

Marc SCHAEFER
Epervier 15
2053 Cernier
Ing. inf. dipl. EPFL

Conseil et réalisations en informatique libre http://www.cril.ch/
--

Email : schaefer@alphanet.ch
Tél. : +41 32 841 40 14
Fax : +41 32 841 40 81
Mobile : +41 79 502 56 92

Introduction à la téléphonie IP Asterisk sous GNU/Linux

cours de base

Marc SCHAEFER
schaefer@alphanet.ch

C+R
Informatique libre

25 avril 2008

Licence et droits d'auteurs

Ce cours est ©2006 CRIL - Marc SCHAEFER. Vous avez cependant le droit de le copier, transmettre, modifier et redistribuer, dans la mesure où vous respectez les termes de la licence GFDL et considérez l'invariant (les 2 premières pages).

Si vous ne désirez pas accepter les termes de la licence, je vous donne malgré tout le droit de consulter ce cours sans restrictions (ce qui devrait être évident !)

Dans tous les cas, vous devez accepter le fait que je décline toute responsabilité quant à l'utilisation que vous pourriez faire de ce cours et ne m'engage en rien à ce propos.

Table des matières

1	Introduction à la technologie voix-sur-IP	4
1.1	Qualité de service	5
1.2	Codecs	13
1.3	Protocoles de gestion de session et audio	17
1.4	Problèmes	23
2	Le central Asterisk	28
2.1	Introduction	30
2.2	Fonctions administratives de base	32
2.3	La console Asterisk	34
2.4	Matériel et interfaces	37
2.5	Le concept de <i>dial plan</i>	44
2.6	La Manager interface	54
3	Résoudre les problèmes	58
3.1	Logs systèmes et Asterisk	59
3.2	Debugging SIP	62
3.3	Debugging ISDN	63
3.4	Echo en ISDN	64
3.5	tcpdump et Ethereal (Wireshark)	65
4	Où trouver de la documentation	66
5	Corrigé des exercices	67

1. Introduction à la technologie voix-sur-IP

Contenu du chapitre

- notions de qualité de service (QoS)
- les codecs
- les protocoles de gestion de session et audio
- les problèmes courants liés à la voix-sur-IP

Buts du chapitre

- connaître les caractéristiques des codecs (notamment en bande passante et qualité)
- connaître les caractéristiques principales des protocoles IAX2, SIP et H.323 ainsi que leurs domaines d'utilisation
- connaître les problématiques de firewall, écho, jitter, traitement des DTMF et bande passante
- savoir dimensionner approximativement une connexion en fonction des codecs utilisés

Ce premier chapitre traite principalement de la technologie voix-sur-IP en général. Les codecs, protocoles et problèmes courants sont traités de manière informative. Quelques exemples de dimensionnement de liaison seront également proposés.

Lectures supplémentaires

- QoS en voix-sur-IP : <http://www.voip-info.org/wiki-QoS>
- codecs : http://en.wikipedia.org/wiki/Speech_encoding
- protocoles
 - IAX2** <http://en.wikipedia.org/wiki/IAX2>
 - SIP** http://fr.wikipedia.org/wiki/Session_Initiation_Protocol
 - H.323** <http://fr.wikipedia.org/wiki/H.323>

Qualité de service – 1.1

Contenu du sous-chapitre

- caractéristiques
- facteurs de délai
- réseau local / fournisseur / global
- qualité de service dans un cas pratique

Caractéristiques importantes

- délai perceptible et variation du délai (*jitter*)
- taux d'erreur ($\ll 1\%$)
- bande passante

Taux d'erreur

Un des paramètres les plus importants est le taux d'erreur. Un taux d'erreur élevé amènera à une communication très fortement dégradée voire impossible.

Une des causes d'un taux d'erreur élevé est la **congestion** : la surcharge temporaire d'une ligne ou d'un équipement, qui supprime alors les datagrammes IP excédentaires (en général sans priorité).

Echo

- un délai de plus de 100 ms environ¹ de bout en bout crée une perception de décalage et augmente la perception de gêne d'un écho résiduel
- en cas de présence d'écho, des délais plus courts sont parfois problématiques.

Bande passante

Le calcul de la bande passante nécessaire dépend du codec utilisé, du nombre de communications maximum et du surdébit de gestion des protocoles. Il faut donc s'assurer qu'en production, le système

¹max selon ITU-T G.114 150ms

n'autorise pas plus de connexions que possible. Des exemples de dimensionnement seront proposés dans le chapitre sur les codecs.

Facteurs de délai

- lignes
- composants/équipements
- surcharge / congestion
- MTU (maximum transmission unit) d'interface
- tampons (buffers) de compensation de jitter et suppression d'écho
- groupement d'échantillons dans un datagramme

3

No. -	Time	Source	Destination	Protocol	Info
6	7.958737	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 5ms N=0
7	7.959676	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 19ms AUTHREQ
8	7.968523	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 9ms AUTHREP
9	7.986108	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 29ms ACCEPT
10	8.004616	192.168.1.1	192.168.1.127	IAX2	Control, source call# 3, timestamp 32ms RINGING
11	8.016480	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 29ms ACK
12	8.032488	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 32ms ACK
13	8.048908	192.168.1.127	192.168.1.1	IAX2	Voice, source call# 2883, timestamp 100ms, Raw A-law data (6.7
14	8.049454	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 100ms ACK
15	8.112535	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 120ms, Raw A-law dat
16	8.112628	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 140ms, Raw A-law dat
17	8.112678	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 160ms, Raw A-law dat
18	8.176597	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 180ms, Raw A-law dat
19	8.176660	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 200ms, Raw A-law dat
20	8.192520	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 220ms, Raw A-law dat
21	8.192625	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 240ms, Raw A-law dat

▶ Frame 17 (206 bytes on wire, 206 bytes captured)					
▶ Ethernet II, Src: HewlettP_c9:c1:f1 (00:15:60:c9:c1:f1), Dst: CameoCom_72:65:f0 (00:40:f4:72:65:f0)					
▶ Internet Protocol, Src: 192.168.1.127 (192.168.1.127), Dst: 192.168.1.1 (192.168.1.1)					
▶ User Datagram Protocol, Src Port: iax (4569), Dst Port: iax (4569)					
▶ Inter-Asterisk eXchange v2					
▶ Data (160 bytes)					

Exercices

1. que peut-on faire pour assurer au trafic voix-sur-IP un fonctionnement correct même en cas de partage de la liaison ?
2. quel serait le MTU nécessaire sur une liaison ADSL 600/100 de manière à garantir un délai maximum dû au MTU de 50 ms ?
3. combien de données utiles dans le datagramme audio de l'exemple IAX2 à la figure ci-dessus ? cela correspond à combien d'échantillons audio (ou quel durée) ? combien d'entêtes ? déduisez le rendement et le débit total nécessaire.
4. quel sont les problèmes d'un MTU trop bas ?

