

Intégration Voix / Données

TP 4 - Voix sur IP (2)

Objectifs du TP

- Comprendre la répartition de services d'une architecture de Voix sur IP multi-sites.
- Implanter les solutions techniques découvertes en cours et TP (QoS)
- Savoir déployer une solution de VoIP répartissant la charge au sein d'un site.
- Comprendre les paramètres influant sur le transport de la voix et savoir les estimer.
- Observer les performances lors d'appels.

Configuration

Configuration matérielle

- 3 stations Linux (par groupe)
- 1 routeur (par groupe)
- 1 switch (par groupe)
- éventuellement SIP-phones

Configuration logicielle

- Wireshark
- Softphone
- Asterisk
- Utilitaires réseaux habituels (ifconfig, ping...)

Cas d'étude

Une architecture de VoIP peut supposer des contraintes de disponibilités ou de performance qui ne peuvent être atteintes par un seul et même serveur VoIP. Pour ce faire, on peut distribuer ces services.

La distribution des services de VoIP peut être vue selon plusieurs points de vue. Une première vision est la distribution intra-établissement, la seconde étant la distribution inter-établissement. Il est par contre à noter que ces deux distributions ne sont pas exclusives et peuvent tout à fait être combinées.

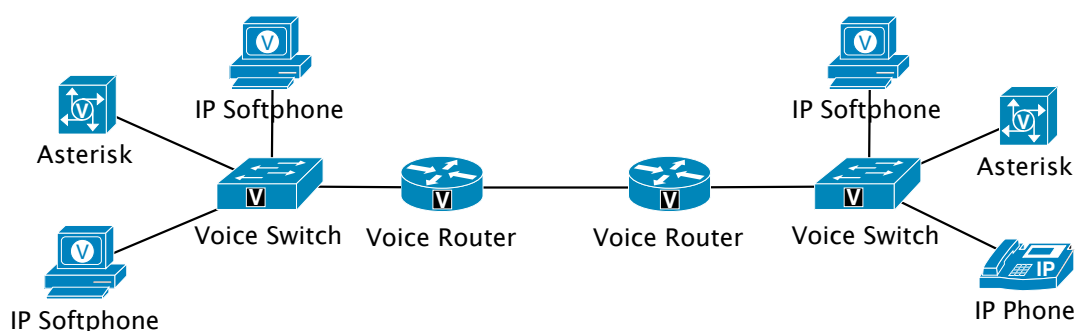
Dans le premier cas, il s'agit de distribuer les différents services de VoIP (comme la messagerie vocale, l'authentification des usagers...) et/ou de multiplier les IP-PBX (pour assurer de la haute disponibilité). La distribution de services dans Asterisk s'effectue en partie par la délégation à des SGBD (via ODBC) d'une partie de son travail. Le déploiement de plusieurs serveurs Asterisk au sein d'une même entité peut se réaliser au moyen d'architectures virtuelles.

La distribution inter-établissement permet de s'affranchir (ou du moins limiter) des contraintes sur des critères de performances comme la latence, la gigue ou le débit. En effet, si pour une même organisation, un seul IP-PBX est présent, sa charge risque d'être importante. De plus, si les deux sites sont reliés par un WAN et que tous les appels (y compris locaux) passent par ce même serveur, la charge réseau risque d'être très importante. Il est de fait important de distribuer cette charge.

Il devient alors important d'interconnecter plusieurs IP-PBX afin d'assurer continuité de service et/ou optimisation des ressources. Plusieurs possibilités sont offertes afin d'interconnecter des IP-PBX : PSTN, IAX et SIP notamment. Dans ce TP, nous étudierons l'interconnexion SIP et IAX avec Asterisk.

I. Architecture VoIP distribuée

Cette deuxième architecture vise à déployer un réseau plus réaliste. Dans cet exercice, le serveur Asterisk n'est plus directement joignable par tous les équipements de téléphonie IP. De plus, nous supposons l'existence de plusieurs sites distincts possédant chacun leur IP-PBX.



II. Interconnexion Asterisk via SIP

Cette interconnexion est très similaire à la configuration de clients SIP. Le principe est de déclarer les deux serveurs Asterisk comme client l'un de l'autre. Deux étapes sont nécessaires : 1- Déclarer les serveurs SIP; 2- Déclarer les utilisateurs SIP.

