We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

6

VoIP System for Enterprise Network

Moo Wan Kim and Fumikazu Iseki
Tokyo University of Information Sciences
Japan

1. Introduction

This chapter describe VoIP system for the enterprise network (e.g. company, university) based on Asterisk(http://www.asterisk.org). Asterisk is a kind of open source software to implement IP-PBX system and supports various necessary protocols to realize the VoIP system such as SIP, H.323, MGCP, SCCP.

First the main ideas and development process are described based on the VoIP system that we have developed by using Asterisk in the Intranet environment. Then the new scheme to realize high security by using Open VPN is described when developing the large scale enterprise network.

2. Basic idea

The following are the main requirements to develop the VoIP system for the enterprise network (Yamamoto et al., 2008)

a. Scalability

In the environment of the enterprize network, it is not easy to anticipate the traffic because there are lots of uncontrollable factors. So developing various scale systems based on the same architecture is necessary to meet the unpredictable change of traffic.

b. Cost

It is obviously desirable to develop the system at reasonable cost because generally the budget is rather limited.

c. High security

Also obviously high security is indispensable.

Considering the above requirements, the following are our basic ideas.

a. Developing VoIP system by using Asterisk as the open software

Obviously considering development cost it is desirable to use the open software. So we have selected three open softwares as candidates, that is, OpenSIPS, FreeSwitch and Asterisk. As the SIP server's viewpoint, OpenSIPS and FreeSwitch are superior to Asterisk in terms of functions, but Asterisk support various protocols (e.g. H.323, MGCP, SCCP) other than SIP and also has lots of additional PBX services (e.g. Voice Conference, Automatic Call distributor). So we have decided to use Asterisk to development VoIP system for the enterprise network.

b. Realizing high security by using Open VPN

When we develop the large scale enterprise network by connecting multiple Asterisk servers located in different sites based on Asterisk proprietary protocol (i.e. IAX2), some method is necessary to realize the high security because the voice data among sites is not encrypted. For this purpose we have introduced a new scheme to establish VPN by using Open VPN.

3. Overview of Asterisk

Asterisk is a kind of open source software executed on Linux to implement IP-PBX system and support various VoIP protocols such as SIP, H.323, MGCP, SCCP. It can be connected with IP network and also can be connected with the existent telephone networks via analog/digital interfaces. Fig.1 shows the architecture of Asterisk. Channel portion in Fig.1 consist of various logical communication interface modules and Application portion consist of the additional PBX service modules. In the following the main modules of the channel and the application are described.
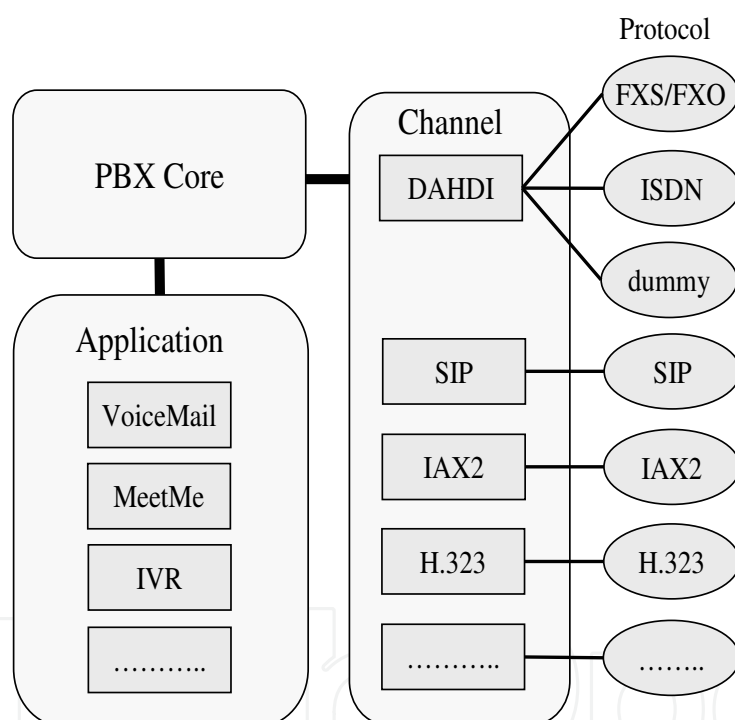

Fig. 1. Architecture of Asterisk

a. Channel modules
   - DAHDI (Digium/Asterisk Hardwae Device Interface): To connect with the ordinary existent telephone terminal it is necessary to insert the telephony card (e.g telephone card of Digium or of Voicetronix) as the physical interface and then the DAHDI interface module will be used. In case of connecting with existing POTS (Plain Old Telephone Service), FXS (Foreign eXchange Subsciber) and FXO (Foreign eXchange Office) interfaces will be used. In case of connecting with ISDN terminal it is necessary to insert the extension card as the physical interface.
   - SIP: This is the most basic signaling protocol to perform call processing in Asterisk and RTP/RTCP are used in order to transmit user data (e.g. voice data).

       - IAX2(Inter-Asterisk eXchange2): IAX2 is Asterisk proprietary protocol to conncet with multiple Asterisk servers located in the diffrent sites. The same port (i.e. 4569 as the default port) is used to transmit the call control signal and voice data.

b.   Application modules

       - Voice Conference: The voice conference service in Asterisk is called as "MeetMe". User can join the conference by inputting the designated number as the service number.

       - Voice Mail: When the called user is absent or busy, voice message can be kept in the voice box as the voice mail.

       - IVR (Interactive Voice Response): The automatic voice response can be performed by integrating voice response data file and dial number plan.

       - ACD (Automatic Call Distributor): The call can be automatically terminated to some terminal in the group based on the registered distribution rule.

       - AGI (Asterisk Gateway Interface): AGI is an API to connect the outside program with Asterisk in order to include some additional functions. Various programming language (e.g. C, Java, Perl, PHP, Bone Shell) are supported.