5. les tampons de traitement d'un téléphone sont de 400 bytes, pour un codec 8 bit à 8000 Hz (G.711 A-law par exemple), que pouvez-vous dire pour le délai ajouté ?

Cas du réseau local

- si possible, utiliser un *réseau séparé*
 - physiquement séparé
 - évt. VLAN avec priorisation du trafic
- sans fil : privilégier les solutions garantissant la QoS (DECT)

Connexion à des fournisseurs

- dépendance du contrat (SLA) avec le fournisseur IP
- pour une meilleure fiabilité, choisir un fournisseur pour la solution complète
- p.ex. liaison ADSL et connexion au réseau téléphonique par le même opérateur, dédié ou partagé
- cas typique : remplacement de ligne ISDN PRI (30 canaux) par ADSL symétrique de capacité suffisante, connexion directe au fournisseur (ou garantie de bande passante ADSL Swisscom)
- réservation, prioritisation de trafic possible (partage), diverses méthodes suivant fournisseur

5

Cas du réseau global

- dépendance également avec d'autres fournisseurs
- cas général : *best effort*
- avoir une solution de secours est recommandé (liaison ISDN p.ex.)
- il est possible de mettre en place des solutions de priorisation de trafic (TOS)

Cas pratique

le problème

- cas *réseau global*
- pas de SLA particulier
- fournisseur distinct
- partage avec utilisation Internet classique

solutions

- routeur intelligent
 - MTU adapté et *tcp-MSS-clamping*
 - forçage du champ TOS pour spécifier la QoS en fonction p.ex. du numéro de port
 - priorisation au niveau du routeur lui-même

Lectures supplémentaires

- HOWTO Advanced Routing/Traffic Shaping Linux : <http://lartc.org/> et notamment l'exemple <http://lartc.org/wondershaper/>

Codecs – 1.2

Contenu du sous-chapitre

- quelques exemples de codecs courants et de leurs caractéristiques
- problème des licences
- dimensionnement de liaisons (bande passante, charge CPU)
- problèmes particuliers (fax, DTMF)

8

Exemples de codecs

nom	bande passante	qualité	licence
G.711 A/U-law	64 kbps	bonne	-
iLBC	15 kbps	correcte	-
G.729	10-30 kbps	bonne	propriétaire
GSM	15-35 kbps	mauvaise	-
FS-1015/LPC10	2.4 kbps	très mauvais	-
G.726 (ADPCM)	16, 24, 32, 40 kbps	bonne	-
Speex	2-44 kbps	bonne à mauvaise	-

9

Problème des brevets/licences

Seul Speex est garanti non encombré de brevets et donc peut être utilisé dans le monde entier sans licences. Les autres codecs sont en général également sans restrictions, avec la notable exception de G.729.

Utilisation des codecs en pratique

En pratique, on utilisera G.711 A-law dans le réseau interne, de manière à éviter une succession de transcodages et à diminuer le CPU nécessaire sur le central. Si des connexions externes ISDN sont utilisées, aucun transcodage n'est nécessaire en G.711 A-law.

Lors de connexions à des fournisseurs (p.ex. SIP, IAX2, etc), tout dépend de la bande passante disponible et du partage éventuel de celle-ci, et de la qualité exigée par le client. Le codec G.726 permet une division par deux de la bande passante nécessaire par rapport à G.711 A-law. Si supporté par le fournisseur, Speex peut être une bonne solution adaptable.

Lors du dimensionnement, on n'oubliera pas de tenir compte du surdébit de gestion des connexions (entêtes, protocoles, etc). En fonctionnement, on limitera le nombre maximum de connexions effectivement utilisées.

Toute conversion de codecs devra faire l'objet d'une évaluation de la charge CPU nécessaire, notamment par mesures.

Transport du fax

- fax analogique G3
- connecté en voix-sur-IP via un ATA, une carte ou un banc analogique, envoi en audio (T.30)
- le protocole G.3 est très résistant aux erreurs, aux prix de la perte de performance
- envoi d'un signal supprimant le traitement de canal
- seuls les codecs *sans perte* sont recommandés : G.711 A-law par exemple
- en dehors du réseau local, il est recommandé de transférer le fax *décodé* : T.38
- aux bornes du réseau voix-sur-ip, recréer le fax sous forme audio, si nécessaire.
- alternative : envoi spécifique sous forme d'image

<http://www.voip-info.org/wiki-Asterisk+fax>

Transport des DTMFs

- combinaison de deux fréquences fondamentales
- transmission en voix-sur-IP soit sous forme audio (**inband**), soit d'un message de contrôle (rfc2833 (RTP), INFO (SIP))
- la forme audio est la plus simple et compatible, mais ne fonctionne de façon relativement fiable qu'avec un codec *sans perte* sont recommandés : G.711 A-law par exemple
- en dehors du réseau local, il est recommandé de transformer en message de contrôle
- aux bornes du réseau voix-sur-ip, générer les fréquences fondamentales pour compatibilité

11

Protocoles de gestion de session et audio – 1.3

Contenu du sous-chapitre

- principes
- IAX2
- SIP
- H.323
- RTP (audio)

12

La figure 1 décrit l'établissement d'un appel IAX2 (trace `ethereal`). La figure vue précédemment est une trace audio.

Les figures 2 et 3 décrivent la même chose pour le protocole SIP.

IAX2

- un seul échange de données (multiplexage contrôle et données audio) sur un port connu, souvent 4569
- plusieurs communications entre deux équipements passeront également par la même connexion
- très efficace pour les *trunks*
- facile à firewaller/NATer.
- protocole spécifique à Asterisk, supporté par quelques équipements
- recommandé dans des liaisons inter-centraux Asterisk

13

No. .	Time	Source	Destination	Protocol	Info
2	0.000673	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 2, timestamp 3ms REGAUTH
3	0.016005	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2882, timestamp 16ms REGREQ
4	0.021384	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 2, timestamp 21ms REGACK
5	0.031967	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2882, timestamp 21ms ACK
6	7.958737	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 3ms NEW
7	7.959676	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 19ms AUTHREQ
8	7.968523	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 9ms AUTHREP
9	7.986108	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 29ms ACCEPT
10	8.004616	192.168.1.1	192.168.1.127	IAX2	Control, source call# 3, timestamp 32ms RINGING
11	8.016480	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 29ms ACK
12	8.032488	192.168.1.127	192.168.1.1	IAX2	IAX, source call# 2883, timestamp 32ms ACK
13	8.048908	192.168.1.127	192.168.1.1	IAX2	Voice, source call# 2883, timestamp 100ms, Raw A-law data (G.711)
14	8.049454	192.168.1.1	192.168.1.127	IAX2	IAX, source call# 3, timestamp 100ms ACK
15	8.112535	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 120ms, Raw A-law data
16	8.112628	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 140ms, Raw A-law data
17	8.112678	192.168.1.127	192.168.1.1	IAX2	Mini packet, source call# 2883, timestamp 160ms, Raw A-law data
[Lateness: -0.002000000 seconds]					
Outbound seq.no.: 0					
Inbound seq.no.: 0					
Type: IAX (6)					
IAX subclass: NEW (1)					
Information Element: Protocol version: 0x0002					
Information Element: Calling number: 700000000					
Information Element: Name of caller: iaxComm User					
Information Element: Desired codec format: Raw A-law data (G.711) (0x00000008)					
Information Element: Actual codec capability: 0x00000008					
Information Element: Username (peer or user) for authentication: voyager					
Information Element: Number/extension being called: 0800800800					
Information Element: Originally dialed DNID: 0800800800					

