



Date de publication :
10 novembre 2014

Contrôle dynamique de ressources Internet - Atouts du protocole PCP

Cet article est issu de : **Technologies de l'information | Réseaux Télécommunications**

par **Mohamed BOUCADAIR, Christian JACQUENET**

Résumé Le double contexte de pénurie d'adresses globales IPv4 et de multiplication de boîtiers intermédiaires (« middleboxes ») complique la gestion des réseaux au quotidien, mais est également de nature à dégrader sensiblement les performances associées à la fourniture d'un service.

Le protocole PCP (Port Control Protocol) constitue aujourd'hui l'une des réponses les plus attractives à ces problématiques complexes. Standardisé par l'IETF, le protocole PCP repose en effet sur une architecture client/serveur simple, tout en offrant une grande souplesse d'évolution fonctionnelle. PCP permet aujourd'hui de contrôler dynamiquement des dispositifs tels que les fonctions NAT (Network Address Translation) ou pare-feu, que ceux-ci soient déployés dans des infrastructures fixes ou mobiles. [...]

Abstract Both the global IPv4 address depletion and the subsequent multiplication of middleboxes like Network Address Translators in the Internet dramatically complicate everyday network management operations, let alone the risk of seriously degrading service performances.

The Port Control Protocol (PCP) that was recently standardized by the Internet Engineering Task Force (IETF) is seen as one of the most attractive responses to this issue: PCP indeed relies upon a simple client/server architecture that aims at dynamically controlling the behavior of the aforementioned middleboxes. PCP is also very flexible from a functional evolution perspective, so that it becomes a de facto standard for the dynamic control of Internet resources. [...]

Pour toute question :
Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Par mail :
infos.clients@teching.com
Par téléphone :
00 33 (0)1 53 35 20 20

Document téléchargé le : **17/02/2024**

Pour le compte : **7200045224 - universite de strasbourg // 130.79.14.140**

Contrôle dynamique de ressources Internet

Atouts du protocole *PCP*

par **Mohamed BOUCADAIR**

Architecte de réseaux et services IP – France Telecom Orange

et **Christian JACQUENET**

Directeur des programmes stratégiques réseaux IP – France Telecom Orange

1. Contexte d'ensemble.....	TE 7 612 - 2
2. Éléments fonctionnels <i>PCP</i>	— 4
3. Transmission des messages <i>PCP</i>	— 5
4. Opérations et options <i>PCP</i>	— 5
5. Fonctions avancées	— 6
6. Exemples d'échanges <i>PCP</i>	— 15
7. Scénarios de déploiement	— 18
8. Voix sur IP	— 20
9. Réseaux mobiles	— 21
10. Continuité de service IPv4 : cas « <i>Lightweight IPv4 over IPv6</i> »	— 22
11. Problèmes que <i>PCP</i> n'est pas censé résoudre.....	— 23
12. Perspectives.....	— 24
Pour en savoir plus	Doc. TE 7 612

Cet article décrit le protocole *PCP* (**Port Control Protocol**) et ses usages. Il est organisé de la manière suivante :

- au § 1, on décrit les motivations et les principes de base du protocole *PCP* ;
- le § 2 introduit les éléments fonctionnels impliqués dans une architecture *PCP* ;
- le § 3 décrit la mécanique protocolaire *PCP*, dont les procédures de découverte de serveur(s) *PCP* et de transmission de messages *PCP* ;
- le § 4 détaille la structure d'un message *PCP*, y compris l'en-tête commun *PCP*. Cette section détaille les deux types de message *PCP* (MAP et PEER), ainsi que l'utilisation des options *PCP* ;
- le § 5 présente les fonctions avancées offertes par *PCP* ;
- le § 6 illustre les usages du protocole *PCP* dans différents contextes ;
- le § 7 présente certains scénarios de déploiement de *PCP* ;
- le § 8 détaille comment *PCP* simplifie les déploiements VoIP (Voix sur IP) ;
- le § 9 se focalise sur le déploiement *PCP* dans les réseaux mobiles ;

- le § 10 précise comment PCP peut être activé pour offrir un service de connectivité IPv4 dans un contexte de pénurie globale d'adresses IPv4 ;
- le § 11 aborde certains des problèmes qui ne peuvent être résolus par la seule activation de PCP. Ces problèmes constituent des effets collatéraux du déploiement de solutions de partage d'adresses IPv4 à grande échelle dans le contexte de la pénurie d'adresses globales IPv4.

1. Contexte d'ensemble

1.1 Motivation

La pénurie d'adresses globales IPv4 est devenue une réalité. La grande majorité des opérateurs et fournisseurs de services IP sont confrontés à ce problème, ou le seront à courte échéance. Pourtant, ces acteurs majeurs de l'Internet doivent être capables de garantir à leurs clients qu'ils pourront toujours accéder à n'importe quel contenu disponible sur l'Internet, quand bien même ces clients ne pourront plus disposer d'une adresse globale IPv4 pour leur seul usage.

■ Répondre à des besoins techniques fondamentaux

- La réponse à cette problématique de continuité de service IPv4 passe aujourd'hui par l'**installation de mécanismes de partage d'adresses IPv4 dans le réseau**. Ces mécanismes de translation d'adresses (appelés *CGN*, *Carrier Grade NAT* – fonctions de translation d'adresses de « qualité opérateur » (cf. encadré 1) c'est-à-dire présentant des performances compatibles avec un environnement opérateur) sont destinés à optimiser l'usage des derniers blocs d'adresses IPv4 disponibles, en acceptant le principe de partager ces ressources, désormais précieuses entre plusieurs utilisateurs.

De plus en plus de fournisseurs de services de connectivité IP déploient ainsi de tels mécanismes. Cette pratique souffre cependant de plusieurs limitations telles que la difficulté à rediriger correctement, et de manière fiable, une partie du trafic en provenance de l'Internet vers une machine du réseau local de l'utilisateur.

- Une autre difficulté caractéristique de l'utilisation d'un mécanisme de partage d'adresses est liée à la **fourniture de services qui reposent sur des protocoles manipulant les adresses IP ou des numéros de port comme le protocole SIP (Session Initiation**

Protocol), car ces protocoles répliquent ces informations dans le contenu du paquet IP, ce qui complique le processus de décision d'acheminement du paquet.

■ Des défis en cascade

Les utilisateurs qui maintiennent du contenu chez eux, et qui souhaitent que ce contenu puisse être accessible depuis l'Internet, posent également des difficultés lorsque ces utilisateurs doivent partager une adresse IPv4 avec d'autres.

D'une manière générale, les difficultés caractéristiques de l'utilisation d'un mécanisme de partage d'adresses globales IPv4 ont été documentées dans [33]. Toutefois, ces complications ne sont pas spécifiques à un contexte global IPv4 : elles concernent également les déploiements IPv6, car l'incapacité de migrer l'intégralité de l'Internet vers IPv6 d'un seul coup de baguette magique impose *de facto* le déploiement de mécanismes de continuité de service IPv4 : c'est en effet le prix que doivent payer certains opérateurs afin de garantir à leurs nouveaux clients qu'ils pourront accéder à n'importe quel contenu ou service disponible sur Internet, que ce contenu ou service soit uniquement disponible en IPv4 ou pas.

Exemple

La figure 1 illustre le cas d'un client qui souhaite rediriger tout le trafic destiné au numéro de port « 1234 » utilisé par l'application embarquée dans la machine « H2 ». En l'absence d'un mécanisme de contrôle du *CGN*, cette règle de redirection ne peut pas être mise en œuvre : les paquets à destination du numéro de port interne « 1234 » peuvent en effet être rejetés par le *CGN* car celui-ci ne dispose pas d'entrée spécifique pour traiter ces paquets.

De plus, les paquets ne peuvent pas être correctement acheminés par le CPE vers le terminal « H2 » car le CPE ne dispose pas de la fonction nécessaire pour prendre une décision d'acheminement de paquet sur la base de numéro de port interne [par exemple, CPE *DS-Lite (Dual-Stack Lite)*], etc.

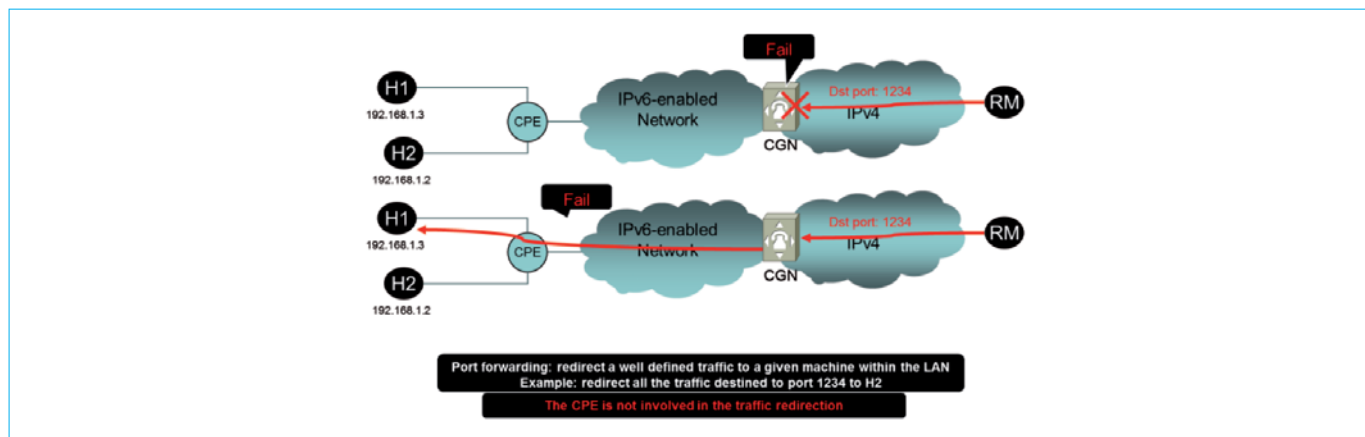


Figure 1 – Problème de redirection de trafic en présence de CGN

Encadré 1 – Fonction CGN (Carrier Grade NAT)

La fonction *CGN* dénote une fonction NAT capable d'optimiser l'usage des derniers blocs d'adresses IPv4 publics disponibles en partageant une même adresse IP entre plusieurs utilisateurs [52]. Un pool d'adresses à partager est ainsi confié à la fonction *CGN* par configuration. Ce pool est constitué d'adresses IP contiguës ou non-contiguës.

On distingue deux types de fonction *CGN* :

- La fonction *CGN* est dite « avec état » s'il n'y a aucune relation algorithmique entre le couple {adresse interne, numéro de port interne} et le couple {adresse externe, numéro de port externe} ; dans ce cas, il devient nécessaire pour le *CGN* de maintenir un état pour chaque session.

Un tel état est instancié lorsque le *CGN* traite un paquet sortant, et il est ensuite utilisé pour traiter un paquet entrant conformément aux informations décrites dans l'entrée maintenue par le *CGN* et correspondant à cet état, afin d'acheminer ce paquet vers le terminal interne *ad hoc*.

- La fonction *CGN* est dite « sans état » si les informations {adresse externe, numéro de port externe} peuvent être déduites à partir des informations {adresse interne, numéro de port interne} et *vice versa*. Aucun état n'est alors maintenu par le *CGN*.

Un *CGN* n'est pas directement contrôlé par les utilisateurs, mais placé sous la responsabilité d'une autre entité administrative (un fournisseur de service de connectivité, typiquement).

Une entrée dans la table NAT correspond à :

{Adresse IP Internet, Numéro de Port Internet, Protocole} ↔
{Adresse IP Externe, Numéro de Port Externe, Protocole}

Une entrée dans une table NAT ou une table de règles mises en œuvre par un pare-feu peuvent être de nature différente :

- « statique » : si l'entrée est permanente ;
- « implicite & dynamique » : si l'entrée est le résultat d'un paquet sortant. Cette entrée a une durée de vie limitée qui n'est pas connue de l'application ;
- « explicite & dynamique » : si l'entrée a été créée par un protocole dédié. Cette entrée a une durée de vie limitée connue de l'application.

Une entrée est par définition « bidirectionnelle » : elle sera consultée par le *CGN* ou le pare-feu pour les paquets en provenance ou à destination de l'Internet, et dont le traitement est associé à ladite entrée.

La figure 2 illustre un problème rencontré pour accéder à un serveur *FTP* (*File Transfer Protocol*) ou une webcam localisés derrière un *CGN*. On suppose que l'adresse IP et le numéro de port de ces serveurs ont été annoncés par l'utilisateur à une machine distante (par exemple, *RM* (*Remote Machine*)) en utilisant un service de rendez-vous. Les paquets émis par *RM* sont rejetés par le *CGN* car celui-ci ne dispose d'aucune entrée pour traiter ces paquets convenablement.

■ Plusieurs solutions envisageables

Pour donner aux clients les moyens de contrôler certaines des fonctions déployées en amont (telles que l'allocation d'un numéro de port spécifique pour accéder depuis Internet à du contenu maintenu par le client), plusieurs solutions peuvent être envisagées.

- L'usage d'*ALG* (*Application Level Gateway*) est une pratique courante pour le traitement de certains trafics par la fonction NAT

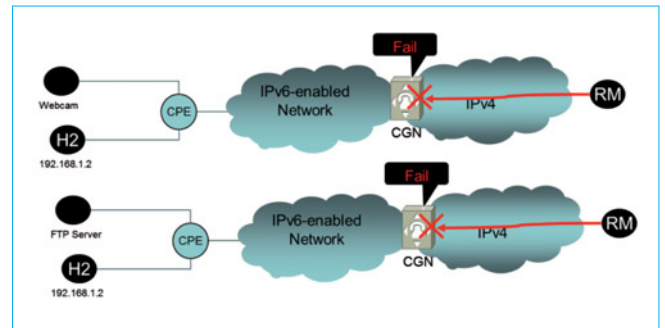


Figure 2 – Problématique de l'accès à du contenu (serveur FTP, caméra) connecté derrière un *CGN* depuis l'Internet

embarquée dans un CPE. Mais de telles structures logicielles ont des performances limitées et se révèlent souvent incompatibles avec un déploiement de fonctions NAT à grande échelle, tels que le *CGN*.

- Une autre solution repose sur le protocole *PCP* (*Port Control Protocol* [67]), qui a été spécifié par l'*IETF* (*Internet Engineering Task Force*). Ce protocole permet de contrôler dynamiquement les états instanciés par une fonction *CGN* ou une fonction pare-feu. L'usage de *PCP* permet ainsi l'acheminement correct de paquets en provenance de l'Internet et à destination d'une machine connectée derrière le *CGN* ou le pare-feu.

PCP est désormais un standard de l'*IETF*. Il a été adopté par la majorité des fabricants d'équipements supportant des fonctions NAT (y compris *CGN*) et pare-feu. Le choix d'une ingénierie *PCP* pour la résolution d'une partie des problèmes soulevés par le besoin de continuité de services IPv4 est également partagé par de nombreux opérateurs. La flexibilité du protocole est un atout majeur pour améliorer la qualité et la robustesse des services caractéristiques de la période de transition durant laquelle les mondes IPv4 et IPv6 vont devoir coexister.

PCP est un protocole prometteur qui peut être utilisé pour des besoins autres que le contrôle dynamique d'une fonction NAT ou d'un pare-feu. En effet, *PCP* peut être considéré comme un mécanisme qui permet d'informer un serveur à propos des caractéristiques d'un flux afin de préparer le réseau à traiter convenablement ce flux. Cette préparation peut consister à installer au préalable un état NAT, une règle de filtrage, une règle de classification, une politique de re-marquage, etc.

1.2 Approche

PCP adopte une architecture client/serveur qui ne nécessite pas de maintenir une session permanente entre le client *PCP* et le serveur *PCP*. Un client *PCP* exprime ses besoins (par exemple allocation d'un numéro de port spécifique) dans une requête, et le serveur *PCP* communique ses décisions au client dans un message réponse.

La prise de la décision relève de la responsabilité du serveur, et non du client, contrairement à d'autres solutions (par exemple, *Universal Plug and Play* (*UPnP*) *Internet Gateway Device* (*IGD*) version 1 [46], *UPnP IGD:2* [64] ou *NAT-PMP* [*NAT Port Mapping Protocol* (*NAT-PMP*) [28]] de prendre les décisions (figure 3).

Le serveur peut satisfaire ou ignorer les préférences du client. Il peut également faire une contre-proposition, selon l'état de la ressource demandée (par exemple, le numéro de port demandé n'est plus disponible, mais tel autre numéro de port peut être réservé).

Ainsi, le client doit s'attendre à recevoir des réponses de la part du serveur qui ne coïncident pas nécessairement avec les préférences indiquées dans la requête initiale émise par le client. Les

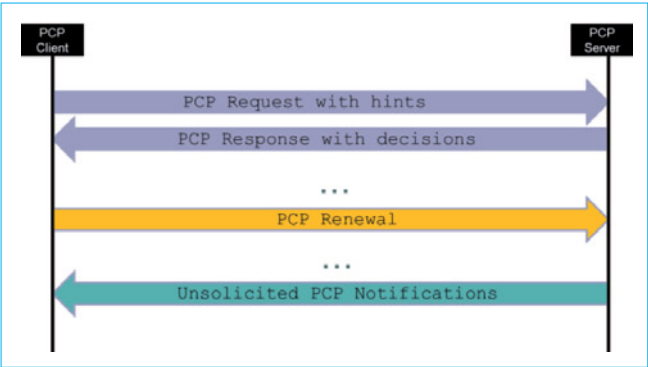


Figure 3 – PCP – Mode opérationnel

décisions du serveur dépendent des politiques définies par l’administrateur et mises en place dans le serveur PCP, notamment.

■ Configuration du serveur

- En effet, plusieurs politiques peuvent être configurées sur un serveur comme par exemple (liste non exhaustive) :
- les protocoles de transport supportés (*UDP, TCP, SCTP*, etc.) ;
 - les numéros de port à ne pas allouer aux clients ;
 - les règles qui consistent à ignorer ou considérer les numéros de port connus (par exemple, *TCP/80 (HTTP), TCP/25 (SMTP)*, etc.) [43] ;
 - le nombre de numéros de port à allouer par client ;
 - la préservation des numéros de port (c’est-à-dire allouer le même numéro pour les ports interne et externe) ;
 - l’affectation aléatoire des numéros de port ;
 - l’allocation de plages de numéros de ports plutôt que des numéros de port individuels ;
 - la préservation de la parité ;
 - la préservation de la contiguïté ;
 - la restriction de certaines adresses IP ;
 - la durée de vie maximum des entrées ;
 - la durée de vie minimum des entrées ;
 - messages *ICMP* ;
 - etc.

Toutes les entrées instanciées par PCP ont une durée de vie limitée. Un client PCP a la responsabilité de rafraîchir les entrées qui lui sont associées et qui sont gérées par un serveur en envoyant des requêtes à cet effet ; ces requêtes de rafraîchissement sont envoyées avec des intervalles qui décroissent exponentiellement. En l’absence de telles requêtes, le serveur nettoie les entrées dont la durée de vie a expiré.

■ Les entrées créées par PCP sont explicites et dynamiques

PCP supporte un mécanisme de retransmission qui s’appuie sur un délai de retransmission qui croît d’une manière exponentielle. PCP permet aussi à un serveur de notifier le (ou les) client(s) de tout changement d’état.

PCP est un protocole extensible. Des extensions peuvent être ajoutées au protocole de base en spécifiant de nouveaux types de message, de nouvelles options ou en définissant certains des bits réservés pour les rendre significatifs.

PCP a été initialement spécifié pour contrôler tout type de fonction pare-feu et de NAT, que cette fonction NAT soit d’ailleurs embarquée dans des passerelles domestiques ou déployée dans les réseaux d’opérateurs.