II.I. Demande d'enregistrement d'un serveur SIP auprès d'un autre serveur SIP

Cette configuration s'effectue au moyen de la primitive `register` dans le fichier `sip.conf` dans le serveur qui s'enregistrera (client). Elle est à placer dans le contexte `general` du fichier.

```
register => nom_utilisateur[:mdp[:login]]@serveur[:port][[/extension]]
```

Les [] dénotent des éléments optionnels

où :

- `nom_utilisateur` est le nom d'utilisateur du serveur dans l'autre serveur,
- `mdp` est le mot de passe associé au compte,
- `login` est le login à utiliser pour l'authentification si celui-ci est différent du nom d'utilisateur.
- `serveur` est l'adresse du serveur (ou son nom symbolique)
- `extension` représente l'extension utilisateur qui sera utilisé dans le message INVITE de SIP.

☒ **Note :** Si l'extension est omise, l'extension 's' sera utilisée. Il est conseillé de préciser le nom de l'utilisateur distant dans l'extension.

☒ Une option générale nous intéresse aussi : `qualify`. Cette option prend la forme suivante : `qualify=yes | no | <valeur>` et indique que Asterisk doit périodiquement tester si la communication est possible avec les pairs. La valeur par défaut est fixée à 60 secondes.

II.II. Déclaration d'un serveur SIP CLIENT dans un autre serveur SIP

La déclaration s'effectue dans le serveur qui reçoit l'enregistrement (serveur) et est semblable à la déclaration d'un utilisateur SIP conventionnel.

➔ Mettre à jour les fichiers `sip.conf` des deux serveurs afin de permettre des communications inter-sites.

II.III. Mise à jour du plan de numérotation

Le plan de numérotation doit maintenant prendre en compte les appels DE et VERS les clients des deux sites.

Afin d'acheminer correctement les appels, l'application `Dial` devra se représenter sous la forme :

`Dial(SIP/utilisateur_distant/extension_distante)` éventuellement complétée des options habituelles.

où :

- `utilisateur_distant` est le nom d'utilisateur du serveur dans l'autre serveur (précisée au §II.I par l'extension de contact),
- `extension_distante` représente le numéro à joindre à distance.

☒ **Attention :** Afin de limiter d'éventuels problème, il convient d'éviter que les appels extérieurs puissent faire d'autres appels extérieurs...

➔ Mettre en place un plan de numérotation permettant les appels inter-sites. Les appels locaux commenceront par 1 et les appels pour les autres sites commenceront par 2, 3, 4... selon le nombre de sites interconnectés. Conseil : pensez au filtrage et notamment à `${EXTEN:aaa}` où `aaa` est le nombre de chiffres à ignorer au début de l'extension.

➔ Observer avec Wireshark, les messages échangés, le nombre de ports utilisés et le coût de cette solution en terme de bande passante.

III. Interconnexion Asterisk via IAX

IAX (Inter-Asterisk eXchange protocol) a été conçu à l'origine pour permettre les communications (voix et contrôle) entre serveurs Asterisk. La version actuelle est IAX2 (désignée IAX dans la suite). Toutefois, aujourd'hui sa portée s'étend aussi aux clients compatibles. IAX a été dès le début pensé pour permettre le transport de la voix, du contrôle même s'il permet le transport d'autres types de données. Il est de fait optimisé pour minimiser la bande passante nécessaire aux appels (via des mécanismes de Trunk ou l'emploi d'un protocole binaire au lieu du textuel SIP), permettre l'échange d'informations de plans de numérotation et surtout limiter les problèmes liés au NAT et aux pare-feux (en utilisant un seul port UDP pour toute les informations, en lieu et place d'un port pour SIP, d'un port pour RTP...). Actuellement, IAX reste un protocole propriétaire mais il a été soumis à l'IETF en vue d'une "standardisation".

III.I. Fichier `iax.conf`

Comme les autres fichiers de configuration d'Asterisk, IAX dispose d'un fichier (`iax.conf`) organisé en sections. La première est `general`. Comme dans `sip.conf`, chaque utilisateur possède sa section.

