



*Université de Caen
UFR des Sciences
Département d'informatique*

*Projet annuel de
Master Informatique
année 2004-2005*

Etude d'architecture VoIP grand-public

Arnaud Houdelette

Mikaël Thibault

Responsables du projet :
Jean Saquet
Jerzy Karczmarczuk

Sommaire

Pourquoi ce projet ?	4
Introduction à la <i>Voip</i>	6
Histoire de la téléphonie.....	7
Exemples d'Architecture Voip	7
Les schémas.....	7
Autres solutions VoIP	9
Skype, sur pc	9
Les Box des FAI	9
Les téléphones VOIP.....	9
1 ^{ère} problématique : localisation et établissement de la session multimédia.	11
Le protocole SIP (Session Initiation Protocol).	11
SIP par l'exemple : un simple appel.	12
Quelques brèves définitions :	13
Les transactions.....	14
Relais des messages et utilité des proxys :	15
Message SIP.....	15
L'en-tête SIP	16
Transport des messages.....	17
Sécurité.....	17
Inconvénients de SIP et adaptations à notre architecture.....	17
Résolution des proxys SIP	17
TCP et TLS	18
2 ^{ème} problématique : Routeurs et partages de connexion : un frein à la VoIP.....	19
Pourquoi je ne peux pas accéder à Internet avec une adresse privée ?	19
NAT statique	20
Quand faire du NAT statique ?.....	21
NAT dynamique	21
Quand faire du NAT dynamique ?	22
Puis-je combiner ces deux méthodes ?.....	23

Inconvénients du Nat pour la VoIP.....	23
Types de NAT :	24
Détection du type de Nat et du couple IP/Port	26
Les différents tests de Stun :	28
NatCheck	30
.....	31
Analyse des impacts sur le trafic SIP selon le type de NAT, et solutions.	32
FullCône :	32
PortRestrictedCône et RestrictedCône :	33
Symétrique :	33
Cas de la communication entre les clients.	33
Cas 1 : FullCône ou SansNat.....	34
Cas 2 : Présence d'un filtre.....	34
Cas 3 : L'un des deux nats ne connaît pas son port public.....	34
Cas 4 : Similaire au précédent, mais.....	34
Cas 5 : Communication indirecte impossible.....	34
Solutions Alternatives	35
UPnP.....	35
Mapping manuel (ou portforwarding).	35
A venir, avenir	36

Pourquoi ce projet ?

La puissance de traitement des processeurs, ainsi que le débit des lignes internet offertes aux particuliers permettent, depuis maintenant quelques années, une transmission quasi-immédiate de la voix d'un point à un autre du globe. De ce fait, on aurait pu espérer que la téléphonie IP point à point se serait dès lors développée pour atteindre le grand public.

Cependant, les solutions aujourd'hui offertes aux particuliers sont pour la plupart directement associées aux fournisseurs d'accès à internet (FAI), liées à un matériel propriétaire et, bien qu'utilisant le réseau IP, ces solutions sont rarement interopérables.

Nous nous sommes donc posé la question suivante : « Est-il possible de mettre en place une architecture permettant des communications IP, qui soit, à la manière de la messagerie électronique (email), indépendante d'un opérateur précis, et donc décentralisée ? »

L'objectif pour nous étant que la configuration faite par l'utilisateur final soit la plus simple possible.

Nous n'avons pas axé notre étude sur le transport de la voix, qui, bien que loin d'être trivial, est un problème aujourd'hui résolu par des codecs (algorithmes de compression/décompression) audio efficaces et des protocoles de transport assurant une communication de qualité.

Nous avons plutôt cherché à résoudre le réel problème de la VoIP, à savoir la mise en relation des interlocuteurs et l'établissement de la communication. La première chose que l'on fait lorsque l'on tente de téléphoner à quelqu'un est de composer le numéro de téléphone identifiant l'abonné à joindre. Dans le cas de la VoIP, un tel numéro n'a pas de signification directe. L'adresse réseau (IP) n'étant pas toujours fixe, l'interlocuteur pouvant être mobile, il est peu concevable d'utiliser cette adresse IP comme identifiant.

Cependant, c'est l'adresse réseau du poste recevant l'appel qui devra être contactée par le poste appelant. Il est donc nécessaire de mettre en place un

service de localisation, afin de faire correspondre à un identifiant l'adresse réseau où joindre le poste de l'appelé.

Ici, un deuxième problème se pose : la multiplication des passerelles résidentielles – faisant en majorité de la translation d'adresse (aussi appelée NAT) – qui masquent les adresses réseau des machines situées derrière elles. Nous allons donc voir en détail le principe de fonctionnement de ces passerelles, leurs différents types, les problèmes qu'elles posent, ainsi que, dans les cas où cela est possible, comment contourner ces problèmes.

Nous avons essayé, tant que possible, de baser notre étude sur des protocoles standards de l'IETF, notamment en ce qui concerne la localisation. Nous verrons par ailleurs pourquoi, dans certains cas, les protocoles existants se prêtent plus ou moins bien au modèle d'infrastructure que nous souhaitons mettre en place.

Introduction à la Voip

Le principe de la voix sur IP est de faire circuler sur Internet, grâce au protocole IP, des paquets de données correspondant à des échantillons de voix numérisée. Reste ensuite à acheminer ces paquets dans le bon ordre et dans un délai raisonnable pour la voix soit correctement restituée. Plusieurs cas de figure peuvent se présenter :

1. Si les deux correspondants possèdent un équipement VoIP relié à Internet, ils pourront communiquer à condition de connaître leurs adresses IP respectives.
2. Si un correspondant utilisant un équipement VoIP connecté à internet souhaite appeler une personne sur son téléphone classique (réseau commuté), il doit passer par un fournisseur de service sur Internet. Ce dernier met en place une passerelle, entre Internet et le RTC (réseau téléphonique commuté), qui gèrera les échanges de données. Dans le sens inverse, le correspondant peut contacter la passerelle de son téléphone.
3. Si les deux correspondants possèdent un téléphone commuté, ils devront chacun passer par une passerelle. Ensuite, les deux passerelles communiquent entre elles par un réseau de type Internet. Ce cas est transparent pour les deux utilisateurs, mais cette solution est souvent adoptée par les opérateurs téléphoniques afin de réduire les coûts.

Dans notre projet nous nous préoccupons du 1^{er} cas, où les correspondants possèdent tout deux un équipement VoIP. Nous ne nous préoccupons pas de la passerelle entre Internet et le RTC.

Histoire de la téléphonie

Du premier télégraphe de Chappe en 1790 au RTC actuel, l'histoire des communications a connu de grands moments et de grandes avancées dû à l'ingéniosité de certains et aux progrès technologiques et électroniques. Nous retiendrons quelques grandes dates tel que :

1837 Premier télégraphe électrique inventé par Samuel Morse

1889 Almon B. Strowger (USA) invente le premier « sélecteur » automatique et donne ainsi naissance à la commutation téléphonique automatique

1938 Alec Reeves (Français) dépose le brevet des futurs systèmes à modulation par impulsion et codage (MIC) : quantification et échantillonnage du signal à intervalles réguliers, puis codage sous forme binaire.

1962 Les premiers systèmes de transmission multiplex de type MIC apparaissent aux Etats-Unis ils permettent une liaison à 24 voies entre centraux téléphoniques, à la même époque en France on installe des MIC à 32 voies.

1970 Un nouveau pas est franchi dans le domaine de la commutation électronique avec la mise en service en France, par le CNET, des premiers centraux téléphoniques publics en commutation électronique temporelle.

