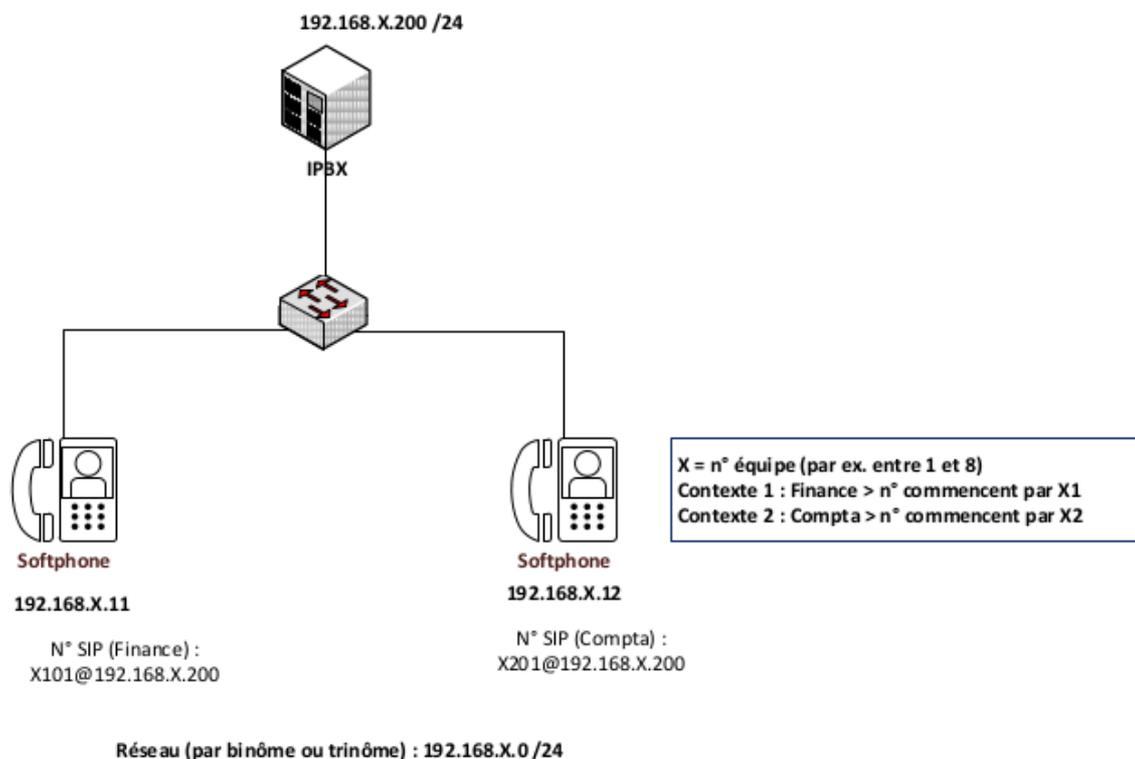


Mise en place et sécurisation d'une infrastructure de téléphonie IP avec Asterisk

Activité 2 – Mise en œuvre d'un plan d'appel sur deux contextes

L'objectif est ici de permettre à des utilisateurs appartenant à des contextes différents de se joindre. La maquette mise en œuvre est identique à celle de l'activité 1.



Adressage et numérotation

Le principe est le même que pour l'activité 1.

CONTEXTES	RÉSEAU IP	N° DE TÉLÉPHONE sur 4 chiffres	N° DE MESSAGERIE	SERVEUR ASTERISK
Finance	192.168.X.0/24	Commencent par X1 Exemple 1101, 1102 etc. pour l'équipe n°1	X199	192.168.X.200/24
Compta		Commencent par X2. Exemple : 1201, 1202, etc. pour l'équipe n° 1	X299	

Le plan d'adressage et de numérotation ci-dessus illustre deux contextes (finance et compta) qui veulent communiquer au sein d'un même site (un seul serveur Asterisk). Dans la numérotation mise en place, X représente un numéro de groupe de travail d'étudiants (équipe) pouvant aller de 1 à 8. Les captures d'écrans réalisées sont associées à l'équipe n°1.

