# R316-ROM
# Ingénierie de la téléphonie sur IP
# Travaux pratiques

Sébastien Peychet et Sami Evangelista

IUT de Villetaneuse

Département Réseaux et Télécommunications

2023–2024

http://www.lipn.univ-paris13.fr/~evangelista/cours/R316-ROM

## Table des matières

# TP 1 — Introduction à Asterisk

Ce TP est une introduction à asterisk, un IPBX logiciel, gratuit et open source, sous Linux. Asterisk peut remplir la plupart des fonctions d'un PABX matériel (serveur SIP, registrar, redirection d'appels, ...). On peut donc l'utiliser en remplacement de ceux-ci dans des environnements de téléphonie basés à 100% sur IP. Dans des environnements mixtes (téléphonie sur IP + téléphonie analogique/numérique) on pourra l'utiliser en combinaison de PABX matériels — asterisk ne pouvant pas être utilisé comme passerelle VoIP.

Le TP est à faire en binôme sur 2 PC sur lesquels on aura chargé une image Debian (version 8 ou supérieure). Chaque binôme utilisera également un téléphone sur IP T42U.

## Exercice 1 — Travail préliminaire

Un des deux postes hébergera le service asterisk. Sur l'autre on installera linphone (un UA SIP).

**I 1.1** Installez le paquet asterisk sur un des 2 PC. Démarrez ensuite le service asterisk puis la CLI asterisk :

```
# asterisk -rv
```

La CLI (Command Line Interface) est un outil permettant d'interagir avec le serveur asterisk. Nous l'utiliserons à la fois pour faire certains tests et pour recharger les fichiers de configuration d'asterisk.

**I 1.2** Installez le paquet linphone sur l'autre PC.

Nous allons maintenant connecter tous les équipements. Notez l'interface qui vous connecte au réseau de l'IUT (avec une IP de la forme 192.168.X.Y). C'est l'autre interface que nous utiliserons dans ce TP. On suppose dans la suite qu'elle se nomme eth0.

**I 1.3** Connectez les équipements (les 2 PC et le téléphone) au switch par leur interface eth0.

**I 1.4** Sur les deux PC, attribuez à l'interface eth0 une IP de la forme 10.N.X.Y (N étant votre numéro de groupe et X et Y deux octets de votre choix) dans le réseau 10.0.0.0/8.

**I 1.5** Vérifiez que les deux PC peuvent s'échanger des messages ping.

## Exercice 2 — Fonctionnement de base d'asterisk

Le service asterisk a deux fichiers de configuration principaux : `sip.conf` et `extensions.conf` ; tous les deux dans le répertoire `/etc/asterisk`. Avant d'éditer ces deux fichiers, il est important de comprendre comment ils sont utilisés par asterisk. Deux exemples simples sont visibles sur la Figure 1.

Chaque téléphone doit avoir un nom. Ce nom peut être un numéro, une chaîne de caractère, ou une combinaison des deux. Dans ce TP, nous utiliserons des noms symboliques. Vous choisirez ces noms comme vous voulez. Dans nos exemples nous les nommerons **Aladdin** et **Genie**.

**Le fichier `sip.conf`** Il définit les noms des téléphones et est divisé en *sections* : une section `general`, puis une section par téléphone. Chaque section débute par son nom entre crochets. Dans une section, chaque ligne précise un *paramètre*.

La section `general` permet de donner des valeurs par défaut aux paramètres qui ne seraient pas définis dans la section propre à un téléphone. Sur la figure, évidemment, le fichier est incomplet : il n'y a rien dans la section `general`, certains paramètres n'ont pas de valeur, ...

Le paramètre `context` est très important car il permet de définir des groupes de téléphones ; chaque groupe aura son propre plan de numérotation, ce qui permet par exemple de donner des droits différents. Tous les téléphones ayant un même contexte font partie du même groupe. Ici, Aladdin et Genie ont le même contexte : `local`.

**Le fichier `extensions.conf`** Il définit le *plan de numérotation* (dialplan), c'est-à-dire ce que doit faire asterisk quand un appel lui parvient. Il est également divisé en sections : une section `general`, une section `globals`, puis une section par contexte (donc par groupe d'utilisateurs). Pour chaque contexte, donc ici dans la section `local`, on définit le plan de numérotation des utilisateurs en écrivant des *extensions*. Nous verrons la syntaxe d'une extension un peu plus loin. Sur l'exemple de la figure, une seule extension a été définie : `exten => 102,1,Dial(SIP/Genie)`. Elle indique que le numéro 102 fera sonner le téléphone `Genie`. On a donc le plan de numérotation suivant :

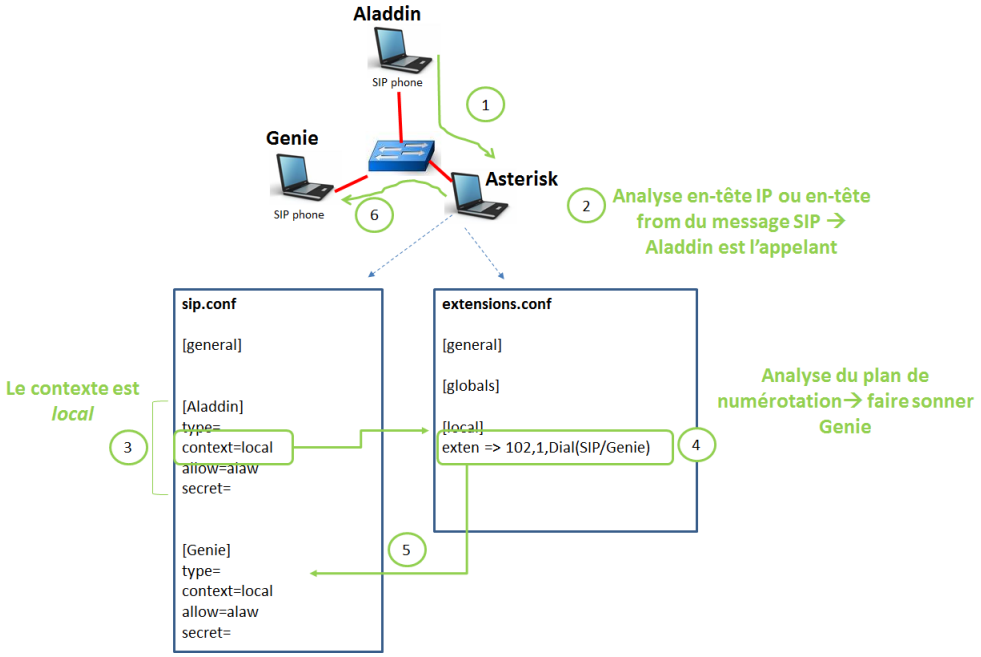



FIGURE 1 – Exemples de fichiers de configuration et déroulement d'un appel

| Nom d'extension (numéro composé) | Application | Paramètre |
|---|---|---|
| 102 | Faire sonner (Dial) | technologie SIP, téléphone Genie (SIP/Genie) |

Pour bien comprendre, voyons ce qui se passe quand Aladdin compose le 102.

1. Une requête SIP INVITE parvient au serveur asterisk, avec comme Request-URI 102@IP_asterisk.
2. Pour connaître le plan de numérotation à utiliser, asterisk doit identifier l'appelant. Deux possibilités :
   — soit il utilise l'en-tête From du message SIP. C'est le cas si le `type` de l'appelant est `friend`;
   — soit il utilise l'adresse IP source du message SIP; et en déduit l'URI de l'appelant par consultation du registrar. C'est le cas si le `type` de l'appelant est `peer`.
3. Dans `sip.conf`, le serveur repère le contexte d'Aladdin : ici `local` (et d'autres paramètres comme le codec).
4. En consultant le plan de numérotation de ce contexte (dans `extensions.conf`) il voit qu'il doit faire sonner Genie.
5. Dans `sip.conf`, il consulte les paramètres de Genie.
6. Il fait suivre la requête SIP INVITE à Genie.

Nous allons maintenant configurer le service asterisk pour que les téléphones puissent passer des appels.

**I 2.1** Renommez les fichiers `sip.conf` et `extensions.conf` en `sip.conf.bak` et `extensions.conf.bak` afin de partir de fichiers de configuration vierges et de sauvegarder les fichiers initiaux.