       - SLA (Shared Line Appearances): Multiple telephone terminals can share a subscriber line.

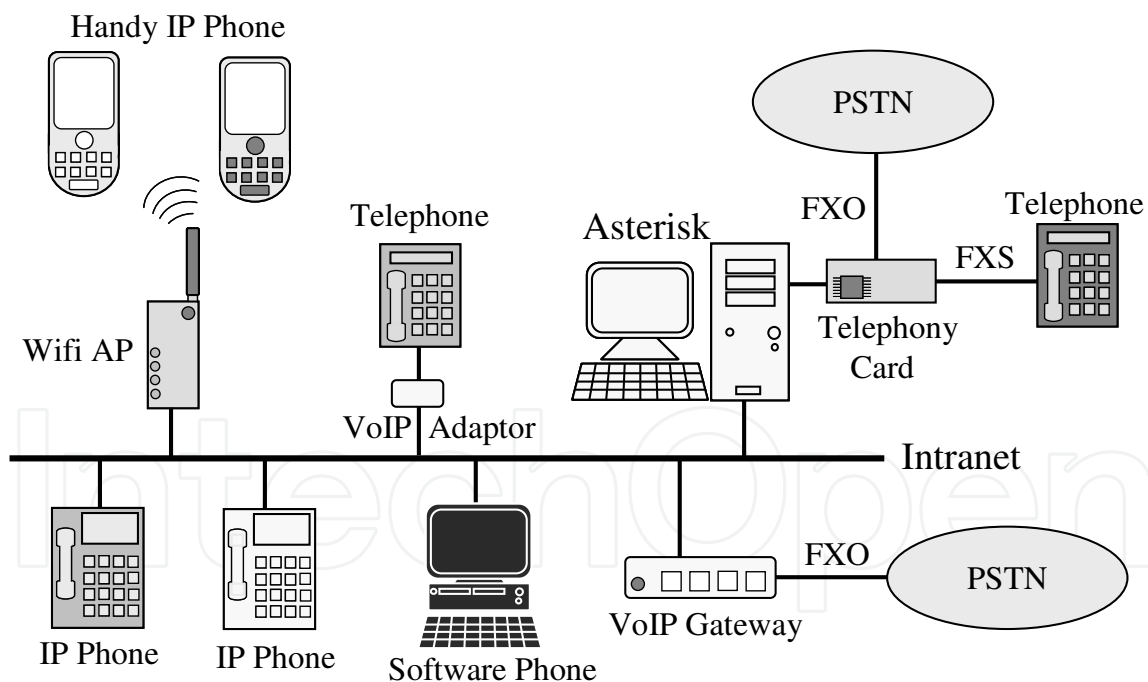More application modules (e.g. Call Parking, Call Queuing, Call Pickup, SNMP support) are also provided.

Fig. 2. VoIP system developed in Intranet

4. VoIP System based on Asterisk

4.1 VoIP system in Intranet

Fig.2 shows the VoIP system that we have developed by using an Asterisk in the Intranet environment (i.e. enterprise network).

Photo 1. Grandstream BT101 as IP Phone

Photo 3. Grandstream HT286 as VoIP Adaptor

Photo 2. Snom 105 as IP Phone

Photo 4. Sipura SPA1000 as VoIP gateway

In the Fig.2 all telephone terminals are connected to one Asterisk server, but it is possible to use multiple Asterisk servers depending on the scale of the Intranet (i.e. the number of terminals). As the IP phones, we have accommodated Grandstream BT101 (Photo 1) and Snom 105 (Photo 2), and also Grandstream HT286 (Photo 3) has been used as VoIP adaptor. Grandstrem HT488 and Sipura SPA1000 (Photo 4) have been used as VoIP gateways to connect with PSTN.