- Ainsi, PCP permet de contrôler plusieurs types de fonctions incluant (mais non limitées à) :
- pare-feu IPv4 ;
 - pare-feu IPv6 [69] ;

Tableau 1 – Exemples de fonctions contrôlées par le protocole PCP			
Fonctions	Adresses internes	Adresses externes	Adresses du correspondant
Pare-feu IPv4	IPv4	IPv4	IPv4
Pare-feu IPv6	IPv6	IPv6	IPv6
NAT44	IPv4	IPv4	IPv4
NAT46	IPv4	IPv6	IPv6
NAT64	IPv6	IPv4	IPv4
NPTv6	IPv6	IPv6	IPv6

- fonction NAT (*Network Address Translator* [61]) embarquée dans un CPE (*Customer Premises Equipment*) ;
- fonction CGN [52] ;
- DS-Lite (*Dual-Stack Lite*, [32]) CGN ;
- DS-Extra-Lite CGN [5] ;
- fonction NAT64 (*Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*) avec état [6] ;
- NAT64 sans état [47] ;
- fonction PRR (*Port Range Router*) d’une architecture A+P [24] ;
- fonction NPTv6 (*IPv6-to-IPv6 Network Prefix Translation*) [65], etc.

■ Fonctions avancées du PCP

- En plus du contrôle des fonctions listées précédemment, le protocole PCP supporte des fonctions avancées telles que :
- réinstaller un contexte ;
 - restaurer rapidement un état ;
 - notifier un changement d’adresse ou de numéro de port externes ;
 - détecter un NAT dans le chemin ;
 - découvrir l’adresse externe ;
 - créer une entrée pour un tiers ;
 - installer des filtres ;
 - corréler des adresses externes avec des adresses internes ;
 - associer une description à une entrée.

Le protocole PCP fonctionne indifféremment sur IPv4 ou IPv6 [37]. Il peut également être déployé dans un contexte de transition, où les mondes IPv4 et IPv6 cohabitent (tableau 1).

2. Éléments fonctionnels PCP

- Le protocole PCP repose sur une architecture client-serveur qui implique les éléments fonctionnels suivants :
- 1 – client PCP est une structure logicielle embarquée dans un terminal fixe ou mobile, un CPE (*Customer Premises Equipment*), voire un équipement du réseau tel qu’un routeur IP, et qui a la responsabilité de générer des requêtes PCP vers un serveur PCP ;
 - 2 – serveur PCP est une structure logicielle responsable du traitement des requêtes émises par un client PCP. Un tel traitement passe en particulier par une interaction avec une fonction particulière telle qu’une fonction de translation d’adresses IP (fonction NAT) ou une fonction pare-feu, de telle sorte que le serveur PCP puisse dynamiquement contrôler ces fonctions conformément à la requête émise par un client PCP. Ces fonctions (NAT, pare-feu, etc.) sont embarquées dans des équipements qui sont donc contrôlés par PCP (notion de « PCP-controlled device »).

Exemple

La requête *PCP* descriptive d'une demande d'allocation de numéro de port donne lieu à une instruction envoyée par le serveur *PCP* à la fonction *NAT* incriminée, de telle sorte que tout trafic utilisant le numéro de port demandé en provenance de l'Internet et à destination du terminal qui supporte le client *PCP* pourra traverser la fonction *NAT* sans difficultés : ce contexte est emblématique d'un environnement où des utilisateurs, raccordés à une fonction *NAT* déployée dans le réseau souhaitent que le contenu qu'ils maintiennent chez eux (par exemple un serveur *FTP*) puisse être accessible depuis l'Internet.

3. Transmission des messages *PCP*

3.1 Procédure de sélection du serveur *PCP*

Un client *PCP* peut accéder à un ou plusieurs serveurs *PCP*. Ces serveurs peuvent être déclarés explicitement à un client *PCP*, soit par configuration statique, soit par un protocole spécifique, tel que *DHCPv4* [30] ou *DHCPv6* [31].

Le document [15] spécifie une option *DHCPv4* et son équivalent pour *DHCPv6* pour déclarer un serveur *PCP* auprès d'un client *PCP*. Plusieurs listes d'adresses IP peuvent être indiquées dans un message *DHCP* en cas de présence de plusieurs serveurs ; chacun de ces serveurs peut être joignable via une liste d'adresses IP.

À noter que plusieurs instances de l'option *PCP* doivent être utilisées pour communiquer autant de listes d'adresses IP avec *DHCPv6*, alors que ces différentes listes sont incluses dans une même instance de l'option *DHCPv4*.

L'option *DHCPv6* peut inclure des adresses IPv6 et IPv4 (encodées comme des « IPv4-mapped IPv6 addresses » [37]) dans l'option *DHCPv6* mais seules des adresses IPv4 peuvent être véhiculées dans l'option *DHCPv4*.

- Si plusieurs listes d'adresses IP sont configurées pour le client *PCP*, celui-ci contacte en parallèle tous les serveurs *PCP* [20]. Si un serveur *PCP* est joignable via plusieurs adresses IP, le client utilise une seule adresse de la liste pour contacter le serveur. En cas d'incapacité pour le client *PCP* de joindre le serveur *PCP* avec une certaine adresse de la liste, le client choisira une autre adresse de la liste [20].

- Si aucun serveur n'est configuré explicitement au client, ce dernier suppose que sa passerelle par défaut est son serveur *PCP* [67].

3.2 Protocole de transport

Tous les messages *PCP* sont transportés sur *UDP* (*User Datagram Protocol* [55]). *PCP* utilise le numéro de port 5351 (unicast) et 5350 (multicast). La taille d'un message *PCP* ne peut excéder 1 100 octets.

4. Opérations et options *PCP*

4.1 Opérations *PCP*

4.1.1 MAP

MAP est un message (ou opération) qui permet d'instancier un état dans le serveur *PCP* pour associer l'adresse IP interne et le numéro de port interne avec une adresse externe et un numéro de

port externe pour un protocole donné. L'instanciation de cet état par un serveur *PCP* permet d'acheminer les paquets destinés à l'adresse et numéro de port internes.

Un exemple typique d'usage de ce message est la demande de contrôle du *CGN* (ou du pare-feu) par le serveur *PCP*, de sorte que le *CGN* puisse laisser passer le trafic correspondant à l'accès au serveur *FTP*. En cas de changement de l'état côté serveur *PCP*, une réponse *MAP* non sollicitée doit être générée par le serveur vers les clients *PCP* concernés.

Une application utilisant le même numéro de port pour les communications sortantes et entrantes, peut d'abord récupérer l'adresse IP et numéro de port externes grâce à un message *MAP* et envoyer ensuite ces informations à ses pairs.

Les messages *MAP* générés de manière spontanée permettent aux équipements qui embarquent un client *PCP* de notifier leur pairs de tout changement d'information de connectivité IP (par exemple adresse IP externe, numéro de port externe, etc.).

En référence à l'exemple de la figure 4, le client *PCP* a émis une requête *MAP* vers un serveur *PCP* pour demander la création d'une entrée *UDP* pour l'adresse IPv6 interne « 2001:db8::1 » et le numéro de port interne « 3938 », et l'adresse « 161.105.194.14 » et numéro de port « 15200 ». La durée de vie de cette entrée, telle que retournée par le serveur, est « 20 000 secondes ».

À noter que les entrées créées par la requête *MAP* sont par définition « *Endpoint-Independent Mappings* » (*EIMs*) avec un filtrage « *Endpoint-Independent Filtering* » (*EIF*).

Plusieurs types de filtrages peuvent être considérés :

- « *Endpoint-Independent Filtering (EIM/EIF)* » : envoyer un paquet sortant depuis un terminal interne vers n'importe quelle adresse externe est suffisant pour acheminer du trafic entrant vers ce même terminal interne à partir de n'importe quelle adresse ;
- « *Address-Dependent Filtering (ADM/ADF)* » : pour recevoir du trafic depuis une adresse donnée, un paquet sortant vers cette adresse doit être envoyé par un terminal interne ;
- « *Address and Port-Dependent Filtering (APDM/APDF)* » : pour recevoir du trafic depuis une adresse donnée, un paquet sortant vers ces adresse et numéro de port doit être envoyé par un terminal interne.

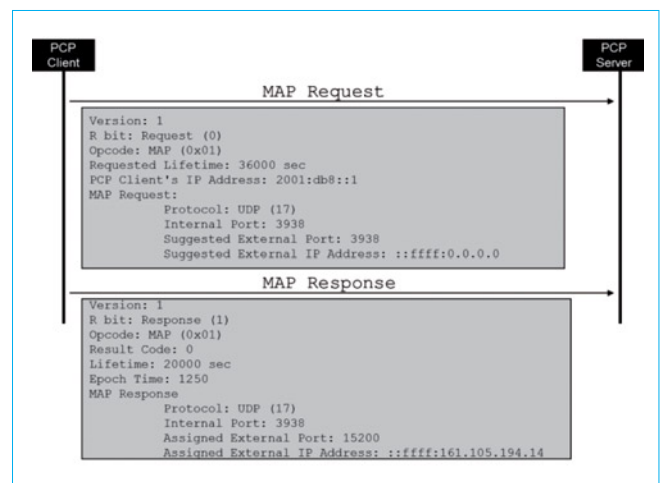


Figure 4 – Exemple d'échange de messages *MAP*

4.1.2 PEER

PEER est un message qui permet de créer une entrée dynamique vers une adresse IP ou un numéro de port. Ce message permet aussi de découvrir la durée de vie d’une entrée déjà maintenue par la fonction contrôlée par le serveur *PCP*.

L’entrée créée par un message PEER est similaire à celle créée par un message TCP SYN.

Contrairement à une requête MAP, une requête PEER ne peut pas être utilisée pour réduire la durée de vie d’une entrée.

Exemple

Si aucune entrée correspondant à ces quintuplets n’est trouvée dans la table, alors le serveur accepte la requête. Une entrée avec les informations du quintuplet est alors ajoutée à la table. Une réponse positive est envoyée au client (cf. l’exemple de la figure 5).

Les entrées créées par la requête PEER sont par définition « *Endpoint-Independent Mappings* » (EIMs) ou *Endpoint-Dependent Mapping* (EDM), avec un filtrage « *Endpoint-Independent Filtering* » (EIF) ou « *Endpoint-Dependent Filtering* » (EDF).

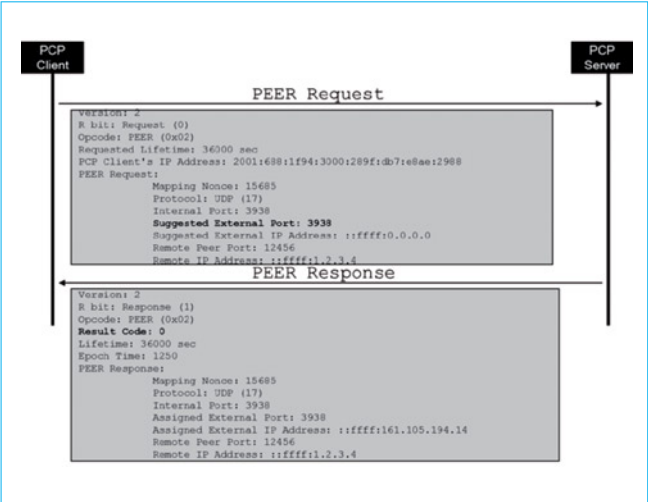


Figure 5 – Exemple d’échange de messages *PEER*

4.2 Options *PCP*

PCP est un protocole extensible. En effet, des extensions peuvent être ajoutées au protocole en définissant de nouvelles options.

Un message *PCP* peut inclure une ou plusieurs options sans aucune condition concernant l’ordre d’apparition dans une requête. Par contre, le serveur *PCP* doit les traiter dans leur ordre d’apparition.

Le tableau 2 énumère une liste d’options supportées par *PCP*.

L’option « *PREFER_FAILURE* » n’est pas obligatoire pour la fonction d’interfonctionnement d’IGD:2 [64] et *PCP*. Pour plus de détails concernant la spécification de la fonction d’interfonctionnement *UPnP IGD/PCP*, voir le document [19].

5. Fonctions avancées

5.1 Réinstaller un contexte

Les clients *PCP* utilisent le paramètre « *Epoch* » pour détecter des anomalies côté serveur.

Exemple

Si le serveur *PCP* a redémarré en raison d’une opération menée par un administrateur ou d’un problème d’alimentation, ou encore si le serveur a perdu ses états, etc., le serveur *PCP* doit réinitialiser le paramètre « *Epoch* » à « 0 ». Le paramètre « *Epoch* » doit aussi être réinitialisé si l’adresse allouée à un client par la fonction contrôlée par le serveur *PCP* a changé. Si une anomalie a été détectée par un client *PCP*, ce dernier doit réinstaller immédiatement les contextes perdus en renvoyant les messages *PCP* correspondants.

À noter qu’une anomalie consiste à détecter un décalage d’une seconde (plus ou moins une seconde) entre l’Epoch maintenu par le client et celui retourné par le serveur.

La figure 6 illustre l’usage du paramètre « *Epoch* ». On suppose qu’à T0, le client *PCP* émet une première requête vers un serveur *PCP*. En plus des informations contenues dans la réponse reçue de la part du serveur, le client sauvegarde aussi la valeur du paramètre « *Epoch* » telle que retournée dans la réponse.

Tableau 2 – Liste d’options <i>PCP</i>	
Option	Description
<i>THIRD_PARTY</i>	Indique que la requête concerne une tierce partie, et non le client <i>PCP</i> (§ 5.6)
<i>PREFER_FAILURE</i>	Cette option est utile pour la fonction d’interfonctionnement IGD:1 [63] et <i>PCP</i> [19] La présence de cette option dans un message <i>PCP</i> indique au serveur qu’il doit obligatoirement satisfaire la demande ; sinon un message d’erreur doit être retourné au client <i>PCP</i>
<i>FILTER</i>	Décrit un filtre à appliquer pour les paquets entrants (§ 5.7)
<i>DESCRIPTION</i>	Associe une description textuelle avec un état maintenu par un serveur <i>PCP</i> (§ 5.9)
<i>PREFIX64</i>	Permet de découvrir le préfixe IPv6 utilisé par une fonction NAT64
<i>PORT_SET</i>	Permet de récupérer une plage de ports

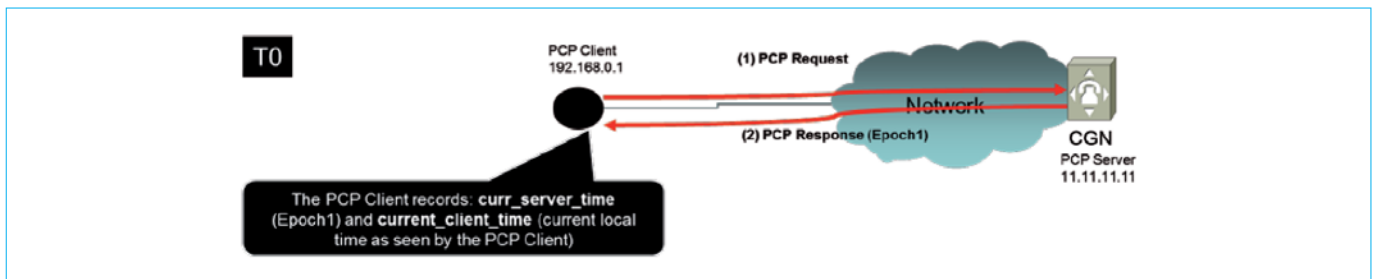


Figure 6 – Première sauvegarde de la valeur du paramètre « Epoch »

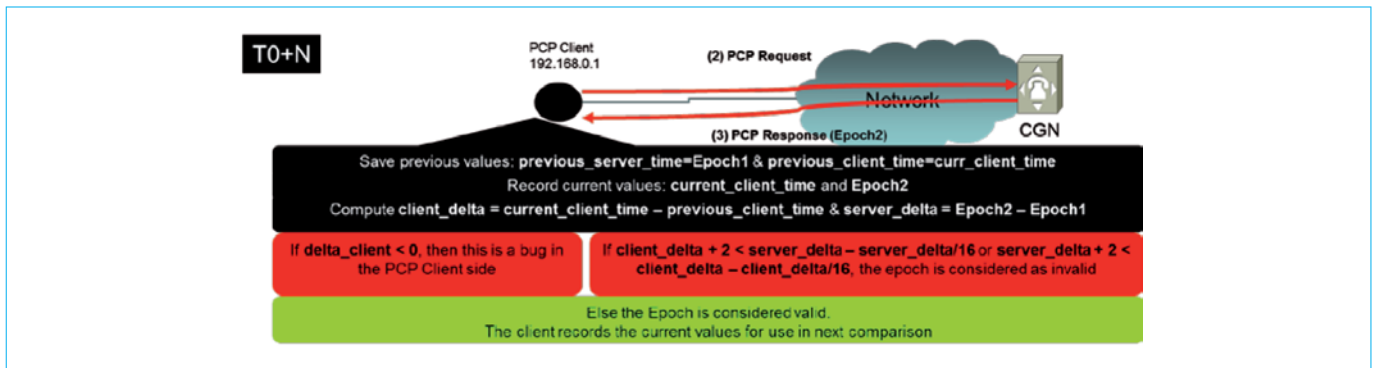


Figure 7 – Procédure de détection d'anomalie utilisant le paramètre « Epoch »

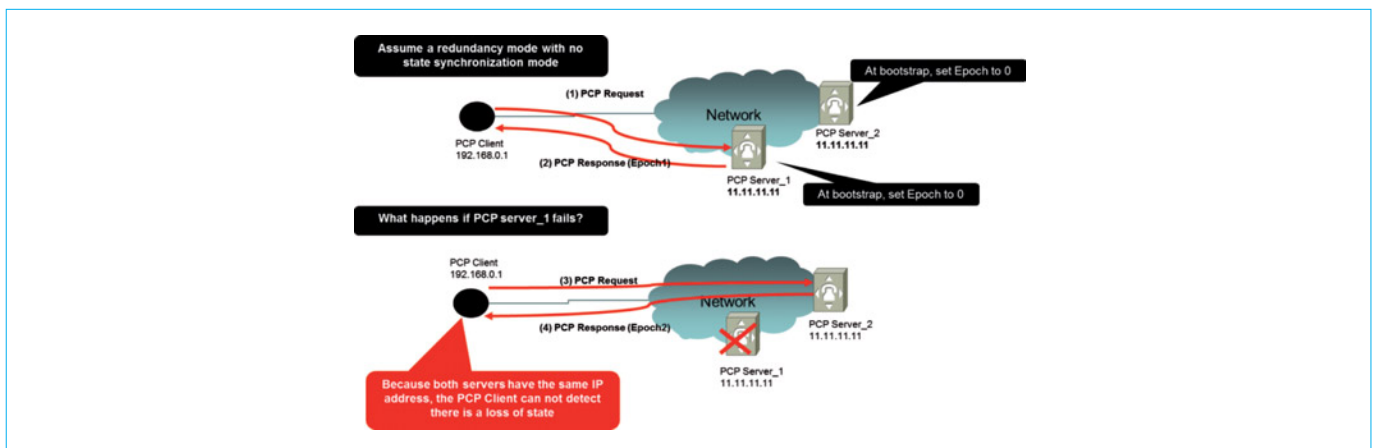


Figure 8 – Cas d'usage : « Epoch »

Après l'écoulement de « N » secondes, on suppose que le client envoie une autre requête vers le serveur comme le montre la figure 7. Dès réception de la réponse du serveur, le client récupère la valeur du paramètre « Epoch » contenue dans la requête et applique la formule de la figure 7.

Cette procédure encore appelé mode « *anycast* » peut être utilisée pour déployer deux serveurs PCP ayant la même adresse IP, mais sans nécessiter de mécanisme de synchronisation d'état entre ces serveurs PCP.