**I 2.2** Éditez le fichier `sip.conf` pour avoir le contenu ci-dessous :

```
[general]
bindport=5060       ; port sur lequel sont attendues les requêtes SIP
bindaddr=0.0.0.0    ; IP de l'interface sur laquelle le serveur va écouter (0.0.0.0 = toutes)
language=fr
dtmfmode=auto
disallow=all        ; tous les codecs interdits
allow=alaw          ; codecs autorisés, par ordre de préférence, ici G711 loi A
allow=gsm

[Aladdin]           ; nom du téléphone (à changer)
context=local
secret=0000         ; mot de passe pour l'enregistrement (à changer)
type=peer           ; c'est l'adresse IP source qui permettra d'identifier l'appelant
host=dynamic        ; adresse IP dynamique => obtenue par consultation du registrar (asterisk)
directmedia=yes     ; le flux RTP n'a pas besoin de passer par le serveur asterisk
directrtpsetup=yes  ; le flux RTP est envoyé sur les paramètres fournis dans le premier INVITE
```

**I 2.3** Modifiez ce fichier pour créer un deuxième téléphone.

**I 2.4** Depuis la CLI, rechargez le fichier `sip.conf` puis affichez les téléphones connus :

```
sip reload
sip show peers
```

Passons maintenant au fichier `extensions.conf`

**I 2.5** Éditez le fichier `extensions.conf` pour obtenir le contenu ci-dessous :

```
[general]
static=yes        ; ces deux lignes donnent la possibilité de
writeprotect=no   ; sauvegarder le plan depuis la CLI asterisk

[globals]

[local]
exten => N02,1,Dial(SIP/Genie)   ; changez N par le numéro du binôme
                                 ; et Genie par le nom du téléphone
```

Une extension définit une ou plusieurs actions, appelées applications, classées par priorité, à effectuer en cas d'appel. La syntaxe est la suivante :

`exten => nom_extension,numéro_de_priorité,application()`

Le nom de l'extension est le numéro appelé. Il peut être une suite de chiffres ou une chaîne de caractères. Pour des parcs de téléphones physiques comme les T42U, il faut choisir des numéros, typiquement sur 3 ou 4 chiffres (car il est difficile de numéroter une chaîne de caractères !) ; pour des softphones (logiciels de ToIP), on peut choisir des numéros ou des chaînes de caractère. L'`application` la plus commune est `Dial()`, qui permet de faire sonner un téléphone, avec en paramètre le protocole (SIP pour nous) et, séparé par un /, le nom du téléphone à faire sonner (ou la Request-URI à utiliser si le téléphone n'est pas dans `sip.conf`).

**I 2.6** Ajoutez une extension pour traiter un appel vers le second téléphone.

**I 2.7** Depuis la CLI, rechargez le fichier `extensions.conf` :

```
dialplan reload
```

Dans la suite du TP il faudra exécuter dans la CLI une des deux commandes ci-dessous en cas de modification d'un fichier de configuration : `sip reload` après modification de `sip.conf` ; et `dialplan reload` après modification de `extensions.conf`. Ceci évite de redémarrer le service asterisk.

## Exercice 3 — Mise en service des téléphones

Nous allons maintenant configurer les deux téléphones (linphone et le T42U) pour qu'ils puissent s'enregistrer auprès d'asterisk et s'appeler entre eux. Commençons par l'application linphone.

**I 3.1** Lancez linphone.

**I 3.2** Sélectionnez *Utiliser un compte SIP*.

**I 3.3** Entrez les paramètres suivants :
- Nom d'utilisateur : nom de la section dans le fichier `sip.conf`
- Domaine SIP : IP du serveur asterisk
- Mot de passe : secret dans le fichier `sip.conf`

puis validez.

**I 3.4** Depuis la CLI, vérifiez que l'enregistrement s'est bien déroulé : `sip show peers`.

Passons maintenant au T42U. Nous allons le redémarrer en *mode usine* pour revenir à une configuration vierge.

**I 3.5** Mettez le téléphone sous tension.

**I 3.6** Redémarrez le téléphone en mode usine : laissez le bouton *OK* appuyé pendant quelques secondes puis confirmez le redémarrage.

Une fois le téléphone redémarré, nous allons d'abord configurer statiquement ses paramètres IP.

**I 3.7** Allez dans le menu *3 Settings* → *2 Advanced settings* (le mot de passe est `admin`) → *2 Network* → *1 WAN Port* → *2 IPv4* → *2 Static IPv4 Client*.

**I 3.8** Choisissez une IP de la forme 10.N.X.Y dans le réseau 10.0.0.0/8, entrez le masque et choisissez une IP quelconque du réseau 10.0.0.0/8 pour la passerelle par défaut. (Il n'y a pas de route vers l'extérieur dans notre réseau mais il est impossible de sauvegarder les paramètres si aucune passerelle par défaut n'a été saisie.)

**I 3.9** Sauvegardez ces paramètres : *Save*.

**I 3.10** Vérifiez que les deux PC peuvent envoyer des pings vers le téléphone.

L'étape suivante est de configurer le compte SIP, un peu comme nous l'avions fait avec linphone.

**I 3.11** Allez dans le menu *3 Settings → 2 Advanced settings → 1 Accounts → 1*.
**I 3.12** Passez *Active line* à *Enabled*.
**I 3.13** Saisissez les paramètres suivants :
- *Display Name* : choisissez le nom qui s'affichera quand le correspondant recevra un appel.
- *Register Name* : nom de la section dans le fichier `sip.conf`
- *User Name* : nom de la section dans le fichier `sip.conf`
- *Password* : secret dans le fichier `sip.conf`
- *SIP Server 1* : IP du serveur asterisk

**I 3.14** Sauvegardez ces paramètres : *Save*.

Une fois ces paramètres saisis, le *Display Name* devrait s'afficher sur l'écran du téléphone. Il nous reste à vérifier que le téléphone s'est bien enregistré auprès d'asterisk et que les appels passent entre les deux téléphones.

**I 3.15** Depuis la CLI, vérifiez que l'enregistrement s'est bien déroulé : `sip show peers`.
**I 3.16** Passez des appels entre les deux téléphones.

## Exercice 4 — Éléments supplémentaires dans le plan de numérotation

**I 4.1** Ajoutez les lignes suivantes au plan de numérotation (`hello-world` est un fichier de son inclus dans asterisk) :

```
exten => 1200,1,Answer()
exten => 1200,2,Playback(hello-world)
exten => 1200,3,Hangup()
```

**I 4.2** Appelez le numéro 1200 depuis un des deux téléphones.

**Q 4.1** Que se passe-t-il ? Concluez sur le rôle des applications `Answer()`, `Playback()` et `Hangup()`.

La numérotation des priorités étant source d'erreur, asterisk permet de remplacer la priorité par `n`. Dans ce cas, `n` correspond à la priorité de la ligne précédente incrémentée de 1. Autre amélioration : si, pour plusieurs lignes, l'extension est la même — par exemple 1200 ci-dessus, — on peut, à partir de la seconde ligne, utiliser `same =>` à la place de `exten => 1200,`. Ceci améliore la lisibilité.

**I 4.3** Testez ces deux améliorations en modifiant les lignes du point I 4.1 puis vérifiez que le résultat est le même.

On peut aussi écrire une extension correspondant non pas à un seul numéro mais à un ensemble de numéros. Pour cela le nom de l'extension doit commencer par un underscore (_). On peut ensuite utiliser :
- `X` : n'importe quel digit de 0 à 9
- `Z` : n'importe quel digit de 1 à 9
- `N` : n'importe quel digit de 2 à 9
- `.` : un (ou plusieurs) caractères, peu importe le(s)quel(s)

**I 4.4** En utilisant cette possibilité, ajoutez des extensions permettant de jouer le son `invalid` si un des utilisateurs appelle un numéro commençant par 0.

## Exercice 5 — Trunk SIP entre serveurs asterisk authentifiés

Dans cet exercice, nous allons mettre en place des trunks SIP entre les serveurs de la salle. Un trunk SIP est une liaison entre deux serveurs SIP. Le trunk peut fonctionner avec ou sans authentification des serveurs. Ici, nous considérerons des trunks avec authentification. C'est typiquement le cas dans des entreprises réparties sur plusieurs sites avec des serveurs asterisk acceptant uniquement le trafic SIP venant des autres serveurs asterisk de l'entreprise.