4.2 Connecting with SIP Server (Fig.3)

Asterisk can connect with existing SIP servers. When SIP server is located in the same network, it is easy to connect with each other. When SIP server is located in the different network (e.g. SIP server located in ISP network across Internet), it is possible to occur the NAT problem because the payload of SIP message can contain private information like private IP address. But even such case it is possible to connect with SIP server if we have selected some appropriate method to solve the NAT problem.
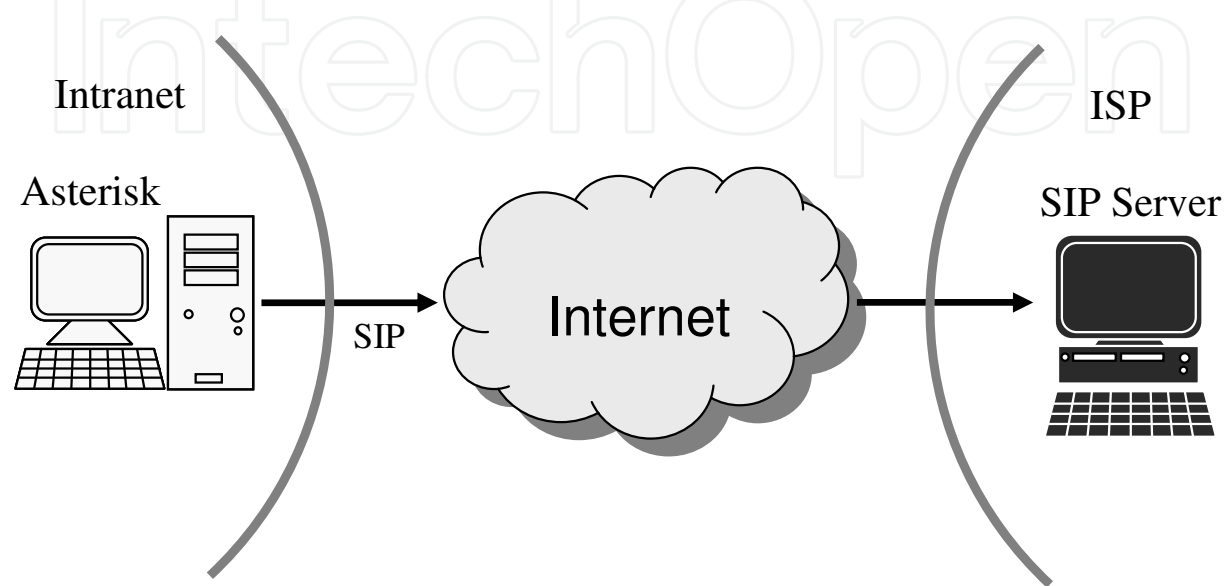
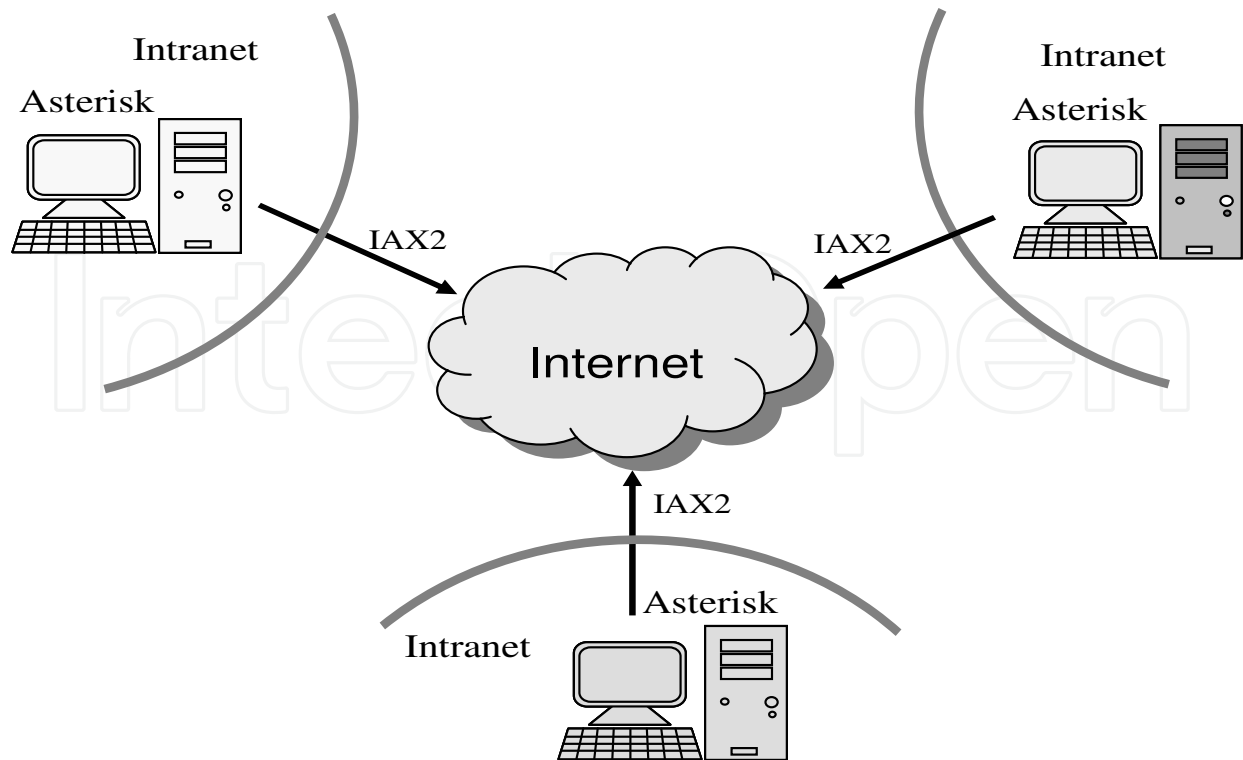Fig. 3. Connecting with SIP server across Internet

Fig. 4. Connecting with multiple Asterisks located in different Intranet

4.3 Multiple location connection by IAX2 (Fig.4)

As described previously IAX2 is the Asterisk proprietary protocol to connect with multiple Asterisk servers. So it is possible to connect with multiple Asterisk servers located in different Intranets easily.

5. Development process of VoIP system

In this section the detailed development process about the VoIP system we have developed is described.

5.1 Basic development process

a.   DAHDI compile and install

First of all Asterisk should be installed, but before that it is necessary to complete the DAHDI compile and install. Fig.5 shows the process of compile and install from the Asterisk cite (http://www.asterisk.org/).

```
# export http_proxy=http://your.proxy.server:port-num/

# tar zfxv dahdi-linux-complete-2.3.0.1+2.3.0.tar.gz

# cd dahdi-linux-complete-2.3.0.1+2.3.0

# make all

# make install
```

Fig. 5. Process of DAHDI compile and install

b.   Asterisk compile and install

Next Asterisk compile and install has been performed as shown in Fig.6.

```
# tar zxfv asterisk-1.6.2.10.tar.gz

# cd asterisk-1.6.2.10

# ./configure

# make

# make install

# make samples

# make config
```

Fig. 6. Process of Asterisk compile and install

Then after the necessary definition is completed, the access to Asterisk has been possible as shown in Fig.7.

c.   Define Dial Plan

Dial Plan is the core portion of the call processing in Asterisk. Dial Plan is defined in /etc/asterisk/extensions.conf. Extensions.conf consist of general section, globals section and context blocks as follows;

- General section: General parametes to cover the whole Dial Plan are defined in this section.
- Globals section: Variables used in the content blocks are defined in this section.