1979 Lancement du minitel en France

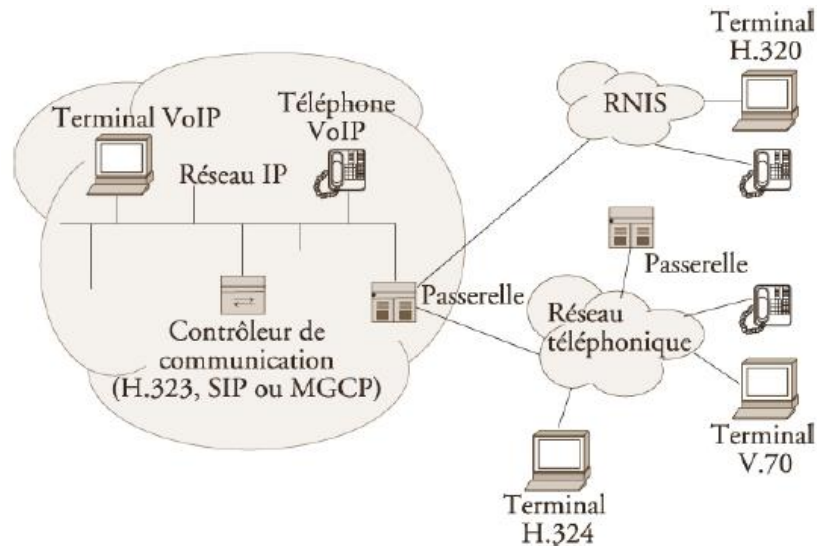
1987 Le RNIS est mis en service en France.

1990 De nouveaux concepts apparaissent tel que la commutation temporelle asynchrone (ATM) et la hiérarchie numérique synchrone.

Exemples d'Architecture Voip

Les schémas

Voici un schéma général de l'utilisation de la Voip en entreprise :



La VoIP a pas encore de standard unique. En effet, chaque constructeur apporte ses normes et ses fonctionnalités à ses solutions. Il existe tout de même des références en la matière. Les trois principales sont H.323, SIP et MGCP/MEGACO.

Le schéma ci-dessus, décrit de façon générale la topologie d'un réseau de téléphonie IP

On retrouve les éléments communs suivants :

- Le routeur : Il permet d'aiguiller les données et le routage des paquets entre deux réseaux.
- La passerelle : il s'agit d'une interface entre le réseau commuté et le réseau IP.
- Le PABX : C'est le commutateur du réseau téléphonique classique. Il permet de faire le lien entre la passerelle ou le routeur et le réseau RTC. Une mise à jour du PABX est aussi nécessaire. Si tout le réseau devient IP, il n'y a plus besoin de ce matériel.
- Les Terminaux : Des PC ou des équipements VoIP.

Tout cela constitue une architecture relativement complexe, qui demande à être administrée par des spécialistes réseau et/ou télécom. Il est donc peu concevable d'utiliser ce genre d'architecture chez un particulier.

Autres solutions VoIP

Skype, sur pc

Skype est un logiciel de partage de fichiers utilisant la technologie P2P Il est donc différent de notre projet car nous n'utilisons pas la technologie P2P.

Une fois inscrit, chaque utilisateur peut discuter avec un autre internaute utilisant lui aussi ce logiciel. La qualité audio est excellente dès lors que l'internaute dispose d'une connexion haut débit (type ADSL). Skype diffère de notre application car il est centralisé, alors que nous souhaitons faire un logiciel décentralisé.

Les Box des FAI

Depuis le 18 mars 2004, Free a décidé de proposer à l'ensemble des nouveaux inscrits son modem propriétaire Freebox qui permet, en plus de se connecter au web en haut débit, de téléphoner gratuitement en utilisant le réseau IP. Suivent Cegetel qui en juin 2004 annonce la sortie de sa set-up box, Neuf Telecom avec sa Neuf Box lui emboite le pas et enfin Wanadoo sort la LiveBox en proposant de la VoIP dans ses forfaits Internet. Le client branche un téléphone classique sur la « Box », et peut appeler. Cependant, une LiveBox ne peut appeler directement une Freebox ou inversement sans passer par le réseau commuté, leurs réseaux VoIP étant indépendants.

Les téléphones VOIP

Pour utiliser la VoIP, il existe deux autres moyens :

Utiliser un adaptateur :



Ou un téléphone VoIP :



Il existe sur le marché plusieurs téléphones VoIP. Il ne faut pas oublier, qu'en plus d'un téléphone VoIP, il est nécessaire de posséder une connexion à internet.

1^{ère} problématique : localisation et établissement de la session multimédia.

Telle que nous la souhaitons, la localisation ne doit pas se faire de manière centralisée (id est : elle ne doit pas dépendre d'un unique serveur, mais de plusieurs, gérés par différents organismes ou sociétés). A l'image du mail, nous avons envisagé un protocole basé sur le DNS (système de nom de domaines) où chaque serveur gère un domaine. Le DNS permet de localiser le serveur gérant un domaine donné.

Nous nous sommes rapidement aperçus qu'il existait déjà un protocole basé sur ce principe : SIP.

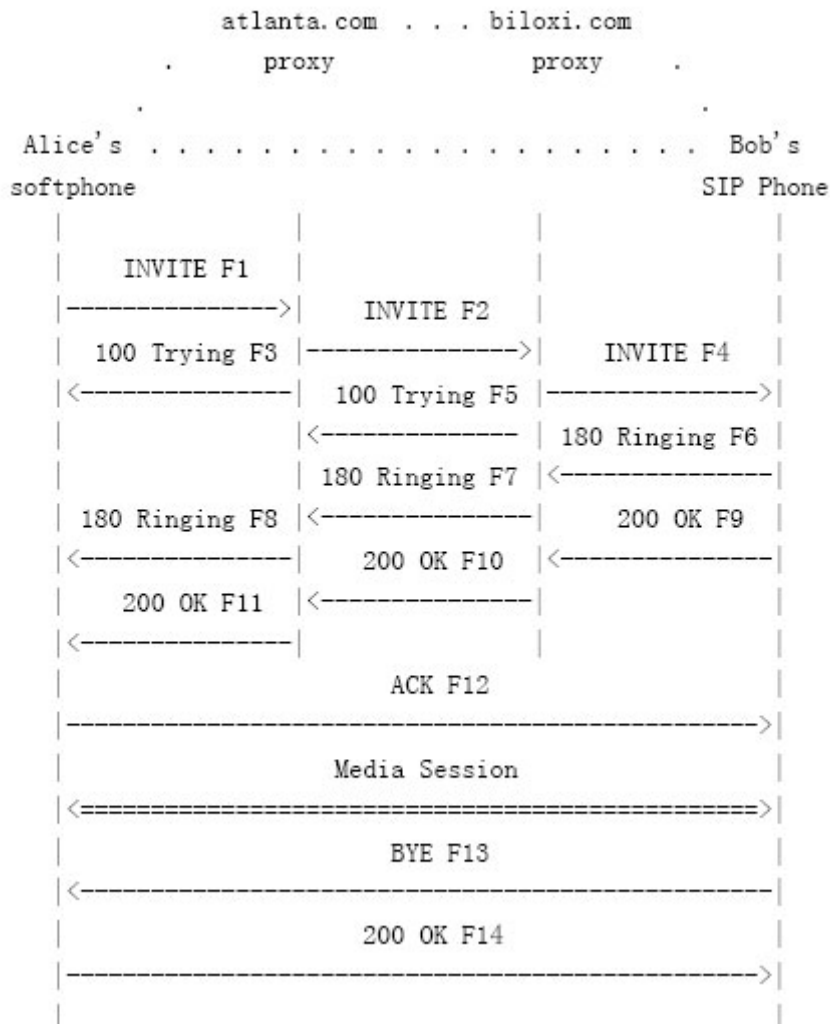
Le protocole SIP (Session Initiation Protocol).

SIP est un protocole de la couche application. Son but est de transmettre des messages entre deux interlocuteurs afin d'établir, et de fermer, des sessions multimédias. Ces sessions peuvent être des appels téléphoniques (communication audio bidirectionnelle).

Les invitations SIP permettent de créer les sessions et permettent à leurs participants de se mettre d'accord sur les paramètres de celles-ci.

SIP permet aussi l'enregistrement sur les serveurs ou "proxys SIP" permettant la localisation des clients.

SIP par l'exemple : un simple appel.



Alice (alice@atlanta.com) appelle bob (bob@biloxi.org) :

1. Le téléphone d'Alice signale régulièrement sa présence au serveur SIP d'atlanta.com. De même, le téléphone de bob signale aussi sa présence au serveur SIP de biloxi.org.