Les binômes seront répartis deux à deux. Un trunk sera créé entre chaque couple de binômes. Chaque binôme aura trois choses à faire :
- ajouter le serveur distant dans le fichier `sip.conf`, en lui donnant un contexte spécifique et un mot de passe notamment ;
- dans `extensions.conf`, dans la section correspondant à ce contexte spécifique, attribuer un plan de numérotation pour gérer les appels entrants (le même que dans la section `local`) ;

— toujours dans `extensions.conf`, ajouter une extension au plan de numérotation du contexte `local` pour gérer les appels sortants vers le serveur distant.
Par exemple, si le binôme 2 souhaite ajouter le binôme 4, dans `sip.conf`, il faudra ajouter :

```
[serveur4]
context=appel_serveur4  ; contexte spécifique pour les appels venant du binôme 4
type=peer
host=10.4.0.X           ; IP du serveur asterisk du binôme 4
secret=toto             ; clé secrète que devra connaître le serveur distant
```

Dans `extensions.conf`, il faut ensuite ajouter la section `appel_serveur4` avec les extensions adéquates.
Enfin, dans la section `local`, il faut ajouter les extensions permettant de rediriger les appels externes vers les serveurs distants. Par exemple, si le binôme 2 veut rediriger les appels commençant par 4 vers le binôme 4, il devra ajouter des lignes ayant la forme ci-dessous :

```
exten => _4XX,1,Dial(SIP/${EXTEN}:pass::username@10.4.0.X)
```

La variable `${EXTEN}` correspond au numéro qui a été composé, et servira donc à construire la Request-URI du message INVITE, la partie domaine étant l'adresse IP du serveur distant. `username` et `pass` sont respectivement le nom de la section dans le `sip.conf` du serveur distant (serveur2) et la clé donnée, toujours dans le fichier `sip.conf` du serveur distant.

**I 5.1** Modifiez `sip.conf` et vérifier votre modification en tapant sur la CLI asterisk :

```
sip show peers
```

**I 5.2** Modifiez `extensions.conf`.
**I 5.3** Quand le binôme distant est prêt, testez les appels entrants et sortants.

## Exercice 6 — Mise en place de la messagerie

Mettez en place et testez la messagerie vocale en vous aidant d'Internet. Les messageries des deux utilisateurs seront protégées par un mot de passe de votre choix. Lors d'un appel, l'appelant basculera automatiquement sur la messagerie si l'appelé ne décroche pas au bout de 5 secondes.

En fin de TP, archivez et sauvegardez sur clé USB le contenu du répertoire `/etc/asterisk` :

```
$ tar czf asterisk.tgz /etc/asterisk
```

Nous réutiliserons ces fichiers dans les TP suivants.

# TP 2 — Analyse d'échanges SIP et RTP

Le but de ce TP est d'étudier les échanges SIP dans le cadre d'appels entre différents UA et serveurs SIP. Comme pour le TP précédent, nous travaillerons en binômes, chaque binôme travaillant sur 2 PC et un téléphone T42U.

## Exercice 1 — Travail préliminaire

**I 1.1** Refaites l'exercice 1 du TP 1 pour installer les différents paquets nécessaires et connecter les équipements.

**I 1.2** Désarchivez le contenu du fichier `asterisk.tgz` obtenu à la fin du TP 1. L'archive doit être extraite à la racine du système de fichiers :

```
# cd /
# tar xzf asterisk.tgz
```

**I 1.3** Refaites l'exercice 3 du TP 1 pour configurer les deux téléphones (linphone sur un des deux PC et le T42U).

## Exercice 2 — L'enregistrement SIP

Le but de cet exercice est d'analyser les échanges SIP lors de l'*enregistrement* d'un UA auprès du registrar (asterisk dans notre cas).

**Q 2.1** À quoi sert l'enregistrement d'un UA SIP ?

**I 2.1** Tout en capturant les trames sur le PC hébergeant le service asterisk, redémarrez linphone après l'avoir fermé depuis la barre des tâches.

**Q 2.2** Quels sont les 4 premiers messages SIP capturés lors de cette opération ? Donner la signification des messages.

**Q 2.3** Dans sa première réponse, asterisk indique à l'UA l'algorithme de chiffrement à utiliser pour chiffrer le mot de passe. Dans quel champ d'en-tête SIP (voir la section `Message Header` dans wireshark) trouve-t-on cette information et quel est cet algorithme ?

**Q 2.4** Dans quel champ d'en-tête SIP l'UA insère-t-il ensuite son mot de passe (chiffré) ?

**Q 2.5** Quelle informations sont, à la fin de l'échange, enregistrées par le registrar et dans quels champs d'en-tête SIP les trouve-t-on ?

## Exercice 3 — Initialisation d'une session SIP

Dans cet exercice nous allons observer les échanges SIP dans le cas d'un appel du linphone vers le T42U.

**I 3.1** Sur le PC avec asterisk, capturez avec wireshark les échanges SIP lors du scénario suivant :

(a) Initier un appel depuis linphone.
(b) Ne pas répondre, laisser une ou deux sonneries.
(c) Avec linphone, raccrocher.

(Il se peut qu'asterisk refuse le premier message INVITE envoyé comme dans l'exercice précédent.)

**Q 3.1** Donnez une copie d'écran du graphique des flux capturés (Dans wireshark, utiliser `sip` comme filtre puis aller dans *Statistiques* → *Graphique des flux*. Cocher ensuite *Limiter au Filtre d'Affichage*).

**Q 3.2** Indiquez le type de message utilisé dans chacun des cas suivants :

(a) Linphone initie l'appel.
(b) Le T42U prévient qu'il a démarré la sonnerie.
(c) Asterisk prévient qu'il tente de contacter le T42U.
(d) Linphone prévient que l'utilisateur a raccroché.
(e) Asterisk acquitte l'annulation de l'appel.

On s'intéresse maitenant à l'en-tête des messages `INVITE`.

**Q 3.3** Retrouvez dans le message `INVITE` relayé par asterisk vers le T42U chacune des informations suivantes et indiquez dans quel champ d'en-tête l'information est disponible :
(a) L'adresse IP du (ou des) proxy(s) par le(s)quel(s) est passé le INVITE.
(b) L'URI de linphone.
(c) L'URI du T42U.
(d) Le langage de description des données dans le corps du message INVITE.
(e) Le nombre maximal de proxys par lequel ce message INVITE peut encore transiter.

Supposons que le serveur Asterisk est en panne.

**Q 3.4** Quelle URI doit-on alors composer pour joindre directement un UA ? Testez.
**Q 3.5** Quel est l'inconvénient de cette méthode ?

## Exercice 4 — La négociation SDP

Le corps d'un message `INVITE` contient des données au format SDP (Session Description Protocol). C'est dans ce corps que l'UA appelant fournit les paramètres RTP (ou autre) qu'il souhaite utiliser pour la session : port UDP sur lequel les paquets de voix seront envoyés et reçus, codec (algorithme de codage de la voix), débit du codec, taille des paquets, …. L'UA appelé consulte ces paramètres, détermine ceux qui sont compatibles avec sa propre configuration (p.ex., les codecs demandés par l'appelant qui sont également installés chez lui) et renvoie ceux-ci dans son message `200 OK`. On parle alors de négociation de paramètres.

Sous wireshark, c'est la ligne `Media Description` qui contient les informations les plus importantes. Elle contient :
— le type de session demandée (audio, vidéo ou les deux) ;
— le port UDP utilisé pour le transport des données (voix ou image) ;
— le protocole utilisé pour transporter les données (RTP, …) ;
— et la liste des numéros des codecs supportés (chaque codec étant identifié par un numéro).

**I 4.1** Sur linphone, changez les préférences pour que seul le codec PCMA soit activé (*Préférences* → *Audio*).
**I 4.2** Sur le PC avec asterisk, capturez avec wireshark les échanges SIP lors d'un appel du T42U vers linphone (arrêtez la capture après avoir décroché et avant de raccrocher afin de ne pas avoir trop de messages SIP).

**Q 4.1** Sélectionnez le message `INVITE` envoyé par le T42U et en consultant la ligne `Media Description` du corps du message (`Message Body` dans wireshark), trouvez :
— le port UDP qu'utilisera le T42U pour le flux RTP ;
— et les numéros des codecs disponibles chez le T42U.
**Q 4.2** Sélectionnez maintenant le message `200 OK` envoyé par l'UA 2 et retrouvez les paramètres fournis par linphone (numéro de port et numéros des codecs disponibles).

