée dans les RFC d'extension. Le tableau 31.3 récapitule les 10 messages COPS regroupés par sens de circulation.

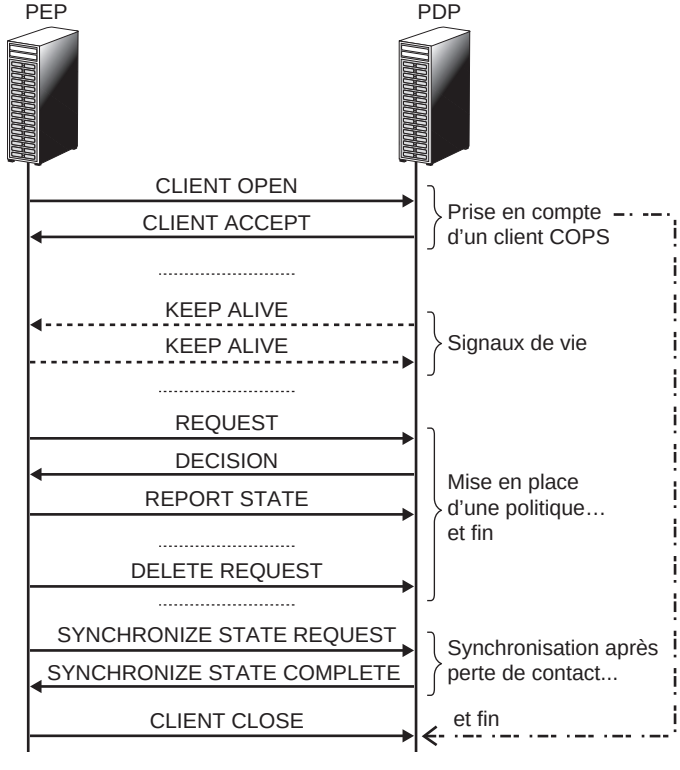
Message PEP ⟶ PDP	Message PDP ⟶ PEP	Message PDP ⟷ PEP
REQUEST (demande de politique)	DECISION (envoi de politique)	CLIENT CLOSE (fin de prise en compte de client COPS)
REPORT STATE (résultat d'installation de politique)	SYNCHRONIZE STATE REQUEST (demande de synchronisation)	KEEP ALIVE (Signal d'existence)
DELETE REQUEST STATE (fin d'application de politique)	CLIENT ACCEPT (prise en compte de client COPS)	
CLIENT OPEN (demande de prise en compte de client COPS)		
SYNCHRONIZE STATE COMPLETE (fin de synchronisation)		

TABLEAU 31.3 • Messages COPS

Scénarios de contrôle de politique

Les scénarios de contrôle de politique COPS dépendent du mode de contrôle de politique. Cependant, on peut illustrer les échanges de messages COPS par la décomposition en étapes de la figure 31.6.

Figure 31.6
Échanges COPS



Les extensions de COPS

Le protocole COPS peut être étendu en introduisant de nouveaux types de clients. Nous ne décrivons ici que les deux extensions les plus répandues, COPS-RSVP et COPS-PR.

COPS-RSVP (COPS usage for RSVP)

La RFC 2749 de janvier 2000 précise les directives d'usage pour le support de COPS dans un environnement RSVP. C'est dans cette première optique que COPS a été développé par le groupe de travail RAP (Resource Allocation Protocol) afin de fournir un mécanisme de contrôle d'admission à partir de requêtes sur les ressources réseau. Cela a donné lieu à la création d'une extension pour RSVP permettant de prendre en charge le contrôle d'admission par politique, qui spécifie notamment l'objet POLICY-DATA transporté par les messages RSVP et utilisé pour le contrôle par politique par les PEP et le PDP. La RFC 2750 décrit cette extension.

Comme expliqué précédemment, le mode de fonctionnement de COPS-RSVP est l'outsourcing, dans lequel les événements déclencheurs sont les messages RSVP.

Les détails de l'architecture COPS-RSVP sont peu développés dans les RFC 2749 et 2750. On peut cependant déduire les informations suivantes :

- Un PEP est un client RSVP.
- Un client RSVP n'est pas forcément un PEP.
- Un client RSVP sur un routeur extrémité du domaine est forcément un PEP.

L'architecture COPS-RSVP est illustrée à la figure 31.7.

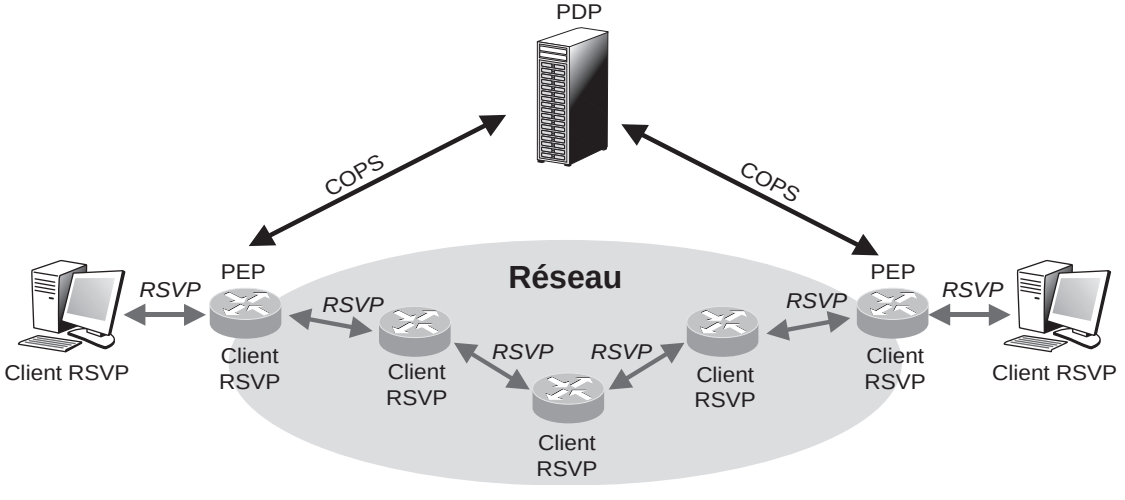


Figure 31.7
Architecture COPS-RSVP

Les types de messages RSVP générant des requêtes COPS sont Path, Resv, PathErr et ResvErr. Tous les objets reçus dans ces messages sont encapsulés dans les requêtes COPS. Trois contextes objet sont définis pour la génération de requêtes COPS du PEP vers le PDP en fonction des événements/actions RSVP :

- Incoming-Message request : lors de l'arrivée d'un message RSVP, une sollicitation pour l'accepter ou le rejeter est envoyée du PEP au PDP.
- Resource-Allocation request : lors de l'arrivée d'un message RSVP Resv, une sollicitation pour injecter (commit) les ressources dans le flux RSVP est envoyée du PEP au PDP.
- Outgoing-Message request : lorsque le PEP doit faire suivre un message RSVP sortant, il sollicite le PDP, qui accepte ou refuse cette sortie et fournit l'objet POLICY-DATA qui sera encapsulé dans le message RSVP.

Lors de l'établissement d'une réservation RSVP, plusieurs sollicitations sont déclenchées. Le nombre de messages COPS engendrés dépend de nombreux paramètres. Dans un fonctionnement normal d'une réservation pour une session point-à-point (unicast), le nombre de ces messages peut être minimisé par regroupement de plusieurs contextes objet dans une même requête COPS. C'est le cas avec le contexte objet combiné In & Allocation & Out pour traiter l'arrivée d'un message Resv et l'affectation des ressources associées et pour le faire suivre.