- Context blocks: Multiple dial plan are defined in the context blocks independently. So Asterisk can realize flexible dail plan by selecting appropriate block based on the conditions. The format of each line in the context block is as follows;

exten => Extension, Priority, Application

Extension in the right side is generally telephone numer and Priority is the order of processing. Application is the processing to be perfoemed to the Extension.

```
# /etc/init.d/asterisk start
# asterisk -r
Asterisk 1.6.2.10, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty'
for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
Connected to Asterisk 1.6.2.10 currently running on star (pid = 2144)
Verbosity is at least 3
star*CLI>
```

Fig. 7. Backgound execution of Asterisk

Fig.8 shows the example of default context which is selected when no context is explictly defined.

```
exten => _1XXX,1,Dial(SIP/${EXTEN}, 30)
exten => _1XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _1XXX,n,Hangup
exten => _1XXX,n(busy),Busy
```

Fig. 8. Default context in estensions.conf

X in _1XXX shows 0~9 and _ shows pattern matching. Dial(SIP/${EXTEN}, 30) in the first line shows to ring the SIP terminal during 30 seconds. GotoIf ($ ["$ {DIALSTATUS}" = "BUSY"] ?busy) in the second line shows that the processing will be terminated to the busy label if the call processing result is busy. ${EXTEN} shows the called party's telephone number and $ {DIALSTATUS} shows the variable to include the previous state.

Also it is necessary to define the channel file(e.g. /etc/asterisk/sip.conf in case of SIP, /etc/asterisl/chan_dahdi.conf in case of DAHDI, ip /etc/asteridk/iax.conf in case of IAX2). Fig.9 shows the call processing flows amoung channels.

d.   Define SIP Terminal

As described previously, /etc/asterisk/extensions.conf and /etc/asterisk/sip.conf should be defined when SIP terminal is used. Fig.10 shows the example of /etc/asterisk/sip.conf. 1000 and 1001 shows the telephone number of SIP terminals and terminals with proper password can only be registered in Asterisk.
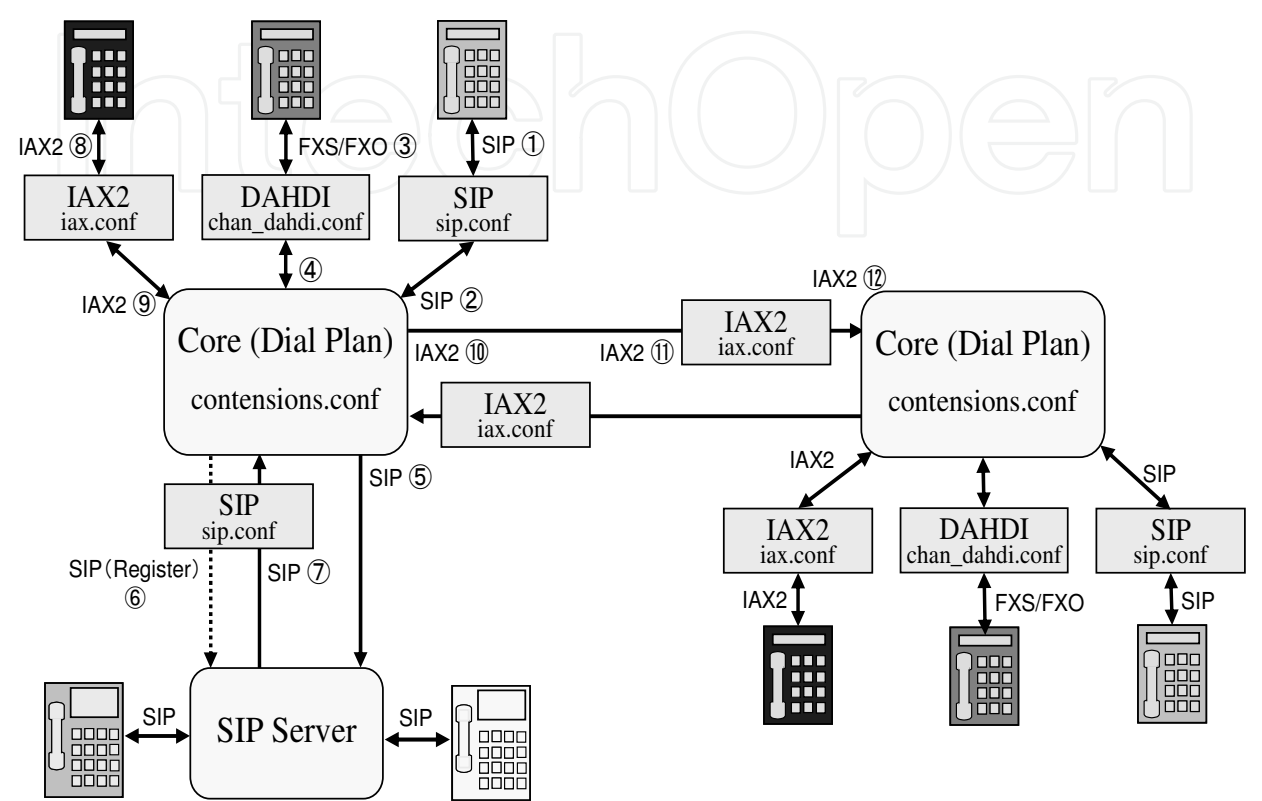

Fig. 9. Call processing flows among channels

```
[1000]
type=friend
secret=password1
host=dynamic

[1001]
type=friend
secret=password2
host=dynamic
```

Fig. 10. Example of /etc/asterisk/sip.conf.

