



Performance Benchmark Test for Asterisk B2BUA

13 November 2007

<http://www.TransNexus.com>

Revision History

Revision	Date of Issue	Changes
0.1	July 16, 2007	Initial Draft.
0.2	July 17, 2007	Minor edits and reformatting
0.3	August 3, 2007	Use sar for test date collection
0.4	September 6, 2007	Add codec translation test case
0.5	September 10, 2007	Add Digium g729 module results
1.0	November 13, 2007	Final draft for publishing

Copyright 2007 TransNexus, Inc.

TransNexus is a registered trademark of TransNexus, Inc.

Asterisk is a registered trademark of Digium, Inc.

Contents

Revision History	2
Contents	3
1 Introduction.....	5
1.1 Summary of Results.....	5
1.1.1 No Transcoding.....	5
1.1.2 G.711 to G.729 Transcoding.....	6
2 Test Bed Diagram	7
2.1 Call Scenarios	7
2.1.1 Call Scenario without Codec Translation	7
2.1.2 Call Scenario with G711 alaw to G729	8
2.1.3 Call Scenario with G729 to G711 alaw	8
2.2 Summary of Test Bed Devices	9
3 Test Bed Configuration	10
3.1 OSP Server.....	10
3.2 SIPp Client.....	10
3.3 CentOS.....	10
3.4 Routing Info for OSP Server.....	10
3.5 Asterisk B2BUA	12
3.5.1 Chan_sip.c.....	12
3.5.2 Modules.conf	12
3.5.3 Sip.conf	12
3.5.4 Osp.conf	25
3.5.5 Extensions.conf.....	26
3.6 SIPp.....	36
3.6.1 Server End.....	36
3.6.2 Client End	40
4 SIPp Client Parameters.....	44
4.1 Transport Mode.....	44
4.2 Call Limit	45
4.3 Timer Resolution	45
4.4 Frequency.....	45
4.5 Call Numbers	45
4.6 Screen Flush Frequency	45
5 Scripts for Running the Test.....	45
5.1 Data Collection Scripts	45
5.1.1 Sar Write Script.....	45
5.1.2 Sar Read Script	45
5.2 SIPp Scripts.....	46
5.2.1 Server End.....	46
5.2.2 Client End	46
6 Test Procedure	46
6.1 Start Test	46
6.2 After Test	46

7	Test Results.....	47
7.1	Actual Results	47
7.1.1	Test Results without Codec Translation	48
7.1.2	Test Results with G711 Alaw to Digium G729	51
7.1.3	Test Results with Digium G729 to G711 Alaw	53

1 Introduction

This document describes a benchmark performance test and results for Asterisk deployed as a SIP Back to Back User Agent (B2BUA). The purpose of the test is to understand the performance of an Asterisk B2BUA in a production environment as a function of host server CPU and memory resources.

To simulate a production environment, the Asterisk B2BUA queries an external OSP server for routing information on each call and then reports call detail records to the external OSP server. The OSP server responds to each OSP route query with five possible destinations which can complete the call in random order. Four of the five destinations simulate call failure scenarios. Therefore the Asterisk B2BUA must, on average, make three call attempts for each completed call.

Section 7 provides the raw data collected from the test.

1.1 Summary of Results

VoIP service providers have two critical questions when evaluating SIP B2BUA alternatives:

1. How many simultaneous calls can the B2BUA handle?
2. What is the cost per port (per simultaneous call)?

This performance test answers these questions for Asterisk configured as a B2BUA on a host server with two, dual core, 2.33 GHz CPUs.

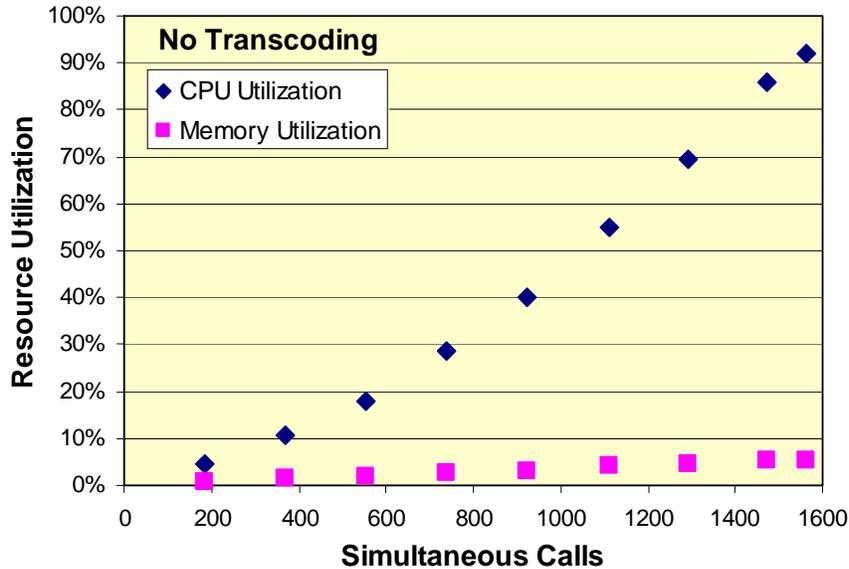
- No transcoding: 1500 simultaneous calls at a cost of \$2 per port
- G.711 to G.729 Transcoding: 400 simultaneous calls at a cost of \$17.50 per port

These test results are extrapolated provide a simple guideline for predicting maximum simultaneous calls based on server CPU resources. Server CPU resources = number of CPUs (or cores) * GHz per CPU (or core). Using this guideline, the host server for this test has $(4 \text{ CPU cores}) * (2.33 \text{ GHz per CPU}) = 9.32 \text{ GHz of CPU resources}$.