COPS-PR (COPS usage for Policy Provisioning)

La RFC 3084 de mars 2001 précise les directives d'usage pour la prise en charge de COPS dans un environnement à base d'approvisionnement de politiques. Cette prise en charge est indépendante du type de la politique devant être approvisionnée (QoS, sécurité, etc.) et développe les mécanismes et conventions utilisés pour l'échange d'information en mode provisioning entre des PEP et des PDP.

Le mode provisioning se différencie du mode outsourcing par le fait qu'il n'y a plus de corrélation entre un événement se produisant dans un PEP et la décision relative du PDP. Le PDP peut envoyer directement des informations de provisionnement au PEP suite à une sollicitation externe ou à un ensemble d'événements s'étant produits dans le PEP ou encore à toute autre combinaison.

Le provisionnement des ressources dans un réseau est souvent fondé sur les SLA et s'opère aux frontières du réseau. Cela confère un aspect statique au modèle COPS-PR, où les échanges entre PEP et PDP sont espacés par des temps longs comparativement au modèle dynamique du mode outsourcing.

Les événements externes susceptibles de déclencher des décisions directes du PDP vers le PEP peuvent être les suivants :

- Utilisateur sollicitant des services réseau *via* une interface Web de l'application centrale de gestion.
- Serveur H.323 sollicitant des ressources pour le compte d'un utilisateur voulant établir une visioconférence.

Ces sollicitations externes arrivent directement au PDP. Cependant, la RFC ne décrit pas le mode de communication entre le serveur H.323 et le PDP. D'un autre côté, le PEP peut lui aussi solliciter directement le PDP. Dès l'ouverture de la connexion globale entre PEP et PDP, c'est-à-dire juste après l'échange Client Open ↔ Client Accept, le PEP sollicite le PDP pour obtenir l'ensemble des politiques à approvisionner en son sein. Il peut ensuite le faire à chaque modification de sa configuration, telle que le retrait d'une carte d'interface.

Pour représenter les informations de politique approvisionnée à échanger entre les PEP et le PDP, une base d'information de politiques PIB (Policy Information Base) est introduite. La PIB est représentée par un arbre, dans lequel les branches identifient les classes de politiques, ou PRC (Provisioning Class), et les feuilles les instances de ces PRC, ou PRI (Provisioning Instance), qui sont échangées entre PEP et PDP. Ces PIB sont stockées à la fois dans les PEP et le PDP.

Si l'on regarde l'application du modèle provisioning à la gestion des politiques de QoS, on voit tout de suite son adéquation avec le modèle DiffServ. Dans le modèle DiffServ, les équipements du réseau sont configurés au préalable pour appliquer des mécanismes de qualité de service à l'ensemble des flux du réseau. L'architecture DiffServ définit deux catégories d'équipements, les routeurs extrémité (edge routers) et les routeurs internes. Les premiers doivent classifier les flux et leur affecter un DSCP (DiffServ Code Points), qui sera utilisé dans la suite par tous les seconds pour traiter les paquets avec le comportement associé à ce DSCP particulier. La mise en place de politiques de contrôle de QoS dans cette architecture se fait par l'implantation d'un PEP COPS-PR au niveau de chaque routeur extrémité.

L'architecture de l'environnement COPS-PR pour un réseau DiffServ est décrite à la figure 31.8.

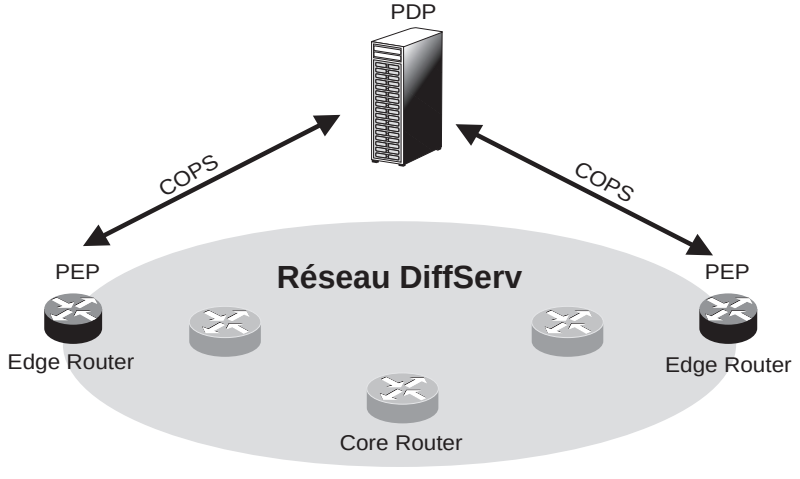


Figure 31.8
Architecture COPS-PR pour un réseau DiffServ

SIP (Session Initiation Protocol)

SIP est un protocole de signalisation dont la première version est apparue sous forme de draft à l'IETF en 1997 dans le groupe de travail MMUSIC. Repris dans le groupe SIP en 1999, il a été normalisé en mars 1999 sous la RFC 2543.

SIP a pour objectif l'établissement, la modification et la terminaison de sessions multimédias entre deux terminaux. Il ne définit pas le corps de ses messages. Le corps, qui contient la description du média (vidéo, audio, codeur, etc.), est décrit par le protocole SDP (Session Description Protocol). Outre la description du flux, SIP peut transporter, les médias utilisés dans une session, ou *session media*, en particulier des informations de QoS ou de sécurité. Le média de session est dissocié des échanges SIP.

Héritant du modèle HTTP, le mode d'échange est de type client-serveur, avec une relation d'égal à égal entre les deux. Les messages SIP sont des requêtes, aussi appelées méthodes, qui engendrent des messages en retour, les réponses. SIP peut fonctionner audessus de plusieurs protocoles de transport. UDP est pour l'instant le plus utilisé, mais l'utilisation de TCP est aussi définie, ainsi que le transport par d'autres protocoles, tel SCTP (Stream Control Transmission Protocol). Avec UDP, SIP assure la fiabilité à partir d'accusés de réception positifs et de temporisateurs.

SIP est conçu pour être évolutif. Seules les fonctions de bases sont obligatoires, des extensions pouvant être supportées ou non par les différentes entités qui s'échangent leur capacité.

Deux modes de communication sont possibles, le réseau décidant du mode :

- Mode direct : les deux entités SIP représentant les terminaux communiquent directement.
- Mode indirect : des entités intermédiaires faisant partie du réseau relaient les messages échangés.

Les entités SIP

SIP comporte plusieurs catégories d'entités, dont les plus classiques sont les entités utilisatrices et les entités réseau. Ces entités s'échangent des messages.

Les entités utilisatrices

Les entités utilisatrices sont appelées agents utilisateur, ou UA (User Agent). Les UA ouvrent, modifient et terminent les sessions pour le compte de l'utilisateur. Concrètement, dans l'application multimédia du terminal utilisateur, il s'agit de la partie du programme qui permet de recevoir et d'établir les sessions. Elle regroupe deux composantes, l'une qui agit en tant que client (UAC) et initie les sessions à la demande de l'utilisateur, et l'autre qui agit en tant que serveur (UAS) et qui est responsable de la réception de toutes les sessions à destination de l'utilisateur. Les UA conservent des informations sur l'état de la session et sont dits pour cela stateful.

Les entités réseau

SIP définit trois entités logiques de type serveur faisant partie du réseau et agissant pour étendre son fonctionnement.

- **Proxy server.** Le proxy server a une fonction de relais. Il accepte les requêtes ou les réponses SIP en provenance d'un UA ou d'un autre serveur proxy et les fait suivre. Ce serveur peut conserver des états de l'avancement des sessions pour lesquelles il intervient. Il est dit dans ce cas stateful. Dans le cas inverse, il est dit stateless.
- **Registrar server.** Les UA de son domaine viennent s'enregistrer auprès de lui. Il renseigne ensuite le service de localisation, qui, lui, n'est pas défini par SIP. Le protocole généralement utilisé pour accéder à ce service est LDAP.
- **Redirect server.** Il répond à des requêtes en donnant les localisations possibles de l'UA recherché.