**I 4.3** Désactivez maintenant tous les codecs sur linphone.
**I 4.4** Refaites le test précédent (capture de trames sur asterisk lors d'un appel).

**Q 4.3** Quel est le code SIP utilisé par linphone pour indiquer qu'il ne supporte aucun des codecs demandés ?

**I 4.5** Réactivez le codec PCMA sur linphone.

## Exercice 5 — Fermeture de session SIP

Sur le PC asterisk, lancer une capture pour observer le traffic SIP lors d'un appel complet (stopper la capture après avoir racroché) entre les deux UA.

**Q 5.1** Quels message SIP indique qu'un correspondant a raccroché ?
**Q 5.2** Avez vous capturé des paquets RTP pendant l'appel ? Pourquoi ?

## Exercice 6 — Les paquets RTP

Capturer maintenant sur un des deux PC avec UA le trafic RTP lors d'un appel.

**Q 6.1** Observer l'en-tête RTP. Le champ `Payload Type` indique le format des données, c'est-à-dire ici le numéro du codec choisi à l'issue de la négociation SDP (voir exercice 4). Quel est ce codec ici ? Quel est son numéro ?

**Q 6.2** Comment varie le numéro de séquence entre un paquet et le suivant ? À quoi sert ce numéro ?

**Q 6.3** Combien d'octets de voix sont contenus dans chaque paquet RTP ? (Attention à retirer les octets d'en-tête du nombre affiché par wireshark (colonne `Length`).)

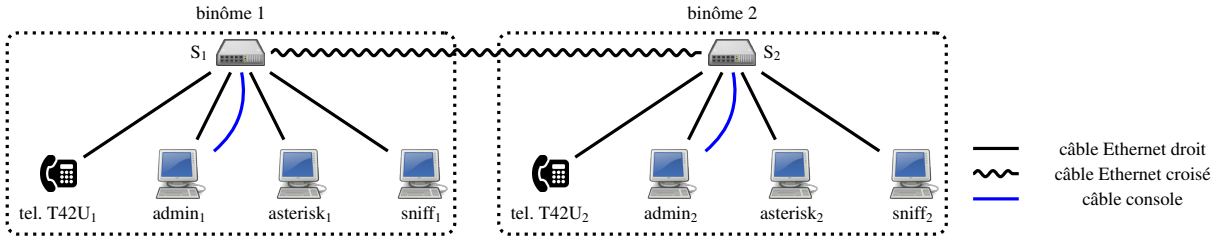
**Q 6.4** Le codec produit un octet toutes les 125 $\mu$s. En déduire le temps nécessaire pour générer un paquet RTP.

**Q 6.5** Quelle est la limite que ce temps ne doit pas dépasser pour avoir une qualité de service optimale ?

# TP 3 — Qualité de service sur un réseau local

L'objectif du TP est :
— de mettre en évidence le problème lié à la cohabitation de la téléphonie et des données sur un réseau local ;
— d'apprendre à résoudre ce problème en utilisant des VLAN.

Nous réaliserons l'architecture de la figure suivante :




Le TP est à faire en binôme et chaque binôme travaillera sur trois PC et un téléphone T42U connectés à un switch lui même connecté au switch d'un autre binôme. Nous nommerons les PC admin, asterisk et sniff. Leurs rôles sont les suivants :
— admin sera utilisé pour administrer le switch du groupe via l'outil minicom et effectuer des transferts de données vers le PC d'un autre binôme ;
— asterisk sera le serveur SIP du binôme ;
— et sniff servira à capturer et analyser le trafic.

Les commandes de configuration du switch sont disponibles dans le document suivant :

`https://www.lipn.univ-paris13.fr/~evangelista/cours/cisco.pdf`

## Exercice 1 — Travail préliminaire

Notez l'interface qui vous connecte au réseau de l'IUT (avec une IP de la forme 192.168.X.Y). C'est l'autre interface que nous utiliserons dans ce TP. On suppose dans la suite qu'elle se nomme eth0.

**I 1.1** Connectez les PC et le téléphone au switch de la baie.
**I 1.2** Installez les paquets suivants :
— sur admin : minicom, iperf ;
— sur asterisk : asterisk ;
— et sur sniff : wireshark.
**I 1.3** Attribuez les IP ($N$ = numéro de groupe) :
— $10.N.0.1/8$ au T42U ;
— $10.N.0.2/8$ à admin ;
— et $10.N.0.100/8$ à asterisk.
(Il est inutile d'attribuer une IP à sniff qui se contentera de capturer des paquets)
**I 1.4** Sur asterisk :
— restaurez votre sauvegarde du TP 1
— apportez les modifications nécessaires aux fichiers `sip.conf` et `extensions.conf` pour pouvoir passer et recevoir des appels vers le téléphone du binôme auquel vous êtes connectés. Dans ces deux fichiers, le serveur asterisk de cet autre binôme devra être désigné uniquement par son IP dans le réseau $10.0.0.0/8$.
**I 1.5** Connectez votre switch au switch de l'autre binôme.
**I 1.6** Passez des appels vers le téléphone du binôme auquel vous êtes connectés.

## Exercice 2 — Observation du phénomène de congestion

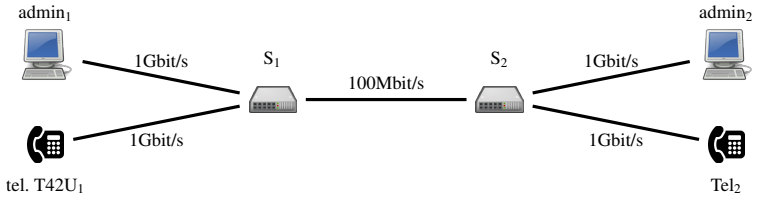
Nous allons provoquer une congestion du réseau en générant beaucoup de trafic entre les PC des deux binômes. En conséquence, les mémoires des switchs seront saturées, ils ne pourront pas mémoriser les paquets reçus et ces paquets seront détruits. Si l'on passe simultanément un appel entre les deux téléphones, la qualité de la communication sera (fortement) dégradée. Pour provoquer cette congestion nous diminuerons le débit sur les liaisons entre les switchs. La première étape et de lancer minicom pour interfacer le PC admin avec le switch.

**I 2.1** Suivez les instructions de la section 1 de la documentation cisco pour vous interfacer avec le switch.

Dans la suite du TP, à chaque fois qu'il sera demandé de configurer le switch (comme, par exemple, au point suivant), cela devra être fait dans le terminal minicom.

**I 2.2** Suivez les instructions de la section 3.6 de la documentation cisco pour passer le débit du port qui vous lie à l'autre switch à 100Mbit/s.

On se retrouve alors dans la configuration suivante :




Ainsi, si $admin_1$ envoie des paquets à $admin_2$ au débit maximal (soit 1Gbit/s), ces paquets ne pourront pas tous être écoulés par $S_1$ sur la liaison vers $S_2$. En moyenne, un paquet sur 10 (le rapport entre les deux débits) devrait passer.

**I 2.3** Initiez un appel vers le téléphone de l'autre binôme puis, sans raccrocher, lancez le transfert de données avec iperf depuis votre PC admin vers le PC admin de l'autre binôme. Il faut pour cela effectuer dans l'ordre :

— Sur le PC admin de l'autre binôme :

```
$ iperf -s -u
```

iperf est ici lancé en mode serveur. Celui-ci attend qu'on lui envoie des données sur le port UDP par défaut.

— Sur votre PC admin :

```
$ iperf -c @IP-de-lautre-PC-admin -u -n 5000M -b 1000M
```

iperf est ici lancé en mode client (option `-c`). Il envoie au serveur iperf à l'adresse indiquée un total de 5 giga-octets de données à un débit de 1Gbit/s.

On doit alors constater une dégradation de la qualité sur un des deux téléphones.

Nous allons maintenant observer les rapports RTCP échangés. Il faut pour cela que le switch redirige vers sniff tous les paquets envoyés ou reçus par le téléphone. Nous utiliserons pour cela la fonction de *monitoring* (ou *mirroring*) de port.

**I 2.4** Suivez les instructions de la section 3.5 de la documentation cisco pour activer le monitoring de port afin que :

— la source soit le port auquel est connecté le T42U ;

— et que la destination soit le port auquel est connectée sniff.