En référence à l'exemple de la figure 8, on considère que deux serveurs redondants (« PCP Server_1 » et « PCP Server_1 ») sont

déployés dans un réseau IP. On suppose que ces serveurs sont joignables avec la même adresse « 11.11.11.11 ».

On suppose en outre que la valeur du paramètre « Epoch » est initialisée au même moment pour les deux serveurs.

Dans ce contexte, les requêtes d'un client PCP sont alors acheminées vers le serveur le plus proche [au sens *SPF* (*Shortest Path First*)]: PCP Server_1 dans cet exemple. Si PCP Server_1 n'est plus joignable, les requêtes seront redirigées automatiquement vers PCP Server_2; cependant, le client ne peut pas détecter le changement de serveur car les deux serveurs sont visibles avec la même adresse IP et la même valeur du paramètre « Epoch » !

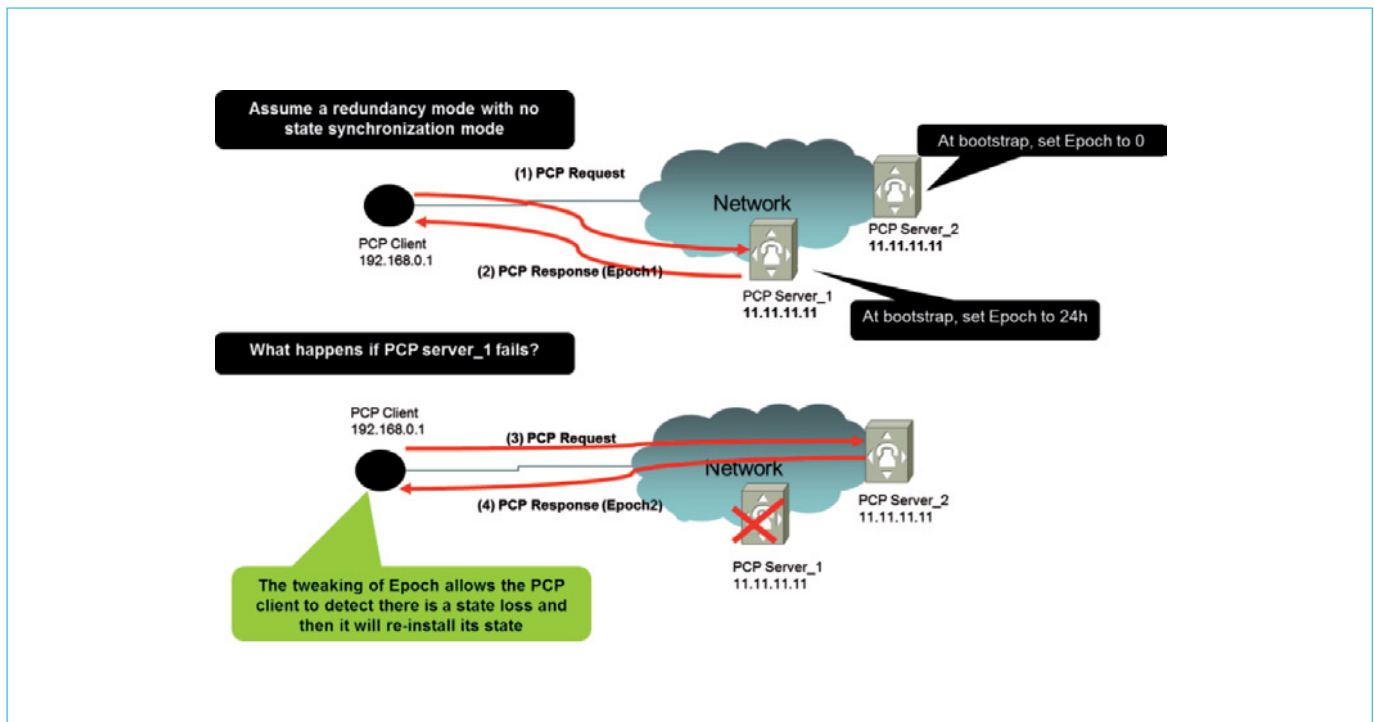


Figure 9 – Gestion de la redondance de serveurs PCP par le paramètre Epoch

Exemple

Pour résoudre ce problème, une configuration alternative est décrite dans la figure 9. Concrètement, la solution consiste à configurer des valeurs du paramètre « Epoch » différentes pour les deux instances du serveur, par exemple avec un décalage de 24 h.

En cas de panne de la première instance du serveur, les requêtes PCP seront redirigées vers le deuxième serveur. Le client pourra facilement détecter une anomalie grâce au calcul de l'Epoch (24 h de décalage). En conséquence, le client PCP réinstallera les états dans la deuxième instance. Ce faisant, la dégradation de service qui résulte de l'indisponibilité d'une fonction réseau est optimisée grâce à PCP.

Cette solution n'est pas limitée au seul cas où deux instances de serveurs PCP sont présentes ; elle peut être généralisée pour un nombre important de serveurs PCP avec un mode d'adressage *anycast* et sans mécanisme de synchronisation d'état entre ces différentes instances.

5.2 Restaurer rapidement un état

Le mécanisme de restauration rapide PCP permet de réinstaller immédiatement un état dans une fonction contrôlée par un serveur PCP lorsque cet état a été supprimé, par exemple lorsque le serveur a redémarré, etc.

Pour notifier cette perte d'état, le serveur PCP doit générer un message « ANNOUNCE » dont le champ « Epoch » est valorisé à « 0 ». Ce message « ANNOUNCE » est envoyé en multicast. À réception de ce message par le (ou les) client(s) PCP, la procédure de détection d'anomalie décrite dans la section précédente permettra aux clients de rétablir leurs états dans le serveur PCP.

Exemple

La figure 10 illustre le cas d'un serveur qui a perdu certaines des entrées instanciées par des clients PCP. Par conséquent, le serveur émet un message « ANNOUNCE » vers le client PCP en utilisant le mode de transmission multicast. Dès réception du message, le client PCP procède à l'installation de l'entrée correspondant au numéro de port interne « 3938.UDP » à l'aide du message MAP.

Le client indique l'adresse IP externe et numéro de port qui ont été auparavant alloués par le serveur (« 161.105.194.14:15000 ») comme étant ses préférences. Le serveur PCP répond positivement à cette requête.

5.3 Notifier un changement d'adresse ou de numéro de port externes

PCP supporte un mécanisme pour permettre aux serveurs PCP de notifier de manière spontanée les clients PCP de toute modification de leurs entrées instanciées par ces serveurs. Un exemple typique est le changement de l'adresse externe allouée par le serveur PCP. Ce mécanisme consiste à envoyer des réponses MAP/PEER non sollicitées aux clients impactés.

Pour minimiser le risque de perte de message généré de manière spontanée, la spécification PCP recommande de transmettre le message plusieurs fois, tout en ajustant la valeur du champ « Epoch » entre les retransmissions.

La spécification PCP recommande également de transmettre le message 3 fois avec un délai minimum de 250 ms entre les deux premières retransmissions, et la troisième retransmission après un délai de 500 ms.



Ce mécanisme permet de détecter rapidement tout changement côté serveur et d'optimiser l'impact sur les services.

La figure **11** décrit un cas qui illustre ce mécanisme lorsqu'une nouvelle adresse est allouée par le serveur. Cet exemple indique que la nouvelle adresse « 1.23.54.2 » et numéro de port « 15000 » sont maintenant alloués au client.

En effet, le serveur compare l'adresse source de la requête *PCP* avec l'adresse IP contenue dans la requête. Dans l'exemple de la figure **12**, l'adresse source « 11.11.11.11 » n'est pas identique à l'adresse « 192.168.0.1 ». Le serveur rejette la requête par un message d'erreur « *ADDRESS_MISMATCH* » sans créer d'état pour le port interne demandé.

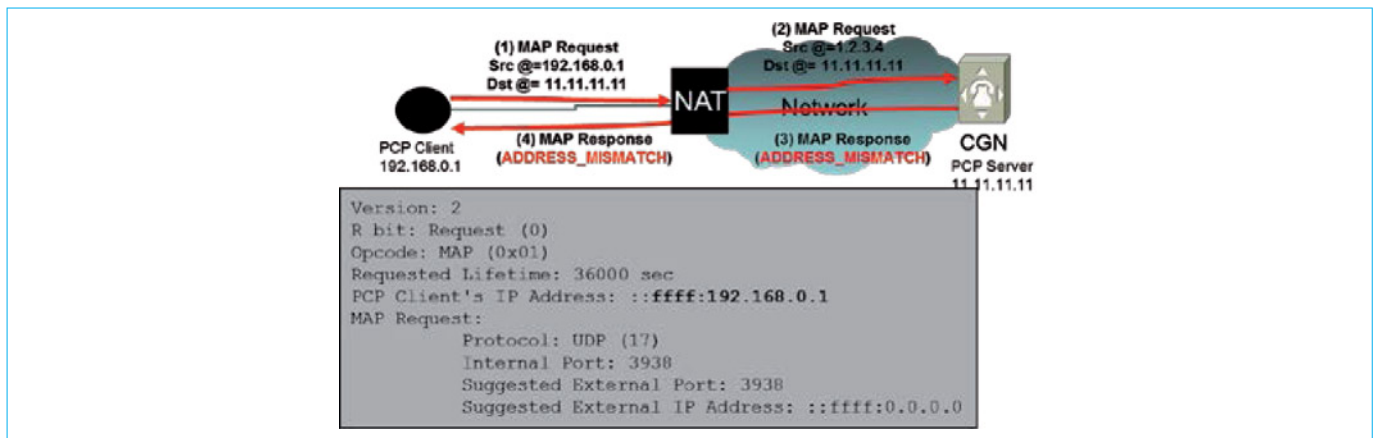


Figure 12 – Détection de NAT

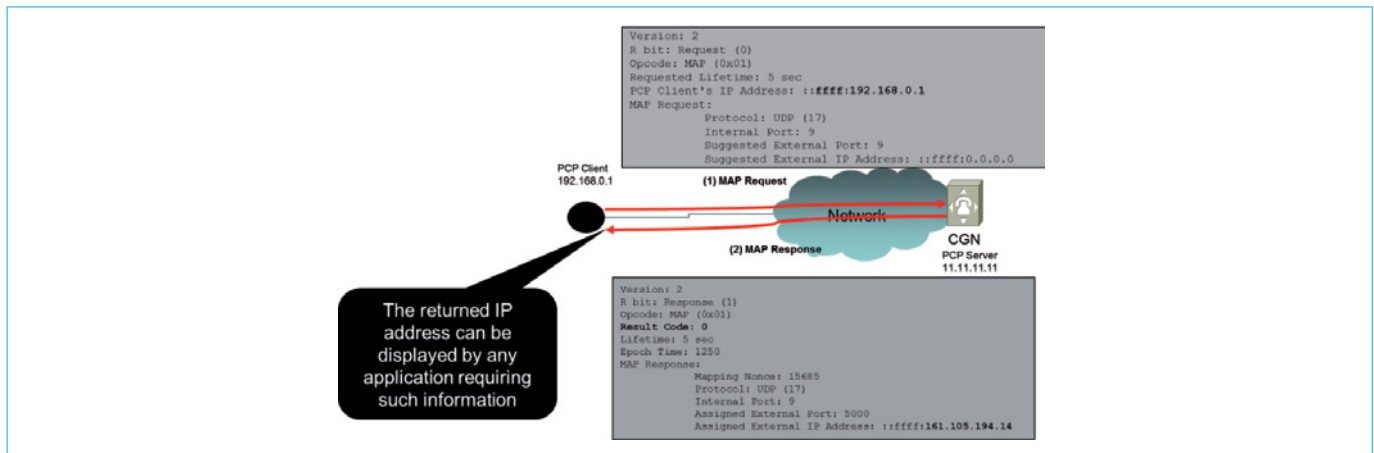


Figure 13 – Découverte d'adresse IP externe

Cette procédure simplifie le diagnostic de problèmes de connectivité qui peuvent être induits par la présence de NAT. Le client peut utiliser ce message d'erreur pour interagir avec le NAT en question, et ainsi résoudre les problèmes de connectivité.

5.5 Découvrir l'adresse externe

Afin de découvrir l'adresse IP externe, un client PCP peut émettre un message MAP avec une durée de vie courte ayant comme numéro de port « 9 (TCP ou UDP) ». Ces deux numéros de port correspondent au « Discard Service » [53]. La figure 13 illustre un cas d'utilisation de cette fonction supportée par le protocole PCP pour découvrir l'adresse externe.

En effet, le client PCP émet une requête PCP dont la durée de vie est de 5 secondes [Étape (1)]. Le serveur retourne l'adresse IPv4 « 161.105.194.14 » [Étape (2)].

L'application peut alors utiliser cette adresse en tant que de besoin.

5.6 Créer une entrée pour un tiers

PCP supporte la fonction qui consiste à instancier des états dans un NAT ou un firewall pour les besoins d'un tiers appartenant au

même client (administratif) : c'est le cas typique d'un CPE embarquant un client PCP et qui agit au nom des terminaux connectés à ce CPE. Cette fonction est réalisée grâce à l'option appelée : « THIRD_PARTY ».

Cette option peut être utilisée pour s'affranchir de la contrainte de mettre à jour tous les équipements connectés à un réseau pour supporter un client PCP. En effet, une interface d'administration peut être hébergée par un CPE ou dans le réseau de l'opérateur. Les demandes effectuées via l'interface d'administration seront traduites en des demandes PCP. L'option « THIRD_PARTY » est utilisée pour désigner le propriétaire de la demande (afin d'éviter que les contextes ne soient associés au client PCP mais au tiers).

Un serveur PCP peut activer ou désactiver cette option.

À noter que l'option « THIRD_PARTY » est une option obligatoire ; si le serveur ne supporte pas cette option ou a été configuré pour l'ignorer, un message d'erreur doit être envoyé au client PCP (figure 14).

En référence à l'exemple de la figure 14, un utilisateur ne peut pas accéder à une Webcam depuis Internet.

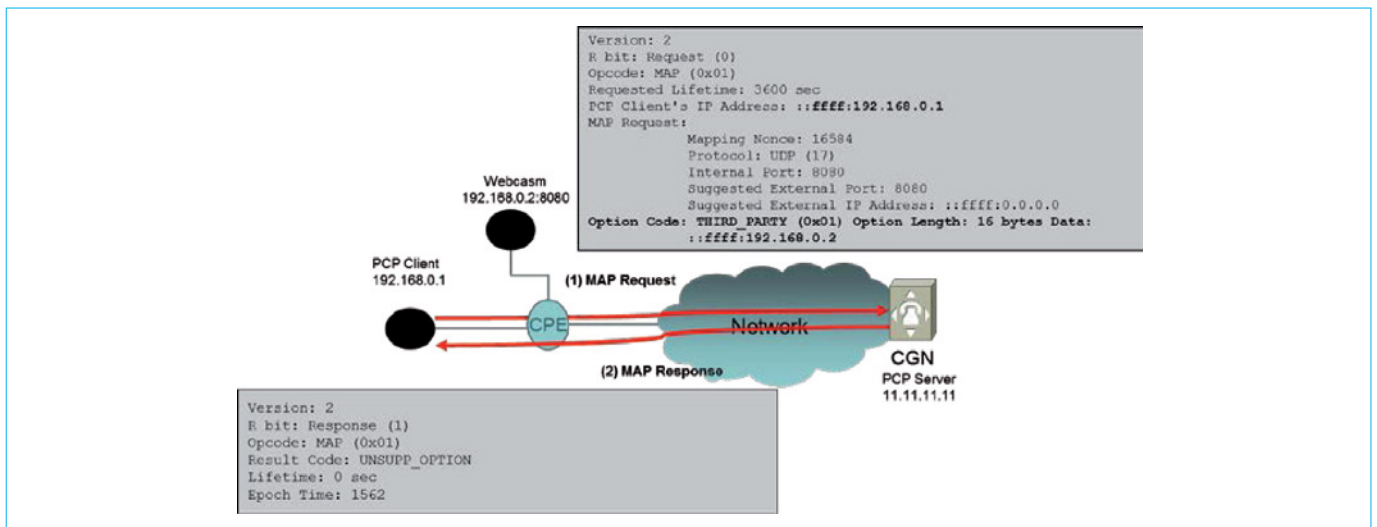


Figure 14 – Exemple d'utilisation de l'option « *THIRD_PARTY* » : message d'erreur

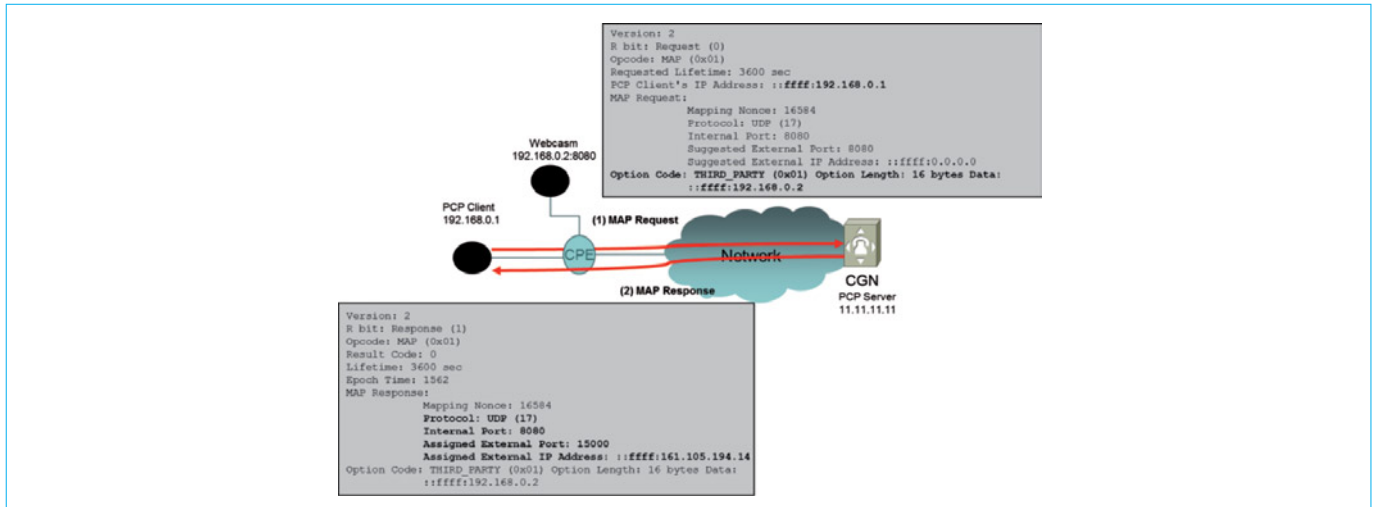


Figure 15 – Exemple d'utilisation de l'option « *THIRD_PARTY* »

On suppose maintenant que le serveur *PCP* active l'option « *THIRD_PARTY* » (figure 15). Un client veut accéder à sa Webcam depuis Internet, mais celle-ci ne supporte pas de client *PCP*. Un client *PCP* localisé dans le *CPE* envoie un message *PCP* avec une option « *THIRD_PARTY* » qui indique l'adresse IP « 192.168.0.2 » de la Webcam ; le numéro de port « 8080 » est inclus dans le corps du message *MAP*.

Lorsque la requête est reçue par le serveur, il détecte la présence de l'option « *THIRD_PARTY* ». Le serveur effectue des vérifications pour s'assurer que les deux adresses (c'est-à-dire celle du client *PCP* ainsi que celle contenue dans l'option « *THIRD_PARTY* ») appartiennent au même client. Dans cet exemple, aucune anomalie n'est détectée par le serveur.

Par conséquent, il accepte la requête, demande au *CGN* d'instancier une entrée spécifique, et renvoie ensuite une réponse indiquant l'adresse et le numéro de port externes relatifs à l'adresse « 192.168.0.2 » et au numéro de port « 8080 ».