SIP fait aussi référence à une quatrième entité, qui n'entre pas dans les dialogues SIP mais offre le service de localisation sur lequel les entités logiques SIP viennent s'appuyer.

Un exemple d'échange entre entités SIP est illustré à la figure 31.9.

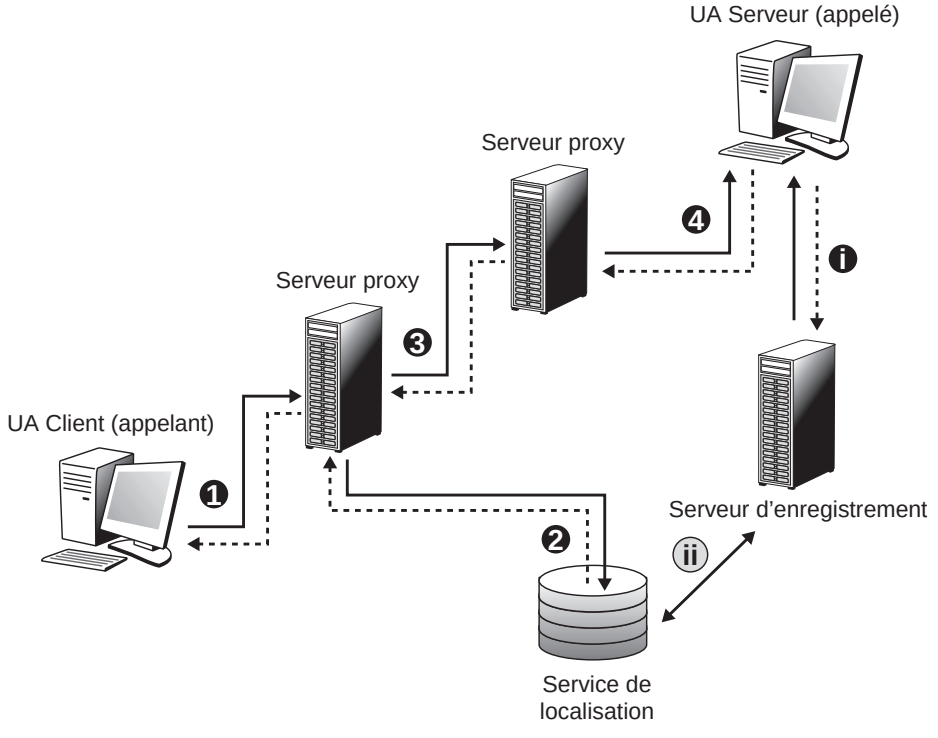
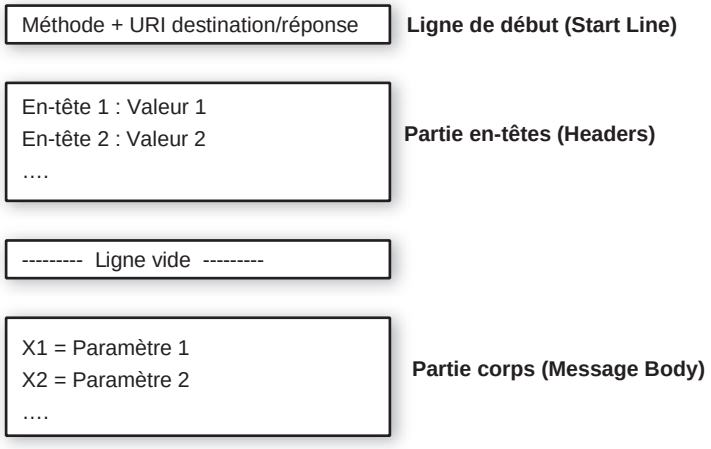


Figure 31.9
Exemple d'échange entre entités SIP lors d'un établissement de session

Les messages SIP

Il existe deux catégories de messages SIP, les requêtes et les réponses. Les messages sont codés en langage textuel. Le message comprend trois parties, comme illustré à la figure 31.10.

Figure 31.10
Structure des messages SIP



Les messages initiés par les UAC (User Agent Client) à destination d'un ou de plusieurs UAS (User Agent Server) sont appelés requêtes ou méthodes, par analogie avec HTTP.

Les méthodes définies dans la version actuelle de SIP sont les suivantes :

- INVITE : ouverture ou modification de session ;
- ACK : acquittement d'une réponse positive à un INVITE ;
- CANCEL : annulation d'une requête en cours ;
- OPTIONS : demande de capacité ;
- BYE : terminaison d'une session ;
- REGISTER : enregistrement d'un UA ;
- INFO : information relative à la session en cours.

Les réponses sont envoyées par un USA ou un proxy en réponse à une requête provenant d'un UAC. Elles sont regroupées en six classes, les xx représentant des codes détaillés :

- 1xx : information sur l'avancée de la requête ;
- 2xx : succès de la requête ;
- 3xx : redirection de la requête ;
- 4xx : erreur client ;
- 5xx : erreur serveur ;
- 6xx : erreur globale.

Les en-têtes SIP sont regroupés en quatre catégories :

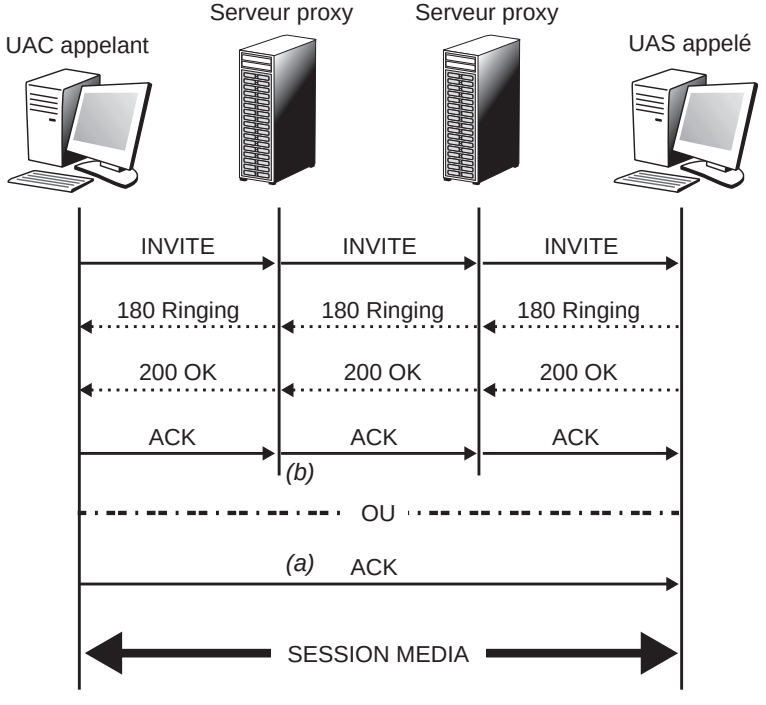
- général : présent dans les requêtes et les réponses ;
- requête : présent uniquement dans les requêtes ;
- réponse : présent uniquement dans les réponses ;
- entité : relatif au corps du message.

Le nom de l'en-tête est suivi de la valeur de l'en-tête.

Les scénarios de session

Différents scénarios d'établissement de médias de session sont possibles. La figure 31.11 illustre un scénario qui met en évidence une caractéristique majeure de SIP : lors de sa réponse, l'UAS peut déclencher un échange direct des messages SIP à venir (a) ; de leur côté, les serveurs proxy peuvent obliger les messages SIP qui vont suivre à passer par eux (b).