FIG. 1 – Appel (contrôle de) IAX2

SIP

- chaque communication nécessite un échange sur le port de contrôle SIP (établissement de session), souvent 5006
- mais aussi deux flux RTP sur des ports dynamiques (ou configurables en fixe)
- protocole standard IETF
- moins efficace qu'IAX2 pour les *trunks*
- plus complexe à firewaller/NATer (il y a des adresses et numéros de port dans les données de contrôles à modifier ; proxy possible)
- protocole ouvert, très compatible équipements
- recommandé pour l'accès client

14

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	80.83.46.147	212.117.200.148	SIP	Request: REGISTER sip:sip.backbone.ch
2	0.038912	212.117.200.148	80.83.46.147	SIP	Status: 200 OK (1 bindings)
3	10.118141	80.83.46.147	212.117.200.148	SIP/SDP	Request: INVITE sip:0800800800@sip.backbone.ch, with session i
4	10.141009	212.117.200.148	80.83.46.147	SIP	Status: 100 trying -- your call is important to us
5	10.152212	212.117.200.148	80.83.46.147	SIP	Status: 401 Unauthorized
6	10.152912	80.83.46.147	212.117.200.148	SIP	Request: ACK sip:0800800800@sip.backbone.ch
7	10.153232	80.83.46.147	212.117.200.148	SIP/SDP	Request: INVITE sip:0800800800@sip.backbone.ch, with session i
8	10.177334	212.117.200.148	80.83.46.147	SIP	Status: 100 trying -- your call is important to us
9	11.726303	212.117.200.148	80.83.46.147	SIP/SDP	Status: 183 Session Progress, with session description
10	11.735325	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42475, T
11	11.743093	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2625, Time
12	11.755016	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42476, T
13	11.762558	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2626, Time
14	11.775013	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42477, T
15	11.782567	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2627, Time
16	11.795000	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42478, T

▶ Frame 1 (638 bytes on wire, 638 bytes captured)
 ▶ Ethernet II, Src: CameoCom_1e:d6:64 (00:40:f4:1e:d6:64), Dst: Cisco_2b:c1:1c (00:04:dd:2b:c1:1c)
 ▶ Internet Protocol, Src: 80.83.46.147 (80.83.46.147), Dst: 212.117.200.148 (212.117.200.148)
 ▶ User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
 ▶ Session Initiation Protocol

FIG. 2 – Session SIP

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	80.83.46.147	212.117.200.148	SIP	Request: REGISTER sip:sip.backbone.ch
2	0.038912	212.117.200.148	80.83.46.147	SIP	Status: 200 OK (1 bindings)
3	10.118141	80.83.46.147	212.117.200.148	SIP/SDP	Request: INVITE sip:0800800800@sip.backbone.ch, with session i
4	10.141009	212.117.200.148	80.83.46.147	SIP	Status: 100 trying -- your call is important to us
5	10.152212	212.117.200.148	80.83.46.147	SIP	Status: 401 Unauthorized
6	10.152912	80.83.46.147	212.117.200.148	SIP	Request: ACK sip:0800800800@sip.backbone.ch
7	10.153232	80.83.46.147	212.117.200.148	SIP/SDP	Request: INVITE sip:0800800800@sip.backbone.ch, with session i
8	10.177334	212.117.200.148	80.83.46.147	SIP	Status: 100 trying -- your call is important to us
9	11.726303	212.117.200.148	80.83.46.147	SIP/SDP	Status: 183 Session Progress, with session description
10	11.735325	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42475, T
11	11.743093	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2625, Time
12	11.755016	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42476, T
13	11.762558	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2626, Time
14	11.775013	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42477, T
15	11.782567	212.117.200.80	80.83.46.147	RTP	Payload type=ITU-T G.711 PCMA, SSRC=449038416, Seq=2627, Time
16	11.795000	80.83.46.147	212.117.200.80	RTP	Payload type=ITU-T G.711 PCMA, SSRC=1514519518, Seq=42478, T

▶ Frame 10 (214 bytes on wire, 214 bytes captured)
 ▶ Ethernet II, Src: CameoCom_1e:d6:64 (00:40:f4:1e:d6:64), Dst: Cisco_2b:c1:1c (00:04:dd:2b:c1:1c)
 ▶ Internet Protocol, Src: 80.83.46.147 (80.83.46.147), Dst: 212.117.200.80 (212.117.200.80)
 ▶ User Datagram Protocol, Src Port: 19942 (19942), Dst Port: 17560 (17560)
 ▶ Real-Time Transport Protocol

FIG. 3 – Données audio RTP SIP

H.323

15

- un annuaire H.323 est nécessaire (*gatekeeper*)
- chaque communication nécessite un échange sur le port de contrôle H.323 (établissement de session)
- mais aussi deux flux de données sur des ports dynamiques (ou configurables en fixe)
- protocole standard ITU-T
- très complexe (similaire à Q.931 d'ISDN)
- très complexe à firewaller/NATer (il y a des adresses et numéros de port dans les données de contrôles à modifier)
- protocole ouvert, relativement compatible équipements
- recommandé pour la connexion de centraux propriétaires ne supportant que ce protocole

Problèmes – 1.4

Contenu du sous-chapitre

- la qualité perçue
- pare-feu (firewall)
- l'écho
- sécurité

16

La qualité perçue

Il est très difficile de déterminer la cause d'un problème de qualité perçue par l'utilisateur. Même si des tests normalisés existent (lecture de phonèmes standards et compréhension), cela reste malgré tout complexe.

Les problèmes de qualité sont dus à :

- un écho perçu trop tard et/ou trop fort (délais), ou l'absence total d'écho
- des gains de transmission ou réception trop élevés ou trop bas
- des blancs, indicateurs d'erreurs de transmission (congestion p.ex.) ou de variation de délai
- l'utilisation de codecs avec déformation (GSM p.ex.)
- des blips ou beeps, parfois dus à une détection de DTMF mal choisie ou configurée (in-band)

Pare-feu (firewall)

problèmes

- multitude des ports à ouvrir / dynamique
- problèmes en combinaison avec NAT/PAT

recommandations

- privilégier les protocoles simples comme IAX2 lorsque les firewalls sont traversés
- pour SIP, utiliser soit des ports de contrôle et de données audio (RTP) fixes, avec règles statiques dans le firewall, soit un proxy SIP.
- les clients SIP peuvent en général modifier les adresses (NAT) via divers protocoles comme STUN.
- éviter absolument les règles dynamiques de NAT, elles risquent de ne fonctionner que pour le premier équipement sur le même port, et causer des pannes difficiles à

déterminer.