Les fonctions de sécurité sont importantes pour se prémunir contre toute tentative de piratage de session avec l'option « *THIRD_PARTY* » [67].

L'utilisateur de la Webcam dispose maintenant de l'adresse et numéro de port externes pour accéder à cette caméra depuis l'Internet : « 161.105.194.14:15000 ».

La figure 16 illustre un exemple de paquet entrant destiné à « 161.105.194.14:15000 ».

Le paquet est d'abord reçu par le *CGN* qui examine ses tables pour retrouver l'adresse et numéro de port internes correspondant à « 161.105.194.14:15000 ». Étant donné qu'un état est présent, le *CGN* procède à la traduction du paquet en remplaçant l'adresse destination par « 192.168.0.2 » et numéro de port interne par « 8080 ». Le paquet ainsi construit est alors acheminé vers la Webcam.

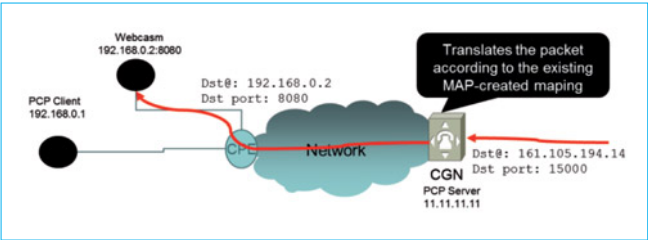


Figure 16 – Accès à une Webcam depuis l’Internet en utilisant l’option « THIRD_PARTY »

À noter que l’option THIRD_PARTY ne doit pas contenir la même adresse IP que celle du client PCP ; sinon un message d’erreur doit être envoyé par le serveur (figure 17).

5.7 Installer des filtres

PCP permet d’installer des filtres pour restreindre les communications à une liste spécifique de pairs. En effet, l’option « FILTER » permet d’instruire le serveur PCP pour configurer un ou plusieurs filtres.

Un serveur PCP peut activer ou désactiver cette option.

À noter que l’option « FILTER » est une option obligatoire ; si le serveur ne supporte pas cette option ou s’il a été configuré pour l’ignorer, un message d’erreur doit être envoyé au client PCP.

La figure 18 illustre un exemple d’utilisation de cette fonction. Cet exemple décrit le cas d’un client PCP qui ne veut recevoir de trafic entrant qu’en provenance de l’adresse source « 1.2.3.4 » associée au numéro de port « 5968 ». La réponse positive du serveur indique que le filtre a été installé correctement par le CGN.

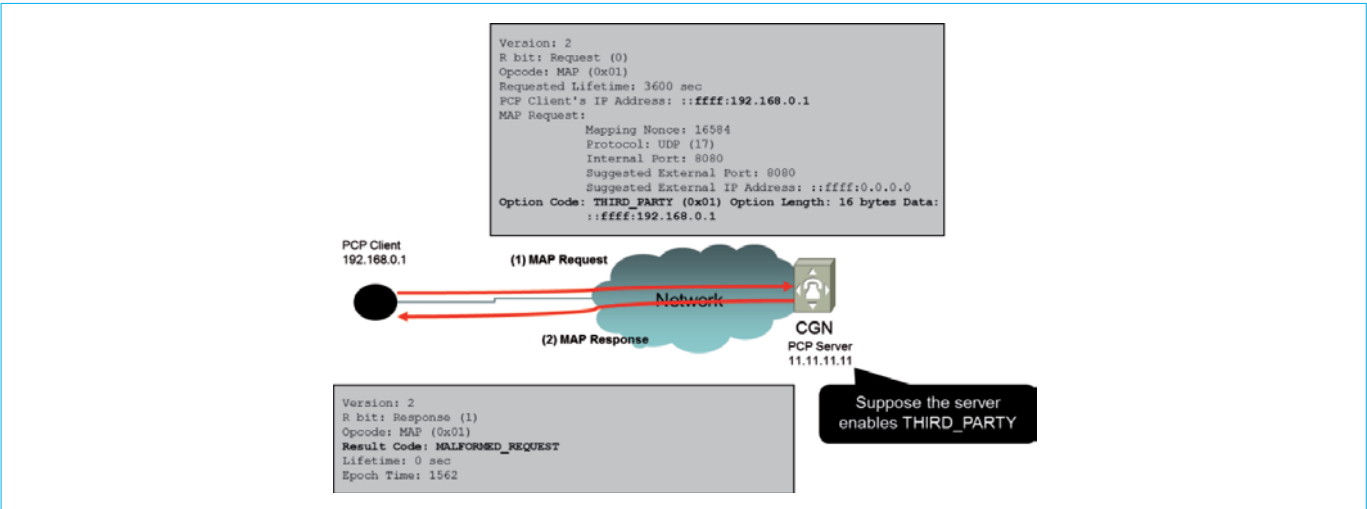


Figure 17 – Exemple d’utilisation de l’option « THIRD_PARTY » : adresse du client PCP

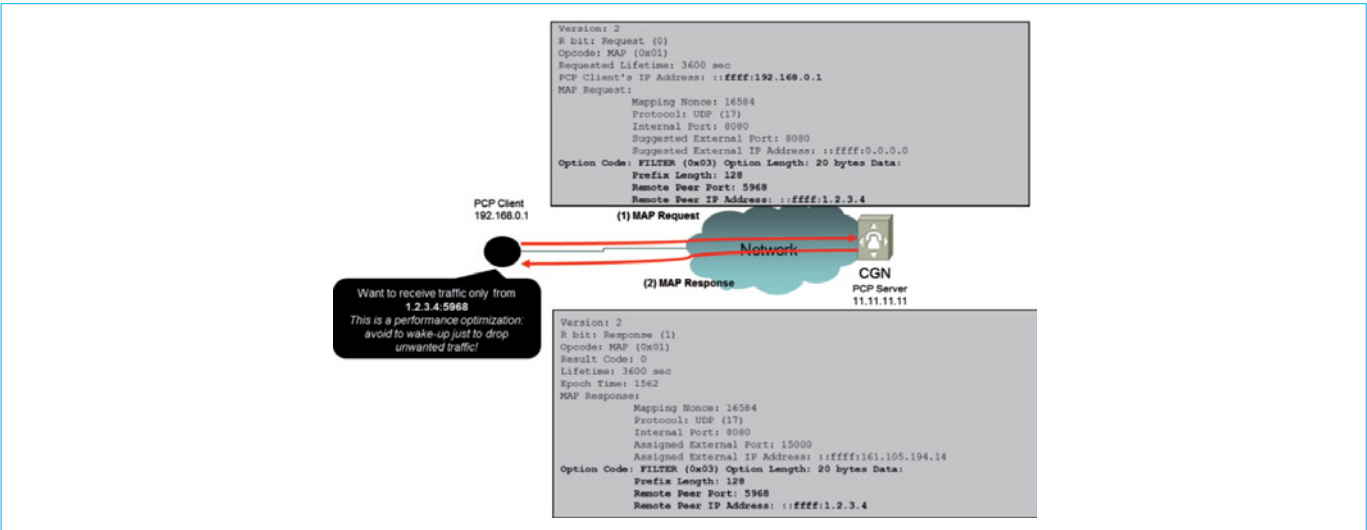


Figure 18 – Exemple d’utilisation de l’option « FILTER »

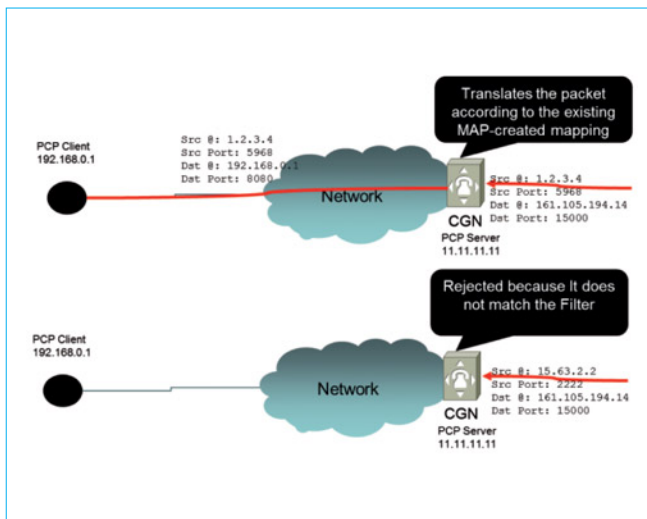


Figure 19 – Exemple d'utilisation de l'option « FILTER » – Traitement de paquet entrant

La figure 19 illustre le comportement du CGN lorsqu'il reçoit un paquet après l'installation du filtre. En effet, le premier paquet ayant comme adresse source « 1.2.3.4 » et numéro de port source « 5968 » est acheminé par le CGN vers sa destination finale, alors que le paquet ayant comme adresse source « 15.63.2.2 » est rejeté par le CGN, car il ne correspond pas au filtre installé par le client.

À noter que :

- Le client PCP peut faire installer plusieurs filtres avec le même message ou avec des messages différents.
- Le client PCP peut demander la suppression de tous les filtres installés en envoyant un message MAP avec une option « FILTER » dont le champ « Remote IP Address » est valorisé à zéro.
- Le client peut faire mettre à jour la liste des filtres en insérant dans le même message une option « FILTER » nulle et d'autres options « FILTER » qui décrivent les nouveaux paramètres de filtrage (figure 20).

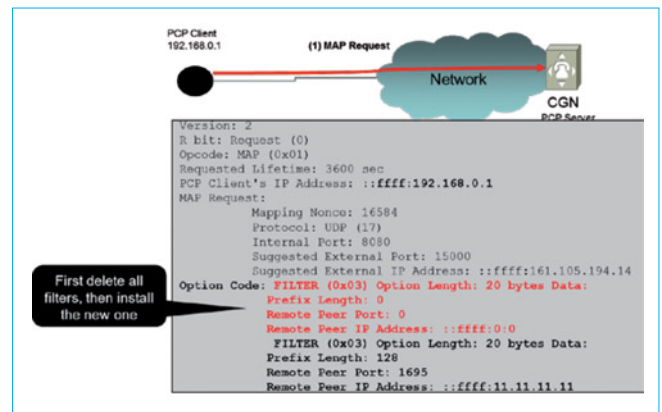


Figure 20 – Exemple de mise à jour de filtres

5.8 Corréler des adresses externes avec des adresses internes

La figure 21 décrit deux scénarios dans lesquels la fonction PCRF (Policy and Charging Rules Function [1]) ne peut pas appliquer convenablement des politiques de service car les informations remontées par la fonction PCEF (Policy and Charging Enforcement Function [1]) ne coïncident pas avec celles remontées par l'AF (Application Function [1]).

■ Scénarios de problèmes

Ces scénarios sont les suivants [21] :

• Scénario 1

La fonction NAT est localisée entre la fonction PCEF et AF. La fonction PCEF envoie au PCRF les informations caractéristiques du flux de paquets identifiés par la paire {192.168.1.1, 5888}, mais à cause de la présence de NAT entre la fonction PCEF et la fonction AF, l'adresse remontée par AF est « 1.2.3.4 ». Le PCRF ne peut pas corréler les informations remontées par le PCEF et AF.

• Scénario 2

La fonction NAT est localisée avant la fonction PCEF et AF. Les deux fonctions PCEF et AF remontent les mêmes informations, mais la fonction PCRF ne peut pas corréler ces informations « 1.2.3.4:2566 » avec l'identifiant interne de l'UE [User Equipment (terminal mobile, typiquement) par exemple, l'identifiant IMSI (International Mobile Subscriber Identity) ou l'adresse IP interne].

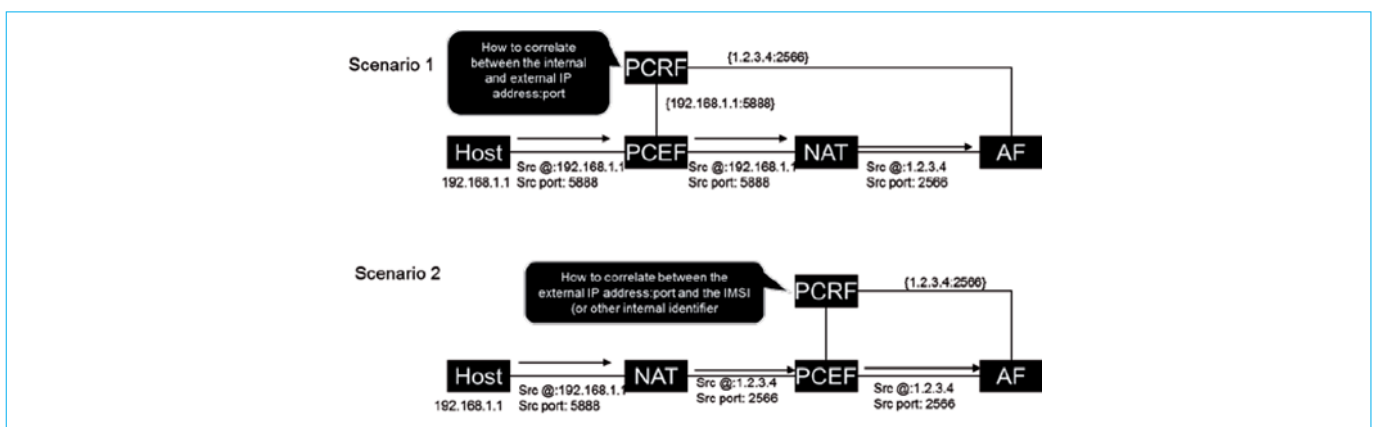


Figure 21 – Problème de corrélation d'adresse interne avec une adresse externe

■ Scénarios de solutions

PCP peut résoudre ces deux problèmes de la manière suivante :

• Scénario 1

La solution PCP consiste à activer un client PCP dans la fonction PCRF et un serveur PCP colocalisé avec la fonction NAT. Quand la fonction PCEF remonte les informations caractéristiques d'un flux de paquets, la fonction PCRF envoie un message MAP avec l'option « THIRD_PARTY » pour récupérer l'adresse et numéro de port externes alloués par la fonction NAT, et associés aux informations remontées par le PCEF (c'est-à-dire au couple {192.168.1.1, 5888}). Le serveur PCP co-localisé avec la fonction NAT renvoie une réponse MAP indiquant les informations externes suivantes : {1.2.3.4, 2566}. Cette réponse permet alors de corréliser avec succès les informations remontées par le PCEF avec celles remontées par la fonction AF (figure 30).

• Scénario 2

Contrairement au scénario précédent, celui-ci nécessite une nouvelle extension appelée « QUERY » (figure 22). La fonction PCRF utilise le message « QUERY » pour récupérer les informations internes associées à une adresse et un numéro de port externes, telles que remontées par la fonction PCEF/AF (figure 31).

5.9 Associer une description avec une entrée

L'option « DESCRIPTION » est utilisée pour associer une description textuelle avec une entrée créée par une application auprès d'un serveur PCP [16]. Cette information est utile dans plusieurs scénarios tels que :

- assurer l'interfonctionnement avec UPnP IGD [19]. Ce faisant, l'information transportée par la variable UPnP IGD « PortMapping Description » est relayée vers le serveur PCP ;
- simplifier certaines opérations de gestion du serveur PCP. Par exemple, la description textuelle évite à un administrateur de supprimer certaines entrées importantes ;
- dans certains déploiements, un portail est présenté aux utilisateurs pour gérer leurs entrées PCP (cf. par exemple, § 5.2 du [13]). Cette option peut être utilisée pour sauvegarder l'identifiant de l'utilisateur ayant instancié l'entrée correspondante. Cette information n'est consultable que par l'administrateur du serveur PCP.

Un exemple d'utilisation de cette option est illustré par la figure 23. L'extrait de la table du serveur PCP indique l'attribut « Description » associé à trois entrées.

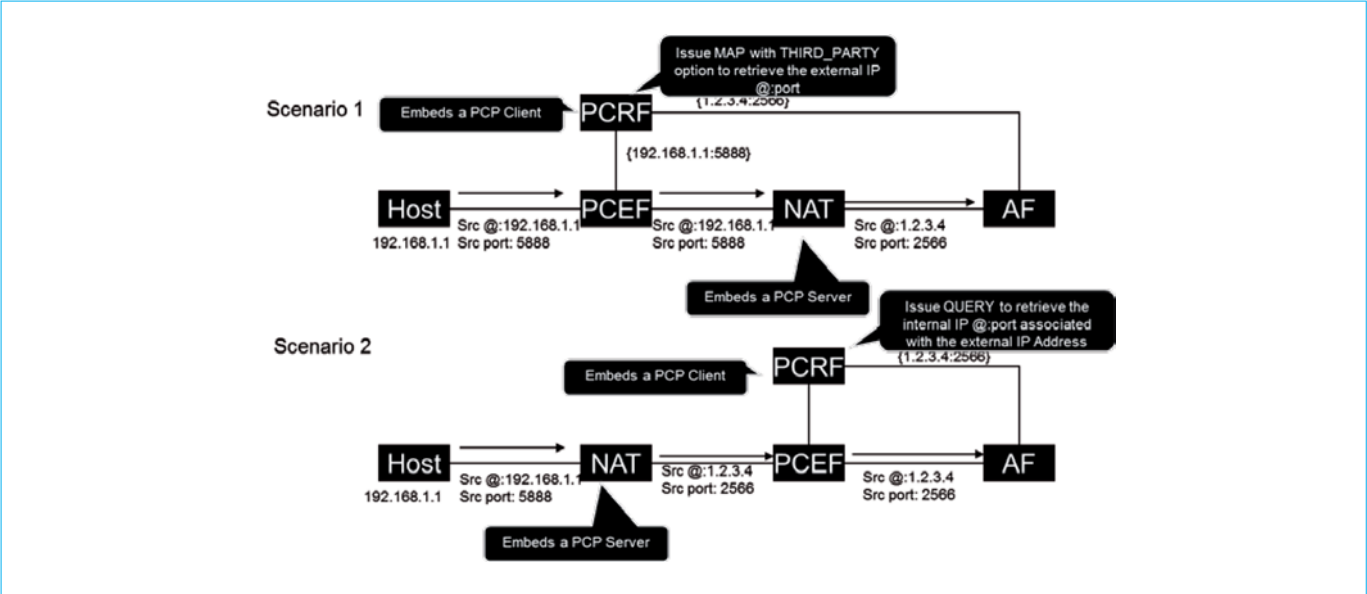


Figure 22 – Solution PCP pour corréliser l'adresse et numéro de port internes avec l'adresse et le numéro de port externes

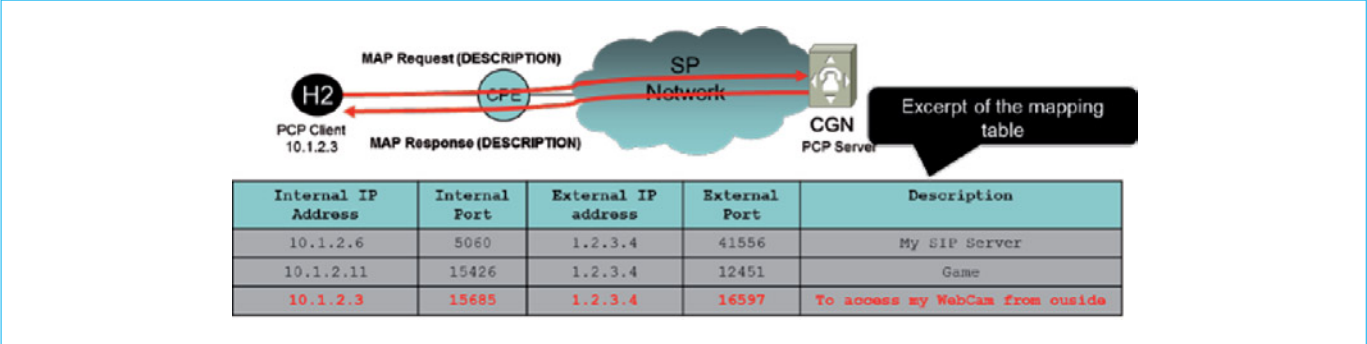


Figure 23 – Exemple d'utilisation de l'option « DESCRIPTION »

6. Exemples d'échanges PCP

Cette section liste plusieurs exemples pour illustrer l'utilisation de PCP. [71] est un recueil exhaustif d'exemples PCP.