Figure 31.11
Exemple d'établissement de session SIP



SDP (Session Description Protocol)

SDP est une syntaxe de description de médias normalisée dans la RFC 2327 d'avril 1998. Cependant, comme nous le verrons un peu plus loin, SDP n'offre pas de possibilité complète de négociation de capacité de média. Issue du groupe MMUSIC de l'IETF précédant SIP, sa première application a été la description de sessions multicast combinée avec le protocole d'annonce de session multimédia SAP (Session Announcement Protocol), qui officie très largement dans le réseau multicast expérimental MBONE (Multicast Backbone).

SDP est ensuite devenu le protocole naturel de description des médias de session intégrés à l'établissement de session réalisé avec le protocole SIP.

La syntaxe de description de SDP suit un codage textuel. Une description de session est en fait une suite ordonnée de lignes, appelées fields, représentées par une lettre. Le nombre de fields est volontairement limité de façon à faciliter le décodage (*via* un parser). Le seul moyen d'étendre SDP est de définir de nouveaux attributs.

Le format général d'un field est x = paramètre1 paramètre2… paramètre*N*.

Les principales informations caractérisant le média de session sont les suivantes :

- adresse IP (ou nom d'hôte) : adresse de réception du flux média ;
- numéro de port pour la réception des flux ;
- type de média : audio, vidéo, tableau blanc, etc. ;
- schéma d'encodage (PCM A-LAW, MPEG-2, etc.

Le format du champ caractérisant le média de session est m = media port transport format-list, avec :

- media : audio, vidéo, application, données, contrôle ;
- port : numéro de port de réception du média ;
- transport : RTP/AVP (Audio Video Profiles) ou UDP ;
- format-list : liste d'informations complémentaires sur le média, ou Media Payload Type.

Plusieurs types de payload peuvent être listés. S'ils sont listés, il s'agit d'un choix proposé. Pour ouvrir *n* canaux audio il faut présenter *n* champs media.

Échange des caractéristiques du média de session

Le besoin de qualité de service nécessaire à la session est déduit des caractéristiques du média de session. Ces caractéristiques définies par SDP et présents dans le corps des messages SIP sont les suivantes :

- extrémités d'échange de flux : adresses IP source et destination et numéros de ports source et destination ;
- type de média échangé : audio, vidéo, etc. ;
- mode de codage utilisé : liste des codecs utilisés pour chaque flux, par exemple G.711 ou G.723.1.

Lors de l'initialisation d'une session SIP, seules les caractéristiques du média de l'émetteur sont connues. Elles sont déterminées par l'UAC en fonction de la demande de l'application utilisateur, application de téléphonie sur IP, par exemple. L'UAC qui initialise la demande d'établissement de session ne connaît pas *a priori* les informations concernant la partie caractéristique du média de l'UAS qu'il sollicite. C'est seulement après que la sollicitation est arrivée à l'UAS que la faisabilité de l'établissement de session est connue et, dans l'affirmative, que les caractéristiques du média de session sont identifiées.

Il ne faut pas perdre de vue qu'un média de session classique résulte de l'échange de deux flux de média :

- flux de média de l'initiateur vers le sollicité ;
- flux de média du sollicité vers l'initiateur.

Dans cette version de SIP (RFC 2543), l'échange des caractéristiques du média de session donne très peu de latitude pour la négociation. En effet, l'UAS qui reçoit les caractéristiques du média de session souhaité dispose du type de média et de la liste des codecs relatifs proposés par l'UAC de départ. Il est précisé que la station qui initialise la communication indique dans sa liste des codecs ceux avec lesquels il veut recevoir le flux média et le fait qu'il aimerait émettre ce flux. Dans sa réponse, l'UAS précise sa liste de codecs, qui peut être ou non un sous-ensemble de la liste de l'émetteur. Il indique lui aussi la liste des codecs avec lesquels il veut recevoir le flux média. La RFC SDP précise que lorsqu'une liste de codecs est donnée, elle spécifie les codecs qui peuvent être utilisés pendant la session avec un ordre de préférence, le premier étant considéré comme le codec par défaut de la session.

Dans le meilleur des cas, l'UAS sollicité accepte la liste des codeurs qui lui sont proposés (flux sollicité → initiateur). Inversement, l'UAC initiateur accepte celle qui lui est proposée (flux initiateur → sollicité). Pour qu'une session puisse avoir lieu, il faut au minimum que chacun accepte le premier codeur spécifié dans la liste qui lui est proposée. Autrement, la session ne peut avoir lieu.

L'échange des caractéristiques de média est illustré à la figure 31.12.

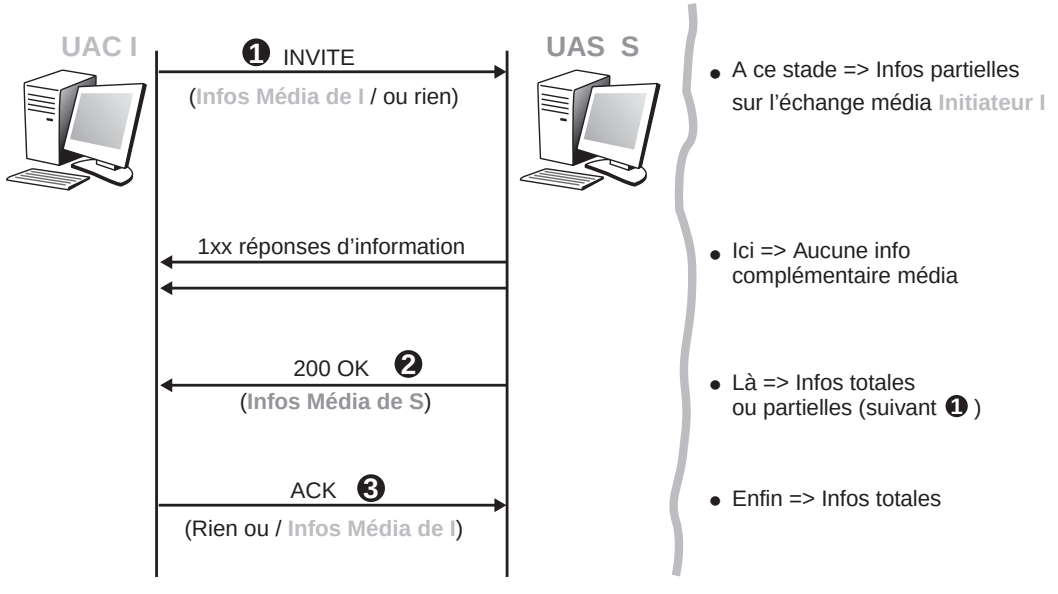


Figure 31.12

Échange des caractéristiques de média dans le protocole SIP actuel

SIP nouvelle version

La RFC 3261 décrit une nouvelle version de SIP. Cette RFC a pour objet de réviser la RFC 2543, devenue de ce fait obsolète, en corrigeant ses erreurs et en apportant des détails sur les scénarios d'utilisation. Ces modifications sont listées au sein même de la RFC.

La liste ci-dessous présente les modifications les plus significatives :

- Recentrage de la RFC sur la partie SIP :
  - Dans les scénarios, le corps SDP n'est plus présenté. Les échanges SDP sont définis directement dans la nouvelle RFC 3264 (An Offer/Answer Model with Session Description Protocol).
  - Les procédures de localisation des serveurs SIP par DNS sont définies dans la nouvelle RFC 3263 (SIP : Locating SIP Servers).
- Obligation pour les UA (User Agent) de supporter TCP en plus d'UDP.
- Support de TLS (Transport Layer Security) et SCTP (Stream Control Transmission Protocol) pour le niveau transport.
- Traitement des routes et de l'enregistrement de route SIP retravaillé et largement détaillé.
- Modification de la sécurité, qui devient plus approfondie :
  - PGP (Pretty Good Privacy) supprimé et remplacé par S/MIME.
  - Basic Authentication supprimé et même interdit.
  - Fonctions de sécurité supplémentaires apportées par TLS.
  - Mécanisme d'authentification des agents utilisateur par serveur remplacé par l'authentification mutuelle de la RFC 2617.