2. Alice téléphone à Bob, en entrant son adresse SIP (bob@biloxi.org).
3. Le téléphone d'Alice va donc envoyer au serveur SIP d'atlanta.com, un message d'invitation à destination de bob@biloxi.org. Ce message contient les paramètres possibles pour la future communication.
4. Recevant le message, le serveur SIP d'atlanta.com va rechercher à qui faire suivre le message, et, à l'aide de DNS, déterminer qu'il doit être transmis au serveur SIP de biloxy.org.
5. Le serveur SIP biloxy.org reçoit le message, adressé à un utilisateur qu'il gère. Il va donc rechercher la dernière position signalée par le téléphone de bob, et passer le message au téléphone de bob.
6. Le téléphone de Bob reçoit le message, se met à sonner et envoie une réponse d'attente, via les deux serveurs.
7. Le téléphone d'Alice reçoit la réponse temporaire, et joue la tonalité de sonnerie.
8. Bob décroche son téléphone, qui envoie une deuxième réponse, toujours via les deux serveurs, indiquant l'acceptation de l'appel. Cette réponse contient les paramètres acceptés par le téléphone de bob.
9. Le téléphone d'Alice reçoit la réponse contenant les paramètres de communication, puis établit cette communication.

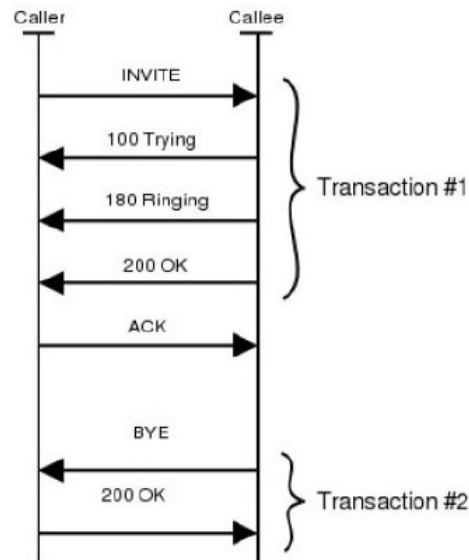
Quelques brèves définitions :

User Agents ou UA : SIP nomme les téléphones "User Agents", ce sont les parties qui établissent la communication.

Proxys SIP : assurent le relai des messages jusqu'à leur destination.

URI SIP : De la forme sip:utilisateur@domaine , il s'agit de l'identifiant d'un utilisateur. C'est cette URI qui sera utilisée pour contacter le téléphone de l'interlocuteur.

Les transactions.



Les transactions sont à la base du protocole SIP. En effet, chaque message SIP fait partie intégrante d'une transaction.

Une transaction commence par un message de requête, envoyé par l'"User Agent Client" (UAC), envoyée (éventuellement via un ou plusieurs proxys SIP) à l'"User Agent Server" (UAS). Si la requête est toujours unique, l'UAS peut envoyer plusieurs réponses (dont des réponses intermédiaires). La transaction se termine par la réception d'une réponse définitive.

La transaction SIP définit donc deux rôles, UAC et UAS, mais il est important de noter que ces rôles peuvent être inversés pour les différentes transactions effectuées au cours d'un appel téléphonique.

SIP définit les transactions suivantes :

- **INVITE**, permettant l'établissement d'une session. C'est la transaction principale du protocole.
- **CANCEL**, permettant d'annuler une transaction en cours.
- **BYE**, terminant une session.
- **OPTION**, permettant la découverte des paramètres de l'UAS, sans établir de session.
- **REGISTER**, permettant l'inscription de l'UAC dans l'annuaire du domaine.

Relais des messages et utilité des proxys :

Les proxys ont essentiellement une fonction de relais de message. Le principe est le suivant :

Un proxy SIP reçoit un message à destination de sip:utilisateur@domaine.com

- Le domaine est géré par le proxy SIP.
 - o Le proxy trouve l'adresse correspondante à "utilisateur".
- Le domaine n'est pas géré par le proxy SIP.
 - o Le proxy recherche le proxy SIP de "domaine.com"

Cependant si le chemin de la requête SIP est calculé, la réponse DOIT passer par les mêmes proxys que la requête. A cette fin, chaque proxy ajoute sa propre adresse dans le message SIP. Ces adresses sont incluses dans la réponse, et sont dépilées jusqu'à l'émetteur de la requête.

Le proxy SIP peut, selon les cas, avoir d'autres rôles, et être configurés pour agir différemment.

- Redirection d'appels en cas d'absence.
- Redirection vers une messagerie vocale.
- Multiplication ("forking") des appels (fait sonner plusieurs postes).
- ...

Message SIP

Le format des messages SIP est hautement similaire aux requêtes et réponses http. Le message SIP est séparé en deux parties : les en-têtes, et le corps du message.

L'en-tête contient les informations nécessaires à l'acheminement du message et a son traitement par les proxys. La nature du corps du message est précisée dans l'en-tête par un type MIME, qui est généralement un message SDP (Session Definition Protocol) indiquant les capacités multimédia, ainsi que les paramètres (IPs et ports) nécessaires à l'établissement d'une future session multimédia.

L'en-tête SIP

Seule la première ligne du message distingue une requête d'une réponse SIP. Une requête est introduite par une méthode, suivie de l'URI éventuellement paramétrée et de la version du protocole (SIP/2.0).

La réponse commence par la version du protocole, suivie d'un code réponse (ou d'erreur) suivie de sa description.

Le reste de l'en-tête est constitué de champs, parmi lesquels on distinguera :

- **Via** : Ce champ permet un retour de la réponse par le même chemin emprunté par la requête. L'émetteur de la requête, ainsi que chaque proxy empile son adresse dans ce champ lors du transfert d'une requête. L'UAS répondant à la requête copie ce champ dans la réponse, et chaque proxy dépile l'adresse du saut suivant.
- **Max-Forwards** : Le message SIP étant "routé" à travers de multiples proxys, une mauvaise configuration de l'un d'eux pourrait amener un message à tourner indéfiniment entre les proxys. Pour éviter cela, le champ Max-Forward est initialisé à l'émission et décrémenté à chaque saut. Si la valeur tombe à zéro, le proxy répond à la requête par une erreur.
- **To** : Adresse (URI) SIP du destinataire demandée originalement par l'émetteur de l'appel. Elle peut différer de l'URI indiquée dans la première ligne de l'en-tête dans les cas où l'appel aura été redirigé par un proxy.
- **From** : Adresse de l'émetteur. Elle peut aussi être différente de l'émetteur réel.
- **Call-ID** : identifie les transactions d'un même appel.
- **CSeq** : Numéro de séquence du message. En effet, les messages SIP pouvant être transmis via UDP, qui n'est pas un protocole de transport fiable, un numéro de séquence est nécessaire pour la détection des retransmissions.
- **Contact** : Ce champ permet à une des parties de préciser une URI SIP temporaire, mais plus directe, afin de se passer des proxys SIP pour les messages à venir.

Transport des messages

Les messages SIP sont généralement transportés via UDP. La version 2 du protocole introduit le support de TCP, et de TLS (connexion TCP chiffrée). Le protocole SIP définit le port 5060 comme port TCP et UDP standard, mais ce port n'est pas imposé, et pourra être précisé dans l'URI SIP.

Sécurité

SIP prévoit deux niveaux de sécurité. L'identification par http-Digest et le cryptage TLS.

Identification : Effectuée à l'aide d'Http-Digest, similaire à l'authentification basique http, elle permet à un client de fournir des informations d'identifications afin d'accéder à certains services ou proxys.

Authentification et Cryptage : Uniquement avec le transport TCP, TLS permet de crypter et de signer, assurant à la fois la confidentialité, l'intégrité, et une assurance sur la provenance des messages. Cependant, une architecture TLS nécessite la mise en place de certificats.

Inconvénients de SIP et adaptations à notre architecture

Résolution des proxys SIP

Comme nous l'avons vu, le rôle majeur de SIP et de ses proxys est de router les messages jusqu'à leur destination. Pour ce faire, le message doit d'abord être acheminé vers le proxy gérant le domaine du destinataire.

Lors de la mise en place d'un proxy SIP. Si celui-ci gère les utilisateurs d'un domaine, il doit, pour que les autres proxys lui relaient les messages qui leurs sont destinés, être référencé d'une manière ou d'une autre.