**I 2.5** Refaites le test du point I 2.3 tout en capturant les trames sur sniff. Attendez la capture de quelques rapports RTCP de type *Sender Report*. (Les messages RTCP capturés sont de différents types : Receiver, Sender, … Pour diminuer la bande passante allouée à RTCP, les paquets RTCP peuvent contenir plusieurs rapports.)

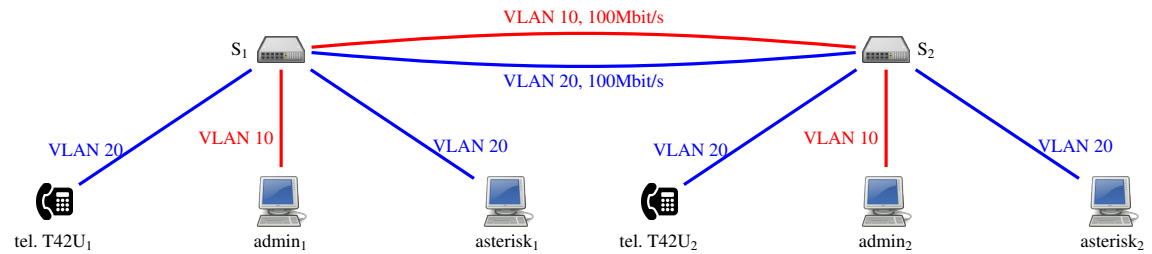
**Q 2.1** Le *Sender Report* doit contenir le nombre de paquets perdus et la gigue observés par l'émetteur du rapport. Comment wireshark nomme-t-il ces informations ?

## Exercice 3 — Résolution du problème avec des VLAN

Pour résoudre le problème constaté à l'exercice précédent, nous allons séparer les flux de données (échange de fichiers, le trafic iperf, …) et les flux de voix (trafic SIP et RTP) en créant deux VLAN : le VLAN 10 pour les données, et le VLAN 20 pour la voix. Le PC admin sera donc sur le VLAN 10 ; le téléphone et le PC asterisk sur le VLAN 20.

On va aussi rajouter un câble entre les deux switchs pour avoir un câble par VLAN.

L'objectif est donc d'arriver à la configuration suivante :



Pour cela :

**I 3.1** Ajoutez un câble entre les deux switchs.
**I 3.2** Passez le débit du port auquel vous avez branché ce nouveau câble à 100Mbit/s.
**I 3.3** Suivez les instructions de la section 3.4 de la documentation cisco pour créer deux VLAN (le VLAN 10 nommé `donnees` et le VLAN 20 nommé `voix`) et pour associer les différents ports aux VLAN selon la figure.

Ainsi, les flux sont séparés et en cas de transfert de données entre les PC admin, seul le VLAN 10 sera engorgé et les paquets SIP ou RTP pourront circuler sur le VLAN 20.

**I 3.4** Refaites le test du point I 2.3 de l'exercice 2 et vérifiez que la qualité de la communication reste bonne pendant le transfert de données.

## Exercice 4 — Résolution du problème avec des VLAN et de la QoS

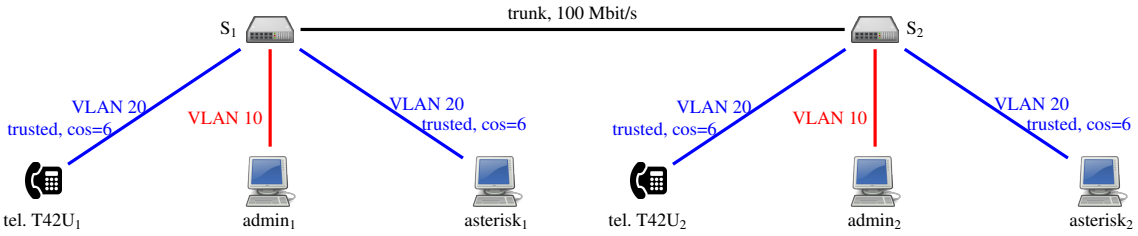
La solution de l'exercice précédent présente l'inconvénient de rajouter du câblage entre les switchs. On peut s'en passer en utilisant un unique câble. Étant donné que les trames échangées par les deux switchs sur ce câble peuvent appartenir à l'un des deux VLAN il faut que cette liaison soit configurée comme une liaison trunk. Pour rappel, les liaisons trunk sont celles sur lesquelles peuvent circuler des trames étiquetées contenant un identifiant de VLAN.

En faisant cela on réintroduit cependant le problème de congestion observé à l'exercice 2 puisque les paquets de voix et de données passeront par le même câble. Il faut donc que sur ce lien, *les paquets de voix aient la priorité sur les paquets de données*. Autrement dit, les switchs retransmettent en priorité les trames du VLAN 20, les trames du VLAN 10 passant en second. Ainsi en situation de congestion, les switchs retransmettront d'abord les trames du VLAN 20 et seules les trames du VLAN 10 pourront être détruites.

Notre réseau doit donc gérer la qualité de service (QoS). Sur un réseau Ethernet, l'information de priorité est portée par un champ de 3 bits de l'étiquette de VLAN parfois appelé CoS (Class of Service). Les trames qui doivent sortir sur une interface donnée d'un switch gérant la QoS sont classées en 4 files d'attente (priorité 0-1, 2-3, 4-5, 6-7), et la file non vide la plus prioritaire voit ses trames émises en premier. Puis, c'est la seconde la plus prioritaire, et ainsi de suite : on parle de priorité stricte.

Les trames émises par le PC et le téléphone ne sont pas étiquetées. C'est lorsque le switch les retransmet sur sa liaison trunk qu'il insère l'étiquette de VLAN contenant l'identifiant de VLAN. Il doit aussi insérer dans cette étiquette la priorité. Nous allons ainsi attribuer une priorité de 6 au téléphone et au serveur asterisk afin que les trames émises par ceux-ci soient prioritaires sur les trames émises par le PC (qui garderont la priorité par défaut de 0). Nous allons donc configurer les ports sur lesquels sont connectés les téléphones et serveurs asterisk comme *trusted ports*.

L'objectif est donc d'arriver à la configuration suivante :



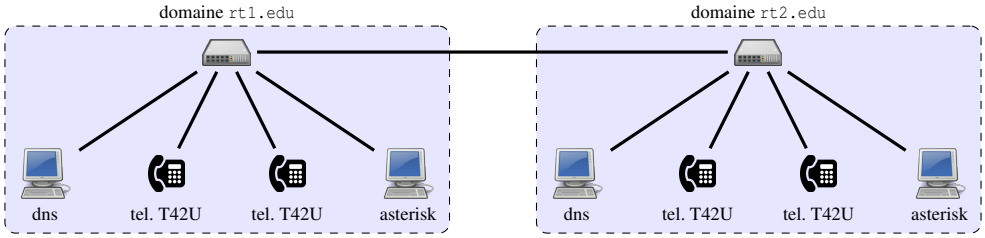

Pour cela :

**I 4.1** Retirez un des deux câbles reliant votre switch à l'autre binôme.
**I 4.2** Passez le port qui vous relie à l'autre binôme en mode trunk.
**I 4.3** Suivez les instructions de la section 3.7 de la documentation cisco pour modifier les paramètres de QoS des ports auxquels sont reliés le téléphone et le serveur asterisk : passer le port en trusted et sa cos à 6.
**I 4.4** Refaites le test du point I 2.3 de l'exercice 2 et vérifiez que la qualité de la communication reste bonne pendant le transfert de données. Durant l'appel, repassez la CoS des téléphones à 0 pour observer les changements dans la qualité de la communication.

Avant de partir, effacer la configuration des VLAN.

# TP 4 — Mise en œuvre d'une architecture de téléphonie

Le but de ce TP est de réaliser une architecture de téléphonie répartie sur plusieurs sites et d'étudier l'utilisation de protocoles de service comme DHCP, DNS, ou FTP dans le cas de la téléphonie sur IP. Le TP est à réaliser sur deux PC sur lesquels on chargera un système debian (9 ou supérieure) et deux téléphones T42U. L'architecture sera la suivante :




Toutes les machines seront sur le réseau 10.0.0.0/8. Chaque groupe administrera le domaine `rtN.edu` (avec `N` = numéro de groupe). Les rôles des différentes machines sont les suivants :

— L'hôte asterisk sera le serveur asterisk du domaine `rtN.edu`. Il hébergera aussi un service DHCP pour configurer automatiquement les téléphones.