D'autres extensions liées à SIP accompagnent le nouveau protocole. Les sections qui suivent présentent les extensions de la RFC qui concernent l'amélioration de la négociation des caractéristiques de média au sein de la phase d'établissement d'appel.

Le modèle offre/réponse

L'objectif du modèle offre/réponse est de permettre à deux entités d'arriver à une vision commune de la session qu'ils vont avoir ensemble. Cette session est décrite à l'aide de SDP et est utilisée par SIP. Il a été défini pour préciser et compléter l'échange des caractéristiques de média lors de l'établissement de la session. Dans ce modèle, un des participants offre à l'autre la description du média de session qu'il désire, et l'autre répond à cette offre. Il est précisé que lorsque plusieurs codecs sont listés, l'offreur indique qu'il est capable d'utiliser chacun d'eux au cours de la session. Celui qui répond peut les utiliser au cours de la session sans renégociation préalable.

Celui qui offre envoie l'ensemble des flux de média et des codecs qu'il veut utiliser, ainsi que l'adresse IP et le numéro de port sur lequel il est prêt à recevoir. Celui qui répond reprend chaque flux de média proposé en indiquant s'il l'accepte ou non, avec la liste des codecs qui seront utilisés ainsi que l'adresse IP et le numéro de port sur lequel il est prêt à recevoir.

Dans le cas général où la session consiste en deux flux, c'est à dire un flux aller et un flux retour (sendrecv), la liste des codecs correspond aux types d'encodage des flux que l'on veut avoir en réception et que l'on préférerait utiliser pour envoyer. Le premier codec listé est le préféré. Il est recommandé que celui qui est sollicité l'utilise. D'une manière plus générale, le sollicité s'attache dans la réponse à suivre l'ordre de la liste des codecs présentée dans l'offre, car cela permet d'utiliser le même codec dans les deux sens. Ce modèle permet aussi la modification de sessions établies.

À n'importe quel moment, l'un ou l'autre des participants peut lancer une nouvelle offre pour modifier les caractéristiques de la session en cours. Il est possible de modifier les paramètres d'un flux média, de détruire un flux existant ou d'ajouter un nouveau flux. Tous les champs de média de la première offre sont repris dans l'ordre, et les nouveaux médias sont ajoutés à la fin.

La nouvelle méthode SIP

L'objectif de cette méthode est de permettre à un UA de modifier les paramètres d'une session, tels que les flux de média et les codecs associés, sans impact sur l'état du dialogue SIP classique. La méthode peut être déclenchée après l'établissement de la session ou pendant cette phase. Cela se révèle très utile pour modifier les informations caractérisant la session avant que celle-ci soit établie, c'est-à-dire au cours de l'établissement. L'exemple présenté est celui du message d'accueil EARLY MEDIA envoyé au correspondant à la place du retour d'appel pendant la phase de recherche du correspondant. Ce flux média est envoyé jusqu'à ce que l'invitation soit acceptée. À ce moment-là, il y a modification du média de session afin de permettre de retirer le message d'accueil et d'ouvrir les flux de communication. Ce processus est réalisé par la méthode UPDATE.

Une extension apportée par la RFC 3262 de juillet 2002 introduit un mécanisme permettant de fiabiliser l'envoi de réponses provisoires. Ces dernières permettent de transporter des informations importantes, telles les réponses aux offres dans le modèle offre/réponse. Ce mécanisme repose sur l'introduction d'un nouvel en-tête, qui est transporté dans la réponse provisoire et qui spécifie la demande pour que cette requête soit fiabilisée.

Échange des caractéristiques du média de session

Les trois extensions introduites à la section précédente enrichissent l'échange des caractéristiques du média de session. La figure 31.13 présente un exemple mettant en évidence tout le potentiel apporté par ces extensions. Ces nouvelles possibilités ont un impact important. En effet, les informations caractérisant le média ne se trouvent plus uniquement dans les messages principaux de l'établissement de session INVITE, 200 OK et ACK mais dans les messages provisoires, tel 180 RINGING, et les messages intermédiaires de modification UPDATE. Cela rend toutefois l'intégration plus complexe.

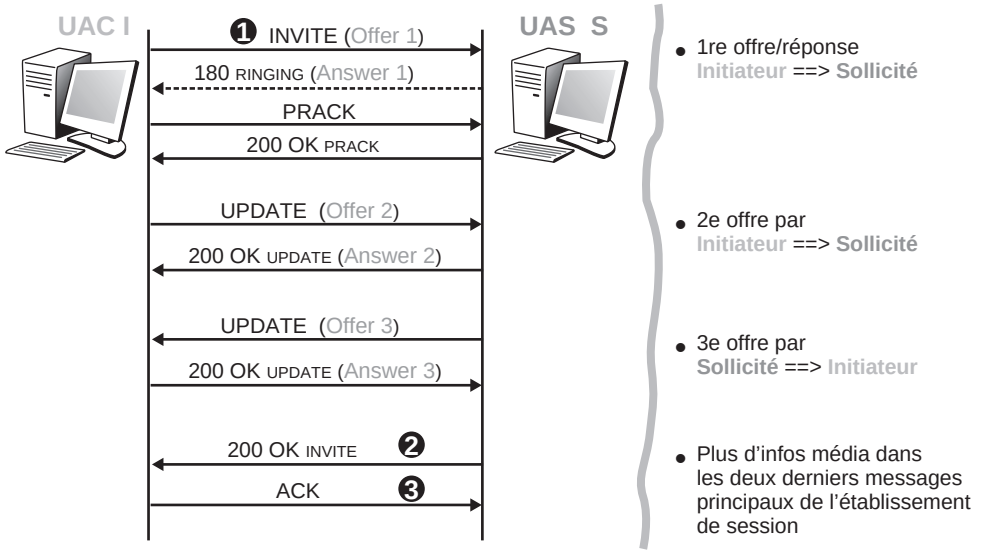


Figure 31.13

Échange des caractéristiques média dans SIP nouvelle version

Le dispositif middle box et l'architecture MIDCOM

Une *middle box* est un dispositif de réseau intermédiaire permettant d'implémenter des services divers tels qu'un filtrage de paquets, un VPN, une détection d'intrusion, une translation d'adresse NAT ou un pare-feu. Une middle box est donc une appliance située entre deux équipements de réseau, d'où son nom de middle box. En fait, middle box et appliance sont deux noms qui désignent pratiquement les mêmes équipements. Dans cette section, nous préférons utiliser middle box, qui est le terme utilisé par l'IETF, appliance étant surtout le terme des industriels commercialisant des boîtiers intermédiaires.

Ces boîtiers nécessitent de l'intelligence pour faciliter la traversée du flux applicatif. Cela rend leur maintenance difficile et dégrade leur performance. D'où l'idée de déplacer l'intelligence de ces boîtiers dans des agents MIDCOM communiquant avec le boîtier à l'aide du protocole MIDCOM. Les agents MIDCOM exécutent des fonctions ALG (Application Level Gateway), qui examinent le flux applicatif et aident la middle box à remplir ses fonctions.

Le protocole MIDCOM s'exécute en trois phases : établissement de la session, session et rupture de la session. La communication entre le boîtier et l'agent se déroule de façon transparente pour l'utilisateur final. Les agents peuvent résider dans les hôtes finals, des serveurs proxy des applications, des passerelles applicatives ou dans la middle box.