Then the Dial Plan is defined for each SIP terminal by editing /etc /asterisk / extensions.conf. There is no context to describe Dial Plan in the example in Fig.10, so the default context is used. Thus it has been possible between 1000 SIP terminal and 1001 SIP terminal by using Asterisk server as the SIP server.
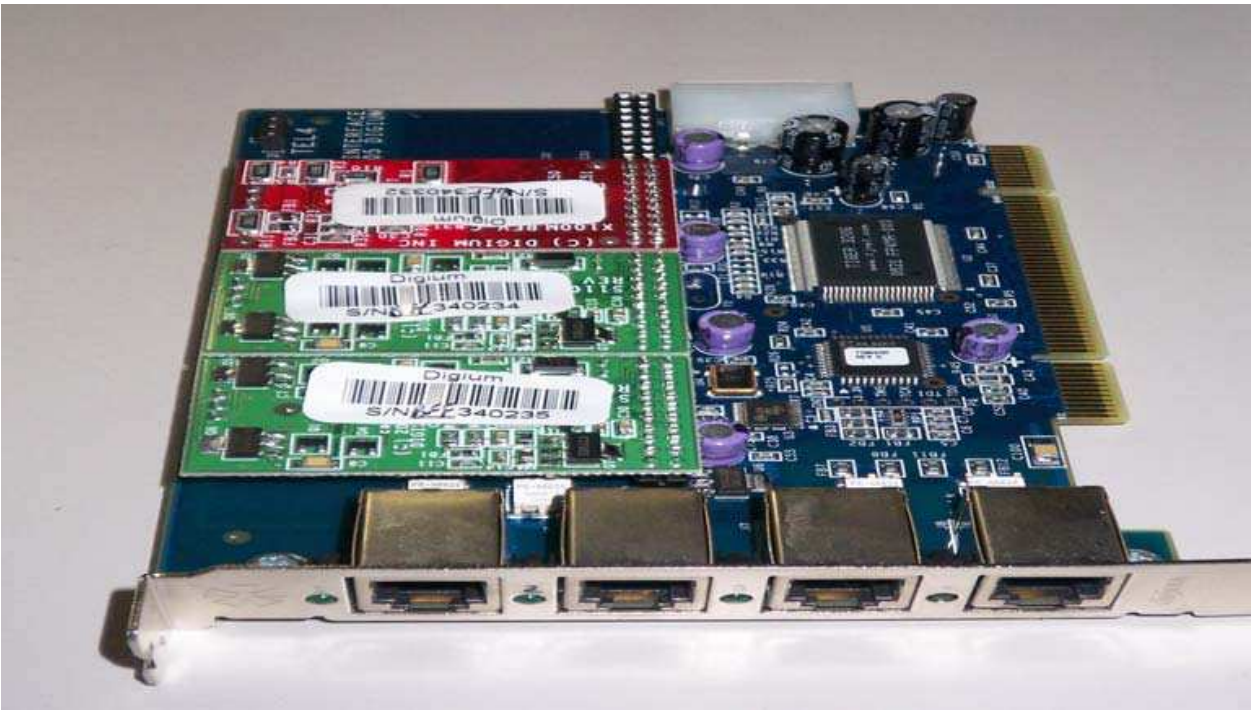
Photo 5. TDM410 as Telephony Card

e.   Define Telephony Card
We have used the Digium's Telephony Card TDM410 (Photo 5) and AEX410 (Photo 6). 1~3
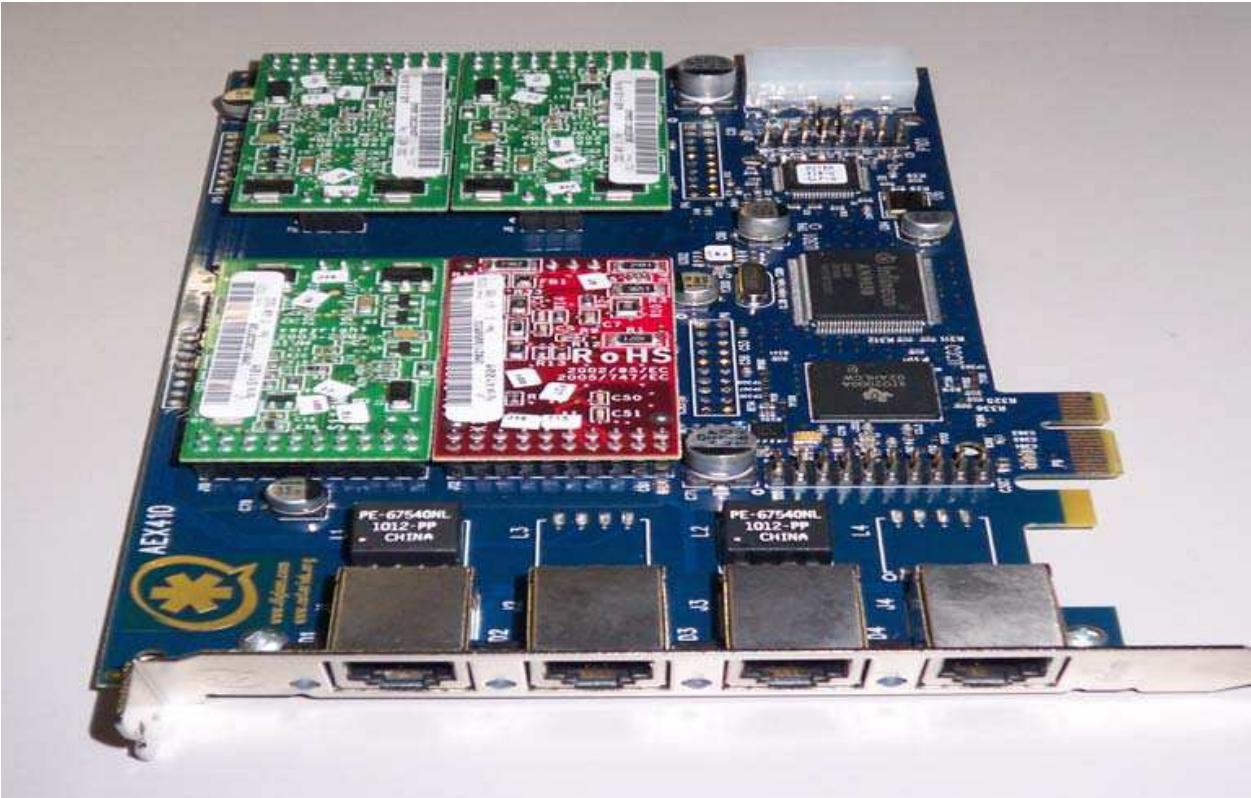ports are FXS and 4 port is FXO.