— L'hôte dns sera le serveur DNS du domaine `rtN.edu`. Il jouera également le rôle de serveur FTP pour que les téléphones puissent automatiquement télécharger certains fichiers de configuration au démarrage.

## Exercice 1 — Travail préliminaire

**I 1.1** Sur dns, installez les paquets wireshark, bind9 (serveur DNS) et tftpd-hpa.
**I 1.2** Sur asterisk, installez les paquets wireshark, asterisk et isc-dhcp-server.
**I 1.3** Sur asterisk, lancez la CLI asterisk (`asterisk -rv`).

Dans la suite, n'oubliez pas de recharger les fichiers de configuration avec les commandes suivantes dans la CLI : `sip reload` après modification de `sip.conf` ; et `dialplan reload` après modification de `extensions.conf`.

Notez l'interface qui vous connecte au réseau de l'IUT (avec une IP de la forme 192.168.X.Y). C'est l'autre interface que nous utiliserons dans ce TP. On suppose dans la suite qu'elle se nomme eth0.

**I 1.4** Réalisez le câblage de la figure *sans relier les switchs*.
**I 1.5** Attribuez l'IP 10.N.0.101/8 à l'interface eth0 d'asterisk.
**I 1.6** Attribuez l'IP 10.N.0.102/8 à l'interface eth0 de dns.
**I 1.7** Sur asterisk, restaurez votre sauvegarde du TP 1 à la racine du système de fichiers :

```
$ cd /
$ tar xzf asterisk.tgz
```

**I 1.8** Modifiez les fichiers asterisk pour que vos deux téléphones aient les numéros N01 et N02.

## Exercice 2 — Mise en service des téléphones avec DHCP et FTP

Au démarrage, un téléphone va dans un premier temps obtenir une configuration IP auprès d'un serveur DHCP. Le téléphone va ensuite contacter le serveur FTP transmis par le serveur DHCP (l'hôte dns dans notre cas) pour télécharger certains fichiers de configuration. Ceux-ci contiendront, par exemple, l'IP et le port d'un proxy SIP, le numéro du téléphone (pour qu'il puisse s'enregistrer automatiquement au démarrage) ou la langue de l'interface du téléphone.

Nous allons dans un premier temps configurer le service DHCP sur asterisk.

**I 2.1** Créez un fichier de configuration DHCP (`/etc/dhcp/dhcpd.conf`) minimal permettant uniquement d'attribuer une centaine d'IP à partir de 10.N.0.1 et contenant les deux lignes ci-dessous :

```
option tftp-server-name "tftp://<ip-de-dns>";
option routers 10.N.0.254;  # routeur inexistant mais demandé par les tél.
```

**I 2.2** Modifiez le fichier `/etc/default/isc-dhcp-server` pour que le serveur écoute sur eth0.
**I 2.3** Redémarrez le service `isc-dhcp-server`.

Les téléphones peuvent maintenant obtenir une configuration IP auprès d'asterisk.

**I 2.4** Tout en capturant les trames sur l'hôte dns, redémarrez les téléphones en mode usine pour vérifier que leur serveur DHCP leur attribue une IP.
**I 2.5** Vérifiez que les deux téléphones répondent aux messages ping.

**Q 2.1** En analysant la capture faite à l'instruction I 2.4, donnez la liste des fichiers demandés par un téléphone au serveur FTP.

Après avec contacté le serveur DHCP, le téléphone va ensuite contacter le serveur FTP. Vous avez dû voir lors de la capture de trames (I 2.4) que les téléphones ont demandé plusieurs fichiers de configuration. En effet les informations nécessaires à la configuration d'un téléphone peuvent être disséminés dans plusieurs fichiers. Pour simplifier nous allons les regrouper dans un seul fichier. Ce fichier est nommé d'après l'adresse MAC du téléphone. Celle-ci étant a priori unique, c'est le moyen utilisé pour identifier le téléphone.

Nous allons configurer les téléphones un par un.

Le répertoire racine du serveur FTP est `/srv/tftp`. C'est ce répertoire qui contient les fichiers servis. Sur l'hôte dns :

**I 2.6** Créez dans le répertoire `/srv/tftp` un fichier `<adresse-mac-du-telephone>.cfg` (sans les deux-points et avec les caractères alphabétiques en minuscules) ayant la structure suivante :

```
1 #!version:1.0.0.1
2 account.1.enable = 1
3 account.1.label = <identifiant-du-compte>
4 account.1.display_name = <nom-qui-sera-affiché>
5 account.1.auth_name = <nom-du-téléphone-dans-le-fichier-sip.conf>
6 account.1.user_name = <nom-du-téléphone-dans-le-fichier-sip.conf>
7 account.1.password = <secret-du-téléphone-dans-le-fichier-sip.conf>
8 account.1.sip_server.1.address = <ip-du-serveur-asterisk>
9 lang.gui = French
```

**I 2.7** Redémarrez le service `tftpd-hpa`.

Quelques remarques sur le contenu du fichier :
— La première ligne est indispensable. Elle définit le numéro de version du format du fichier.
— Les lignées préfixées par `account.1` permettent de configurer le premier compte SIP du téléphone. En effet plusieurs comptes peuvent être configurés. On retrouve sans surprise les mêmes informations que celles demandées lors de la configuration manuelle du compte SIP (voir exercice 3 du TP 1).
— La deuxième ligne active le compte au démarrage.
— Plusieurs serveurs SIP peuvent être renseignés pour un compte (p.ex., pour pouvoir contacter un autre serveur SIP en cas de panne du premier). La ligne 8 définit le premier serveur du premier compte.

Le serveur FTP est maintenant prêt à servir le premier téléphone pour que celui-ci soit automatiquement configuré.

**I 2.8** Redémarrez le téléphone en mode usine.
**I 2.9** Vérifiez depuis la CLI asterisk que le téléphone est bien enregistré.

Passons maintenant au deuxième téléphone.

**I 2.10** Suivez à nouveau les instructions I 2.6 à I 2.9 pour le deuxième téléphone.
**I 2.11** Vérifiez que les deux téléphones peuvent bien s'appeler (dans les deux sens).

## Exercice 3 — Mise en place des serveurs de noms

**I 3.1** Reliez maintenant votre switch au switch d'un autre binôme. Il faut que cet autre binôme ait d'abord terminé l'exercice précédent.
**I 3.2** Modifiez au besoin vos fichiers asterisk pour que les numéros et IP qui y apparaissent soient cohérents avec notre plan d'adressage et de numérotation.
**I 3.3** Testez en passant des appels entre les téléphones des deux binômes.

Actuellement les serveurs asterisk distants sont référencés dans `extensions.conf` par leurs IP. En pratique, on utilise plutôt le nom du domaine SIP. Il faut pour cela que chaque domaine de notre réseau (`rt1.edu`, `rt2.edu`, ...) ait un serveur DNS permettant de renseigner les autres serveurs SIP sur le nom et l'IP du serveur SIP de son domaine.

Nous allons pour cela procéder en deux temps. Dans un premier temps, nous allons configurer le serveur DNS du domaine afin de pouvoir renseigner les serveurs asterisk des autres réseaux. Puis, dans un second temps, nous allons modifier le serveur asterisk afin qu'il utilise des noms de domaine plutôt que des IP. Sur l'hôte dns :

**I 3.4** Dans le fichier des options du serveur DNS (`/etc/bind/named.conf.options`), ajoutez les options suivantes (entre les accolades de `options { ... };`) :

```
recursion yes;                 // active la résolution récursive
dnssec-validation no;          // désactive DNSSEC
directory "/var/cache/bind";   // fixe le répertoire de travail du serveur
```

Il se peut qu'une option soit déjà présente dans le fichier auquel cas il suffira de s'assurer que la valeur courante est celle donnée plus haut ou de la modifier si ce n'est pas le cas.

**I 3.5** Ajoutez à la fin du fichier `/etc/bind/named.conf` la définition de deux zones (en changeant `N` par votre numéro de binôme et `P` par le numéro du binôme auquel vous êtes connectés) :

```
zone rtN.edu. {  // la zone sur laquelle on fait autorité
  type master;
  file "/etc/bind/rtN.edu";  // chemin du fichier de zone
};
zone rtP.edu. {  // la zone de l'autre binôme
  type forward;  // on forwarde les requêtes concernant cette zone ...
  forwarders {
    10.P.0.102;  // ... vers cette IP
  };
};
```

*Remarque.* En situation réelle, il ne serait pas nécessaire de déclarer la zone `rtP.edu.` car le DNS obtiendrait l'IP du DNS de cette zone directement auprès du serveur de plus haut (le DNS de `edu.` dans notre cas).