Seuls des agents MIDCOM autorisés peuvent influencer le fonctionnement du boîtier. L'autorisation d'un agent requiert une inscription, pendant laquelle les agents suppléent leur profil à la middle box ou au MIDCOM PDP que la middle box consulte. Ce profil détermine les opérations autorisées. L'inscription est souvent une opération manuelle.

L'architecture d'une middle box est illustrée à la figure 31.14.

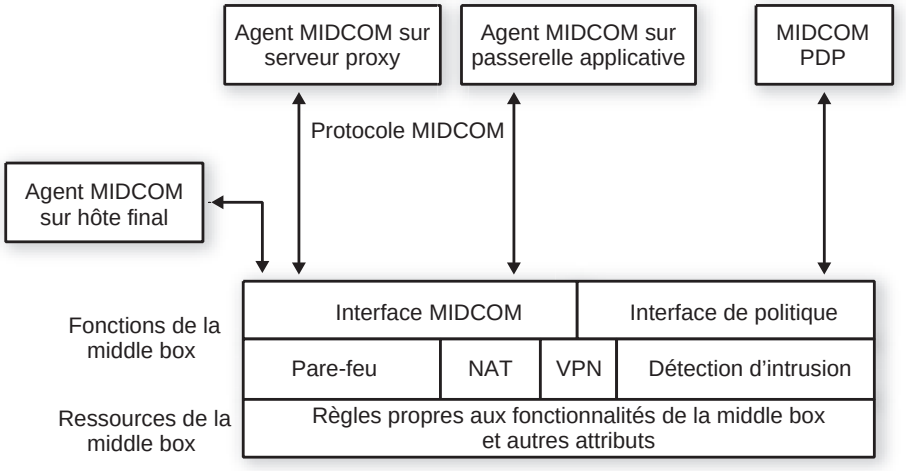


Figure 31.14
Architecture d'une middle box

Un agent MIDCOM assistant un pare-feu, par exemple, peut lui demander d'autoriser l'accès au trafic d'une application.

Les sections suivantes illustrent le fonctionnement temporel d'une *middle box* par des exemples d'applications temps réel, notamment avec l'implémentation de fonctionnalités de pare-feu et de NAT dans une middle box pour la téléphonie SIP.

Implémentation d'une fonction de pare-feu dans une middle box

Prenons le cas d'un téléphone SIP externe au réseau qui souhaite communiquer avec un téléphone interne. L'agent MIDCOM résidant dans le SIP proxy demande à la middle box pare-feu du réseau de débloquer les ports nécessaires aux flux RTP (Real-time Transport Protocol) et RTCP (Real-Time Control Protocol) dans les deux sens. La figure 31.15 en illustre le fonctionnement.

Implémentation de NAT dans une middle box

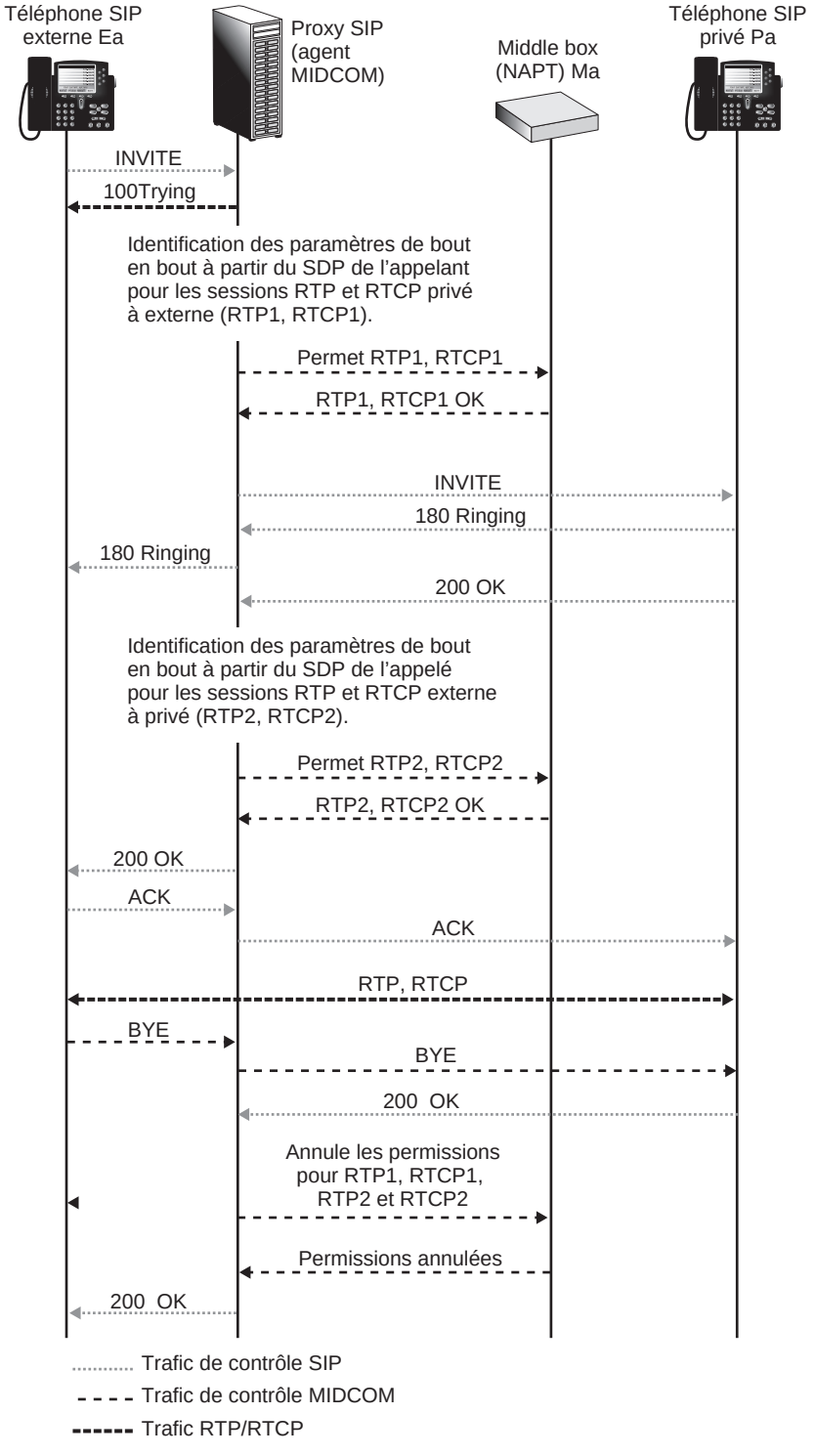
La middle box est configurée pour rediriger les appels SIP entrants vers l'adresse privée du téléphone SIP. La commande INVITE est destinée à l'adresse NAT externe. Les appels SIP sont des sessions TCP/UDP sur le port 5060.

Nous utilisons la notation suivante :

- Ma : adresse externe de la middle box ;
- Pa : adresse interne du téléphone SIP ;
- Ea : adresse du téléphone SIP externe.

Figure 31.15

Téléphonie SIP à travers une middle box implémentant la fonction de pare-feu



Le proxy SIP demande les descripteurs de la session NAT pour les deux flux entrant et sortant. Les ports dynamiques utilisés pour le flux média sont contenus dans la partie SDP du message SIP. Après le 200 OK reçu par le proxy du téléphone privé, l'agent demande à la middle box d'allouer des descripteurs de session NAT pour le trafic entrant de sorte que les ports réservés pour RTP2 et RTCP2 soient contigus. Bien que les flux média entrants et sortants soient indépendants, ils sont liés à la même session SIP. Quand le message BYE est envoyé, toutes les ressources sont libérées. La figure 31.16 illustre l'interaction entre un proxy SIP et une middle box implémentant la fonctionnalité NAT.