Photo 6. AEX410 as Telephony Card

Telephony Card has been defined in chan_dahdi.conf as shown in Fig.11.

```
context=default
signaling=fxo_ks
usecallerid=no
callwaiting=yes
echocancel=no
threewaycalling=yes
transfer=yes
channel => 1-3
;
context=incoming
signaling=fxs_ks
usecallerid=no
callwaiting=no
echocancel=no
transfer=no
channel => 4
```

Fig. 11. Example of chan_dahdi.conf

When the Dial Plan is defined in extensions.conf as shown in Fig.12, the call to 1 will be terminated to the terminal connected with port 1 and the call to 0 will be terminated to the port 4 (i.e. PSTN). That is, originating call to 0 means the call to connect ouside PSTN.

```
exten => 1,1,Dial(DAHDI/1, 30)
exten => 1,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => 1,n,Hangup
exten => 1,n(busy),Busy

exten => 0,1,Dial(DAHDI/4, 30)
exten => 0,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => 0,n,Hangup
exten => 0,n(busy),Busy
```

Fig. 12. Dial Plan in extensions.conf

When the call from PSTN should be terminated to the port 4, incoming context is defined in the extensions.conf as shown in Fig.13.

```
[incoming]
exten => s,1,Dial(SIP/1000)
exten => s,n,Dial(SIP/1001)
exten => s,n,Hangup
```

Fig. 13. incoming context for FXO port

f.   Connecting with SIP server

As previously described, communication between Asterisk and SIP server can be possible. When terminal connected with Asterisk communicate with the terminal connected with SIP server, Dial Plan is defined as shown in Fig.14.

```
exten => _3XXX,1,Dial(SIP/${EXTEN}@sip.server.address,30)
exten => _3XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _3XXX,n,Hangup
exten => _3XXX,n(busy),Busy
```

Fig. 14. Dial Plan for the call from Asterisk to SIP server

When terminal connected with SIP server communicate with the terminal connected with Asterisk, Asterisk should perform registration process as SIP client. So in order to perform the registration process, data is defined in the general section of sip.conf as shown in Fig.15. 4000 and 4001 are telephone numbers to be registered in the SIP server. 1000 and 1 are extensions to be used in the Dial Plan of Asterisk. Authid and password are ID and password to be registered in the SIP server, sip.server.address is IP address of the SIP server.

```
register => 4000:authid:password@sip.server.address/1000
register => 4001:authid:password@sip.server.address/1
```

Fig. 15. Data defined in general section for the registration process

g.   Connecting with multi Asterisk based on IAX2

When IAX2 telephone terminal is registered in the Asterisk, /etc/asterisk/iax.conf is defined as shown in Fig.16.

```
[6000]
type=friend
host=dynamic
secret=iax2pass
[6001]
type=friend
host=dynamic
secret=iax2pass
```

Fig. 16. Example of iax.cong

When the call is terminated to the IAX2 terminal, Dial Plan is defined in extension.conf as shown in Fig.17.

```
exten => _6XXX,1,Dial(IAX2/${EXTEN},30)
exten => _6XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _6XXX,n,Hangup
exten => _6XXX,n(busy),Busy
```

Fig. 17. Dial Plan for IAX2 terminal

As described previously, Asterisks located in multi Intranets can be connected with each other based on IAX2. In order to connect with Asterisk located in other Intranet, Dial Plan of the originating Asterisk should be defined as shown in Fig.18. Whole calls to 8XXX will transfer to the Asterisk whose address is defined as other.asterisk.address.

```
exten => _8XXX,1,Dial(IAX2/iax2id:iax2pass@other.asterisk.address/${EXTEN})
exten => _8XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _8XXX,n,Hangup
exten => _8XXX,n(busy),Busy
```

Fig. 18. Dial Plan for originating Asterisk

Also Dial Plan of the terminating Asterisk located in other Intranet should be defined as shown in Fig.19.

```
exten => _8XXX,1,Dial(SIP/${EXTEN}, 30)
exten => _8XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _8XXX,n,Hangup
exten => _8XXX,n(busy),Busy
```

Fig. 19. Dial Plan for terminating Asterisk

5.2 Application development process
a.   MeetMe (Voice conference)

MeetMe, voice conference service, can be easily realized in Asterisk. In order to regist the service, first, registration data is defined in the [room] section of /etc/asterisk/meetme.conf as shown in Fig.20. The right side of **=>** in Fig.20 include conference number, PIN, and PIN of administrator. PIN is the password to enter the conference and telephone number can be usually used as the conference number.

```
[room]
conf => 2000
conf => 2001,2456,7889
```

Fig. 20. Example of meetme.conf

Next in order to start the service, Dial Plan is defined as shown in Fig.21.

```
exten => _200X,1,Meetme(${EXTEN})
```

Fig. 21. Dial Plan for MeetMe

b.   Voice Mail

When the called user is absent or busy, voice message can be kept as the voice mail. In order to start the voice mail service, first the mail box is defined in /etc/asterisk/voicemail.conf as shown in Fig.22. 1000 and 1001 show the numbers of mail boxes, and telephone number is used usually as the number of mail box. 1212 and 2875 in Fig.21 show the passwords.

```
[default]
1000 => 1212,SIP User1
1001 => 2875,SIP User2
```

Fig. 22. Example of voicemail.conf

Next the service function is defined in extensions.conf as shown in Fig.23. The argument of Voicemail shows the number of the mail box and ${EXTEN} means that telephone number can be used as the number of mail box.

```
exten => _1XXX,1,Dial(SIP/${EXTEN},5)
exten => _1XXX,n,GotoIf($["${DIALSTATUS}"="BUSY"]?busy)
exten => _1XXX,n,GotoIf($["${DIALSTATUS}"="NOANSWER"]?busy)
exten => _1XXX,n,Hangup
exten => _1XXX,n(busy),Voicemail(${EXTEN})
```