Par exemple, 1101 représente le numéro du premier téléphone de l'équipe 1 appartenant au contexte *finance*. 1202 représente le numéro d'un deuxième téléphone de l'équipe 1 associé au contexte *compta*.

Vous disposez d'un document illustrant les différentes étapes à suivre.

Travail à faire

À l'aide du dossier documentaire fourni, vous devez réaliser l'ensemble des travaux. Vous prendrez soin de rédiger une documentation au fur et à mesure de votre avancement. Lors de chaque étape, vous devez indiquer les commandes utilisées vous permettant de tester vos configurations. Vos captures d'écran ne devront prendre en compte que la zone d'affichage nécessaire à vos démonstrations.

Travail à faire 1 : pré-requis

Dans cette première partie, vous devez commencer par vérifier que les travaux associés à l'activité 1 (plan d'appel sur un seul contexte) sont opérationnels. Comme pour l'activité 1, il faut utiliser le softphone Ekiga.

Q1.1. Commencer par vérifier que deux softphones peuvent se joindre au sein de chaque contexte. Pour cela, configurer les deux softphones au sein du contexte *finance* en testant un appel et un message vocal.

Q1.2. Réaliser la même manipulation que la question précédente en configurant les deux softphones au sein du contexte *compta*.

Q1.3. Vérifier que l'outil Wireshark est disponible sur les machines associées aux deux softphones.

Travail à faire 2 : liaison entre contextes

Dans cette deuxième partie, vous devez configurer une liaison entre le contexte *finance* et le contexte *compta*.

Q2.1. Adapter la configuration de votre serveur en conséquence. Tester un appel entre un softphone du contexte *finance* et un softphone du contexte *compta*. Faire les tests dans les deux sens. Tester la messagerie vocale.

Q2.2. Capturer, sur la console du serveur, la trace des échanges lors d'un appel entre deux contextes. Vérifier le bon fonctionnement dans les deux sens : un appel de *finance* vers *compta* et inversement.

Q2.3. Utiliser Wireshark pour reporter sur votre documentation les trames échangées.

Dossier documentaire

Document 1 – Appels entre contextes

D1.1 – Configuration des échanges entre contextes

Pour que des appels puissent aboutir entre deux utilisateurs appartenant à des contextes différents, il convient de créer des extensions supplémentaires indiquant au serveur le routage à appliquer pour joindre les postes demandés.

Dans notre cas, deux contextes sont à modifier : *finance* et *compta*. Ainsi, pour que les utilisateurs du contexte *finance* arrivent à joindre ceux du contexte *compta*, on peut ajouter l'extension suivante dans le contexte *finance* du fichier *extensions.conf*.

```
[finance]
....
;jonction entre les utilisateurs des contextes finance et compta.
exten => _12XX,1,Goto(compta,${EXTEN},1)
```

Il convient de faire de même dans le bloc associé au contexte *compta* en adaptant les paramètres. Bien entendu, il ne faut pas oublier de recharger la configuration pour que les modifications soient prises en compte par le serveur.

D1.2 - Trace des échanges entre contextes

La capture suivante montre la trace des échanges lorsque l'utilisateur *Utilisateur1* du contexte *finance* (numéro 1101) appelle l'utilisateur *Utilisateur5* du contexte *compta* (numéro 1201).

```
Connected to Asterisk 11.13.1~dfsg-2+b1 currently running on asterisk (pid = 1051)
== Using SIP RTP CoS mark 5
-- Executing [1201@finance:1] Goto("SIP/1101-00000004", "compta,1201,1") in new stack
-- Goto (compta,1201,1)
-- Executing [1201@compta:1] Dial("SIP/1101-00000004", "SIP/1201,20") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/1201
-- SIP/1201-00000005 is ringing
-- SIP/1201-00000005 answered SIP/1101-00000004
-- Remotely bridging SIP/1101-00000004 and SIP/1201-00000005
> 0x7f3b4803ec40 -- Probation passed - setting RTP source address to 192.168.1.11:5072
== Spawn extension (compta, 1201, 1) exited non-zero on 'SIP/1101-00000004'
```