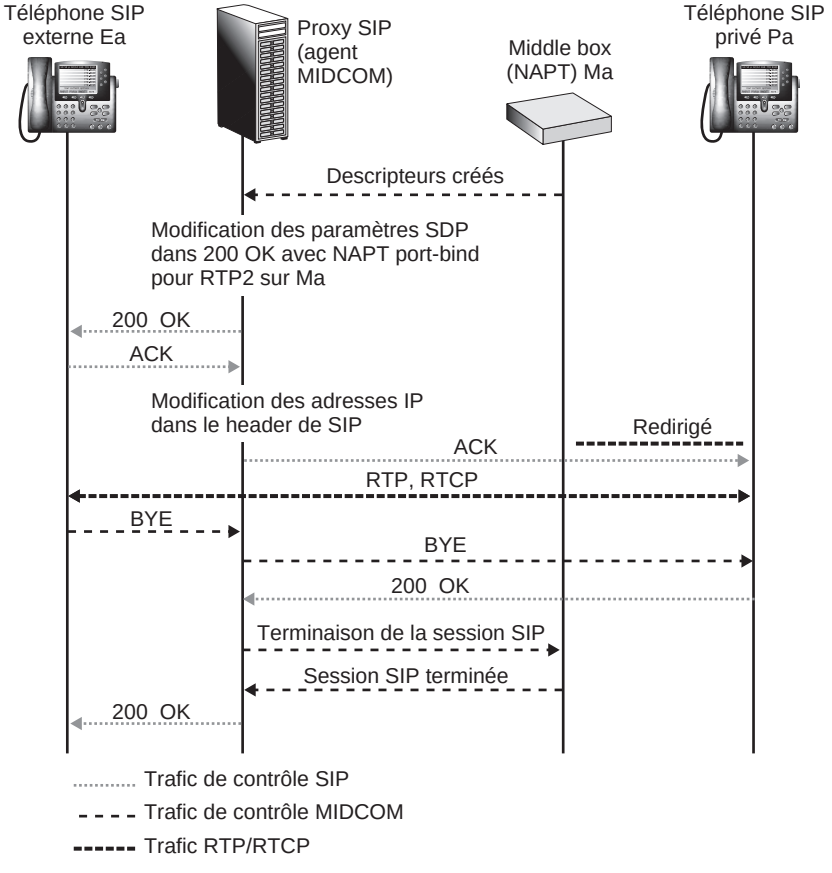


Figure 31.16

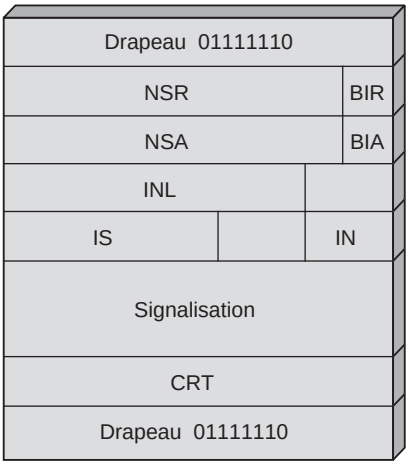
Téléphonie SIP à travers une middle box implémentant la fonction NAT

La signalisation CCITT n° 7

Le protocole CCITT n° 7 a été mis au point par l'UIT-T dans le cadre de la transmission de la signalisation sur les réseaux publics. Le protocole LAP-D, que nous verrons par la suite, véhicule la signalisation sur la terminaison d'abonnés. Au moment de leur passage dans le réseau public proprement dit, les informations de supervision sont prises en charge par un réseau spécifique de type datagramme, le réseau sémaphore, qui suit la recommandation CCITT n° 7 décrivant les couches du protocole. Cette architecture est compatible avec le modèle de référence.

Le protocole de niveau 2 est de type HDLC. Il a été légèrement modifié pour prendre en compte les contraintes temps réel de la signalisation. Tous les algorithmes sont semblables à ceux de HDLC, excepté celui des reprises sur erreur. La détection se fait toujours par la zone de contrôle. La structure de la trame CCITT n° 7 est illustrée à la figure 31.17.

Figure 31.17
La trame CCITT n° 7



Trois types de trames sont disponibles dans la procédure :

- Les PDU de signalisation sans champ d'information.
- Les PDU avec un champ d'information, qui servent aux contrôles de la procédure elle-même. C'est par ce type de trame que le contrôle de flux de la liaison est effectué. Lorsque la procédure n'a pas de signalisation utilisateur à transmettre, elle émet en continu des trames de ce type, en acquittant la dernière trame bien reçue. On a ainsi une duplication des acquittements, ce qui est très utile en cas de perte d'acquittement. Un autre avantage de ces trames est qu'elles détectent presque instantanément une rupture de la liaison.
- Les PDU avec un champ d'information, qui transportent la signalisation de bout en bout. Pour cette catégorie, on trouve un numéro de trame sur 7 bits situé dans le deuxième octet de la trame, juste derrière le drapeau, ainsi qu'un deuxième numéro de séquence dans le troisième octet de la trame. Ces deux numéros, associés aux bits BIR (indicateur de bit arrière) et BIA (indicateur de bit avant), permettent un contrôle avant et arrière de la procédure.

Les trames contiennent encore un indicateur de longueur sur 6 bits, le champ INL, un indicateur de service dans le champ IS sur 4 bits et un indicateur national IN sur 2 bits.

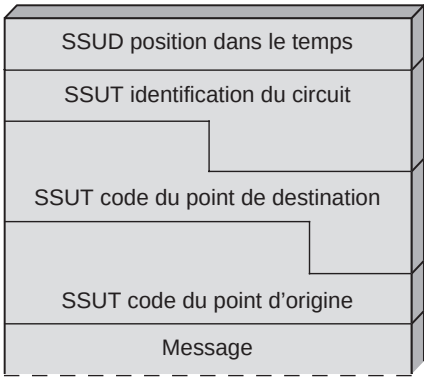
Deux techniques de reprise sur erreur sont disponibles. La première est conforme à la procédure HDLC. La seconde permet une récupération plus rapide et une duplication des réémissions. Cette seconde technique est particulièrement appréciable dans les réseaux où le temps de propagation est très long, comme les réseaux satellite. À chaque réémission, si le support est libre, on retransmet toutes les trames depuis la trame en erreur, et l'on recommence jusqu'à ce qu'il y ait une nouvelle signalisation à émettre. Cette politique permet de dupliquer ou tripliquer, c'est-à-dire faire trois copies, les reprises et de prévoir, le cas échéant, plusieurs trames successives erronées.

Pour compléter les caractéristiques de la procédure CCITT n° 7, indiquons que les coupleurs extrémité possèdent des compteurs d'erreur, qui comptabilisent le nombre de trames erronées par unité de temps. Si le compteur dépasse une valeur déterminée par le gestionnaire du réseau, la liaison est fermée.

Le niveau réseau de la recommandation CCITT n° 7 spécifie un réseau datagramme puisque les données à transmettre sont extrêmement courtes, de l'ordre de quelques octets. Nous allons décrire brièvement le protocole de niveau réseau.

Les avis Q.702 à Q.704 de l'UIT-T décrivent le protocole CCITT n° 7. Le niveau 3 prend surtout en charge le problème de l'adressage. Celui-ci est décrit dans les recommandations Q.711 à Q.714. En particulier, la norme 84 définit le sous-système de commande des connexions sémaphore SCCP (Signaling Connection Control Part). Le paquet de niveau 3 est illustré à la figure 31.18.

Figure 31.18
Paquet de niveau 3 du protocole CCITT n° 7



SSUT (sous-système utilisateur téléphonique)
SSUD (sous-système utilisateur informatique)

Deux sous-systèmes ont été normalisés : le sous-système correspondant aux applications téléphoniques, dans les avis Q.721 à Q.725, et celui correspondant aux applications informatiques. Ils sont appelés respectivement TUP (Telephone User Part) et DUP (Data User Part).