- une option est d'utiliser un VPN sous forme de tunnel pour transporter le tout sans interaction avec le firewall.

Echo

- problème très complexe
- en général, supprimer l'écho le plus près de sa source (écho local ; pas écho distant)
- les équipements entièrement numériques n'ont en général pas d'écho
- l'écho est souvent causé par des parties analogiques (micro, ligne analogique) et aggravé par les délais introduits
- on peut jouer sur les gains et sur les supprimeurs d'écho logiciels intégrés dans les équipements ou logiciels.

19

Sécurité

le problème

- la plupart des protocoles supportent une authentification simple basée sur clés publiques
- une fois le système identifié, il n'y a en général pas de sécurité pour p.ex. les transferts d'appels ou la fin de connexion
- les données audio ne sont pas chiffrées
- en conséquence : attaques de **confidentialité** et de **déni de service** possibles, facilité par l'utilisation de numéros de ports connus et fixés

solutions possibles

- version sécurisée de RTP (SRTP) pour le flux audio
- protocoles spéciaux (p.ex. PGP ZFon)
- VPN sous-jacent

20

2. Le central Asterisk

Contenu du chapitre

- introduction (licence, packaging)
- fonctions administratives de base
- la console Asterisk
- matériel et interfaces
- le concept de *dial plan*
- les applications standard
- exercices pratiques
- liens à d'autres applications

Ce chapitre traite du central Asterisk et de ses interconnexions à des protocoles et matériels. Il traite également du plan de numérotation (*dial plan*) et de configurabilité en temps réel.

Lectures supplémentaires

- console Asterisk et principales commandes : <http://www.voip-info.org/wiki/index.php?page=Asterisk+CLI>
- le *dial plan* : <http://www.voip-info.org/wiki/view/Asterisk+howto+dial+plan>
- les applications : <http://www.voip-info.org/wiki/index.php?page=Asterisk+-+documentation+of+application+commands>

Buts du chapitre

- savoir installer, mettre à jour, démarrer, arrêter et configurer la base d'Asterisk
- pouvoir interagir avec la console, déterminer les configurations, interfaces et connexions
- connaître le support matériel et sa configuration, et quelques cas particuliers de services spécifiques
- concevoir et réaliser un *dial plan*, y compris quelques spécialités utilisant des macros, du traitement d'erreur ou du contexte (variables)

22

Introduction – 2.1

Licence

double (*dual-licensed*)

- Asterisk est développé par Digium, Inc.
- soit libre (GPL), soit propriétaire (OEM, version testée, certains pilotes restreints, codecs problématiques)

Branche bristuff

- développée par un spécialiste d'ISDN
- GPL (uniquement, la version propriétaire de Digium n'a pas le support notamment pour les chips HFC via interface Zaptel)
- bien testée

*Packaging***CRIL propose :**

- version GPL 1.0.9, complétée bristuff, testée
- maintenue par un spécialiste d'ISDN
- packaging compatible Debian (/etc/apt/sources.list)

```
deb http://packages.cril.ch/cril/debian/packages/ \
    sarge cril/ASTERISK
```
- pilotes kernel HFC zaptel :

```
http://login.alphanet.ch/~schaefer/tmp/asterisk-hfc-drivers-1.0.9rc8j-easier-to-build.tar.gz
```
- préconfiguration standard
- si désiré, support FAI (Fully Automatic Installation) ou images d'installation.

Cartes HFC sans bristuff

Il est certes possibles de faire fonctionner les cartes HFC également avec la version propriétaire (ou la version GPL non bristuff), mais cela nécessite du support dans le kernel (mISDN) qui n'est pas encore complet ni fiable, même en kernel Linux 2.6.

Fonctions administratives de base – 2.2

Démarrage et arrêt

- démarrage
 - manuel : `/etc/init.d/asterisk start`
 - automatique via `init.d` : voir `/etc/rc2.d/S*asterisk`
- arrêt
 - manuel : `/etc/init.d/asterisk stop`
 - en console Asterisk : `stop now`
 - automatique via `init.d` : voir `/etc/rc?.d/K*asterisk`

25

Configuration de base

- `zaphfc` : `/etc/init.d/hfc, /etc/zaptel.conf, /usr/local/HFC`
- `zapata` : `/etc/asterisk/zapata.conf`
- `SIP` : `/etc/asterisk/sip.conf`
- `composants logiciels` : `/etc/asterisk/modules.conf`
- `dial plan` : `/etc/asterisk/extensions.conf`
- `voicemail` : `/etc/asterisk/voicemail.conf`
- `conférences` : `/etc/asterisk/meetme.conf`
- `musiques d'attente` : `/etc/asterisk/musiconhold.conf`

26

Exercices

1. vérifiez que Asterisk est démarré (`ps auxw | grep asterisk`)
2. stoppez le, vérifiez, puis redémarrez via `init.d`
3. lancez la console Asterisk : `asterisk -rc`, documentez-vous sur la commande `stop` (`help stop`) et stoppez Asterisk via la console de manière à ce que l'arrêt se produise lorsqu'il n'y aura plus d'appels en cours, puis redémarrez via `init.d`
4. assurez-vous que le service est démarré au démarrage de la machine
5. consultez rapidement :
 - `/etc/asterisk/modules.conf`,
 - `/etc/asterisk/extensions.conf`
 - `/etc/asterisk/sip.conf`
6. consultez <http://www.voip-info.org/wiki-Asterisk+GUI>

La console Asterisk – 2.3

Connexion

- asterisk -rc
- options supplémentaires
 - v** niveau de debug (plusieurs spécifiables)
 - n** texte seulement

27

Commandes principales

commande	description
help et help commande	aide interne
sip show peers	voir les téléphones et fournisseurs SIP
sip show registry	voir les enregistrements à des fournisseurs SIP
stop now	arrêt (voir aussi help stop pour les variantes)
show channels	voir les communications en cours
show channel XXXX	détails sur un canal donné
zap show channels	canaux Zaptel
pri show span 1	détails sur un canal Zaptel (ISDN) donné
show codecs	voir les numéros de codecs
show dialplan	voir les différents contextes et extensions

28

Prise en compte de nouvelles configurations

commande	description
reload	reconfiguration globale
reload CONFIG	reconfiguration d'une configuration particulière
sip reload	reconfiguration SIP

29

Exercices

1. connectez-vous à la console
2. listez les téléphones SIP connectés ; faites un appel, vérifiez que l'appel est visible. Déterminez le codec et d'autres paramètres pour l'appel en cours.
3. modifiez le *dial plan*, activez la modification et testez.