6.1 Préférences honorées par le serveur

La figure 24 illustre un échange PCP dans lequel le client demande explicitement l'allocation du numéro de port « 15685 ». Cette requête est satisfaite par le serveur.

6.2 Problème de validation de « Nonce »

L'échange de la figure 25 illustre un scénario dans lequel la demande ne peut être satisfaite car la valeur demandée n'est pas autorisée, conformément au code d'erreur NOT_AUTHORIZED.

6.3 Numéro de port demandé déjà alloué par le serveur

Lorsque la requête PCP exprime une préférence pour l'allocation d'un numéro de port spécifique, la réponse à cette requête peut faire l'objet d'une autre proposition.

C'est ce qu'illustre la figure 26, où la requête PCP exprimait une préférence pour le port « 3938 », alors que le serveur PCP a commandé l'allocation du port « 15685 ».

6.4 Exemple d'utilisation de « PREFER_FAILURE »

L'option « PREFER_FAILURE » est souvent utilisée dans un contexte PCP où une fonction d'interfonctionnement IGD/PCP est mise en œuvre.

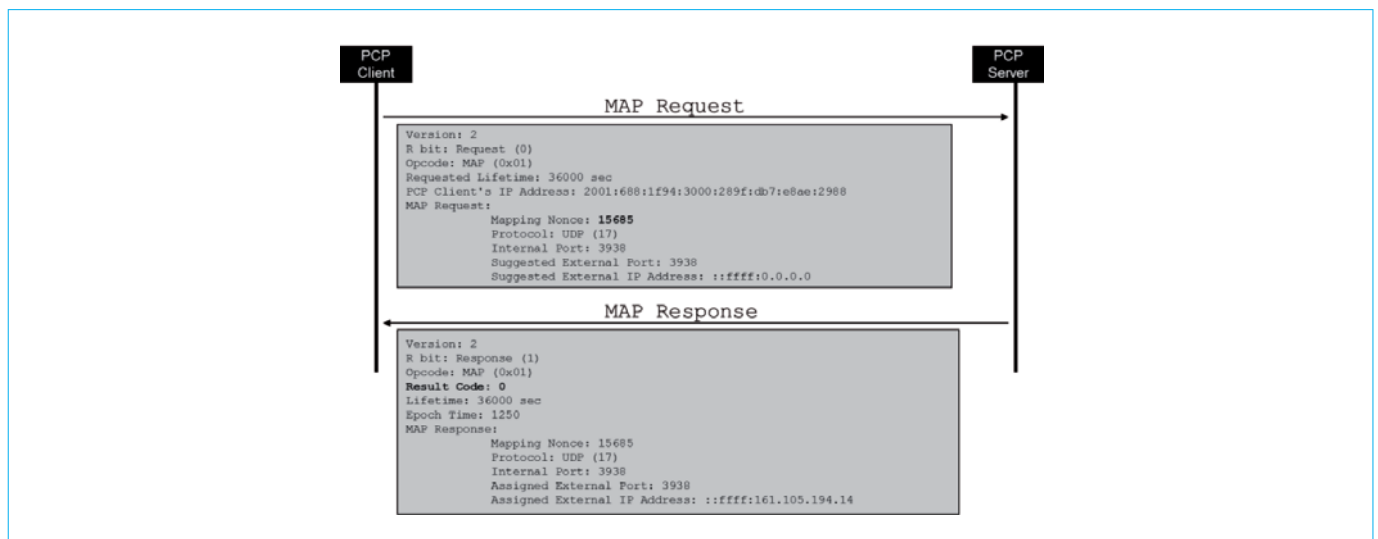


Figure 24 – Exemple de requête satisfaite par le serveur

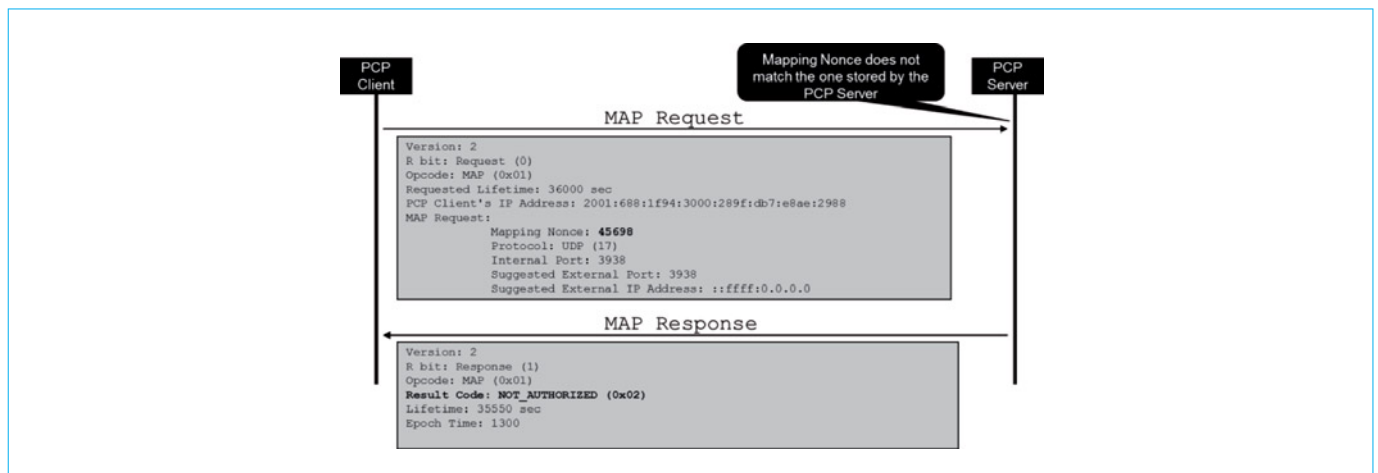


Figure 25 – Exemple de problème de validation de « Nonce »

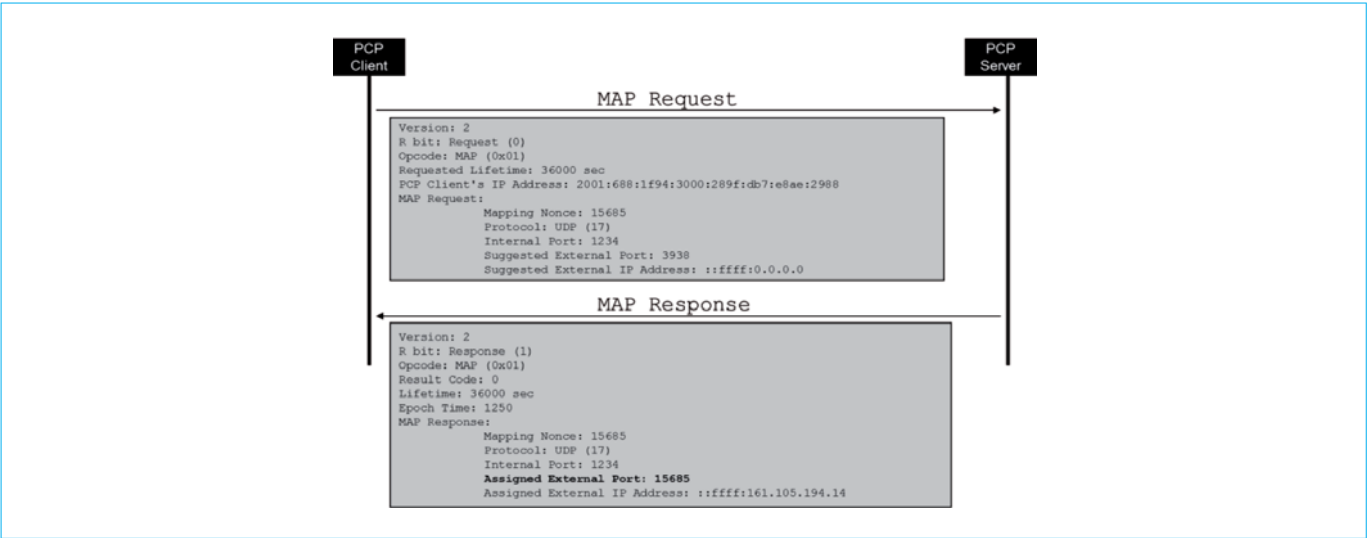


Figure 26 – Exemple d’échange quand le numéro de port est déjà alloué

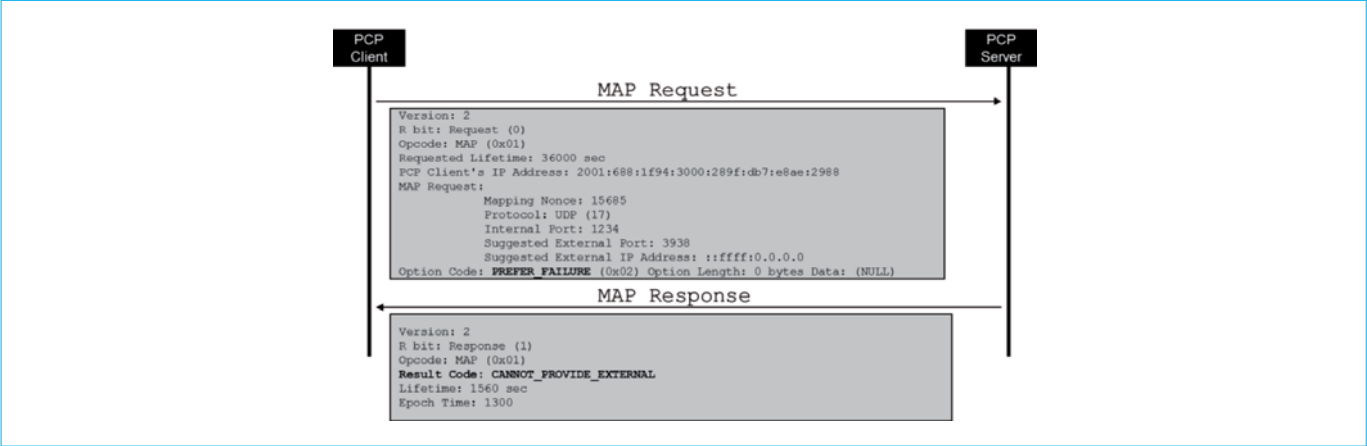


Figure 27 – Exemple d’utilisation de l’option « *PREFER_FAILURE* »

Dans l’exemple de la figure 27, la demande explicite d’allocation de numéro de port exacerbée par la présence de l’option « *PREFER_FAILURE* » ne peut être satisfaite et occasionne la transmission d’un message d’erreur par le serveur *PCP*.

6.5 Contrôle d’une entrée existante

La figure 28 reflète le cas où un client *PCP* souhaite faire confirmer la validité d’une entrée existante. Cette confirmation de l’existence de l’entrée correspondant au numéro de port TCP « 1234 » de l’exemple est explicitement mentionnée dans la réponse envoyée par le serveur.

6.6 Autoriser le trafic entrant associé à un protocole donné

En référence à la figure 29, le client *PCP* demande la création d’une entrée pour l’ensemble du trafic reposant sur le protocole de

transport *UDP*. À réception de cette requête, le serveur *PCP* va commander la fonction contrôlée correspondante (par exemple un pare-feu) afin qu’elle établisse l’entrée demandée. Si cette requête se réfère à un protocole qui n’est pas supporté par la fonction contrôlée, le serveur *PCP* renverra un message avec le code d’erreur « *UNSUPP_PROTOCOL* ».

La demande de création d’une entrée pour un protocole donné peut être étendue à n’importe quel protocole, de sorte que la fonction contrôlée peut autoriser l’acheminement de l’intégralité du trafic entrant, comme l’illustre la figure 30.

6.7 Quota de ports consommés

Lorsqu’un terminal (ou un *CPE*) a consommé l’ensemble des numéros de ports qui lui ont été attribués, le client *PCP* embarqué dans ce terminal (ou *CPE*) ne sera plus en mesure d’obtenir un (ou des) numéro(s) de port(s) supplémentaire(s). C’est ce qu’illustre la figure 31, dans laquelle la réponse fournie par le serveur *PCP* à la requête initiale comporte le code d’erreur « *USER_EX_QUOTA* ».