Tel que défini dans la RFC3263, SIP utilise le système de nom de domaines DNS, et plus particulièrement des entrées DNS de type SRV (RFC2782) et NAPTR (RFC2915). Cependant, de nombreuses implémentations des serveurs et clients DNS ne supportent pas ce type d'enregistrements. Il est donc peu pratique d'utiliser ces enregistrements pour référencer les proxys SIP.

La version 1 de SIP, plus simpliste sur ce sujet, permet l'utilisation d'un champ classique A (Nom → IP) ou CNAME (Alias) spécial : `_sip.domaine.com`.

Afin de simplifier la mise en place de SIP sur un domaine où le serveur DNS ne supporte pas les enregistrements SRV et NAPTR, nous pensons important de conserver ce comportement, et de rechercher de tels enregistrements (A ou CNAME) lors de la résolution.

TCP et TLS

La version 2 de SIP impose TLS. Or TLS, basé sur une architecture de certificats est relativement lourd à mettre en place. En effet, chaque proxy, ainsi que chaque client, doit être muni d'un certificat valide, afin d'assurer l'identité des interlocuteurs lors des communications. Si cette précaution est nécessaire, et relativement aisée à mettre en place dans un environnement contrôlé (administré par des techniciens spécialisés), elle devient relativement onéreuse et compliquée pour proposer un service grand public.

De plus le transport TCP (imposé par TLS) amène deux inconvénients majeurs :

- comme nous allons le voir par la suite, pose des problèmes avec la translation d'adresse des passerelles résidentielles.
- TCP est orienté connexion. Chaque connexion consommant des ressources, un proxy SIP gérant de nombreux téléphones devra maintenir ces connexions.

De ce fait, nous limiterons notre architecture à l'utilisation d'UDP.

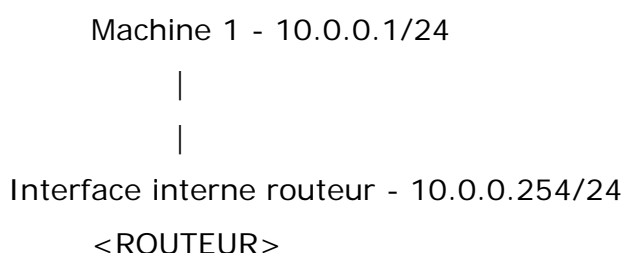
2^{ème} problématique : Routeurs et partages de connexion : un frein à la VoIP.

Aujourd'hui, les fournisseurs d'accès ne fournissent donc plus qu'une IP aux particuliers, qui doivent, s'ils veulent connecter plusieurs PC ou autres équipements à internet via leur réseau local (filaire ou sans-fil), utiliser une passerelle Nat. On peut distinguer deux solutions proposées : le partage de connexion, où l'un des PC est relié à internet et au réseau local, et s'occupe de la translation, et l'utilisation d'un matériel dédié (routeur) qui s'occupera seul de cette tâche (les fournisseurs d'accès proposent de plus en plus leurs propres routeurs). Dans les deux cas, le partage de connexion ou le routeur utilise le mécanisme de translation d'adresses.

Le mécanisme de translation d'adresses (en anglais Network Address Translation noté NAT) a été mis au point afin de répondre à la pénurie d'adresses IP avec le protocole IPv4. En effet, en adressage IPv4 le nombre d'adresses IP routables (donc uniques sur la planète) n'est pas suffisant pour permettre à toutes les machines nécessitant d'être connectées à Internet de l'être. Le principe du NAT consiste donc à utiliser une adresse IP routable (ou un nombre limité d'adresses IP) pour connecter l'ensemble des machines du réseau en réalisant, au niveau de la passerelle de connexion à internet, une translation (littéralement "une traduction") entre l'adresse interne (non routable) de la machine souhaitant se connecter et l'adresse IP de la passerelle.

Pourquoi je ne peux pas accéder à Internet avec une adresse privée ?

On prend l'exemple suivant:



Interface externe routeur - 193.22.35.42/24



La machine 1 veut envoyer un paquet sur Internet, vers www.unicaen.fr, par exemple. Donc dans l'en-tête IP, l'adresse en destination est celle de www.unicaen.fr, et en source c'est 10.0.0.1. Si jamais il n'y avait pas de translation d'adresse, le paquet arriverait bien à la machine www.unicaen.fr, mais celle-ci essaierait de renvoyer sa réponse à 10.0.0.1 qui n'est pas une adresse routée sur Internet !! (Elle fait partie d'une classe d'adresses réservées pour les réseaux privés). Et notre machine 1 n'obtiendrait jamais de réponse... Ainsi, une machine ayant une adresse privée ne pourra pas recevoir de réponse à ses requêtes sans un mécanisme supplémentaire.

NAT statique

Le principe du NAT statique consiste à associer une adresse IP publique à une adresse IP privée interne au réseau. Le routeur (ou plus exactement la passerelle) permet donc d'associer à une adresse IP privée (par exemple 192.168.0.1) une adresse IP publique routable sur Internet et de faire la traduction, dans un sens comme dans l'autre, en modifiant l'adresse dans le paquet IP.

Le NAT statique permet ainsi de connecter des machines du réseau interne à Internet de manière transparente mais ne résout pas le problème de la pénurie d'adresse dans la mesure où n adresses IP routables sont nécessaires pour connecter n machines du réseau interne. Dans la suite du rapport lorsqu'on dit qu'un ordinateur est derrière est Nat, cela veut dire que l'ordinateur utilise un Nat pour se connecter à Internet.

Quand faire du NAT statique ?

Nous avons vu que la NAT statique permettait de rendre disponible une machine sur Internet, mais qu'il fallait par contre une adresse IP pour que ce serveur soit joignable. Il est donc utile d'utiliser la NAT statique quand vous voulez rendre une application disponible sur Internet, comme un serveur web, mail ou un serveur FTP.

NAT dynamique

Le NAT dynamique permet de partager une adresse IP routable (ou un nombre réduit d'adresses IP routables) entre plusieurs machines en adressage privé. Ainsi toutes les machines du réseau interne possèdent virtuellement, vu de l'extérieur, la même adresse IP. C'est la raison pour laquelle le terme de "mascarade IP" (en anglais IP masquerading) est parfois utilisé pour désigner le mécanisme de NAT.

Afin de pouvoir multiplexer les différentes adresses IP sur une ou plusieurs adresses IP routables le NAT dynamique utilise la translation de port (PAT - Port Address Translation), c'est-à-dire l'affectation d'un port source différent à chaque requête de telle manière à pouvoir maintenir une correspondance entre les requêtes provenant du réseau interne et les réponses des machines sur Internet, toutes adressées à l'adresse IP du routeur.

Quand faire du NAT dynamique ?

La NAT dynamique permet d'une part de donner un accès à Internet à des machines possédant des adresses privées, et d'autre part d'apporter un petit plus en terme de sécurité. Elle est donc utile pour économiser les adresses IP, donner un accès à Internet à des machines qui n'ont pas besoin d'être joignables de l'extérieur (comme la plupart des utilisateurs). D'autre part, même quand on possède assez d'adresses IP, il est souvent préférable de faire de la NAT dynamique pour rendre les machines injoignables directement de l'extérieur. Par exemple, pour un usage personnel de partage de l'ADSL ou du câble, on utilise souvent la NAT dynamique pour partager son accès, étant donné que les machines n'ont pas besoin d'être jointes de l'extérieur.

Puis-je combiner ces deux méthodes ?

Oui, et c'est même souvent la meilleure solution lorsque l'on a à la fois des machines offrant un service, et d'autres qui n'ont besoin que de se connecter à Internet. Ainsi, on économisera les adresses IP grâce aux machines NATées dynamiquement, et on utilisera exactement le bon nombre d'adresses IP publiques dont on a besoin. Il est donc très intéressant de combiner ces deux méthodes.

Pour la suite de notre rapport nous considérerons nos NAT comme un combiné des deux méthodes.