Matériel et interfaces – 2.4

Concept de channel

- un channel est une interface (SIP, ISDN, IAX2, etc) activée
- on réfère aussi aux pilotes concernés (p.ex. `chan_sip`)

Interfaces téléphones

- analogique (FXO ; carte Digium, interface Zaptel) : Zap/num
- ISDN : rare, voir interface fournisseurs (avec inversion et NT, possible sur HFC)
- voix-sur-IP
 - SIP (analogique via ATA, ou téléphone SIP, softphone) : SIP/ident
 - IAX2, H.323

31

Exemples de configuration

Téléphone SIP (sip.conf)

```
/etc/asterisk/sip.conf
```

```
[200]
callerid=200
type=friend
context=local-sip-in
host=dynamic
secret=DEMOPW
username=200
dtmfmode=rfc2833
qualify=1000
disallow=all
allow=ulaw
allow=alaw
canreinvite=no
nat=no
mailbox=200
```

Téléphone IAX2 (iax.conf)

Attention, il y a deux versions du protocole IAX ! Certains softphones ne comprennent que la version 1, qui est obsolète aujourd'hui.

```
/etc/asterisk/iax.conf
```

```
[220]
type=friend
mailbox=220
accountcode=220
username=220
host=dynamic
auth=md5,plaintext,rsa
secret=DEMOPW
context=local-sip-in
disallow=all
allow=alaw
allow=ulaw
```

Interfaces fournisseurs

- analogique : rare en Suisse (FXS, carte Digium)
- ISDN
 - BRI (MultiLineISDN, P2MP, bus S) : cartes HFC
 - BRI/DDI (BusinessISDN, DDI, P2P, T) : cartes HFC
 - PRI (E1) : cartes HFC PRI
- voix-sur-IP (p.ex. SIPCALL, Stanaphone, Intertel, etc)
 - SIP SIP/ident, register =>
 - IAX2, H.323

Fournisseur SIP

```
/etc/asterisk/sip.conf
```

```
[general]
useragent=irgendwas

register => 41325112156:DEMOPW@sip.backbone.ch:5060

[sipcall]
type=peer
secret=DEMOPW
username=41325112156
fromuser=41325112156
callerid=41325112156
host=sip.backbone.ch
qualify=no
nat=no
disallow=all
allow=alaw
canreinvite=no
context=sipcall-dial-in
insecure=very
```


Note : pour différencier entre plusieurs comptes du fournisseur, suivant les cas on pourra différencier avec le numéro appelé. Sinon, le plus simple est d'ajouter /1 au `register` ci-dessus et le numéro appelé sera 1 (respectivement 2, etc).

Fournisseur ISDN (BRI)

```
/etc/init.d/hfc
```

```
ZAPHFC_MODE=0
```

```
/etc/zaptel.conf
```

```
defaultzone=us
```

```
# First board (TE)
span=1,1,3,ccs,ami
bchan=1-2
dchan=3
```

```
# Second board (NT)
span=1,1,3,ccs,ami
bchan=4-5
dchan=6
```

```
/etc/asterisk/zapata.conf
```

```
; TE mode
pridialplan=local
prilocaldialplan=unknown
immediate=no
callerid=asreceived
signalling = bri_cpe_ptmp
context = isdn-dial-in
```

```
echocancel=yes
echotraining=no
echocancelwhenbridged=no
```

```
rxgain=-5.0
txgain=-7.0
```

```
group = 1
channel => 1-2
```

Interfaces centraux et équipements

- similaire au problème de la connexion de téléphones
- en ISDN, support inversion NT avec HFC
- cas particuliers : interfaces GSM (ISDN BRI/PRI ou SIP), racks de raccordement analogiques ou ISDN (ISDN PRI, SIP)

33

Téléphone ISDN

```
/etc/init.d/hfc
```

```
ZAPHFC_MODE=2
```

```
/etc/asterisk/zapata.conf
```

```
; p2mp NT mode  
signalling = bri_net_ptmp  
context=isdn-local-bus  
callerid=7516862  
pridialplan=local  
prilocaldialplan=local ; remove addition of first 0  
echocancel=no  
  
immediate=no  
  
group = 2  
channel => 4-5
```

Exercices

1. configuration d'un téléphone SIP (codec, écho, silence, passage DTMF correct et testé, appel, réception d'appel, dial-plan interne éventuel)
2. configuration d'une liaison ISDN ; configuration du MSN en sortie et du groupe de lignes.
3. configuration d'un lien SIP en sortie sur un fournisseur, vérification d'enregistrement (`registry`), test d'appel et de réception d'appel.

Le concept de dial plan – 2.5

Concepts

- le plan de numérotation est un fichier de configuration, `extensions.conf`, qui décrit les numéros composables (entièrement ou via des expressions génériques simples)
- des séquences complexes peuvent être définies, par exemple sous forme de macros.
- on peut le voir comme un système *event driven*
- des variables de contexte peuvent être consultées, affectées ou effacées (soit pour cette session, soit globalement)
- deux concepts importants :

context un ensemble d'extensions définies, servant à définir des sous-ensembles activables suivant un type d'appel p.ex..

extension définition d'une réaction correspondante à une action de l'utilisateur (composition d'un numéro, touche étoile, etc

Exemple d'un dial plan

Généralités

```
/etc/asterisk/extensions.conf
```

```
[service-numbers]
exten => 303,1,Meetme(1234)
exten => 304,1,VoicemailMain
exten => 305,1,Goto(demo,s,1)
exten => 307,1,MusicOnHold(default)
exten => 308,1,Echo
exten => 311,1,Directory

[local-extensions]
include => service-numbers
exten => _2XX,1,Dial,Sip/${EXTEN}
exten => _4XX,1,Dial,Zap/g2/${EXTEN}

[local-sip-in]
include => local-extensions
include => isdn-dial-out
```

- génération : idéalement généré dynamiquement, p.ex. d'une base de données (soit par fichiers générés, concept de `pbx-admin`, soit via lien direct à la base de données).