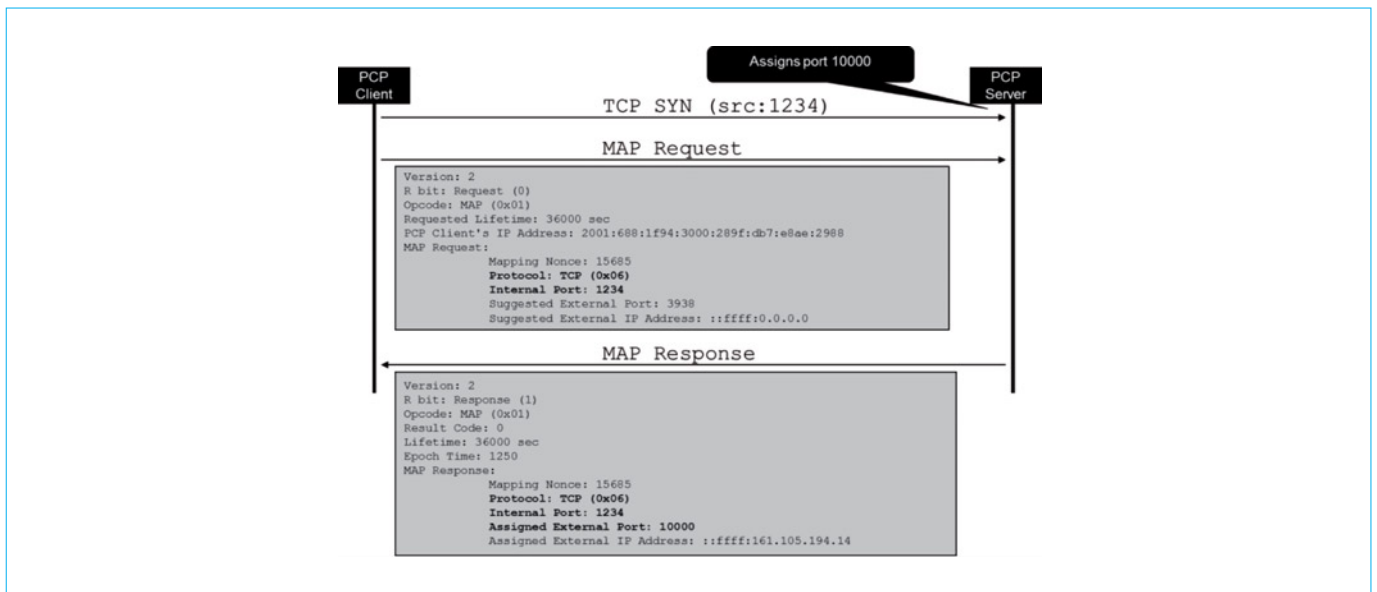


Figure 28 – Exemple de contrôle d'une entrée explicite existante

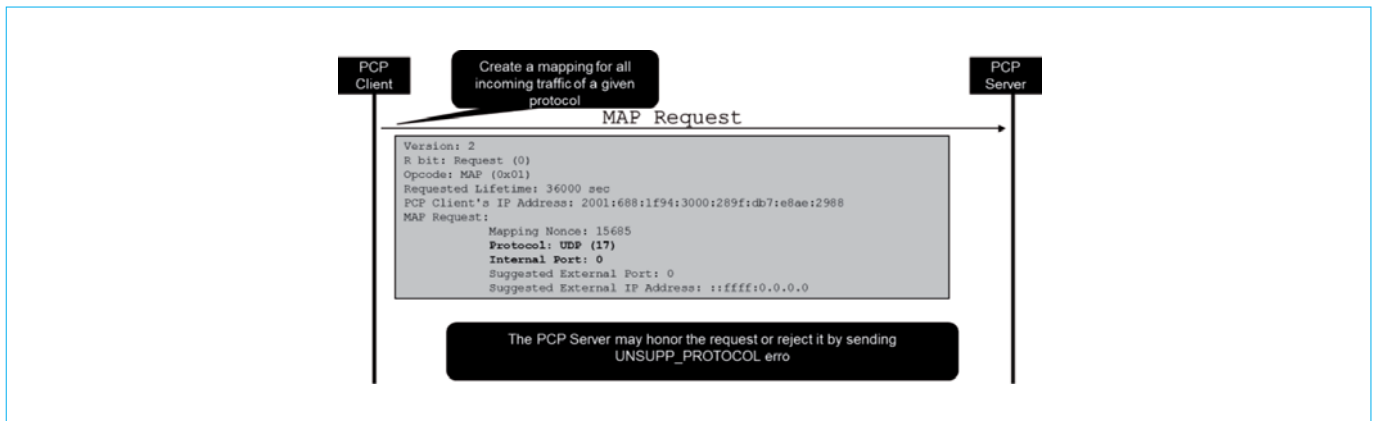


Figure 29 – Créer une entrée pour tout le trafic UDP entrant

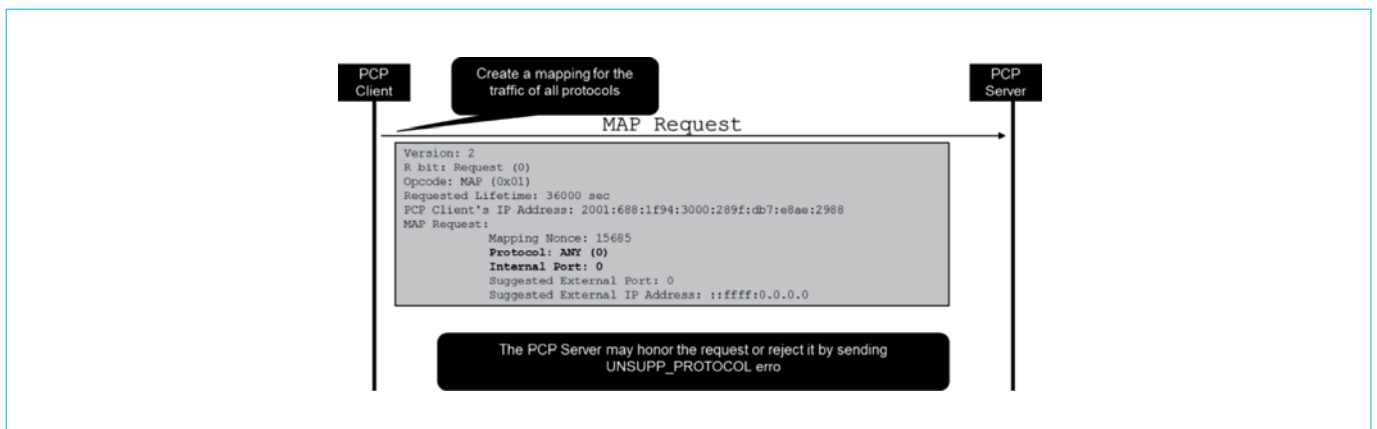


Figure 30 – Créer une entrée pour tout le trafic entrant

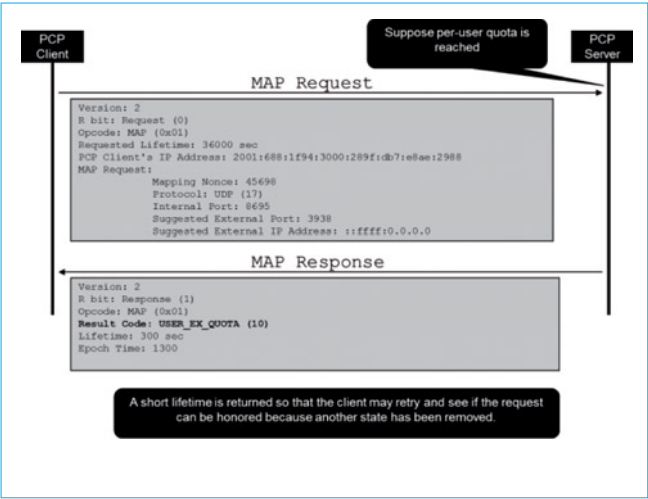


Figure 31 – Exemple de quota de port(s) consommé(s) par un client

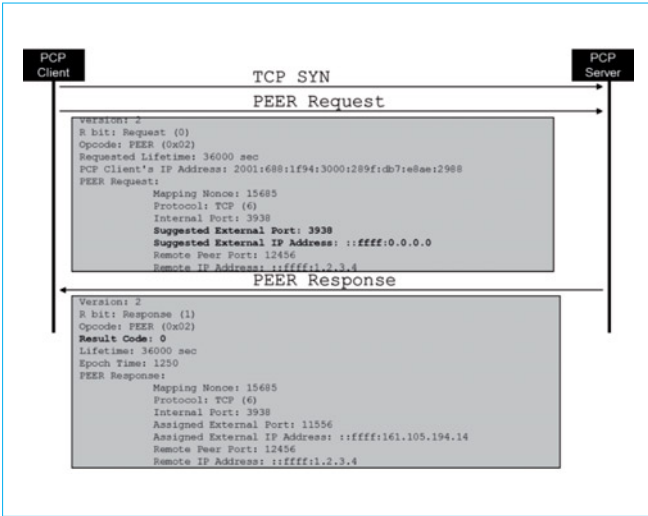


Figure 33 – Exemple pour étendre la durée de vie d’une entrée implicite

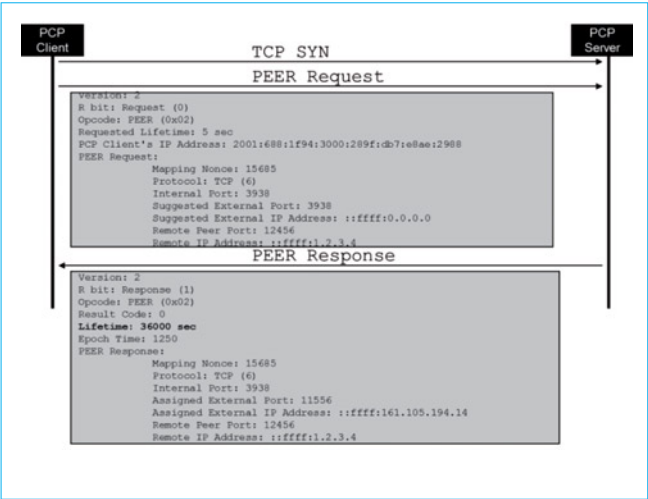


Figure 32 – Exemple pour découvrir la durée de vie d’une entrée

Toutefois, la réponse du serveur *PCP* peut comporter une valeur du paramètre « *Lifetime* » suffisamment courte pour inciter le client *PCP* à renouveler sa requête ultérieurement, c’est-à-dire dans des délais acceptables pour l’utilisateur.

6.8 Découvrir la durée de vie d’une entrée

En utilisant une requête *PEER*, un client *PCP* a la possibilité de découvrir la valeur de la durée de vie associée à une entrée spécifique. C’est ce qu’illustre la figure 32 (36 000 s dans l’exemple).

6.9 Étendre la durée de vie d’une entrée existante

La figure 33 illustre un exemple d’un client *PCP* qui demande l’extension de la durée de vie d’une entrée existante.

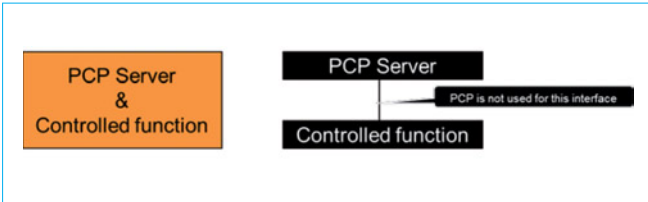


Figure 34 – Localisation du serveur *PCP*

7. Scénarios de déploiement

Cette section identifie plusieurs scénarios de déploiement *PCP*, selon l’usage et le placement du serveur *PCP* [13].

7.1 Placement du serveur *PCP*

D’une manière générale, un serveur *PCP* pourrait être co-localisé avec la ou les fonctions contrôlées (figure 34) pour une simple question d’efficacité. D’un point de vue technique, il est effectivement essentiel de préserver le niveau de qualité associé à un service dont le bon fonctionnement est en partie conditionné par la capacité de *PCP* à répondre favorablement à une requête émise par le client.

En particulier, le traitement de la requête *PCP* intègre notamment l’instruction de la commande par la fonction contrôlée et générée par le serveur *PCP*. Cette commande peut être formatée selon la sémantique caractéristique de la technologie mise en œuvre par la fonction contrôlée : c’est par exemple l’utilisation d’une syntaxe *CLI* (*Command Line Interface*) destinée à activer un filtre supplémentaire dans un pare-feu.

Le temps de traitement d’une telle commande contribue ainsi à l’efficacité de la machinerie protocolaire *PCP*, ce qui suggère naturellement que ce temps puisse être aussi optimisé que possible, grâce à la co-localisation du serveur *PCP* avec la (ou les) fonction(s) contrôlée(s) par le serveur.

7.2 CPE embarquant un serveur PCP

Les CPE activent un ensemble de fonctions susceptibles d'être contrôlées par un serveur PCP, afin de répondre aux besoins des terminaux connectés aux CPE et embarquant un client PCP (figure 35). Le serveur PCP du CPE permet en particulier de contrôler les fonctions NAT et pare-feu en fonction, par exemple, des contenus maintenus par les terminaux et qui doivent être accessibles depuis l'Internet.

L'usage d'un serveur PCP embarqué dans un CPE est également motivé par la capacité de PCP à contrôler dynamiquement les filtres mis en place et maintenus par le pare-feu embarqué dans le CPE, par exemple dans un contexte de contrôle parental ou de gestion d'une transaction financière dont l'établissement est conditionné par l'identification explicite de sites commerciaux réputés fiables.

De plus, un serveur PCP permet une gestion temporelle des fonctions NAT et pare-feu, et permet ainsi à un serveur de fichiers connecté au CPE de ne pouvoir être sollicité depuis l'Internet que pendant une période restreinte et éventuellement renouvelable.

7.3 Serveur localisé dans le réseau opérateur

Ce modèle suppose qu'un ou plusieurs serveurs PCP sont localisés dans le réseau exploité par l'opérateur (figure 36). Ces clients PCP doivent être capables d'acquérir les adresses pour contacter ces serveurs et de décider si l'ensemble de ces serveurs doivent être sollicités ou seulement certains de ces serveurs [20]. En particulier, le déploiement de PCP dans une infrastructure mobile suppose généralement que le client PCP est embarqué dans le terminal mobile. Un tel contexte se révèle particulièrement intéressant pour une gestion optimisée des ressources énergétiques du terminal mobile, comme discuté au § 9.

7.4 Cacher l'identité du serveur PCP

Ce modèle de déploiement suppose que l'identité ou la localisation d'un serveur PCP ne doit pas être connue des clients PCP afin de minimiser les risques d'attaque de déni de service, par exemple.

Dans ce cas, l'ingénierie PCP repose sur l'utilisation de fonctions « Proxy » (figure 37) qui se chargent de relayer les requêtes émises par les clients PCP vers le (ou les) serveur(s) PCP, tout en se comportant comme un serveur PCP vis-à-vis de ces clients.

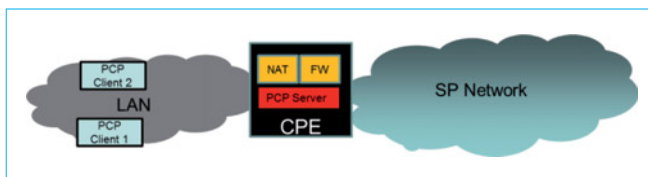


Figure 35 – Utilisation de PCP pour le contrôle d'un CPE

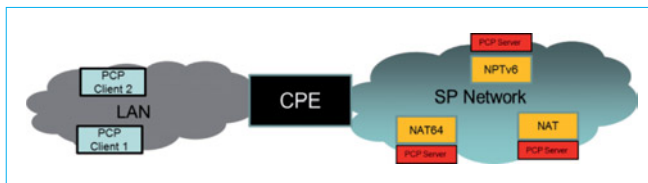


Figure 36 – Serveur PCP hébergé par un fournisseur de service Internet

La fonction « Proxy » est chargée de relayer les requêtes PCP émises par un client vers un serveur PCP. Cette fonction « Proxy » se comporte comme un serveur PCP vis-à-vis des clients PCP à l'origine d'une requête. La fonction « Proxy PCP » se comporte ensuite comme un client PCP chargé de relayer cette requête vers le serveur PCP *ad hoc*.

L'introduction d'une telle fonction « Proxy » au sein d'une architecture PCP se justifie en particulier par :

- la capacité pour un terminal embarquant un client PCP de pouvoir découvrir de façon dynamique le ou les serveur(s) PCP disponibles. Le client PCP n'a pas connaissance de l'identité de(s) serveur(s) PCP ;

- la capacité de pouvoir s'accommoder des éventuelles fonctions NAT et pare-feu embarquées dans l'équipement qui supporte la fonction « Proxy PCP » de façon à garantir l'acheminement des requêtes PCP émises par les terminaux connectés à l'équipement qui embarque la fonction « Proxy PCP » vers le ou les serveurs PCP disponibles. La fonction « Proxy » se charge alors de commander la création des entrées correspondantes dans les tables maintenues par les fonctions NAT et pare-feu avec lesquelles elle cohabite.

7.5 CPE embarquant un « proxy PCP »

Ce scénario est typique d'une ingénierie PCP faisant appel à une fonction « Proxy ». Celle-ci est embarquée dans le CPE, qui se charge donc de relayer les requêtes émises par les clients PCP embarqués dans les terminaux (figure 38) vers l'un ou l'autre des serveurs PCP déployés dans le réseau, en fonction de la nature de la requête ou de la fonction contrôlée par PCP.

La fonction « Proxy » peut être co-localisée avec une fonction NAT ou pare-feu.

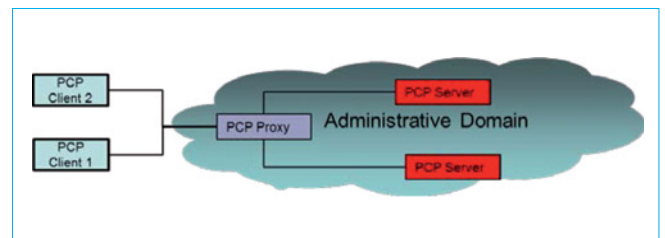


Figure 37 – Cacher l'identité du serveur PCP

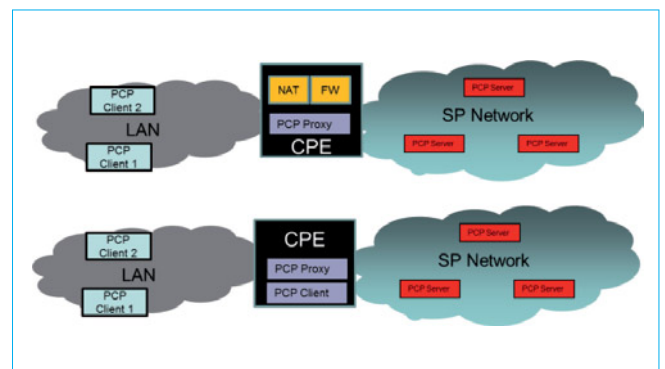


Figure 38 – Exemple d'un CPE qui embarque une fonction « proxy PCP »

8. Voix sur IP

8.1 Introduction

SIP (*Session Initiation Protocol* [58]), protocole largement adopté par les fournisseurs de service de voix sur IP [*VoIP* pour (*Voice over IP*)], est un mécanisme de gestion de sessions et qui s'appuie sur d'autres protocoles comme *SDP* (*Session Description Protocol* [34]). *SDP* est utilisé pour décrire les informations média d'un agent *SIP* à un pair *SIP* distant.

Cette description inclut les informations de connectivité IP. En particulier, la partie *SDP* d'un message d'établissement de session *SIP* (par exemple « *INVITE* ») indique, en plus des informations décrivant les codecs (Codeur/Décodeur) audio ou vidéo supportés, l'adresse IP et un ou plusieurs numéros de port à utiliser pour établir les sessions *RTP* (*Realtime Transport Protocol* [60]). La plupart des déploiements *SIP* n'incluent que le numéro de port *RTP* dans un message *SDP*; le numéro de port *RTCP* est déduit d'une manière algorithmique à partir de celui de *RTP* conformément à [60] (c'est-à-dire le numéro de port *RTCP* est le numéro de port *RTP* + 1).

- **À noter** que rares sont les déploiements qui utilisent l'attribut « *a=rtcp* » [39] pour expliciter le numéro de port *RTCP* ou qui utilisent le même numéro de port pour multiplexer les flux *RTP* et *RTCP*.
- **À noter** que ces problèmes sont valides pour tout protocole utilisant des références (« *referrals* ») pour notifier un correspondant des informations de connectivité IP [26].

Plusieurs solutions ont été envisagées par la communauté *SIP* pour permettre la fourniture de services *SIP* en présence de *NAT* ou de pare-feu (par exemple [38] [44] [45] [57]). Ces solutions souffrent de plusieurs limitations comme la complexité des serveurs *SIP*, la surcharge des *NAT*/pare-feu et des serveurs *SIP*, l'incapacité de découvrir la durée de vie des entrées de *NAT* ou de pare-feu pour les sessions *SIP* et *RTP*, la complexité des agents *SIP*, etc.

Contrairement à ces solutions, l'utilisation de *PCP* dans un environnement *SIP* présente les avantages suivants [10] :

- éviter l'activation d'*ALG* (*Application Level Gateway*) dans un *NAT* ou un pare-feu ;
- rendre obsolète l'activation de mécanismes de traversée de *NAT* supplémentaires comme, par exemple, l'activation de la fonction *HNT* (*Hosted NAT Traversal* [44]) par un serveur *SIP*. De ce fait, l'utilisation de *PCP* rend la présence de *NAT* et de pare-feu transparente pour la plate-forme de service *SIP* ;
- éviter de surcharger le réseau et la plate-forme de service VoIP avec les messages dits « *keep-alive* » ; ces messages sont émis fréquemment pour maintenir les entrées dans un *NAT* ou un pare-feu ;
- éviter l'activation de la fonction « *Symmetric RTP/RTCP* » [66] par les agents et serveurs *SIP* ;
- éviter l'activation de la fonction « *Symmetric SIP* » par les agents *SIP* (c'est-à-dire le support de l'attribut *SIP* « *rport* » [56]) ;
- permettre l'établissement de sessions média unidirectionnelles (par exemple contacter un serveur d'annonce ou de messagerie vocale) ;
- permettre l'échange des flux média avant l'établissement de la session *RTP*. Ces flux média sont appelés « *early media* » [25] ;
- optimiser les ressources d'interconnexion IPv4-IPv6 grâce à l'activation de *PCP* conjointement avec le support de l'attribut *ALTC* (*Alternate SDP connectivity* [14]) ;
- ne pas induire de délai supplémentaire pour établir les sessions *SIP* contrairement à *ICE* (*Interactive Connectivity Establishment* [57]) par exemple.

L'activation des *ALG* dans des éléments intermédiaires peut constituer un obstacle à l'évolution du service. L'activation des *ALG* n'est pas recommandée, compte tenu du risque avéré de dégradation des performances.

PCP peut être envisagé dans le contexte de déploiements *SIP* « classiques » ([2] [10]) ou de services émergents comme *Web Real-time Communication* (*WebRTC* [3]). À ce titre [51] discute plus en détail comment *PCP* simplifie la fourniture des services *WebRTC*.

8.2 Contexte *DS-Lite*

8.2.1 Rappel

Afin d'illustrer l'utilisation de *PCP* pour la fourniture de service *SIP* en présence de *NAT*, nous considérons l'exemple d'une architecture *DS-Lite* [32].

DS-Lite est une solution de continuité de services IPv4 déployée au-dessus d'une infrastructure IPv6. Comme le montre la figure 39, l'architecture *DS-Lite* déporte la fonction *NAT* du *CPE* dans un équipement localisé dans le réseau de l'opérateur.

Les paquets IPv4 reçus par le *CPE* sont encapsulés dans des paquets IPv6 à destination de la fonction *CGN* [encore appelé *AFTR* (*Address Family Transition Router*)]. La fonction *CGN* maintient des entrées *NAT*, notamment pour pouvoir acheminer le trafic retour vers le bon *CPE*.

Une pluralité de fonctions *CGN* peut être déployée par un fournisseur de connectivité IP.

8.2.2 Recommandations *PCP* pour *DS-Lite*

Les fonctions *PCP* utiles dans un contexte *DS-Lite* sont les suivantes :

- utilisation du mode de transport IPv6 pour communiquer avec le serveur *PCP* qui contrôle le *CGN DS-Lite* ;
- le client *PCP* utilise l'adresse IPv6 du *CGN* comme adresse du serveur *PCP* ;
- le support de l'option « *PREFER_FAILURE* » est obligatoire pour les besoins d'interfonctionnement UPnP *IGDI/PCP*. Cette fonction d'interfonctionnement est embarquée dans le *CPE* ;
- le support de l'option « *THIRD_PARTY* » est obligatoire ;
- le *CPE* embarque aussi un client *PCP* associé avec une interface d'administration *HTTP*.

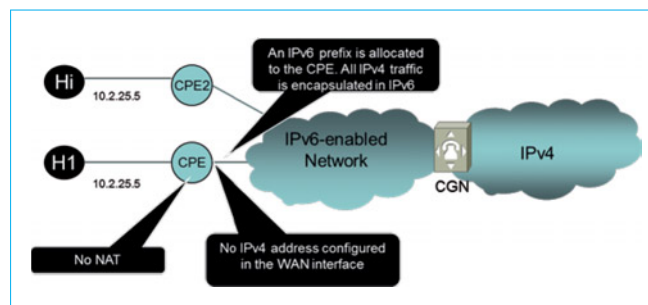


Figure 39 – Architecture de référence *DS-Lite*

8.2.3 Exemple d'établissement de session SIP

Pour établir une session SIP, le client PCP récupère une paire de ports en insérant l'option « *PORT_SET* » dans le message MAP [Étapes (1) et (2) de la figure 40]. On suppose que, suite à ces échanges, le client PCP récupère l'adresse IP externe « 1.2.3.4 » et la paire de numéros de port (« 18684 », « 18685 »).

En référence à la figure 40, l'agent SIP construit un message « *INVITE* » pour établir une session SIP avec « *hostA@example.com* ». Ce message « *INVITE* » ne contient que l'adresse et les numéros de port alloués par le CGN.

L'agent SIP transmet le message « *INVITE* » au CGN qui le traduit et le transmet ensuite vers le SBE sans modifier le contenu des en-têtes SIP, ni les lignes SDP (figure 41). Le message « *INVITE* » est traité par la plate-forme de service SIP avant acheminement vers l'agent SIP appelé.