**I 3.6** Créez le fichier de zone `/etc/bind/rtN.edu` donnant les informations DNS de votre zone :
- Le nom et l'IP du serveur DNS de la zone.
- Une description textuelle de la zone.
- Le nom et l'IP du serveur SIP de la zone (voir le cours).

**I 3.7** Redémarrez le service `bind9`.

Sur asterisk :

**I 3.8** Désactivez eth1 puis modifier le fichier `/etc/resolv.conf` afin que dns devienne le serveur DNS du serveur asterisk. (Il est nécessaire de désactiver eth1 car sinon, le client DHCP associé à eth1 écrase périodiquement ce fichier avec les informations DNS fournies par le serveur DHCP de l'IUT.)

**I 3.9** Testez en interrogeant le serveur DNS sur le nom du serveur SIP+UDP de la zone :

```
$ host -t SRV _sip._udp.rtN.edu.
```

Le serveur DNS est maintenant opérationnel. Il reste maintenant à modifier la configuration du serveur asterisk pour qu'il utilise des noms plutôt que des IP. Sur asterisk :

**I 3.10** Dans `extensions.conf` : modifiez l'extension qui vous permet de router les appels vers le serveur SIP de l'autre binôme en changeant l'IP de ce serveur par son nom de domaine (`rtP.edu`).

**I 3.11** Apportez les deux modifications suivantes dans `sip.conf` :
- Dans la section `[general]`, ajoutez le paramètre ci-dessous qui active la recherche DNS :

```
srvlookup=yes
```

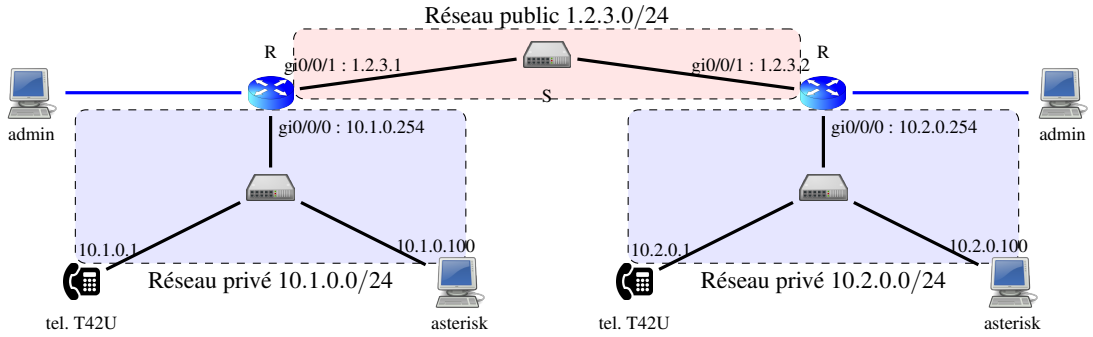
- Dans la section qui déclare le serveur SIP de l'autre binôme changez la valeur du paramètre `host` : indiquez le nom de domaine de l'autre binôme plutôt que l'IP.

**I 3.12** Une fois que l'autre binôme en est au même point, testez en passant un appel vers un téléphone de l'autre binôme.

**Q 3.1** Observez, avec wireshark, les échanges DNS ayant lieu lors d'un appel entre deux téléphones de domaines distincts : quels sont les messages DNS échangés entre un serveur asterisk et son serveur DNS ? Donnez des copies d'écran des messages DNS capturés.

# TP 5 — SIP, RTP et NAT

Le but de ce TP est d'étudier les problèmes liés à l'utilisation de la translation d'adresses dans le cas de la téléphonie sur IP et de voir une solution proposée par Asterisk pour résoudre ce problème. Le TP est à réaliser en binôme sur un téléphone T42U, un routeur et deux PC sur lesquels on chargera un système debian (version 9 ou supérieure). Nous réaliserons l'architecture ci-dessous :




Chaque groupe administrera le réseau 10.N.0.0/24 (avec N = numéro de groupe) connecté au réseau 1.2.3.0/24 via le routeur R qui fera également office de passerelle NAT. Chaque binôme travaillera avec un autre binôme. L'hôte admin servira à administrer le routeur.

Les commandes de configuration du routeur sont disponibles dans le document suivant :

`https://www.lipn.univ-paris13.fr/~evangelista/cours/cisco.pdf`

## Exercice 1 — Travail préliminaire

Notez l'interface qui vous connecte au réseau de l'IUT (avec une IP de la forme 192.168.X.Y). C'est l'autre interface que nous utiliserons dans ce TP. On suppose dans la suite qu'elle se nomme eth0.

**I 1.1** Sur asterisk, installez les paquets wireshark et asterisk.
**I 1.2** Réalisez le câblage de la figure en introduction.
**I 1.3** Sur asterisk, restaurez votre sauvegarde du TP 1.
**I 1.4** Attribuez l'IP 10.N.0.100 à l'interface eth0 d'asterisk, comme indiqué sur la figure.
**I 1.5** Modifiez le fichier `extensions.conf` pour que votre téléphone ait le numéro N01.
**I 1.6** Suivez les instructions I 3.5 à I 3.14 du TP 1 pour que votre téléphone ait l'IP 10.N.0.1, la passerelle par défaut 10.N.0 254 et s'enregistre auprès du serveur asterisk.
**I 1.7** Vérifiez :
— que le téléphone et asterisk peuvent s'échanger des pings ;
— et, dans la CLI asterisk, que le téléphone est bien enregistré.

## Exercice 2 — Mise en place du routage

Nous allons maintenant configurer le routage pour permettre les communications entre différents réseaux. Il est pour cela nécessaire de s'interfacer avec le routeur.

**I 2.1** Suivez les instructions de la section 1 de la documentation cisco pour vous interfacer avec le routeur.

Dans la suite du TP, à chaque fois qu'il sera demandé de configurer le routeur (comme, par exemple, au point suivant), cela devra être fait dans le terminal minicom.

**I 2.2** Suivez les instructions de la section 3.8.1 de la documentation cisco pour attribuer des IP aux deux interfaces de votre routeur, comme indiqué sur la figure en introduction.

Dans un premier temps, nous n'allons pas mettre en place de translation d'adresse. Ainsi, chaque serveur asterisk connaîtra l'IP privée du serveur asterisk de l'autre binôme. De même pour les routeurs : chaque routeur connaîtra l'IP privée du réseau se trouvant derrière le routeur distant.

**I 2.3** Sur asterisk, ajoutez une route vers le réseau (privé) du binôme auquel vous êtes connectés :

```
$ ip route add <ip-reseau-destination>/<masque> via <ip-routeur>
```

**I 2.4** Suivez les instructions de la section 3.8.2 de la documentation cisco pour ajouter sur votre routeur une route vers le réseau (privé) du binôme auquel vous êtes connectés.

**I 2.5** Vérifiez que votre serveur asterisk peut bien envoyer un ping au serveur asterisk de l'autre binôme.

**I 2.6** Modifiez vos fichiers asterisk pour que les numéros et IP qui y apparaissent soient cohérents avec notre plan d'adressage et de numérotation.

**I 2.7** Testez en passant des appels entre les téléphones des deux binômes.

## Exercice 3 — Mise en place de la translation d'adresse

Nous allons, dans cet exercice, mettre en place sur R la translation d'adresses afin de masquer les réseaux privés. Les hôtes d'un réseau privé, par exemple, asterisk, ne connaîtront les hôtes d'un autre réseau privé que par leur adresse publique (sur le réseau 1.2.3.0/24).

Il faut, dans un premier temps, faire en sorte que les adresses privées de l'autre binôme n'apparaissent plus, ni dans les tables de routage, ni dans les fichiers de configuration d'asterisk. Seule l'adresse publique doit apparaître.

**I 3.1** Supprimez les routes vers le réseau (privé) du binôme auquel vous êtes connectés :
- sur asterisk, il suffit de refaire la commande de l'instruction I 2.3 en remplaçant `add` par `del` ;
- et sur le routeur, cela se fait en refaisant la commande de l'instruction I 2.4 en préfixant la commande par `no`.