35

Fournisseur SIP

```
/etc/asterisk/extensions.conf
```

```
[sipcall-dial-in]
exten => s,1,Dial(Sip/200)

[sipcall-dial-out]
exten => _9100XXXXXXXXXX,1,SetCallerID(41325112156)
exten => _9100XXXXXXXXXX,2,SetCIDName(41325112156)
exten => _9100XXXXXXXXXX,3,Dial(Sip/${EXTEN:2}@sipcall)
```

Fournisseur ISDN (BRI)

```
/etc/asterisk/extensions.conf
```

```
[isdn-dial-out]
exten => _0.,1,Dial(Zap/g1/${EXTEN})
exten => _1.,1,Dial(Zap/g1/${EXTEN})

[isdn-dial-in]
exten => 7516862,1,Dial(Sip/200)
```

Téléphone ISDN

```
/etc/asterisk/extensions.conf
```

```
[isdn-local-bus]
exten => s,1,DigitTimeout,3
exten => s,2,ResponseTimeout,5
exten => s,3,SetCallerId(400)
include => local-sip-in
```

Traitement des extensions

- dans un *contexte*
- format
 - exten => ID, SEQUENCE, ACTION, paramètre, paramètre, ...
 - ID : numéro, touche ou expression générique *wildcard*, ou encore lettre spéciale (s : défaut, i : touche invalide, t : timeout, o : operator, T : global timeout)
 - SEQUENCE : numéro de séquence (cf BASIC)
 - ACTION : composition, réponse, raccrochage, application, etc
 - paramètres : variables suivant l’ACTION
- traitement en séquence (sauf traitement d’erreur d’applications et branchements avec Goto)
- préfixage : on peut spécifier un numéro p.ex. par un préfixe

Exemples

```
[demo1]
exten => s,1,Answer
exten => s,2,Wait(1)
exten => s,3,Playback(bienvenue)

; juste touche 1
exten => 1,1,Echo
exten => 1,2,Hangup

; touche *
exten => *,1,Hangup

; numéro commençant par 2, à 4 chiffres
; (compose les 3 derniers chiffres, en Sip)
exten => _2XXX,1,Dial(Sip/${EXTEN:1})

exten => t,1,Goto(demo1,s,3) ; timeout: retour au message de bienve

exten => i,1,Playback(invalid)
exten => i,2,Goto(demo1,s,3) ; touche invalide: retour
```

Traitement d'erreur

En général, le saut est fait à extension + N, où N est défini dans la documentation de la fonction utilisée.

```
exten => s,1,SetVar(line=${CHANNEL})
```

```
exten => s,2,DBget(temp=MORE-MSN/${line}/${EXTEN})
```

```
; en cas d'erreur (variable non existante), saute à 2 + 101
```

```
exten => s,3,SetVar(EXTEN=${temp})
```

```
exten => s,4,Goto(internal-extensions,${EXTEN},1)
```

```
; default extension
```

```
exten => s,103,DBget(temp=MORE-MSN/${line}/DEFAULT)
```

```
exten => s,104,SetVar(EXTEN=${temp})
```

```
exten => s,105,Goto(s,3)
```

Exercices

1. concevez un dial-plan entier pour l'entreprise (choix de préfixe(s) de sortie, plan de numérotation d'internes, numérotation abrégée, routage d'appel SIP/ISDN, services.
2. configurez l'extension 9 via l'opérateur Swisscom (10740), de manière à ce que 90328414014 devienne 107400328414014 lors d'un appel ISDN.
3. configurez un opérateur SIP qui demande le numéro en format international (41328414014), sans le double 0, dans tous les cas.
4. faites un appel parallèle sur les postes 200 et 201 pour une extension donnée, pendant 30 secondes, puis un appel de 203 indéfiniment si n'a pas fonctionné (indication : &)

Macros

Remplaçons le `Dial` dans l'exemple par un appel de macro :

```
exten => _2XXX,1,Macro(std-dial,${EXTEN},Sip/${EXTEN:1})

    [macro-std-dial]
    ; ${ARG1}: extension; ${ARG2}: device
    exten => s,1,Dial,${ARG2}|60|r
    exten => s,2,SetVar(PRI_CAUSE=${HANGUPCAUSE})
    exten => s,3,Goto(s-${DIALSTATUS},1)

    exten => s-NOANSWER,1,Voicemail(u${ARG1})
    exten => s-NOANSWER,2,Hangup
    exten => s-BUSY,1,Voicemail(b${ARG1})
    exten => s-BUSY,2,Hangup

    exten => _s-.,1,Hangup
```

38

Applications standard

Dial Composition.

Directory Annuaire automatique basé sur `voicemail.conf`

Voicemail Messages vocaux, par utilisateur.

Echo Test d'écho.

MusicOnHold Musique d'attente.

MeetMe Conférence simple (ne marche bien qu'avec zaptel en fonction)

39

Exercices

1. testez le voicemail (y compris envoi par e-mail)
2. déposez un nouveau fichier son (attention aux droits d'auteur) comme musique d'attente.
3. testez le mode conférence.
4. faites en sorte que la fonction de répertoire (311) fonctionne.

Données contextuelles

- SetVar : configurer une variable de session
- DBget, DBset, DBdel : variables persistantes (astdb) et globales
- un moyen simple de paramétrer Asterisk du point de vue de l'utilisateur :
 - redirections d'appel configurables
 - MSN en sorties suivant numéro interne
 - délais, groupes d'appel, etc

40

Exemple d'utilisation des données contextuelles : redirection d'appel configurable**Fonctions de configuration**

```
[call-forwarding-setup-support]
exten => *50,1,DBdel(CF/${CALLERIDNUM})
exten => *50,2,Answer
exten => *50,3,Wait(1)
exten => *50,4,Playback(forwarding-cleared)
exten => *50,5,Hangup

exten => *51,1,DBput(CF/${CALLERIDNUM}=voicemail)
exten => *51,2,Answer
exten => *51,3,Wait(1)
exten => *51,4,Playback(forwarding-to-voicemail)
exten => *51,5,Hangup

exten => *_52X.,1,DBput(CF/${CALLERIDNUM}=${EXTEN:3})
exten => *_52X.,2,Goto(call-forwarding-setup-support,*59,1)

exten => *53,1,DBput(CF/${CALLERIDNUM}=no-answer)
```

```
exten => *53,2,Answer
exten => *53,3,Wait(1)
exten => *53,4,Playback(forwarding-no-answer)
exten => *53,5,Hangup

exten => *59,1,Answer
exten => *59,2,Wait(1)
exten => *59,3,DBget(temp=CF/${CALLERIDNUM})
exten => *59,4,GotoIf("${temp}" = "voicemail"?20)
exten => *59,5,GotoIf("${temp}" = "no-answer"?30)
exten => *59,6,Playback(forwarding-to-number)
exten => *59,7,SayDigits(${temp})
exten => *59,8,Hangup

exten => *59,20,Goto(call-forwarding-setup-support,*51,4)

exten => *59,30,Goto(call-forwarding-setup-support,*53,4)

exten => *59,104,Playback(forwarding-none)
exten => *59,105,Hangup
```

Fonctions d'utilisation

```
[macro-call-forwarding-context]
; ${ARG1} - Extension
; ${ARG2} - where to dial
exten => s,1,DBget(temp=CF/${ARG1})
exten => s,2,GotoIf("${temp}" = "voicemail"?20)
exten => s,3,GotoIf("${temp}" = "no-answer"?30)

; Dial specified number
exten => s,4,GotoIf("${temp:0:1}" = "0"?10)
exten => s,5,Macro(dial-internal,${temp})

exten => s,10,Goto(isdn-dial-out,${temp},1)

; Go voicemail
exten => s,20,Voicemail(u${ARG1})

; No answer
exten => s,30,Hangup

; No entry; go default.
exten => s,102,Macro(dial-internal,${ARG1},${ARG2})
```

Exercices

1. faites en sorte que le numéro (MSN) utilisé comme source en sortie dépende de l'extension d'où provient l'appel.
2. mettez en place de la redirection d'appel configurable.