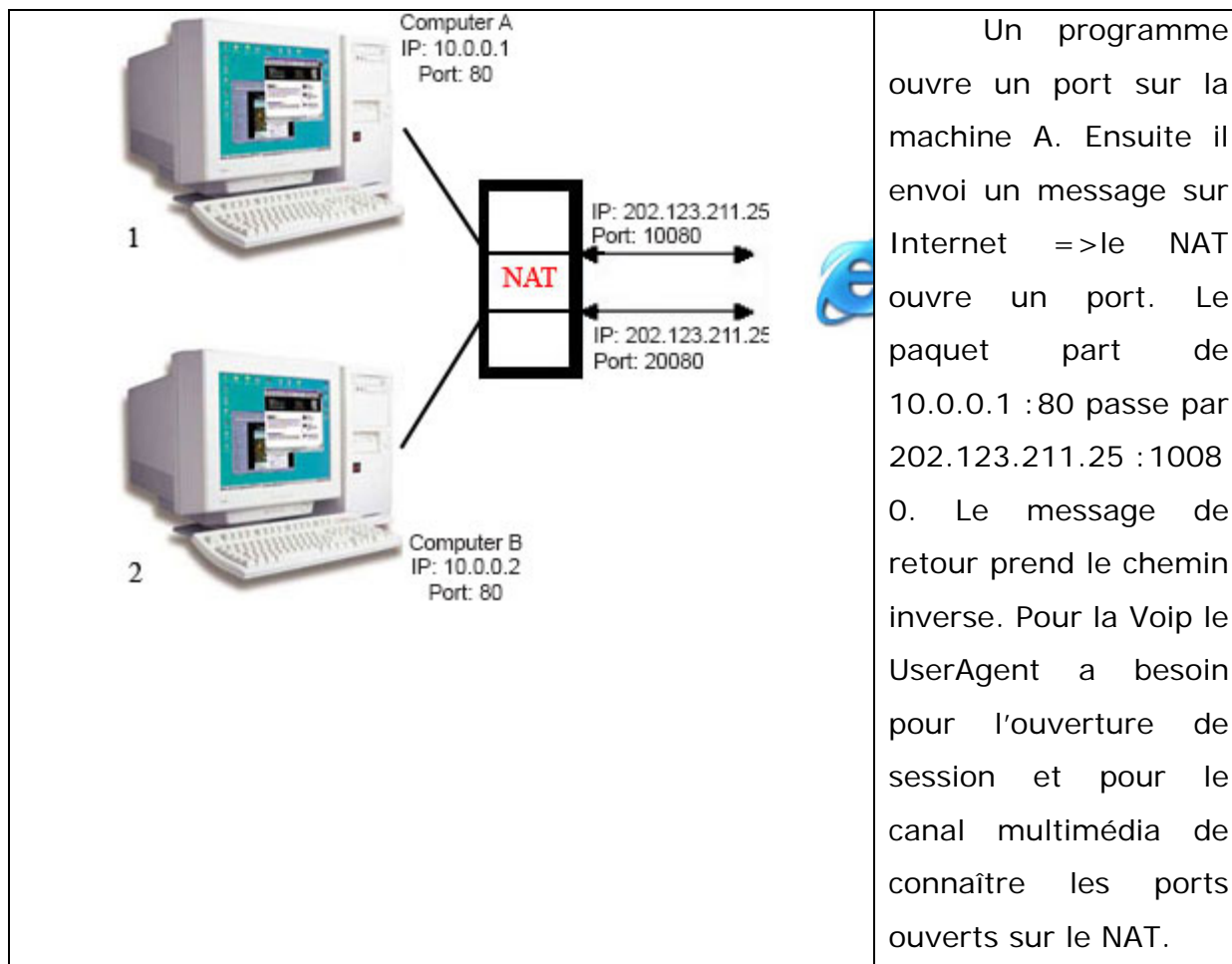
Inconvénients du Nat pour la VoIP

Comme nous l'avons vu, le Nat ne pose aucun problème lorsqu'il s'agit d'établir une communication vers Internet, lorsque cette communication est à l'initiative d'une machine derrière le Nat.

Cependant, un appel n'est pas forcément attendu. Le téléphone doit donc pouvoir être joint à tout moment, et pour ça, doit être en mesure de fournir, lors de l'enregistrement, une adresse et un port où le joindre.

De la même manière, pour qu'une communication multimédia puisse s'établir entre deux interlocuteurs, ils doivent tout deux être en mesure de se fournir mutuellement une adresse et un port "multimédia".

Le premier objectif de notre recherche a été de trouver une solution au problème du NAT.



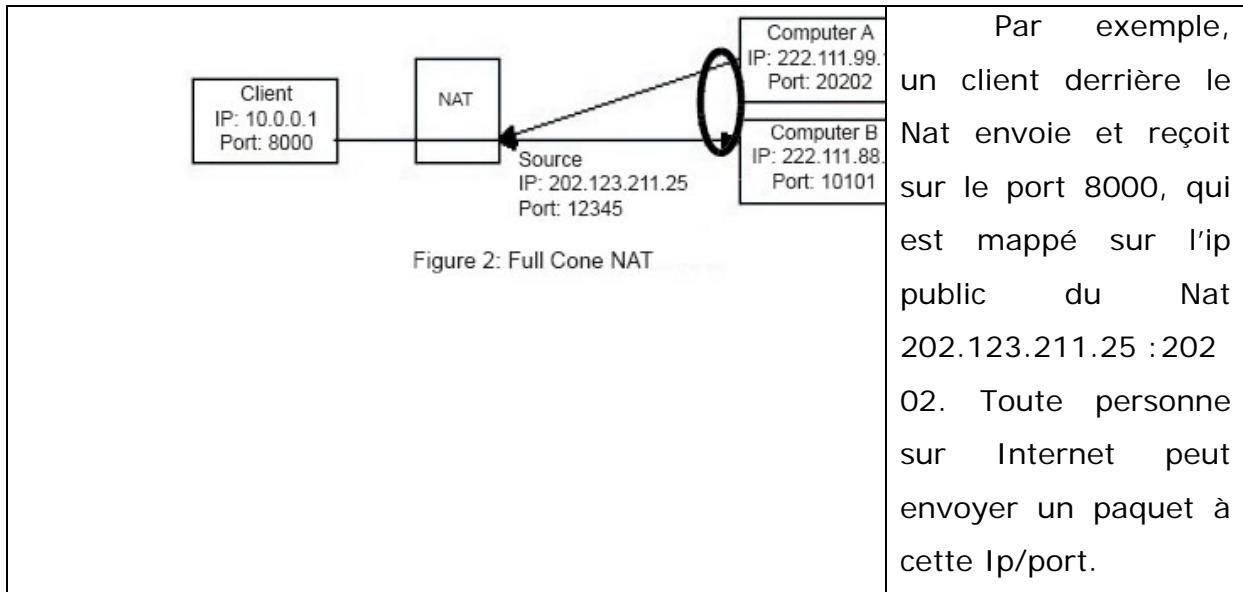
Types de NAT :

UDP étant un protocole sans connexion, La réaction des Nats face au trafic UDP est variée. Nous avons donc besoin de distinguer différents types de Nats en fonction de leur réaction au trafic UDP, et de leur manière de gérer les mappings correspondants.

1. Full Cone:

Dans le cas du Full Cone, un mapping est établi entre le Nat et le UserAgent. N'importe quelle machine d'Internet peut contacter le UserAgent à travers le Nat dès lors qu'il connaît le couple Ip/port publique du Nat.