Une option nous intéresse actuellement : `autokill`.

- Cette option prend la forme suivante : `autokill=yes | no | <valeur>`

Cette option indique que Asterisk doit annuler les opérations en cours si il n'a aucune réponse (ACK) en moins de 2 secondes (`yes`) ou autre (`valeur`).

III.I.Demande d'enregistrement d'un serveur IAX auprès d'un autre serveur IAX

Cette configuration s'effectue au moyen de la primitive `register` dans le fichier `iax.conf` dans le serveur qui s'enregistrera (client). Elle est à placer dans le contexte `general` du fichier.

```
register => nom_utilisateur[:mdp]@serveur[:port]
```

Les [] dénotent des éléments optionnels

où :
`nom_utilisateur` est le nom d'utilisateur du serveur dans l'autre serveur,
`mdp` est le mot de passe associé au compte,
`serveur` est l'adresse du serveur (ou son nom symbolique).

III.II.Déclaration d'un serveur IAX CLIENT dans un autre serveur IAX

La déclaration s'effectue dans le serveur qui reçoit l'enregistrement (serveur) et est similaire à la déclaration d'un utilisateur SIP conventionnel (dans `sip.conf`). Pour chaque utilisateur, il faut créer une section portant son identifiant (unique). Quelques options configurables sont :

Les options se notent sous la forme suivante : `nom_option=valeur`

Les options et valeurs suivantes sont utilisables (liste non exhaustive) :

- `type=friend | user | peer`
- `context=xxx`
- `host=@IP | dynamic`
- `mask=masque_reseau`
 - Permet de préciser l'option `host`.
- `auth=plaintext | md5 | rsa`
 - Permet de choisir le type d'authentification de l'utilisateur. Nous nous limiterons à `plaintext` dans ce TP.
- `secret=mon_code_secret_rien_qu_a_moi`
- `callerid=nom_entre_guillemets numero_entre_<>`
- `language=code_pays`
- `allow=codec`
- `disallow=codec`
- `permit=@ip | @ip_reseau`
 - Permet de spécifier l'adresse IP (hôte ou du réseau) à partir desquelles l'utilisateur peut être appelé.
- `deny=@ip | @ip_reseau`
 - Permet de spécifier l'adresse IP (hôte ou du réseau) à partir desquelles l'utilisateur NE peut PAS être appelé.



Attention : Asterisk lit le fichier par ordre de lecture standard, ce qui peut influencer sur les autorisations et interdictions...

➡ Mettre à jour les fichiers `iax.conf` des deux serveurs afin de permettre des communications inter-sites.

III.III.Mise à jour du plan de numérotation

Le plan de numérotation doit maintenant prendre en compte les appels DE et VERS les clients des deux sites.

Afin d'acheminer correctement les appels, l'application `Dial` devra se représenter sous la forme :

`Dial(SIP/@IP_relais/extension_distante)` éventuellement complétée des options habituelles.

où :
`@IP_relais` est l'adresse IP du serveur destinataire des appels,
`extension_distante` représente le numéro à joindre à distance.

- ➔ Remplacer l'utilisation de SIP pour les communications inter-sites par des communications IAX2.
- ➔ Observer avec Wireshark, les messages échangés, le nombre de ports utilisés et le coût de cette solution en terme de bande passante.

IV. Trunk

Un des problèmes rencontré lors de la répartition de serveurs VoIP sur SIP est la multiplicité des communications. Prenons l'exemple de deux sites possédant chacun leur IP-PBX et interconnectés par SIP. Les communications entre les clients de chaque site passent chacune par les deux IP-PBX. Lorsque plusieurs communications sont simultanées entre ces sites, de multiples «connexions» sont établies (autant que de communications téléphoniques). Le problème qui se pose alors est le surcoût d'en-tête (notamment) pour les liaisons entre les deux IP-PBX. Une solution à ce problème consiste à multiplexer ces communications au sein d'une seule et même liaison entre les IP-PBX. C'est le but des `trunks`.

L'établissement des `trunks` suppose l'utilisation d'IAX entre les PBX.