La Manager interface – 2.6

41

- contrôle détaillé d'Asterisk et suivi temps-réel des actions
- mode interactif : soit configuration dans `manager.conf` puis accès via TCP (port 5038)
- mode batch : dépôt d'un fichier formaté dans `/var/spool/asterisk/outgoing` (pas de retour !)
- package CRIL en Perl d'abstraction : `asterisk-perl`, exemples en CGI dans `pbx-admin`
- interface TAPI pour Microsoft (`asttapi`, <http://www.voip-info.org/wiki/view/Asterisk+TAPI>)
- en Asterisk 1.4 : AJAM (<http://www.voip-info.org/wiki/view/Aynchronous+Javascript+Asterisk+Manager+%28AJAM%29>) pour applications WWW

Exemple de contrôle d'appel

```

schaef@shakotay:~/asterisk-perl$ telnet localhost 5038
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Asterisk Call Manager/1.0
Action: login
Username: alphanet
Secret: DEMOPASSWORD

Response: Success
Message: Authentication accepted

Action: Command
Command: Show Channels

Response: Follows
Channel (Context Extension Pri ) State Appl. Data
IAX2/cflash-local/1 ( s 1 ) Up Bridged
SIP/200-53c7 (macro-dial-extension s 1 ) Up Dial 200
2 active channel(s)
--END COMMAND--

```

```
Action: Originate  
Channel: SIP/200  
Context: sip-dial-out  
MaxRetries: 1  
RetryTime: 60  
WaitTime: 30  
Extension: 0795025692  
Priority: 1
```

Exemples d'événements

```
Event: Link  
Channel1: SIP/200-53c7  
Channel2: IAX2/cflash-local/1  
Uniqueid1: asterisk-19452-1159193538.40  
Uniqueid2: asterisk-19452-1159193538.41
```

Application : fax

rxfax Réception de fax, par une Application du *dial plan*.

txfax Envoi de fax, via la Manager Interface.

42

Réception de fax

```
[macro-faxreceive]
exten => s,1,SetVar(FAXFILE=/var/spool/asterisk/tmp/${UNIQUEID}.tif)
exten => s,2,SetVar(EMAILADDR=schaefer@alphanet.ch)
exten => s,3,rxfax(${FAXFILE})

[fx]
exten => s,1,Macro(faxreceive)
exten => h,1,system(/usr/local/sbin/mailfax ${FAXFILE} ${EMAILADDR} \
    "${CALLERIDNUM} ${CALLERIDNAME}")
```


Envoi de fax

Il faut passer par l'interface de gestion d'Asterisk (*manager interface*).

```
cat > /var/spool/asterisk/tmp/tmp_$$ <<EOF
Channel: Zap/g2/0328414081
MaxRetries: 0
WaitTime: 20
Application: txfax
Data: /tmp/bla.tiff|caller
EOF
```

```
mv /var/spool/asterisk/tmp/tmp_$$ \
   /var/spool/asterisk/outgoing/fax_$$
```

Noter que cette version ne supporte pas de notification du résultat. A part l'interface Manager de notification, ou le parsing des logs, la solution la plus simple est d'appeler un contexte et une séquence.

3. Résoudre les problèmes

Contenu du chapitre

- logs systèmes, logs asterisk
- le debugging avec la console Asterisk
- cas particuliers : SIP et ISDN
- écho en ISDN
- tcpdump et ethereal

Buts du chapitre

- être capable de déterminer la raison d'un problème de configuration ou d'interopérabilité simple

Ce chapitre propose quelques pistes et méthodes générales de debugging de base en cas de problèmes avec Asterisk. Un debugging plus profond nécessite des connaissances étendues en protocoles IP ou ISDN par exemple et n'est pas l'objet de ce chapitre. Quelques exercices pratiques sont proposés.

Logs systèmes et Asterisk – 3.1

Où se trouvent les logs

44

- on peut configurer Asterisk pour passer par le daemon système `syslogd` (pour centralisation, log à distance, etc)
- par défaut Asterisk écrit ses logs directement dans `/var/log/asterisk/`
- on peut configurer le niveau de détail pour Asterisk
- on peut afficher les logs interactivement avec la console, et augmenter le niveau de debugging si nécessaire, soit par argument, soit par commande interne
- les logs systèmes (p.ex. pilotes matériels Zap/HFC) sont dans `/var/log/syslog`, `/var/log/dmesg` (et/ou commande `dmesg` pour le kernel debug ring buffer)
- certains logs (p.ex. d'appel) sont disponibles en CSV (fichier texte formaté), soit via une connexion ODBC sur un SGBD.

Augmentation du niveau de debugging

méthode 1 : configuration, non interactive

- modifier `/etc/asterisk/logger.conf`
 - changer dans la section `[general]`, la variable `messages` (ou choisir un autre nom, sauf `console`, pour le nouveau fichier sous `/var/log/asterisk/`)
 - exemple : `messages => notice,warning,error,debug,verbose`
- effectuer la commande de la console d'Asterisk : `logger reload` (ou relancer `asterisk`)
- pour modifier le niveau de debugging par défaut de la console, utiliser le nom `console`

méthode 2 : interactive

- lancer la console asterisk avec autant d'options `-v` que désiré
- par exemple : `asterisk -rcvvvvv`

Lancement contrôlé d'Asterisk

Il est également possible de lancer Asterisk *manuellement* de manière à ce que la console ne détache pas et soit accessible, ou à défaut les messages d'erreurs directement.

Exemple :

```
su --shell=/bin/sh - asterisk -c "/usr/sbin/asterisk -cnvvvv"
```

```
Parsing '/etc/asterisk/asterisk.conf': Found  
Asterisk 1.0.9-BRIstuffed-0.2.0-RC8j, Copyright (C) 1999-2004 Digium.  
Written by Mark Spencer <markster@digium.com>
```

```
[ ... ]
```

```
*CLI>
```

La seule façon d'arrêter est alors via `stop now`.