La réponse à ce message « *INVITE* » est relayée vers l'agent SIP appelant (figure 42). Les messages SIP échangés avant l'établissement des sessions RTP/RTCP [c'est-à-dire correspondant aux étapes (3) à (8)] ne sont pas modifiés par le CGN. Les sessions RTP/RTCP peuvent alors être établies avec succès en présence de CGN et sans avoir besoin d'activer une ALG dans le CGN ni de fonction particulière côté plate-forme de service SIP.

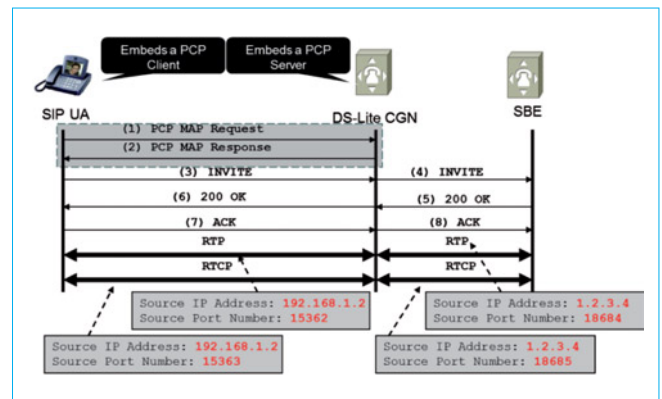


Figure 42 – Exemple de sessions RTP/RTCP dans un contexte DS-Lite

9. Réseaux mobiles

Les réseaux mobiles souffrent de plusieurs problèmes liés à la fourniture de services à valeur ajoutée en présence de NAT et de pare-feu. Ces problèmes peuvent être résolus grâce à l'activation de PCP. En effet, l'activation de PCP dans les réseaux mobiles permet de :

- simplifier la fourniture de services nécessitant des communications entrantes (par exemple, « *Push Mail* », messagerie instantanée (par exemple *MSN*, *QQ*), applications P2P (*Peer to Peer*) comme *BitTorrent* [23]) ;
- éviter le piratage de sessions grâce à la capacité de supprimer les états lorsqu'une nouvelle adresse est allouée au terminal mobile ;
- optimiser la consommation de la batterie d'un UE (*User Equipment*) mobile (cf. l'exemple qui suit) ;
- restaurer les entrées NAT et pare-feu en cas de redémarrage du terminal ou lorsqu'une nouvelle adresse est allouée, ou en cas de redémarrage d'une fonction NAT ou d'un pare-feu ;
- découvrir le préfixe utilisé par une fonction NAT64 pour construire des adresses IPv6 embarquant des adresses IPv4 [9]. L'utilisation de PCP est plus efficace que l'heuristique spécifiée par [59], comme discuté dans [12]. Le lecteur est invité à lire [46] pour se familiariser avec les problèmes rencontrés en présence de la fonction NAT64.

Exemple

Un client VPN embarqué dans un UE n'a pas la capacité de découvrir la durée de vie des entrées NAT ou pare-feu. Par précaution, des messages « *keep-alive* » (appelés aussi messages de rafraîchissement) sont envoyés toutes les 20 s (valeur par défaut pour IPsec [40]).

D'après une étude [36], la consommation de la batterie avec un intervalle « *keep-alive* » de 20 s est 29 mA (2G)/34 mA (3G). La consommation est réduite à 16 mA (2G)/24 mA (3G) si la fréquence de rafraîchissement est de 40 s. La consommation est réduite à 9,1 mA (2G)/16 mA (3G) si la fréquence d'envoi des messages « *keep-alive* » est de 150 s. Elle est réduite à 7,3 mA (2G)/14 mA (3G) si la fréquence de rafraîchissement est de 180 s.

À noter que si aucun message de rafraîchissement n'est envoyé par l'application, la consommation de la batterie est de 5,2 mA (2G)/6,1 mA (3G). Cette consommation peut être plus conséquente si plusieurs applications (par exemple, IPsec, SIP) émettent des messages « *keep-alive* ».

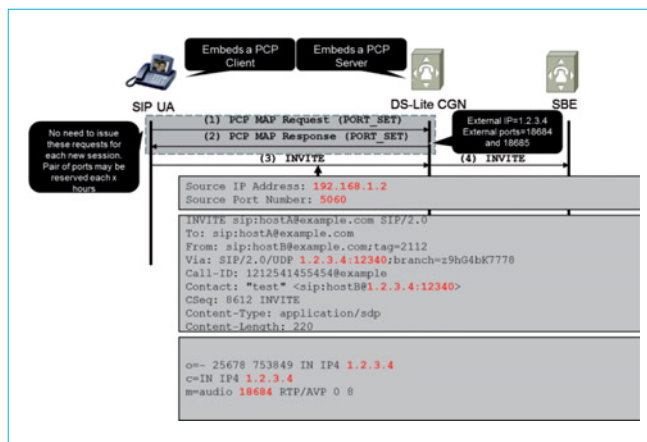


Figure 40 – Exemple de message « *INVITE* » (avant NAT)

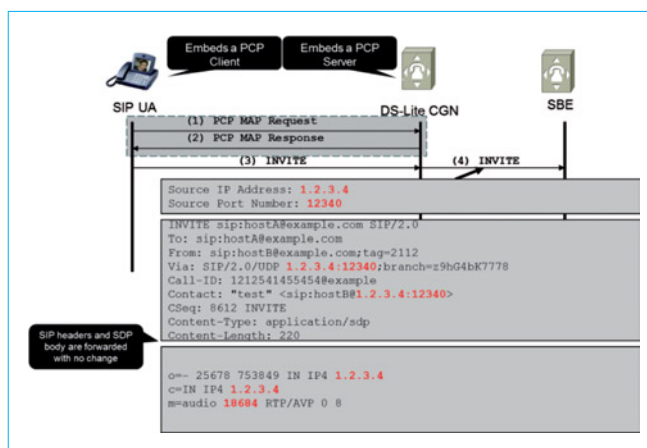


Figure 41 – Exemple de message « *INVITE* » (après NAT)

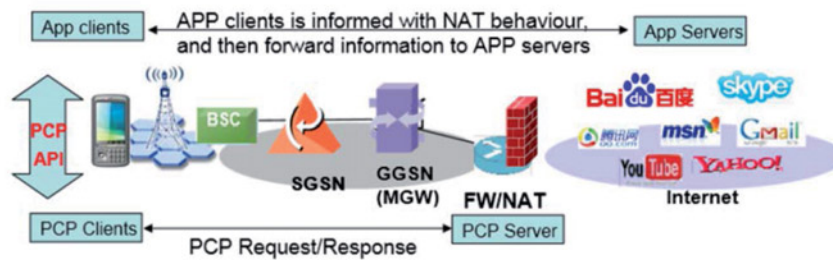


Figure 43 – Activation de PCP par les terminaux mobiles

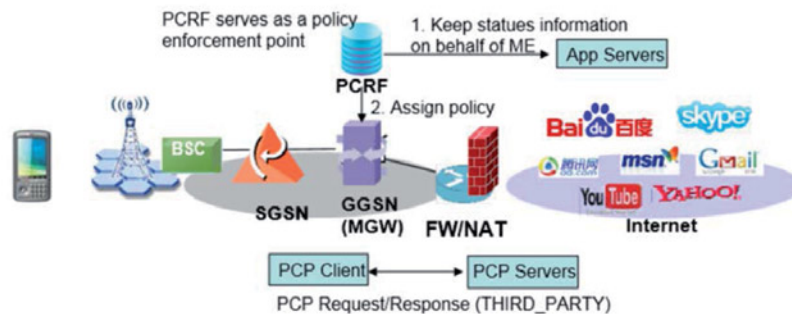


Figure 44 – Activation de PCP dans les passerelles du réseau mobile

Deux modes de déploiement PCP dans les réseaux mobiles peuvent être envisagés [27] :

- la figure 43 illustre le mode où un client PCP est embarqué dans un terminal mobile alors qu'un serveur PCP est embarqué dans un nœud du réseau. Le client PCP découvre le serveur PCP par des mécanismes spécifiques aux réseaux mobiles [par exemple, via un objet PCO (Protocol Configuration Option)]. Le client PCP peut instancier des entrées en interagissant directement avec le serveur PCP ;

- la figure 44 illustre un mode alternatif où le client PCP est sous la responsabilité de l'opérateur mobile. Ce mode s'affranchit de la contrainte du support d'un client PCP par les terminaux mobiles.

10. Continuité de service IPv4 : cas « Lightweight IPv4 over IPv6 »

L'IETF a spécifié une solution de continuité de service IPv4, appelée « Lightweight IPv4 over IPv6 » [29]. Cette solution consiste à allouer la même adresse publique IPv4 à plusieurs clients ; ces clients reçoivent également des plages de ports non recouvrants.

Ces plages de port sont utilisées à des fins de démultiplexage des communications des clients partageant la même adresse IPv4. La plage de ports affectée à un client constitue ainsi un moyen unique d'identification destiné à garantir la fiabilité de l'acheminement du trafic retour à destination de ces clients.

Exemple

- **Côté réseau**, une fonction appelée *Lightweight AFTR (lwAFTR)* doit être activée pour permettre d'acheminer les paquets vers leurs destinations parmi celles partageant la même adresse IPv4. En effet, la fonction *lwAFTR* doit maintenir une entrée par client pour renseigner l'adresse IPv6 associée à une adresse IPv4 et une plage de ports, caractéristiques d'un client donné. La décision d'acheminement du trafic retour repose sur l'analyse de l'adresse destination et du numéro de port.
- **Côté client**, une fonction *Lightweight B4 (lwB4)* doit être activée pour restreindre les numéros de ports à la plage allouée. Concrètement, la fonction *NAT* embarquée dans le *CPE* doit être mise à jour pour restreindre les numéros de port à la plage allouée par l'opérateur.

À noter que les modules *lwAFTR* et *lwB4* sont des encapsulateurs/décapsulateurs IPv4-in-IPv6.

PCP peut être utilisé pour récupérer une plage de ports comme le montre l'exemple de la figure 45. Pour ce faire, *lwB4* récupère l'adresse IPv4 externe ainsi que la plage de ports en envoyant au *lwAFTR* un message *MAP* incluant l'option « *PORT_SET* ».

En référence à l'exemple de la figure 45, l'adresse « 192.1.2.3 » et la plage « 1000-1999 » sont alloués à un *CPE* identifié par l'adresse « 2001:db8::1 » alors que l'adresse « 192.1.2.3 » et la plage « 2000-2999 » sont alloués au *CPE* identifié par l'adresse « 2001:db8::2 ».

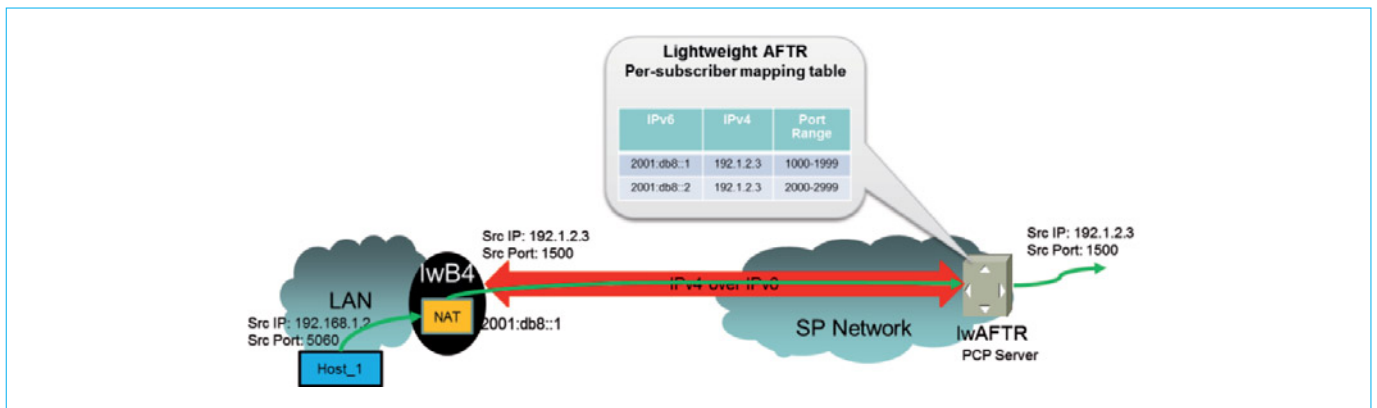


Figure 45 – Architecture « Lightweight IPv4 over IPv6 »

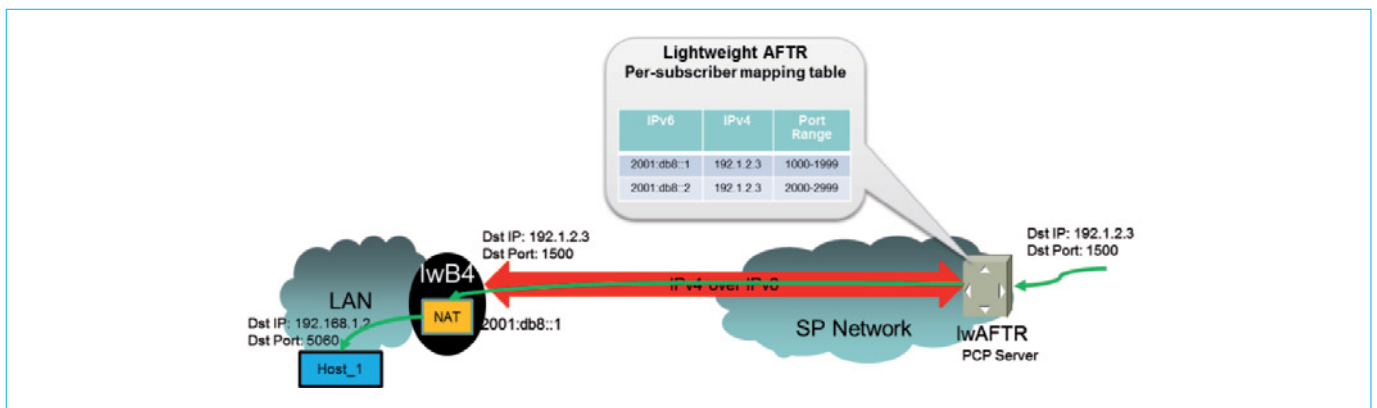


Figure 46 – Architecture « Lightweight IPv4 over IPv6 » – exemple de traitement d'un paquet entrant

On suppose que le terminal « Host_1 » émet un paquet avec comme adresse source « 192.168.1.2 » et numéro de port source « 5060 ». Ce paquet est intercepté par la fonction NAT du CPE qui traduit l'adresse source à « 192.1.2.3 » et choisit un numéro de port de la plage « 1000-1999 ».

Le paquet ainsi transformé est transmis au module lwb4 qui l'encapsule dans un paquet IPv6 ayant comme adresse destination l'adresse IPv6 du lwaFTR. Quand le paquet est reçu par le lwaFTR, il le décapsule pour extraire le paquet IPv4 ; il transmet ensuite le paquet IPv4 vers le prochain saut. À noter que le paquet IPv4 décapsulé n'est pas modifié par le lwaFTR (c'est-à-dire que ce paquet ne fait l'objet d'aucune fonction NAT dans le module lwaFTR).

La figure 46 illustre le traitement d'un paquet IPv4 destiné à l'adresse « 192.1.2.3 ».

Ce paquet est intercepté par la fonction lwaFTR qui consulte sa table d'associations afin de déduire l'adresse IPv6 du CPE auquel l'adresse « 192.1.2.3 » et une plage de ports incluant « 1500 » ont été allouées. Suite à cette opération, l'adresse « 2001:db8::1 » est retournée. Le paquet IPv4 est encapsulé dans un paquet IPv6 ayant comme adresse destination l'adresse « 2001:db8::1 ».

Ce paquet, une fois reçu par le CPE, le décapsule pour extraire le paquet IPv4, vérifie si une entrée NAT est présente dans sa table. Étant donné qu'un paquet a déjà été émis (figure 46), une entrée NAT est présente ; le paquet est traduit selon les informations contenues dans cette entrée : l'adresse destination est remplacée

par « 192.168.1.2 » et le numéro de port destination par « 5060 ». Le paquet ainsi traduit est acheminé vers « Host_1 ».

11. Problèmes que PCP n'est pas censé résoudre

Dans un contexte de partage d'adresses IP à large échelle, des questions récurrentes sont soulevées. Cependant ces questions ne peuvent en aucun cas être résolues par PCP.

Quelques exemples de ces questions sont présentés dans ce paragraphe.

■ « Tous les clients connectés derrière un CGN demandent le numéro de port 8080. Comment faire pour leur allouer ce numéro de port ? » : on ne peut pas satisfaire cette demande car c'est un des effets de bord de tout mécanisme de partage d'adresses à grande échelle [33]. PCP peut satisfaire autant de clients que d'adresses IP distinctes, mais en aucun cas tous les clients.

■ « Je ne peux pas accéder à un service utilisant l'adresse source IP pour des besoins d'identification implicite » : cela est un effet de bord du partage d'adresses à large échelle [33].

[22] identifie une liste de solutions candidates pour résoudre ce problème. Un fournisseur de service d'accès à Internet peut déployer l'une de ces solutions.

■ « Je ne peux pas accéder à un service même si je n’ai pas eu de comportement illicite » : cela peut être dû à un autre utilisateur qui partage la même adresse et qui, lui, a un comportement illicite. La pratique courante consiste à bannir certains utilisateurs sur la base de l’adresse IP source.

Cette pratique a une incidence sur l’ensemble des clients partageant cette même adresse. Afin de limiter l’incidence d’une telle pratique dans un contexte de partage d’adresses, une des solutions décrites dans [22] peut être mise en œuvre. PCP n’est nullement censé résoudre ce problème.

12. Perspectives

De nouvelles applications du protocole PCP ont été promues récemment :

■ ASDN (Application-Enabled Software-Defined Networking [50])

Cette application s’inscrit dans le contexte du Framework SDN [11]. La proposition consiste à utiliser PCP pour interagir avec un PDP (Policy Decision Point) et à installer au préalable des entrées dans le réseau pour accommoder plusieurs cas d’usage comme la bande passante à la demande ou l’envoi de données analytiques. Cette application repose sur une nouvelle extension PCP nommée « FLOWDATA » [68]. Cette option consiste à rapporter à un serveur les caractéristiques d’un flux en plus des besoins de capacité.

■ Envoyer des informations de feedback à un serveur

Cette extension PCP consiste à définir une nouvelle extension PCP nommée « FEEDBACK » [49]. Cette extension a pour but d’améliorer la QoE (Qualité d’expérience) telle que perçue par le client, en fournissant au serveur des informations utiles qui lui permettront d’ajuster le service fourni en s’adaptant aux nouvelles contraintes telles que le changement de terminal ou de réseau d’accès, le niveau de la batterie, etc.