Le service de transport de la recommandation CCITT n° 7 assure pour le compte du niveau session un service de transport de bout en bout des TSDU. Il offre cinq classes de services très différentes de celles proposées par le modèle de référence lui-même, mais compatibles avec lui. Ces cinq classes sont les suivantes :

- classe 0 : sans connexion et sans identification de lien de signalisation ;
- classe 1 : toujours sans connexion mais avec identification de lien de signalisation ;
- classe 2 : avec connexion ;
- classe 3 : avec connexion et contrôle de flux ;
- classe 4 : avec connexion, contrôle de flux et détection et récupération d'erreur.

Cette dernière classe est assez semblable à la classe 4 du protocole UIT-T X.224.

La signalisation dans les réseaux ATM

Les réseaux ATM doivent respecter une signalisation précise pour mettre en place les circuits virtuels et les maintenir.

La signalisation du réseau ATM se sert de l'infrastructure physique du réseau large bande. Les cellules destinées à la signalisation empruntent des circuits virtuels spécifiquement destinés à leur transport. Ce sont des circuits virtuels permanents ayant des numéros spécifiques. Le protocole utilisé au niveau AAL s'appelle SSCOP (Service Specific Connection Oriented Protocol). La signalisation par elle-même est une extension de la signalisation actuelle CCITT n° 7 pour la partie opérateur interne et une extension du LAP-D pour la partie interface.

De façon plus précise, sur l'interface UNI, la signalisation vers l'utilisateur est décrite dans la recommandation Q.2931 de l'UIT-T, qui a été elle-même étendue dans l'environnement ATM pour supporter les connexions point-à-multipoint (UNI 3.1, UNI 4.0). Ces signalisations ne faisant aucune distinction entre les utilisateurs, des extensions, finalisées au cours de l'année 2004, ont été ajoutées pour tenir compte de l'application et adapter le contrôle aux besoins du client.

Les fonctionnalités ajoutées sont notamment les suivantes :

- appel simple d'un point vers un multipoint ;
- généralisation des procédures multipoint-multiconnexion ;
- ajout de modifications demandées en cours de session ;
- ajout de paramètres additionnels des catégories CBR, VBR et ABR ;
- ajout de fonctionnalités pour la VoD (Video on Demand).

Pour la partie interne du réseau, la signalisation est une extension du protocole CCITT n° 7, appelée B-ISUP et définie dans la recommandation Q.3686. Cette signalisation relaie les demandes des extrémités au travers du réseau.

En règle générale, le circuit virtuel de signalisation est ouvert, mais il peut se produire de nombreux problèmes, en particulier l'absence de circuit ouvert. Lorsque le circuit virtuel de signalisation n'est pas ouvert, il faut pouvoir en créer un nouveau afin de faire passer la signalisation. C'est une procédure de métasignalisation.

Conclusion

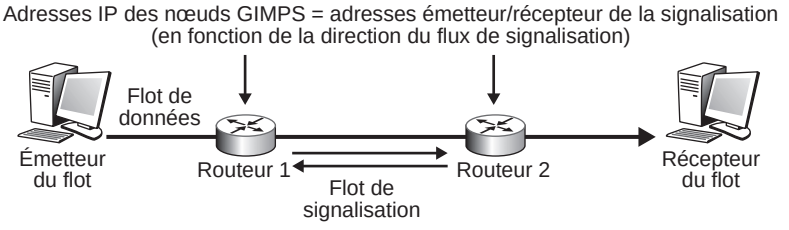
Nous avons examiné dans ce chapitre les principales caractéristiques des protocoles de signalisation. Nous avons vu quelques exemples avec les protocoles de signalisation les plus connus, comme RSVP, COPS et SIP, qui se situent à différents niveaux de l'architecture : le niveau 3 pour RSVP et COPS, qui cherchent à effectuer une allocation de ressources de niveau 3, et le niveau 7 pour SIP.

Le nombre de protocoles de signalisation du monde IP est très important du fait que la normalisation s'est effectuée en suivant les services de niveaux réseau et application et que chaque technologie a introduit ses propres protocoles de signalisation. Le groupe de travail NSIS (Next Step In Signaling) de l'IETF essaie d'introduire un protocole de signalisation unique. Plus exactement, il s'agit d'un protocole de signalisation à deux niveaux, le premier au niveau transport, qui introduit la normalisation de la structure du

paquet de signalisation transportant les différentes catégories de signalisation, et le second au niveau de la couche application pour la signalisation du service.

L'architecture du protocole GIMPS (General Internet Messaging Protocol for Signaling), qui semble devoir être le prochain protocole normalisé, est décrite à la figure 31.19.

Figure 31.19
Architecture du protocole GIMPS



Références

Très bon livre sur le protocole CCITT n° 7 et ses extensions dans les réseaux de type circuit virtuel :

R. J. BATES – *Signaling System 7*, McGraw-Hill, 2002

Un livre qui commence à dater mais dont l'information n'a que peu varié :

U. D. BLACK – *ISDN and SS7: Architectures for Digital Signaling Networks*, Prentice Hall, 1997

Un livre complet sur le CCITT n°7 :

L. DRYBURGH, J. HEWETT – *Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Applications*, Pearson Education, 2003

Cet excellent livre aborde les principaux protocoles de signalisation dans la téléphonie sur IP :

J. F. DURKIN – *Voice-Enabling the Data Network: H.323, MGCP, SIP, QoS, SLAs, and Security*, Pearson Education, 2002

Livre général sur l'ensemble des solutions de signalisation :

L. HARTE – *Introduction to SS7 and IP: Call Signaling using SIGTRAN, SCTP, MGCP, SIP, and H.323*, Althos, 2003

Un livre très complet sur le CCITT n°7 :

L. HARTE, R. DREHER, D. BOWLER – *Signaling System 7 (SS7) Basics*, Althos, 2003

Un livre beaucoup plus général que la seule signalisation consacré à l'ensemble des techniques de base dans les couches basses de l'architecture des réseaux :

L. HARTE – *Telecom Basics: Signal Processing, Signaling Control, and Call Processing*, Althos, 2003

Le protocole SIP est traité en détail dans ce livre ainsi que l'ensemble de la problématique de la téléphonie sur IP :

L. HARTE, D. BOWLER – *Introduction to SIP IP Telephony Systems: Technology Basics, Services, Economics, and Installation*, Althos, 2004

La signalisation est particulièrement importante dans les réseaux de mobiles comme le GSM, le GPRS ou l'UMTS. Ce livre est plus spécifiquement destiné au GPRS :

G. HEINE – *GPRS - Signaling and Protocol Analysis - Volume 2: The Core Network,* Artech House, 2003

Livre très détaillé sur SIP. À lire pour aller plus loin dans ce domaine :

A. B. JOHNSTON – *SIP: Understanding the Session Initiation Protocol, Second Edition,* Artech House, 2004

Un très bon livre sur la signalisation numéro 7 :

T. RUSSELL – *Signaling System # 7,* McGraw-Hill, 2002

Le protocole SIP est à l'origine de nombreux livres, dont le suivant :

H. SINNREICH, A. B. JOHNSTON – *Internet Communications Using SIP,* Wiley, 2001

Un excellent livre sur la signalisation classique du monde des télécommunications :

J. VAN BOSSE – *Signaling in Telecommunication Networks*, Wiley-Interscience, 1997

Un livre technique qui permet d'entrer dans la technique des réseaux en ce qui concerne le contrôle de flux.

M. WELZL – *Scalable Performance Signaling and Congestion Avoidance,* Kluwer Academic Publishers, 2003