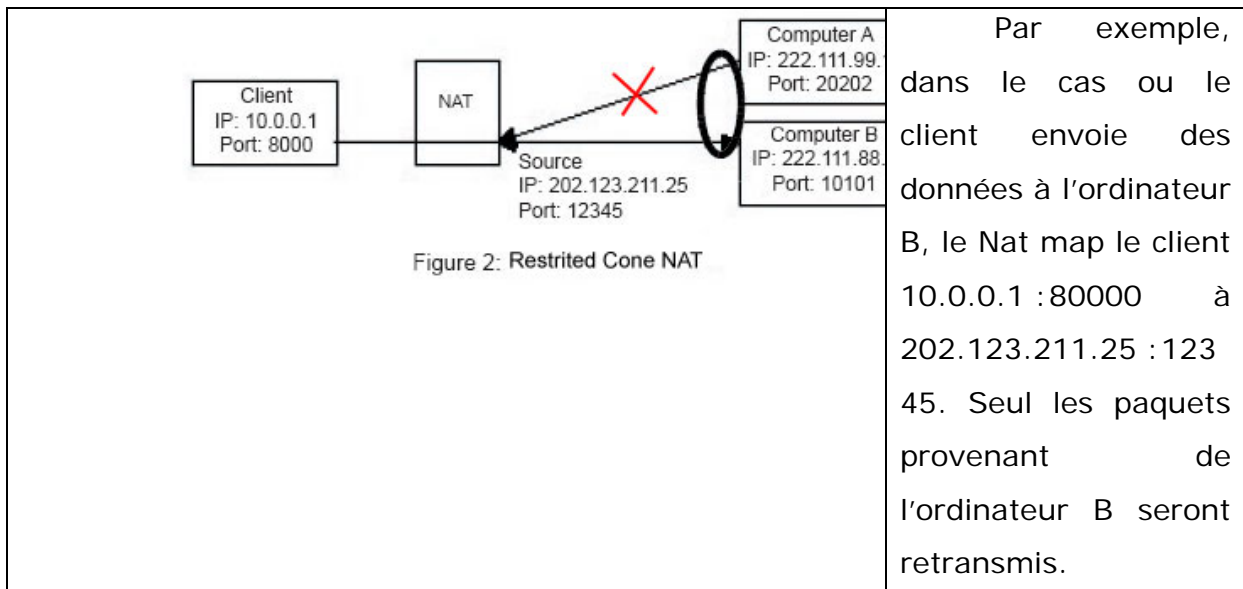
Ex :



2. Restricted Cone

Dans le cas du Restricted Cone Nat, le couple Ip/port externe est ouvert pour une adresse spécifique. Cependant, pour une seconde communication utilisant le même port privé, le Nat utilisera le même port public.

Ex :



3. Port Restricted Cone

Le port restricted cone Nat est Presque identique au Restricted cone Nat, mais dans se cas le Nat va bloquer tous les paquets s'ils ne viennent pas d'un couple ip/port que le client n'a pas précédemment contacté.

4. Symmetric

Le dernier type de Nat est différent des 3 premiers, en effet, le mapping spécifique entre un couple Ip/port interne et l'adresse Ip publique du Nat dépend de la destination : adresse IP où le paquet est envoyé.

Par exemple, si le client envoie de 10.0.0.1:8000 à l'ordinateur B, il peut être mappé à 202.123.211.25:12345, alors que si le client envoie avec le même port a une IP différente, il va être mappé différemment.

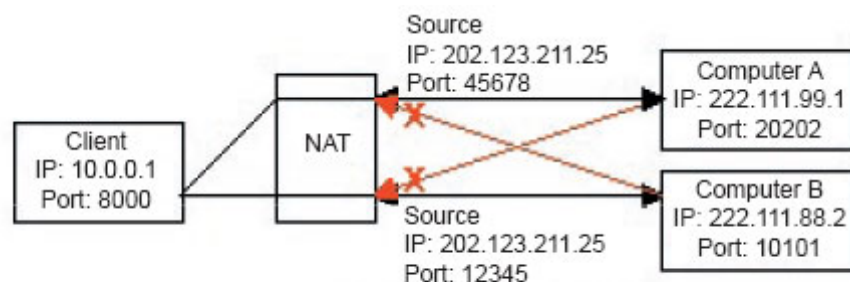


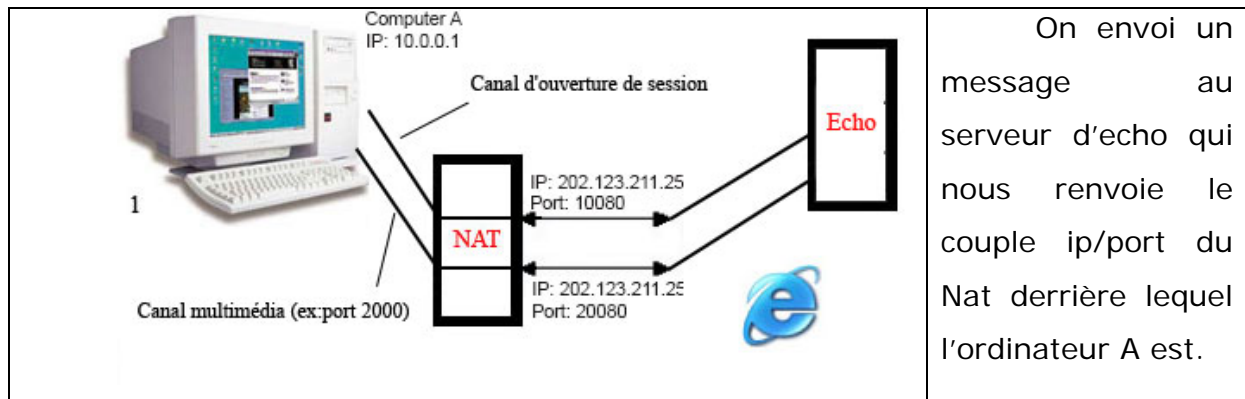
Figure 3: Symmetric NAT

L'ordinateur B ne peut répondre qu'à travers son mapping, de même pour l'ordinateur A. Si chacun veut envoyer un paquet à un autre ip/port, ce paquet sera ignoré par le Nat. Comme dans le cas du restricted cone, la paire Ip/port externe est seulement ouverte une fois.

Détection du type de Nat et du couple IP/Port

Afin de pouvoir s'enregistrer auprès de son proxy SIP, et afin de pouvoir être contacté par son correspondant un "point de contact", le UserAgent VoIP à besoin de fournir un couple IP/Port public. Il doit donc être en mesure de prendre connaissance de ce port publique.

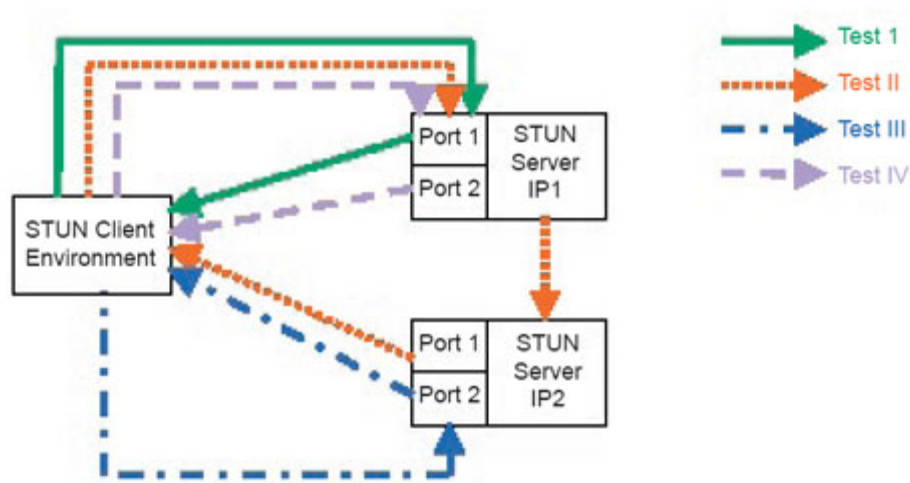
Pour cela, nous avons imaginé un serveur d'écho :



On envoie un message au serveur d'écho qui nous renvoie le couple ip/port du Nat derrière lequel l'ordinateur A est.

Au moment d'implémenter notre serveur d'écho, nous nous sommes aperçu qu'une solution à notre problème venait d'être apportée : STUN. C'est un protocole simple qui propose des méthodes permettant de détecter pour un port UDP, la présence, et si présence il y a, le type de NAT. STUN permet, dans les cas où il est fixe, de déterminer le mapping UDP, et donc au UserAgent de connaître son couple IP/Port public.

Les différents tests de Stun :



1. Le test 1 est exécuté.

Si pas de réponse, alors le UserAgent sait qu'il est derrière un firewall qui bloque UDP.

2. Si une réponse est reçue, l'adresse IP dans le champ MAPPED-ADDRESS du message Stun est testé sur se que le serveur pense comme être son adresse ip.
3. Si l'adresse est bonne, exécution du test II

~ Si pas de réponse, alors le UserAgent est derrière un Nat symétrique UDP, le Firewall va seulement laisser passer UDP, dans le cas ou il a préalablement envoyé un message au destinataire.

~ Si le UserAgent reçoit une réponse, alors il est sur une connexion Internet ouverte et pas bloquée.