Glossaire	
BIB (Binding Information Base)	Table maintenue par une fonction NAT. Les entrées de cette table représentent l’ensemble des connexions actives à un instant donné et qui font l’objet d’une opération de translation d’adresses éventuellement couplée à une opération de translation de numéros de port
CPE (Customer Premises Equipment)	Équipement installé sur un site client et qui permet l’accès à Internet. La grande majorité des CPE actuellement déployés embarquent en particulier une fonction NAT et une fonction pare-feu qui sont activées par défaut

Glossaire (suite)	
CGN (Carrier-Grade NAT)	Fonction NAT déployée dans un réseau et présentant des performances caractéristiques d’un environnement d’opérateur, c’est-à-dire capable de maintenir plusieurs dizaines de milliers d’entrées dans la table BIB maintenue par la fonction NAT sans dégradation notable de performances. Il existe plusieurs techniques de CGN, dont la technique DS-Lite
DS-Lite (Dual Stack Lite)	Technique mise en œuvre par certains CGN, et qui consiste à ce que les paquets IPv4 utilisant un plan d’adressage privé soient encapsulés dans des paquets IPv6 par l’équipement CPE pour être ensuite acheminés vers l’une ou l’autre des fonctions CGN DS-Lite disponibles dans le réseau. À réception de ces paquets IPv6, le CGN en extrait le paquet IPv4 et procède ensuite à une opération de NAT classique. Les entrées de la table BIB maintenues par un CGN DS-Lite contiennent une information spécifique (typiquement l’adresse IPv6 utilisée par le CPE pour envoyer les paquets IPv4 encapsulés vers le CGN) afin de garantir l’acheminement du trafic retour vers le bon CPE
Flux (flow)	Ensemble d’unités de données de protocoles élémentaires communément appelés paquets qui partagent au moins une caractéristique commune telle que l’adresse destination, l’adresse source ou le même numéro de port caractéristique du protocole de la couche transport. Un flux représente souvent le trafic généré par une application
IETF (Internet Engineering Task Force)	Organisme en charge de la standardisation de la suite de protocoles TCP/IP
Network Address Translator (NAT, translateur d’adresse réseau)	Fonction utilisée pour traduire une adresse IP en une autre adresse IP, par exemple à des fins de rationalisation de l’usage d’adresses publiques IPv4 devenues une ressource rare. La fonction NAT peut également se charger de traduire un numéro de port caractéristique du protocole utilisé dans la couche transport
Pare-feu (firewall)	Fonction de sécurité destinée à protéger l’accès à un site au moyen de l’activation de filtres capables d’autoriser ou pas l’acheminement de trafic à destination du site ou émis par le site
Protocole de Contrôle de Port (PCP, Port Control Protocol)	Protocole de communication reposant sur une architecture client-serveur et destiné à faciliter le contrôle dynamique de ressources réseau, tels que des pare-feu ou des équipements de translation d’adresses

Contrôle dynamique de ressources Internet

Atouts du protocole *PCP*

par **Mohamed BOUCADAIR**

Architecte de réseaux et services IP – France Telecom Orange

et **Christian JACQUENET**

Directeur des programmes stratégiques réseaux IP – France Telecom Orange

Sources bibliographiques

- [1] 3GPP. – *Policy and charging control architecture*. Sept. 2012.
- [2] ABDESSELAM (M.), BOUCADAIR (M.), HAS-NAOUI (A.) et QUEIROZ (J.). – *PCP NAT64 Experiments*. Draft-boucadair-pcp-nat64-experiments (work in progress), sept. 2012.
- [3] ALVESTRAND (H.). – *Overview : real time protocols for browser-based applications*. Draft-ietf-rtcweb-overview (work in progress), oct. 2014.
- [4] ARENDS (R.), AUSTEIN (R.), LARSON (M.), MASSEY (D.) et ROSE (S.). – *Resource records for the DNS security extensions*. RFC 4034, mars 2005.
- [5] ARKKO (J.), EGGERT (L.) et TOWNSLEY (M.). – *Scalable operation of address translators with per-interface bindings*. RFC 6619, juin 2012.
- [6] BAGNULO (M.), MATTHEWS (P.) et VAN BEIJNUM (I.). – *Stateful NAT64 : network address and protocol translation from IPv6 clients to IPv4 servers*. RFC 6146, avr. 2011.
- [7] BAGNULO (M.), SULLIVAN (A.), MATTHEWS (P.) et VAN BEIJNUM (I.). – *DNS64 : DNS extensions for network address translation from IPv6 clients to IPv4 servers*. RFC 6147, avr. 2011.
- [8] BAO (C.), HUITEMA (C.), BAGNULO (M.), BOUCADAIR (M.) et LI (X.). – *IPv6 addressing of IPv4/IPv6 translators*. RFC 6052, oct. 2010.
- [9] BINET (D.), BOUCADAIR (M.), VIZDAL (A.), BYRNE (C.) et CHEN (G.). – *An internet protocol version 6 (IPv6) profile for 3GPP mobile devices*. Draft-ietf-v6ops-mobile-device-profile (work in progress), sept. 2014.
- [10] BOUCADAIR (M.). – *PCP for IPv6-enabled SIP deployments*. Draft-boucadair-pcp-sip-ipv6 (work in progress), oct. 2014.
- [11] BOUCADAIR (M.) et JACQUENET (C.). – *Software-defined networking : a service provider's perspective*. RFC 7149, fév. 2014.
- [12] BOUCADAIR (M.). – *Learn NAT64 PREFIX64s using PCP*. RFC 7225, mai 2014.
- [13] BOUCADAIR (M.). – *PCP deployment models*. Draft-boucadair-pcp-deployment-cases-01 (work in progress), juillet 2014.
- [14] BOUCADAIR (M.), KAPLAN (H.), GILMAN (R.) et VEIKKOLAINEN (S.). – *The session description protocol (SDP) alternate connectivity (ALTC) attribute*. RFC 6947, mai 2013.
- [15] BOUCADAIR (M.), PENNO (R.) et WING (D.). – *DHCP options for the port control protocol (PCP)*. RFC 7291, juillet 2014.
- [16] BOUCADAIR (M.), PENNO (R.) et WING (D.). – *PCP description option*. RFC 7220, mai 2014.
- [17] BOUCADAIR (M.), PENNO (R.) et WING (D.). – *Port control protocol (PCP) proxy function*. Work in progress, fév. 2014.
- [18] BOUCADAIR (M.), PENNO (R.) et WING (D.). – *Some Extensions to port control protocol (PCP)*. Draft-boucadair-pcp-extensions (work in progress), avr. 2012.
- [19] BOUCADAIR (M.), PENNO (R.) et WING (D.). – *Universal plug and play (UPnP) Internet gateway device (IGD) – Port control protocol (PCP) interworking function*. RFC 6970, juil. 2013.
- [20] BOUCADAIR (M.), PENNO (R.), WING (D.), PATIL (P.) et REDDY (T.). – *PCP server selection*. Draft-ietf-pcp-server-selection (work in progress), août 2014.
- [21] BOUCADAIR (M.), REDDY (T.) et WING (D.). – *Using PCP to reveal a host behind NAT*. Draft-boucadair-pcp-nat-reveal (work in progress), mai 2013.
- [22] BOUCADAIR (M.), TOUCH (J.), LEVIS (P.) et PENNO (R.). – *Analysis of solution candidates to reveal a host identifier (HOST_ID) in shared address deployments*. RFC 6967, avr. 2013.
- [23] BOUCADAIR (M.), ZHENG (T.), DENG (X.) et QUEIROZ (J.). – *Behavior of BitTorrent service in PCP-enabled networks with address sharing*. Draft-boucadair-pcp-bittorrent (work in progress), mai 2012.
- [24] BUSH (R.). – *The address plus port (A+P) approach to the IPv4 address shortage*. RFC 6346, août 2011.
- [25] CAMARILLO (G.) et SCHULZKRINNE (H.). – *Early media and ringing tone generation in the session initiation protocol (SIP)*. RFC 3960, déc. 2004.
- [26] CARPENTER (B.), BOUCADAIR (M.), HALPERN (J.), JIANG (S.) et MOORE (K.). – *A generic referral object for Internet entities*. Draft-carpenter-behave-referral-object (work in progress), oct. 2009.
- [27] CHEN (G.), BOUCADAIR (M.), VIZDAL (A.) et THIEBAUT. – *Analysis of port control protocol in mobile network*. Draft-chen-pcp-mobile-deployment, juil. 2013.
- [28] CHESHIRE (S.) et KROCHMAL (M.). – *NAT port mapping protocol (NAT-PMP)*. RFC 6886, avr. 2013.
- [29] CUI (Y.), QIONG (Q.), BOUCADAIR (M.), TSOU (T.), LEE (Y.) et FARRER (I.). – *Lightweight 4over6 : an extension to the DS-Lite architecture*. Draft-ietf-softwire-lw4over6 (work in progress), oct. 2014.
- [30] DROMS (R.). – *Dynamic host configuration protocol*. RFC 2131, mars 1997.
- [31] DROMS (R.), BOUND (J.), VOLZ (B.), LEMON (T.), PERKINS (C.) et CARNEY (M.). – *Dynamic host configuration protocol for IPv6 (DHCPv6)*. RFC 3315, juil. 2003.
- [32] DURAND (A.), DROMS (R.), WOODYATT (J.) et LEE (Y.). – *Dual-stack lite broadband deployments following IPv4 exhaustion*. RFC 6333, août 2011.
- [33] FORD (M.), BOUCADAIR (M.), DURAND (A.), LEVIS (P.) et ROBERTS (P.). – *Issues with IP address sharing*. RFC 6269, juin 2011.
- [34] HANDLEY (M.), JACOBSON (V.) et PERKINS (C.). – *SDP : session description protocol*. RFC 4566, juil. 2006.
- [35] HAUTAKORPI (J.), CAMARILLO (G.), PENFIELD (R.), HAWRYLYSHEN (A.) et BHATIA (M.). – *Requirements from session initiation protocol (SIP) session border control (SBC) deployments*. RFC 5853, avr. 2010.
- [36] HAVERINEN (H.), SIREN (J.) et ERONEN (P.). – *Energy consumption of always-on applications in WCDMA networks*. In Proceedings of the 65th Semi-Annual IEEE Vehicular Technology Conference Dublin, Ireland, avr. 2007.
- [37] HINDEN (R.) et DEERING (S.). – *IP Version 6 addressing architecture*. RFC 4291, fév. 2006.

- [38] HOLMBERG (C.). – *Indication of support for keep-alive*. RFC 6223, avr. 2011.
- [39] HUITEMA (C.). – *Real time control protocol (RTCP) attribute in session description protocol (SDP)*. RFC 3605, oct. 2003.
- [40] HUTTUNEN (A.), SWANDER (B.), VOLPE (V.), DIBURRO (L.) et STENBERG (M.). – *UDP encapsulation of IPsec packets*. RFC 3948, janv. 2005.
- [41] IANA. – *Port control protocol (PCP) parameters*. (2014)
<http://www.iana.org/assignments/pcp-parameters/pcp-parameters.xhtml>
- [42] IANA. – *Protocol numbers*. (2014)
<http://www.iana.org/assignments/protocol-numbers>
- [43] IANA. – *Service name and transport protocol port number registry*. (2014)
<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>
- [44] IVOV (E.), KAPLAN (H.) et WING (D.). – *Latching: hosted NAT traversal (HNT) for media in real-time communication*. RFC 7362, sept. 2014.
- [45] JENNINGS (C.), MAHY (R.) et AUDET (F.). – *Managing client-initiated connections in the session initiation protocol (SIP)*. RFC 5626, oct. 2009.
- [46] KORHONEN (J.) et SAVOLAINEN (T.). – *Analysis of solution proposals for hosts to learn NAT64 prefix*. RFC 7051, nov. 2013.
- [47] LI (X.), BAO (C.) et BAKER (F.). – *IP/ICMP translation algorithm*. RFC 6145, avr. 2011.
- [48] MALAS (D.) et LIVINGOOD (J.). – *Session PEERing for Multimedia INterconnect (SPEERMINT) architecture*. RFC 6406, nov. 2011.
- [49] MOUSTAFA (H.), MOSES (D.) et BOUCADAI (M.). – *PCP extension for signaling feedback information from the end-user application to the application sever and to the network*. Draft-mou-pcp-application-newtwork-feed-back, nov. 2013.
- [50] PENNO (R.), REDDY (T.), BOUCADAI (M.), VINAPAMULA (S.) et WING (D.). – *Application-enabled SDN*. Draft-penno-pcp-asdn, sept. 2013.
- [51] PENNO (R.), REDDY (T.), WING (D.) et BOUCADAI (M.). – *PCP considerations for WebRTC usage*. Draft-penno-rtcweb-pcp (work in progress), mai 2013.
- [52] PERREAULT (S.) éd., YAMAGATA (I.), MIYAKAWA (S.), NAKAGAWA (A.) et ASHIDA (H.). – *Common requirements for carrier-grade NATs (CGNs)*. BCP 127, RFC 6888, avr. 2013.
- [53] POSTEL (J.). – *Discard protocol*. RFC 863, mai 1983.
- [54] POSTEL (J.). – *Transmission control protocol*. STD 7, RFC 793, sept. 1981.
- [55] POSTEL (J.). – *User datagram protocol*. STD 6, RFC 768, août 1980.
- [56] ROSENBERG (J.) et SCHULZRINNE (H.). – *An extension to the session initiation protocol (SIP) for symmetric response routing*. RFC 3581, août 2003.
- [57] ROSENBERG (J.). – *Interactive connectivity establishment (ICE): a protocol for network address translator (NAT) traversal for offer/answer protocols*. RFC 5245, avr. 2010.
- [58] ROSENBERG (J.), SCHULZRINNE (H.), CAMARILLO (G.), JOHNSTON (A.), PETERSON (J.), SPARKS (R.), HANDLEY (M.) et SCHOLLER (E.). – *SIP: Session initiation protocol*. RFC 3261, juin 2002.
- [59] SAVOLAINEN (T.), KORHONEN (J.) et WING (D.). – *Discovery of the IPv6 prefix used for IPv6 address synthesis*. RFC 7050, nov. 2013.
- [60] SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.) et JACOBSON (V.). – *RTP: a transport protocol for real-time applications*. STD 64, RFC 3550, juil. 2003.
- [61] SRISURESH (P.) et EGEVANG (K.). – *Traditional IP network address translator (traditional NAT)*. RFC 3022, janv. 2001.
- [62] SUN (Q.), BOUCADAI (M.), SIVAKUMAR (S.), ZHOU (C.), TSOU (T.) et PERREAULT (S.). – *Port control protocol (PCP) extension for port set allocation*. Draft-ietf-pcp-port-set (work in progress), fév. 2014.
- [63] UPnP Forum. – *WANIPConnection : 1 service template version 1.01*. Nov. 2001
<http://upnp.org/specs/gw/UPnP-gw-WANIP-Connection-v1-Service.pdf>
- [64] UPnP Forum. – *WANIPConnection : 2 service*. Sept. 2010
<http://upnp.org/specs/gw/UPnP-gw-WANIP-Connection-v2-Service.pdf>
- [65] WASSERMAN (M.) et BAKER (F.). – *IPv6-to-IPv6 Network prefix translation*. RFC 6296, juin 2011.
- [66] WING (D.). – *Symmetric RTP/symmetric RTCP*. RFC 4961, juil. 2007.
- [67] WING (D.), CHESHIRE (S.), BOUCADAI (M.), PENNO (R.) et SELKIRK (P.). – *Port control protocol (PCP)*. RFC 6887, avr. 2013.
- [68] WING (D.), PENNO (R.) et REDDY (T.). – *PCP flowdata option*. Draft-wing-pcp-flowdata (work in progress), juil. 2013.
- [69] WOODYATT (J.). – *Recommended simple security capabilities in customer premises equipment (CPE) for providing residential IPv6 internet service*. RFC 6092, janv. 2011.
- [70] YERGEAU (F.). – *UTF-8, a transformation format of ISO 10646*. STD 63, RFC 3629, nov. 2003.
- [71] BOUCADAI (M.). – *Port control protocol (PCP) flow examples*. Draft-bouca-dai-pcp-flow-examples, déc. 2013.

À lire également dans nos bases

- [72] ROS (D.). – *TCP: performance et évolution du protocole*, [TE 7 572], 2006.
- [73] LOYE (S.). – *Le multicast IP: principes et protocoles*, [TE 7 527], 2015.

Gagnez du temps et sécurisez vos projets en utilisant une source actualisée et fiable



RÉDIGÉE ET VALIDÉE
PAR DES EXPERTS




MISE À JOUR
PERMANENTE



100 % COMPATIBLE
SUR TOUS SUPPORTS
NUMÉRIQUES



SERVICES INCLUS
DANS CHAQUE OFFRE

- + de 340 000 utilisateurs chaque mois
- + de 10 000 articles de référence et fiches pratiques
- Des Quiz interactifs pour valider la compréhension 

SERVICES ET OUTILS PRATIQUES



Questions aux experts*

Les meilleurs experts techniques et scientifiques vous répondent



Articles Découverte

La possibilité de consulter des articles en dehors de votre offre



Dictionnaire technique multilingue

45 000 termes en français, anglais, espagnol et allemand



Archives

Technologies anciennes et versions antérieures des articles



Info parution

Recevez par email toutes les nouveautés de vos ressources documentaires

*Questions aux experts est un service réservé aux entreprises, non proposé dans les offres écoles, universités ou pour tout autre organisme de formation.

Les offres Techniques de l'Ingénieur



INNOVATION

- Éco-conception et innovation responsable
- Nanosciences et nanotechnologies
- Innovations technologiques
- Management et ingénierie de l'innovation
- Smart city – Ville intelligente



MATÉRIAUX

- Bois et papiers
- Verres et céramiques
- Textiles
- Corrosion – Vieillessement
- Études et propriétés des métaux
- Mise en forme des métaux et fonderie
- Matériaux fonctionnels. Matériaux biosourcés
- Traitements des métaux
- Élaboration et recyclage des métaux
- Plastiques et composites



MÉCANIQUE

- Frottement, usure et lubrification
- Fonctions et composants mécaniques
- Travail des matériaux – Assemblage
- Machines hydrauliques, aérodynamiques et thermiques
- Fabrication additive – Impression 3D



ENVIRONNEMENT – SÉCURITÉ

- Sécurité et gestion des risques
- Environnement
- Génie écologique
- Technologies de l'eau
- Bruit et vibrations
- Métier : Responsable risque chimique
- Métier : Responsable environnement



ÉNERGIES

- Hydrogène
- Ressources énergétiques et stockage
- Froid industriel
- Physique énergétique
- Thermique industrielle
- Génie nucléaire
- Conversion de l'énergie électrique
- Réseaux électriques et applications



GÉNIE INDUSTRIEL

- Industrie du futur
- Management industriel
- Conception et production
- Logistique
- Métier : Responsable qualité
- Emballages
- Maintenance
- Traçabilité
- Métier : Responsable bureau d'étude / conception



ÉLECTRONIQUE – PHOTONIQUE

- Électronique
- Technologies radars et applications
- Optique – Photonique



TECHNOLOGIES DE L'INFORMATION

- Sécurité des systèmes d'information
- Réseaux Télécommunications
- Le traitement du signal et ses applications
- Technologies logicielles – Architectures des systèmes
- Sécurité des systèmes d'information



AUTOMATIQUE – ROBOTIQUE

- Automatique et ingénierie système
- Robotique



INGÉNIERIE DES TRANSPORTS

- Véhicule et mobilité du futur
- Systèmes aéronautiques et spatiaux
- Systèmes ferroviaires
- Transport fluvial et maritime



MESURES – ANALYSES

- Instrumentation et méthodes de mesure
- Mesures et tests électroniques
- Mesures mécaniques et dimensionnelles
- Qualité et sécurité au laboratoire
- Mesures physiques
- Techniques d'analyse
- Contrôle non destructif



PROCÉDÉS CHIMIE – BIO – AGRO

- Formulation
- Bioprocédés et bioproductions
- Chimie verte
- Opérations unitaires. Génie de la réaction chimique
- Agroalimentaire



SCIENCES FONDAMENTALES

- Mathématiques
- Physique Chimie
- Constantes physico-chimiques
- Caractérisation et propriétés de la matière



BIOMÉDICAL – PHARMA

- Technologies biomédicales
- Médicaments et produits pharmaceutiques



CONSTRUCTION ET TRAVAUX PUBLICS

- Droit et organisation générale de la construction
- La construction responsable
- Les superstructures du bâtiment
- Le second œuvre et l'équipement du bâtiment
- Vieillessement, pathologies et réhabilitation du bâtiment
- Travaux publics et infrastructures
- Mécanique des sols et géotechnique
- Préparer la construction
- L'enveloppe du bâtiment
- Le second œuvre et les lots techniques