Cette façon de faire génère beaucoup de sortie. L'utilitaire `script` permet de sauvegarder celle-ci dans un fichier, par défaut `typescript` :

```
script
```

```
... commandes ...
```

```
exit
```

```
more typescript
```

Exercices

1. ajouter une ligne `load => whatever.so` dans le fichier `/etc/asterisk/modules.conf`. Redémarrer Asterisk (`/etc/init.d/asterisk restart`). Vérifier ensuite qu'Asterisk n'a pas redémarré (`ps auxw | grep asterisk`). Trouver le problème dans les logs, puis lancez Asterisk *manuellement*. Sauvegardez (utilitaire `script`, ou fonction de logging de `screen`) toute la sortie du programme et consultez-là. corriger et s'assurer que tout marche à nouveau.
2. connectez-vous à la console Asterisk à distance (remote) (`asterisk -rcvvv`), puis déconnectez-vous (`quit`)

Debugging SIP – 3.2

- sip debug ip HOST[:PORT] : debug restreint à une adresse/un port
- sip debug peer NAME : limité à un peer
- sip debug : global
- sip no debug : suppression debugging

46

aussi possible via tcpdump ou Ethereal

Debugging ISDN – 3.3

- `pri debug span N` : **seulement** sur canal N
- `pri intense debug span N` : **debug** additionnel
- `pri show span N` : **informations**
- `pri no debug span N` : **suppression** debugging span N

Echo en ISDN – 3.4

Déterminer et diminuer ou supprimer l'écho dérangeant

- cas le plus courant : écho du côté analogique (p.ex. central ISDN/analogique)
- certains téléphones (SIP, DECT) ont de la suppression d'écho intégrée
- on peut adapter les gains Zapata
- on peut aussi activer la suppression d'écho logicielle (Zapata) d'Asterisk

48

Méthode

- déterminer le niveau de transmission et de réception : outil `ztmonitor` (pendant une communication problématique)
- adapter `txgain` et `rxgain` dans `/etc/asterisk/zapata.conf`
- activer la suppression d'écho logicielle : `echocancel=yes`
- s'assurer que l'apprentissage est désactivé : `echotraining=no`
- attention aux gains internes de certains périphériques

Exemple

Soit le canal (span) Zapata connecté numéro 1 :

```
# /usr/local/HFC/SRC/zaptel-1.0.9/ztmonitor 1 -v
```

```
Visual Audio Levels.
```

```
-----
```

```
Use zapata.conf file to adjust the gains if needed.
```

```
( # = Audio Level * = Max Audio Hit )
```

```
<----- (RX) -----> <----- (TX) ----->
```

```
#####
```

on pourrait diminuer les gains ainsi :

```
rxgain=-5.0
```

Voir aussi <http://www.asteriskdocs.org/modules/tinycontent/content/docbook/current/docs-html/x1695.html> et <http://www.voip-info.org/wiki/view/Grandstream-GXP-2000+-+Solving+Echo+Problems>

tcpdump et Ethereal (Wireshark) – 3.5

– exemple d'usage de `tcpdump` pour capturer des trames :

```
tcpdump -i eth0 -n -s 0 -w capture-file
```

– visualisation graphique avec Ethereal (aussi sur une autre machine) :

```
ethereal capture-file
```

– utilisation de `tcpdump` pour visualisation locale :

```
tcpdump -i eth0 -n -vvv
```

– restrictions diverses (filtres de capture) :

```
tcpdump -i eth0 -n -vvv 'udp && port 4569'
```

– voir toutes les données décodées :

```
tcpdump -i eth0 -n -vvv -X -s 0 'udp && port 4569'
```

Ethereal est plus simple à utiliser (p.ex. sous client Microsoft Windows), mais permet également de visualiser des captures effectuées sur un serveur distant avec le simple outil `tcpdump` !

Exercices

1. visualiser un appel ISDN de l'extérieur, notamment le MSN présenté et le numéro appelé. Déconfigurer le debugging.
2. tracer l'entier d'un appel SIP et trouver les informations intéressantes (avec `tcpdump`/Ethereal et debugging Asterisk)

4. Où trouver de la documentation

- Asterisk et voix-sur-IP : <http://www.voip-info.org/> (Wiki)
- http://fr.wikipedia.org/wiki/Asterisk_%28logiciel%29
- <http://fr.wikipedia.org/wiki/VoIP>

5. Corrigé des exercices

Introduction à la technologie voix-sur-IP

Facteurs de délai

1. utiliser des queues de priorité par classe de trafic, adapter le MTU. Dans le cas de l'ADSL, demander éventuellement au fournisseur s'il peut changer le mode de correction d'erreur Reed-Solomon en mode *fast-path*¹ de manière à gagner 30-45 ms.
2. on a
$$m = 100 \text{ kBit/s} * 50 \text{ ms} = 5000 \text{ bit}$$
donc le MTU vaut au maximum 500 bytes (Ethernet 1500, ADSL 488 bien souvent, donc ok dans ce cas)
3. données utiles : 160 bytes (20 ms à 8000 Hz x 8 bits) ; données d'entête 46 bytes (206-160, dont 14 bytes entête Ethernet, 20 bytes entête IP, 8 bytes entête UDP, 4 bytes entête IAX2 mini-paquet) ; rendement 78%, débit total en A-law avec entêtes : $1.29 * 64 \text{ kbit/s}$ soit 83 kbit/s.
4. en règle générale, cela baisse l'efficacité, car le rapport entre données utiles et entêtes baisse ; l'autre problème est que les datagrammes UDP sont perdus si taille < MTU, et qu'il y a en général plusieurs échantillons audio dans un datagramme UDP.
5. 8000 bytes par seconde, donc 400 bytes correspond à 50 ms.

Le central Asterisk

Configuration de base

1. `ps auxw | grep asterisk`
2. `/etc/init.d/asterisk stop; /etc/init.d/asterisk start`
3. stop when convenient
4. `ls /etc/rc?.d/S*asterrisk`
5. doc
6. doc

Prise en compte de nouvelles configurations

1. `asterisk -rc`
2. `sip show peers, show channels, show channel XXX, show codecs`
3. `reload extensions`

¹http://fr.wikipedia.org/wiki/Asymmetric_Digital_Subscriber_Line#Latence_de_transmission

Interfaces centraux et équipements

1. manipulation
2. manipulation
3. manipulation

Traitement d'erreur

1. manipulation
2. `exten => _9.,1,Dial(Zap/g1/10740${EXTEN:1})`
3. `exten => _00.,Dial(Sip/${EXTEN:2}@telco)`
`exten => _0.,Dial(Sip/41${EXTEN:1}@telco)`
4. `exten => 34,1,Dial(Sip/200&Sip/201|30|r)`
`exten => 34,2,Dial(Sip/203)`

Applications standard

1. manipulation
2. manipulation
3. manipulation
4. manipulation

Données contextuelles

1. manipulation
2. manipulation

Résoudre les problèmes

Augmentation du niveau de debugging

1. manipulation
2. manipulation

tcpdump et Ethereal (Wireshark)

1. `pri debug span N,pri no debug`
2. `sip debug`