4. Si l'adresse IP du 2. n'est pas la même, le test II est exécuté.

- ~ Si le UserAgent reçoit la réponse, alors il se trouve derrière un Nat Full Cone

5. S'il ne reçoit pas de réponse, le UserAgent exécute le test III et test si l'adresse retournée dans le champ MAPPED-ADDRESS du message Stun égale celle retournée au test I.

- ~ Si les deux adresses ne sont pas les mêmes, alors le UserAgent est derrière un Nat Symétric.

6. Si les deux adresses IP sont les mêmes, alors on exécute le test IV

- ~ Si on reçoit une réponse, le UserAgent est derrière un Nat Restricted.

- ~ Si on ne reçoit rien, le UserAgent est derrière un Nat Restricted Port

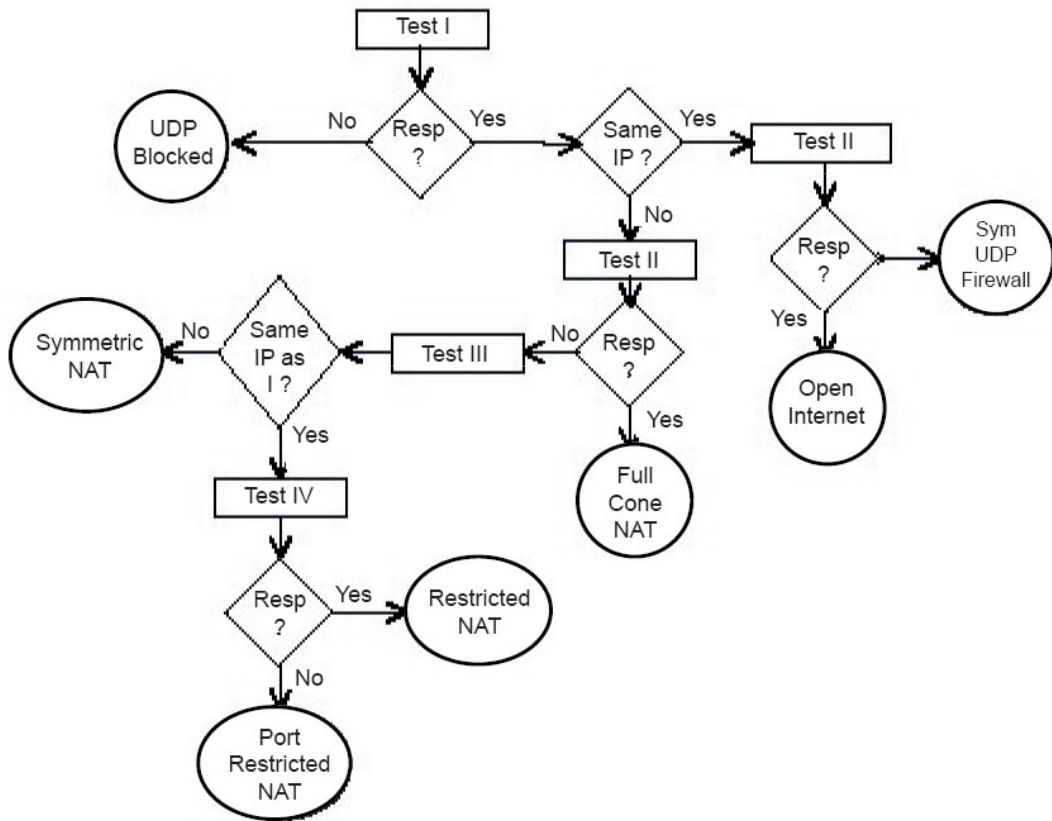


Fig 5 : Les différents tests

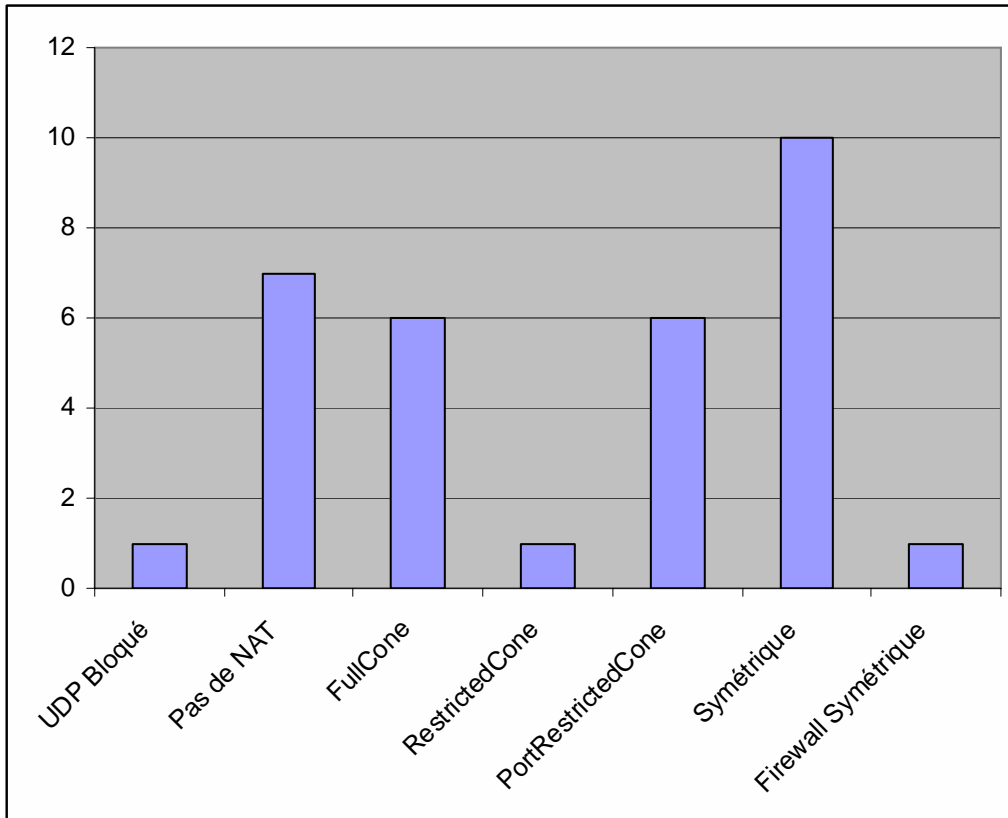
NatCheck

Afin de se faire une idée de la répartition des types de Nats, nous avons créé un programme dont le but est de déterminer automatiquement si il est derrière un NAT ou non, et si oui, de trouver de quel type de NAT il s'agit.

Le programme, rendu disponible à cette adresse : <http://crashmachine.tzim.net/tmp/testnat.zip>, a été publié sur différents forums et testé par une trentaine de personnes.

```
d:\TEMP\Rar$EX10.890\Test.exe
Detection du type de NAT
Port UDP local 1420
STUN Server : larry.gloo.net <66.7.238.210:3478>
Test 1 : 66.7.238.210:3478 reporte le port UDP 82.229.57.51:61428
Serveur 2 : 66.7.238.213:3479
Test 2 : 66.7.238.213:3479 reporte le port UDP 82.229.57.51:61428
Mapping ouvert -> FullCone
Type de NAT : FullCone
```

```
C:\DOCU...~1\MIKAL~1\LOCALS~1\Temp\Rar$EX00.265\Test.exe
Detection du type de NAT
Port UDP local 1916
STUN Server : larry.gloo.net <66.7.238.210:3478>
Test 1 : 66.7.238.210:3478 reporte le port UDP 82.233.220.83:1916
Serveur 2 : 66.7.238.213:3479
Test 2 : Pas de réponse.
Test 3 : 66.7.238.213:3479 reporte le port UDP 82.233.220.83:17440
Mapping Symétrique -> NAT Symétrique.
Type de NAT : SymetricNat
```



Résultats des tests

Le nombre de tests n'est pas assez conséquent pour tirer de conclusions sur la répartition réelle, mais nous a permis de constater que les Nats Symétriques, qui comme nous allons le voir par la suite, sont les plus problématiques, étaient beaucoup plus nombreux que nous le pensions au départ.