Dans ce TP, nous allons créer des `trunks` en passant par des liaisons unidirectionnelles (en différenciant entrée et sortie de sites). Deux moyens sont offerts :

- Activation générale du mode `trunk` d'IAX.
- Activation sélective du mode `trunk` sur une base utilisateur.

Pour chacune des deux options il faut ajouter l'option suivante `trunk=yes` au bon endroit du fichier `iax.conf` :

- à la section `general` pour une activation générale
- à chaque section utilisateur concernée pour une activation sélective.

☒ **Note** : On peut aussi désactiver le mode `trunk` de manière sélective via l'option `trunk=no` dans une section utilisateur.

☒ **Attention** : Recharger les fichiers de configuration ne suffit pas pour activer le mode `trunk`. Il est conseillé de relancer le service Asterisk.

- ➔ Configurer le mode `trunk` pour les liaisons inter-sites. Ces `trunks` devront être unidirectionnels.
- ➔ Observer avec Wireshark, les messages échangés, le nombre de ports utilisés et le coût de cette solution en terme de bande passante.

V. Exercice

Interconnecter tous les IP-PBX de la salle de TP en prenant garde de bien répartir le plan de numérotation. Les liaisons inter-sites doivent toutes passer par IAX avec un secours SIP en cas de problème.

VI. Commandes CLI utiles

VI.1. Commandes Asterisk générales

<code>!</code>	Lancer une commande shell
<code>cdr status</code>	Affiche le statut du module CDR
<code>feature show</code>	liste les fonctionnalités configurées
<code>help</code>	Affiche l'aide générale ou l'aide spécifique à une commande
<code>originate</code>	Débuter un appel
<code>reload</code>	Recharge le fichiers de configuration d'Asterisk
<code>restart now</code>	Redémarre Asterisk <u>immédiatement</u>
<code>restart when convenient</code>	Redémarre Asterisk <u>lorsque le serveur sera inactif</u>
<code>stop now</code>	Stoppe Asterisk <u>immédiatement</u>
<code>stop when convenient</code>	Stoppe Asterisk <u>lorsque le serveur sera inactif</u>

VI.2. Commandes IAX

<code>iax2 debug</code>	Active le mode <i>debug</i> d'IAX
<code>iax2 no debug</code>	Désactive le mode <i>debug</i> d'IAX
<code>iax2 set jitter</code>	Permet le réglage des buffers anti-gigue
<code>iax2 show cache</code>	Affiche le cache du plan de numérotation d'IAX

<code>iax2 show channels</code>	Affiche les canaux IAX actifs
<code>iax2 show netstats</code>	Affiche les statistiques de gigue d'IAX pour le réseau pour les appels IAX actifs
<code>iax2 show peers</code>	Affiche les pairs IAX définis
<code>iax2 show registry</code>	Affiche l'état des enregistrements IAX
<code>iax2 show users</code>	Affiche les utilisateurs IAX définis
<code>iax2 trunk debug</code>	Active le mode <i>debug</i> pour les <code>trunks</code> IAX

VI.3. Commandes spécifiques à SIP

<code>sip history</code>	Active l'historique SIP
<code>sip history off</code>	Désactive l'historique SIP
<code>sip notify</code>	Envoi d'une notification (paquet <i>notify</i>) à un pair SIP
<code>sip reload</code>	Demande à Asterisk de recharger les configurations SIP - Utile en cas de modifications du fichier <code>sip.conf</code> lorsque le serveur fonctionne
<code>sip show channels</code>	Liste les canaux SIP actifs
<code>sip show channel x</code>	Affiche les informations détaillées d'un canal SIP particulier
<code>sip show domains</code>	Liste les domaines SIP locaux
<code>sip show history</code>	Affiche l'historique SIP
<code>sip show peers</code>	Liste les pairs SIP
<code>sip show peer x</code>	Affiche les informations détaillées d'un pair SIP particulier
<code>sip show settings</code>	Affiche les informations générales SIP
<code>sip show users</code>	Liste les utilisateurs SIP
<code>sip show user x</code>	Affiche les informations détaillées d'un utilisateur SIP particulier

Vos notes :



Les fichiers de configuration d'Asterisk sont à rendre par courrier électronique à la fin des TP.