Fig. 23. Example of extensions.conf to define service function

In order to confirm the stored voice mail, the designated number is defined in extensions.conf as shown in Fig.24. If user input 999, the number of mail box and password, the stored voice messages can be confirmed.

```
exten => 999,1,VoiceMailMain()
```

Fig. 24. Example of extensions.conf to define the number to access voice mail

c.   IVR (Interactive Voice Response)

In order to realize the automatic voice response service, detailed Dial Plan should be defined in /etc/asterisk/extensions.conf as shown in Fig.25. Answer in the [incoming] context shows that Asterisk will perform automatic response processing. The function of Background shows that the voice file of vm-enter-num-to-call will be played and that the control signal from the terminal can be processed even during the voice response. Playback is also a kind of function to play the voice file, but the user's signal cannot be processed during the voice response. WaitExtern is a function that suspend the signal processing for the defined time.

```
[default]
……………
exten => 9000,1,Goto(incoming,s,1)

……………

[incoming]
exten => s,1,Answer()
exten => s,n,Wait(1)
exten => s,n(again),Background(vm-enter-num-to-call)
exten => s,n,WaitExten(10)
exten => s,n,Playback(vm-goodbye)
exten => s,n,Hangup()
;
exten => i,1,Playback(invalid)
exten => i,n,Goto(s,again)
;
exten => 1,1,Dial(SIP/1000)
exten => 2,1,Dial(SIP/1001)
exten => 3,1,Dial(DAHDI/1)
```

Fig. 25. Example of IVR definition

6. Open VPN

In order to realize high security to connect multiple Asterisks located in different Intranets, we have implemented VPN capability. In this section the development process is described.
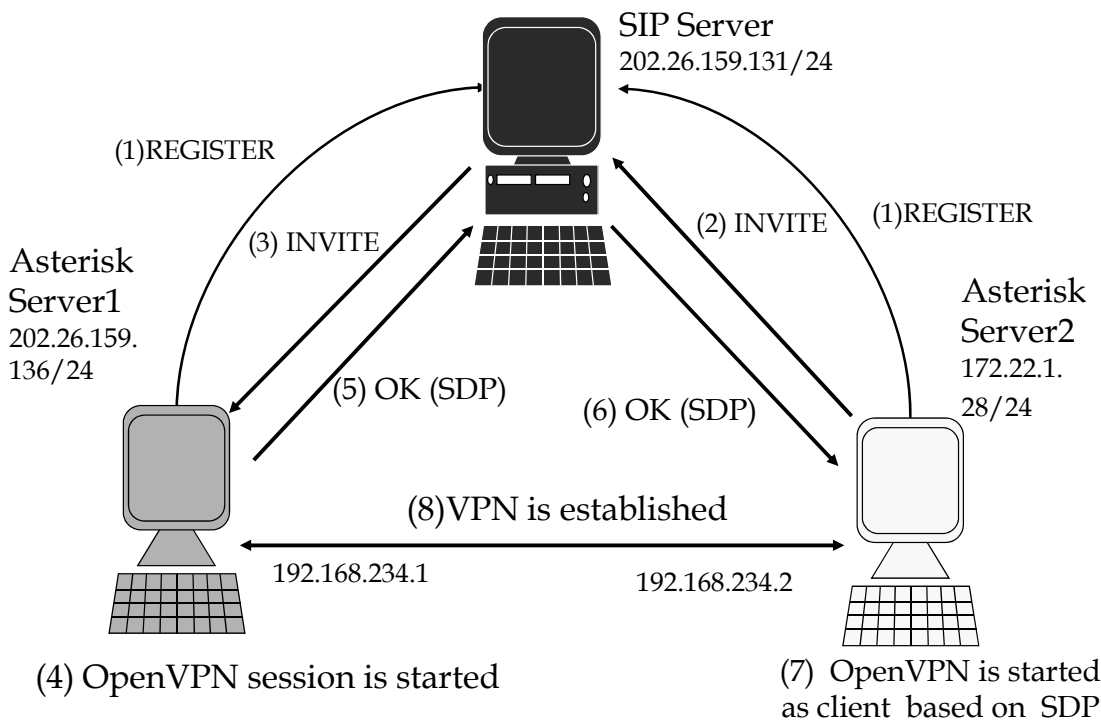

Fig. 26. VPN establishing procedure

Fig.26 shows the procedure to establish VPN between two Asterisk servers by using OpenVPN (http://openvpn.net/) based on the regular SIP sequence. To realize this procedure we have developed a program (i.e. sip_app) to have SIP client function with the function to invoke the external application. It is developed by using oSIP2 (http://www.gnu.org/software/osip) and eXosip2 (http://www.antisip.com/as/en/products.php) libraries in GNU, and has the SIP client function, SDP control function and the function to invoke the external process as child process. In the Asterisk server1, OpenVPN is registered as the external process and sip_app send the REGISTER message to SIP server (1). In the Asterisk server2, sip_app send the REGISTER message to SIP server (1) and send INVITE message to the Asterisk server1(2, 3). Asterisk server1 invoke the OpenVPN as the server mode (4) and reply 200 OK after inserting the necessary connection information into "a" record in SDP (5,6). Asterisk server2 invoke OpenVPN as the client mode after getting the necessary information from "a" record in SDP (7). OpenVPN in the Asterisk server2 communicate with OpenVPN in the Asterisk server1 and VPN between two servers has been established(8).

Table 1 shows the values of SDP at the process (6) in Fig.26. Record "m" shows media type (i.e. application/VPN) and the kind of protocol (i.e. OpenVPN). Record "a" is used by sip_app to control external process invoke. IP4 in Table1 is the IP address of the Asterisk server1 and PORT is the port to receive OpenVPN connection of Asterisk server1. VPN_LOCAL_ADDR is the IP address of Asterisk server1 and VPN_REMOTE_ADDR is the IP address of Asterisk server2.