Au passage, nous avons noté, lorsque le modèle nous a été fourni par le tester, de quels de types étaient les routeurs testés :

FullCone	<ul style="list-style-type: none">• Partage de connexion à internet windows• IPTables dans sa configuration la plus courante.
----------	--

RestrictedCone	<ul style="list-style-type: none">• Linksys BEFSR41• routeur 3com (?)• Routeur SMC (?)
PortRestrictedCone	<ul style="list-style-type: none">• freebox en mode routeur• ZyWALL70 ver 3.64(WM.0)• netgear dg834g• Linksys WRT54G• Firewall Cisco Pix
Symétrique	<ul style="list-style-type: none">• LiveBox• ZyWALL70 ver 3.63(WM.2)• Netopia 3366ENT• Freebox en mode routeur• D-Link 707p sur une box club-internet• Netopia Cyman 3346

Certains résultats peuvent paraître contradictoires. En effet, entre deux versions du firmware (logiciel interne) d'un même routeur, les résultats varient. La configuration joue aussi (notamment lorsque le routeur comporte un firewall intégré).

Analyse des impacts sur le trafic SIP selon le type de NAT, et solutions.

Comme nous l'avons vu plus haut, le client SIP doit pouvoir, à tout moment, être contacté par UDP. Le port et l'IP doivent pouvoir être détectés pour être envoyés au proxy

FullCône :

Les nats fullcône permettant une détection du port par le client, et n'appliquant pas de filtres particuliers une fois le mapping effectué, permettent une utilisation totalement transparente des ports UDP. Le client UDP pourra simplement indiquer le couple IP/Port publique détecté lors de son enregistrement, et l'utiliser dans le champ contact, permettant aux autres clients de le contacter directement sur ce port.

PortRestrictedCône et RestrictedCône :

Dans ces cas, le port peut toujours être détecté, mais le NAT (ou le firewall associé) limitant les paquets entrants aux IPs déjà contactées, le port SIP du client ne pourra être contacté que par le proxy auprès duquel il s'enregistre. (Le mapping s'effectuant lors de l'enregistrement, à l'initiative du client).

Symétrique :

Ici, nous ne pouvons détecter le port public. De ce fait, le client ne peut même pas enregistrer son propre port public, ne le connaissant pas.

Le protocole SIP permet au proxy de passer outre en détectant lui-même le port SIP, sur indication du client. Dès lors, le proxy peut recontacter le client tant que le mapping est actif.

Comme nous l'avons vu, les mappings ne sont pas éternels. Il conviendra donc de ré effectuer régulièrement (60s) l'enregistrement, afin de maintenir le mapping ouvert.

Cas de la communication entre les clients.

La communication entre les clients consiste en un flux de paquets UDP transitant directement d'un client à l'autre. Nous avons analysé l'impact des Nats en fonctions de leurs types de chacun des cotés de cette communication :

	Sans NAT	FullCône	RestrictedCône	Port - RestrictedCône	Symétrique
Sans NAT	Cas 1	Cas 1	Cas 2	Cas 2	Cas 3
FullCône	Cas 1	Cas 1	Cas 2	Cas 2	Cas 3
RestrictedCône	Cas 2	Cas 2	Cas 2	Cas 2	Cas 4
PortRestrictedCône	Cas 2	Cas 2	Cas 2	Cas 2	Cas 5
Symétrique	Cas 3	Cas 3	Cas 4	Cas 5	Cas 5

Cas 1 : FullCône ou SansNat.

Dans ce cas, chaque UserAgent connaît au préalable le port public multimédia de son interlocuteur, les mappings ayant été initiés et détectés par STUN. Il n'y a donc aucun obstacle à la communication.

Cas 2 : Présence d'un filtre.

Dans ce cas, l'un des Nats, ou les deux, n'acceptera pas les paquets arrivant de l'autre interlocuteur tant que l'UserAgent derrière ce Nat n'aura pas lui-même envoyé un premier paquet vers cet interlocuteur. Il y'a donc risque d'une perte des premiers paquets, si l'un des deux clients initie la communication avant l'autre. Cependant, les protocoles de transport temps réels supportent relativement bien cette perte.

Cas 3 : L'un des deux nats ne connaît pas son port public.

Dans ce cas, c'est au UserAgent qui peut fournir son port public d'attendre l'initiative de la communication par l'autre UserAgent. Il suffira alors de répondre l'adresse d'où est venu cette initiation.

Cas 4 : Similaire au précédent, mais...

Ici, l'UserAgent qui connaît son port public est derrière un nat qui filtre les paquets UDP. L'UserAgent peut alors tenter d'ouvrir le filtre en envoyant un paquet UDP à l'IP de l'interlocuteur (qui doit être connue), en utilisant un port aléatoire, avant d'attendre l'initiative de la communication. Il s'agit d'une solution un peu extrême, peu évidente à implémenter, mais qui permet d'écarter un autre cas de disfonctionnement.

Cas 5 : Communication indirecte impossible.

Soit l'un des deux UserAgent connaît son port public, mais ne peut désactiver le filtre du nat, ne connaissant pas celui de son interlocuteur, soit les

deux UserAgents n'ont pas connaissance de leur port public. Dans les deux cas, la communication directe est impossible.

Solutions Alternatives

Ces deux alternatives ont l'inconvénient de dépendre des possibilités du routeur utilisé et d'une configuration qui peut dépasser les capacités du grand public.

UPnP

UPnP (Universal plug&play) est une technologie permettant aux applications tournant derrière un Nat de contacter celui-ci afin d'effectuer eux même un mapping, leur permettant par la même de connaître leur port public, sans même utiliser de réflexion. Cependant, cette technologie est loin d'être présente sur tout les nats, et sur la majorité des routeurs, est désactivée par défaut (les failles de sécurité des premières implémentations ayant rendu les constructeurs frileux).

Mapping manuel (ou portforwarding).

Ici, il s'agit tout simplement de fixer le port privé de l'UserAgent, et d'indiquer au nat d'utiliser un mapping constant sur ce port.

A venir, avenir ...

Avec le débit augmentant, les connexions devenant permanentes, de nouvelles applications basées sur la communication se mettent à la portée du grand public. Paradoxalement, pour profiter de cette connexion sur toutes les machines de la maison, il est nécessaire d'avoir recours à la translation d'adresses, qui du coup limite grandement les possibilités, et deviens un casse tête pour les utilisateurs et les développeurs d'applications.

Le salut pourrait venir d'IPv6, où, le nombre d'adresses étant quasi-illimité, le Nat perd sa raison d'être, et chaque machine peut, dès lors, directement être contactée.

La mise en place d'IPv6 risque cependant d'être longue, le manque d'applications supportant IPv6 ne pousse pas les fournisseurs d'accès à investir dans des infrastructures IPv6, et le peu d'offres IPv6 par les fournisseurs d'accès n'encourageant pas non plus les développeurs à supporter l'IPv6.

Remerciements :

Nous tenons à remercier Mr Jean Saquet pour son soutien, son aide et le temps qu'il a consacré tout au long de notre projet.

Bibliographie

- **VOIP-Info**, un Wiki sur le thème de la VoIP
<http://www.voip-info.org>
- **Understanding IPv6**, de Joseph Davies, Microsoft Press
- **Encyclopedia of Networking** de Mitch & Ingrid Tuloch, Microsoft Press
- **La voix sur IP** d'Olivier Hersent, David Gurle, Jean-Pierre Petit, aux éditions Dunod
- **RFC 2543** : *SIP : Session Initiation Protocol (obsolète, version 1)*
<http://zvon.org/tmRFC/RFC2543/Output/index.html>
- **RFC 3261** : *SIP : Session Initiation Protocol (standard Tracks)*
<http://zvon.org/tmRFC/RFC3261/Output/index.html>
- **RFC 3263** : *Session Initiation Protocol (SIP): Locating SIP Servers*
<http://zvon.org/tmRFC/RFC3263/Output/index.html>
- **RFC 3489** : STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)
<http://zvon.org/tmRFC/RFC3489/Output/index.html>
- **Nat Traversal in SIP** :
[http://www.sipcenter.com/sip.nsf/html/WEBB5YN5GE/\\$FILE/SIPNAT traversal.pdf](http://www.sipcenter.com/sip.nsf/html/WEBB5YN5GE/$FILE/SIPNAT%20traversal.pdf)
- **Cafzone.net**, forum, résultats des tests de natcheck
<http://www.cafzone.net/ipb/lofiversion/index.php/t25279.html>