**I 3.2** Sur asterisk :
- ajoutez une route vers le réseau 1.2.3.0/24 ;
- et remplacez, dans les fichiers d'asterisk, l'IP privée du serveur asterisk de l'autre binôme par son IP publique.

Après la suppression des routes, et étant donné que nous n'avons pour l'instant pas configuré le NAT, toute communication du réseau privé vers l'extérieur est impossible.

**I 3.3** Sur asterisk, envoyez un ping vers l'IP publique de l'autre binôme. La commande ne devrait pas fonctionner.

En effet, lorsque le routeur reçoit la demande d'écho ICMP envoyée par ping, l'adresse source est une adresse privée. Le routeur n'a donc aucune ligne dans sa table de routage lui permettant d'envoyer sa réponse.

Passons maintenant à la configuration du NAT sur le routeur. La première étape est de spécifier l'interface interne (celle qui connecte le routeur au réseau privé) et l'interface externe (celle qui le connecte au réseau public).

**I 3.4** Suivez les instructions de la section 3.12 de la documentation cisco pour définir les interfaces interne et externe.

Par la suite nous aurons à déterminer l'ensemble des adresses sur lesquelles la translation sera activée. Cela se fait par la création d'une ACL (*Access Control List*).

**I 3.5** Suivez les instructions de la section 3.9 de la documentation cisco pour créer l'ACL 1 contenant toutes les IP de votre réseau privé, *et celles-ci uniquement*.

Il faut enfin activer la translation d'adresses pour cette ACL. Cela signifie :
- que quand un paquet ayant pour source une IP de cette ACL est routé de l'interface interne vers l'interface externe, le routeur modifie l'IP source (privée) par l'IP publique associée à l'interface externe et rajoute une ligne dans sa table de translation NAT (comme vu en cours) ;
- et que quand un paquet est routé dans le sens inverse (de l'interface externe vers l'interface interne), l'IP destination (publique) est modifiée par l'adresse IP privée trouvée dans la table.

**I 3.6** Suivez les instructions de la section 3.12 de la documentation cisco pour que l'ACL 1 soit associée par translation à l'IP de l'interface externe (publique).

À présent les adresses privées ne devraient plus apparaître dans les paquets circulant sur le réseau public.

**I 3.7** Sur asterisk, envoyez un ping vers l'IP publique de l'autre binôme. La commande devrait fonctionner.

**I 3.8** Suivez les instructions de la section 3.12 de la documentation cisco pour afficher le contenu de la table NAT. Vous devriez voir une ligne ajoutée par le routeur lorsque la demande d'écho ICMP est sortie du réseau privé.

## Exercice 4 — Solution au problème posé par la translation d'adresse

Dans cet exercice nous allons mettre en place une solution proposée par asterisk. Il faut au préalable désactiver une fonctionnalité du routeur (appelée SIP ALG) qui entre en conflit avec cette solution.

**I 4.1** Sur le routeur, exécutez :

```
R(config)# no ip nat service sip udp port 5060
```

Le premier problème qui se pose une fois le NAT mis en place est qu'un serveur asterisk n'est plus joignable depuis l'extérieur. Il suffit de passer un appel pour constater le problème.

**I 4.2** Appelez le téléphone de l'autre binôme. Celui-ci ne devrait pas sonner.

En effet, lorsque le routeur de l'autre binôme reçoit le paquet SIP INVITE destiné au port UDP/5060 (port SIP par défaut), il ne sait pas que ce paquet doit être redirigé vers le serveur asterisk se trouvant sur son réseau privé. Le paquet est donc ignoré. Pour résoudre ce premier problème, il faut rajouter dans la table NAT du routeur une ligne statique spécifiant que tout paquet reçu sur l'interface externe et destiné au port UDP/5060 doit être redirigé vers le serveur asterisk.

**I 4.3** Suivez les instructions de la section 3.12 de la documentation cisco pour que les paquets destinés au port UDP/5060 soit routés vers le serveur asterisk.

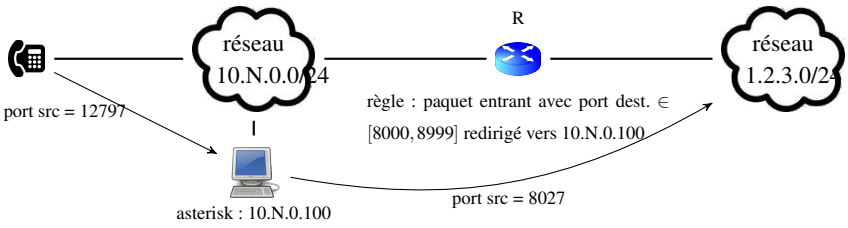
La règle ajoutée permet l'échange de paquets SIP entre les serveurs asterisk mais pas de paquets RTP entre les UA.

**I 4.4** Démarrez une capture sur les serveurs asterisk.
**I 4.5** Appelez le téléphone de l'autre binôme. Celui-ci devrait maintenant sonner. Décrochez. Aucun son ne devrait être entendu (quel que soit le téléphone), ce qui signifie qu'aucun paquet RTP ne circule.
**I 4.6** Arrêtez la capture.

**Q 4.1** En analysant le corps du paquet INVITE envoyé par asterisk, trouvez l'information qui explique le problème.

Une solution proposée par asterisk au problème que nous avons observé est de faire passer le flux RTP par le serveur asterisk. Au lieu que les paquets RTP soient directement échangés entre les téléphones, on rajoute un (ou des) intermédiaire(s) : le(s) serveur(s) asterisk. En procédant de cette manière on peut identifier un intervalle de ports UDP qui seront utilisés par le serveur asterisk pour le flux RTP. Ceci permet ensuite de modifier le comportement de la passerelle NAT afin qu'elle redirige tout le flux RTP entrant (identifié par un port UDP appartenant à l'intervalle défini) directement vers le serveur asterisk. C'est ensuite au serveur asterisk de rediriger le flux vers la bonne IP et le bon port sur son réseau privé.

Le fonctionnement du serveur asterisk lors de l'envoi d'un paquet RTP du téléphone vers le réseau public peut alors être résumé à l'aide de la figure ci-dessous (les numéros de port apparaissant sur la figure sont des exemples) :




— Le téléphone envoie d'abord le paquet RTP au serveur asterisk (au lieu de l'envoyer directement au destinataire).
— Asterisk relaie le paquet RTP après avoir modifié le port source par un port dans l'intervalle $[8000, 8999]$ qui est l'intervalle des ports autorisés au niveau de la passerelle NAT.

On peut maintenant configurer le NAT sur asterisk.

**I 4.7** Ajoutez les 3 lignes suivantes dans la section `general` du fichier `sip.conf` afin d'indiquer au serveur asterisk son IP publique et l'IP de son réseau privé :

```
nat=yes
externip=<ip-publique-du-réseau>
localnet=<ip-privée-du-réseau>/<masque-du-réseau-privé>
```

Dans le fichier `rtp.conf` d'asterisk les paramètres `rtpstart` et `rtpend` définissent l'intervalle des ports UDP utilisés par asterisk pour le flux RTP. Pour chaque communication avec l'extérieur, asterisk choisira un port dans cet intervalle.

**I 4.8** Modifiez les valeurs de ces deux paramètres afin d'utiliser l'intervalle de ports $[50000, 50004]$ pour le flux RTP.
**I 4.9** Fixez à `no` la valeur des paramètres `directmedia` et `directrtpsetup` dans `sip.conf`. Ceux-ci déterminent si le flux RTP va directement entre les deux UA mis en relation par asterisk (`yes`) ou s'il passe par asterisk (`no`).
**I 4.10** Redémarrez le serveur asterisk.

Il reste à rediriger les paquets entrants dans l'intervalle choisi vers le serveur asterisk.

**I 4.11** Réexécutez les commandes de l'instruction I 4.3 en changeant le numéro de port pour rediriger les 5 ports UDP de l'intervalle $[50000, 50004]$ vers le serveur SIP.

(Il existe un moyen de rediriger un intervalle de ports sans spécifier les ports un par un mais c'est un peu plus compliqué à réaliser.)

Il reste maintenant à tester que les appels passent bien.

**I 4.12** Vérifiez que le son passe bien lors d'un appel.

**Q 4.2** Lors d'un appel quelles différences observe-t-on dans le contenu du message SIP INVITE avant et après le relais de ce message par asterisk ?
**Q 4.3** Quels sont les inconvénients de cette solution ?