La capture suivante montre la trace des échanges lorsque l'utilisateur *Utilisateur5* du contexte *compta* (numéro 1201) appelle l'utilisateur *Utilisateur1* du contexte *finance* (numéro 1101).

```
Connected to Asterisk 11.13.1~dfsg-2+b1 currently running on asterisk (pid = 1051)
== Using SIP RTP CoS mark 5
-- Executing [1101@compta:1] Goto("SIP/1201-00000006", "finance,1101,1") in new stack
-- Goto (finance,1101,1)
-- Executing [1101@finance:1] Dial("SIP/1201-00000006", "SIP/1101,20") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/1101
-- SIP/1101-00000007 is ringing
-- SIP/1101-00000007 answered SIP/1201-00000006
-- Remotely bridging SIP/1201-00000006 and SIP/1101-00000007
> 0x7f3b480346d0 -- Probation passed - setting RTP source address to 192.168.1.12:5074
== Spawn extension (finance, 1101, 1) exited non-zero on 'SIP/1201-00000006'
```

D1.3 - Traces associées à la messagerie vocale

Pour parvenir plus rapidement sur la messagerie vocale, il suffit que le destinataire refuse l'appel.



Dans cette capture d'écran, l'utilisateur *Utilisateur1* du contexte *finance* laisse un message vocal à l'utilisateur *Utilisateur5* du contexte *compta*.

```
== Using SIP RTP CoS mark 5
-- Executing [1201@finance:1] Goto("SIP/1101-0000000e", "compta,1201,1") in
new stack
-- Goto (compta,1201,1)
-- Executing [1201@compta:1] Dial("SIP/1101-0000000e", "SIP/1201,20") in new
stack
== Using SIP RTP CoS mark 5
-- Called SIP/1201
-- SIP/1201-0000000f is ringing
-- Got SIP response 603 "Decline" back from 192.168.1.12:5060
-- SIP/1201-0000000f is busy
== Everyone is busy/congested at this time (1:1/0/0)
-- Executing [1201@compta:2] VoiceMail("SIP/1101-0000000e", "1201@compta") i
n new stack
-- <SIP/1101-0000000e> Playing 'vm-intro.gsm' (language 'en')
-- > 0x7f3b480334d0 -- Probation passed - setting RTP source address to 192.
168.1.11:5086
-- <SIP/1101-0000000e> Playing 'beep.gsm' (language 'en')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/compta/1201/tmp/CnqSbd
format: wav, 0x7f3b48037788
-- User hung up
== Spawn extension (compta, 1201, 2) exited non-zero on 'SIP/1101-0000000e'
asterisk>CLI>
```

Dans cette capture d'écran, l'utilisateur *Utilisateur5* du contexte *compta* laisse un message vocal à l'utilisateur *Utilisateur1* du contexte *finance*.

```
== Using SIP RTP CoS mark 5
-- Executing [1101@compta:1] Goto("SIP/1201-00000012", "finance,1101,1") in
new stack
-- Goto (finance,1101,1)
-- Executing [1101@finance:1] Dial("SIP/1201-00000012", "SIP/1101,20") in ne
w stack
== Using SIP RTP CoS mark 5
-- Called SIP/1101
-- SIP/1101-00000013 is ringing
-- Got SIP response 603 "Decline" back from 192.168.1.11:5060
-- SIP/1101-00000013 is busy
== Everyone is busy/congested at this time (1:1/0/0)
-- Executing [1101@finance:2] VoiceMail("SIP/1201-00000012", "1101@finance")
in new stack
-- <SIP/1201-00000012> Playing 'vm-intro.gsm' (language 'en')
-- > 0x7f3b480120e0 -- Probation passed - setting RTP source address to 192.
168.1.12:5086
-- <SIP/1201-00000012> Playing 'beep.gsm' (language 'en')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/finance/1101/tmp/ognaZ
format: wav, 0x7f3b4800d5e8
-- User hung up
== Spawn extension (finance, 1101, 2) exited non-zero on 'SIP/1201-00000012'
asterisk>CLI>
```