Record Type	Value
v	0
o	2500 1169538046 1169538046 IN IP4 202.26.159.131
s	-
t	0 0
m	application/VPN 7084 OpenVPN 0
c	IN IP4 202.26.159.131
a	IP4:202.26.159.136
a	PORT:8000
a	VPN_LOCAL_ADDR:192.168.234.1
a	VPN_REMOTE_ADDR:192.168.234.

Table 1. Record value of SDP

Fig.27and Fig 28 show the detailed SIP messages at the process (5) , (6) in Fig.26.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
202.26.159.131:5060;branch=z9hG4bKd5494712271eafdca196759bbcd82500
Via: SIP/2.0/UDP
202.26.159.131:5060;branch=z9hG4bKdf1ab35628fe284df07a0549b85b5d31
Via: SIP/2.0/UDP 172.22.1.28:5060;rport;branch=z9hG4bK1068437359
Record-Route:<sip:siproxd@202.26.159.131:5060;lr>
From:<sip:2501@202.26.159.131>;tag=1297171609
To:<sip:2500@202.26.159.131>;tag=1988095920
Call-ID: 1815073903@172.22.1.28
CSeq: 20 INVITE
Contact:<sip:2500@202.26.159.136:5060>
User-Agent: SIP for APP b1 rev.45
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY,MESSAGE,INFO,
REFER, UPDATE
Content-Type: application/sdp
Content-Length: 245

v=0
o=2500 1169538046 1169538046 IN IP4 202.26.159.136
s=-
t=0 0
m=application/vpn 8000 OpenVPN 0
k=DH:crypt code
c=IN IP4 202.26.159.136
a=IP4:202.26.159.136
a=PORT:8000
a=VPN_LOCAL_ADDR:192.168.234.1
a=VPN_REMOTE_ADDR:192.168.234.2
```

Fig. 27. SIP message at (5) in Fig.26

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.22.1.28:5060;rport;branch=z9hG4bK1068437359
Record-Route:<sip:siproxd@202.26.159.131:5060;lr>
From:<sip:2501@202.26.159.131>;tag=1297171609
To:<sip:2500@202.26.159.131>;tag=1988095920
Call-ID: 1815073903@172.22.1.28
CSeq: 20 INVITE
Contact:<sip:2500@202.26.159.131>
User-agent: SIP for APP b1 rev.45
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, MESSAGE,INFO,
REFER, UPDATE
Content-Type: application/sdp
Content-Length: 245

v=0
o=2500 1169538046 1169538046 IN IP4 202.26.159.131
s=-
t=0 0
m=application/vpn 7084 OpenVPN 0
c=IN IP4 202.26.159.131
k=DH:crypt code
a=IP4:202.26.159.136
a=PORT:8000
a=VPN_LOCAL_ADDR:192.168.234.1
a=VPN_REMOTE_ADDR:192.168.234.2
```

Fig. 28. SIP message at (6) in Fig.26

7. Conclusion

This chapter describe VoIP system for the enterprise network (e.g. company, university) that we have developed based on Asterisk which is a kind of open source software to implement IP-PBX system. Through the development and evaluation, we have confirmed that VoIP system based on Asterisk is very powerful as a whole and most PBX functions to be required for the enterprise network can be realized.

Compared with the general SIP server, it can be said that Asterisk is more focused on providing basic functions. But Asterisk can connect with SIP server easily, so it is possible to implement the necessary additional functions by just connecting with other outside SIP servers. Also Asterisk can connect with the existent PSTN by using FXO telephony card, so it is possible to be used as the VoIP gateway.

When developing the large scale enterprise network by connecting multiple Asterisk servers located in different sites based on IAX2, to realize high security is the issue because the voice data is not encrypted. To solve this issue, we have proposed the method to establish VPN by using Open VPN and have also described the development process in detail.

8. References

http://www.asterisk.org
http://openvpn.net/
http://www.gnu.org/software/osip
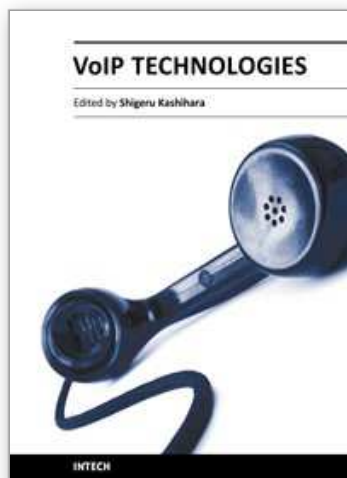http://www.antisip.com/as/en/products.php
Yamamoto et al.(2008). Validation of VoIP System for University Network, Proceedings of
        ICACT2008, 9C-2, Phoenix Park, Feb.2008, Korea

VoIP Technologies

Edited by Dr Shigeru Kashihara

ISBN 978-953-307-549-5

Hard cover, 336 pages

Publisher InTech

Published online 14, February, 2011

Published in print edition February, 2011

This book provides a collection of 15 excellent studies of Voice over IP (VoIP) technologies. While VoIP is undoubtedly a powerful and innovative communication tool for everyone, voice communication over the Internet is inherently less reliable than the public switched telephone network, because the Internet functions as a best-effort network without Quality of Service guarantee and voice data cannot be retransmitted. This book introduces research strategies that address various issues with the aim of enhancing VoIP quality. We hope that you will enjoy reading these diverse studies, and that the book will provide you with a lot of useful information about current VoIP technology research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Moo Wan Kim and Fumikazu Iseki (2011). VoIP System for Enterprise Network, VoIP Technologies, Dr Shigeru Kashihara (Ed.), ISBN: 978-953-307-549-5, InTech, Available from: http://www.intechopen.com/books/voip-technologies/voip-system-for-enterprise-network

INTECH
open science | open minds