- No transcoding: 160 simultaneous calls per 1 GHz of CPU resource.
- G.711 to G.729 Transcoding: 40 simultaneous call per 1 GHz of CPU resource.

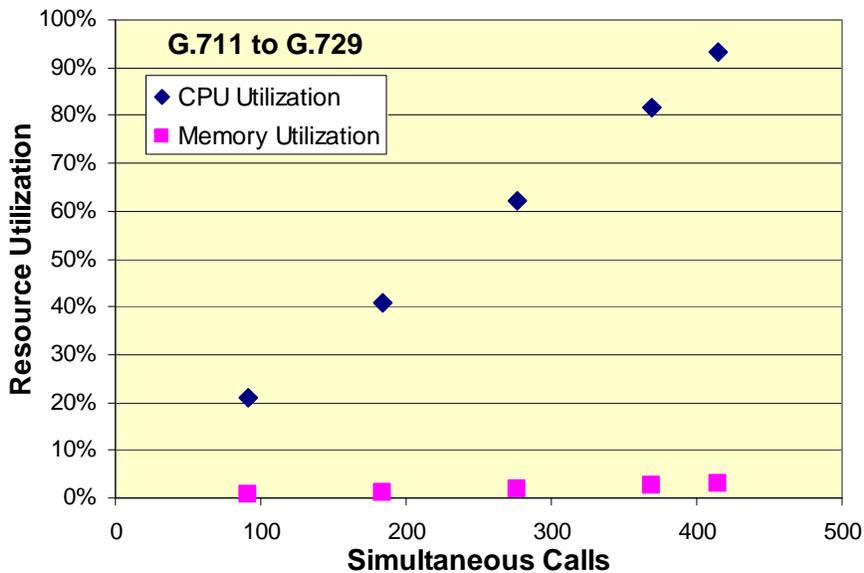
1.1.1 No Transcoding

The following chart plots CPU and memory utilization of the server hosting the Asterisk B2BUA as a function of simultaneous calls. Calls to and from the B2BUA used the G.711 codec and all RTP packets flowed through Asterisk. Memory utilization is minimal and CPU utilization grows linearly with the number of simultaneous calls. At 1560 simultaneous calls, the CPU utilization reaches 92%. The original retail price, in early 2007, for this server was \$3000. The \$3000 hardware cost divided by 1500 simultaneous calls equals \$2 per B2BUA port.



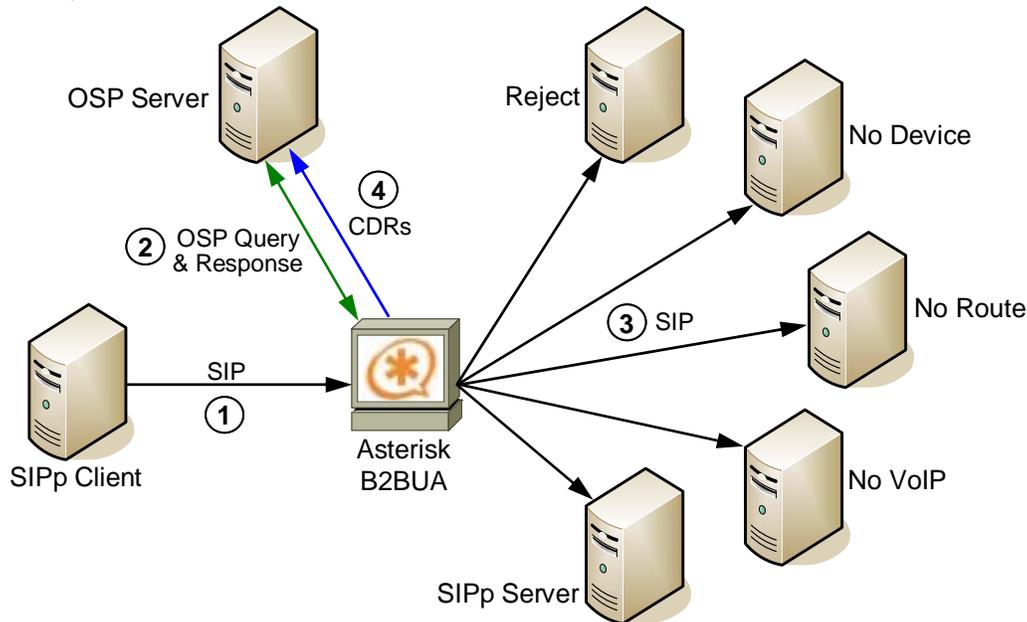
1.1.2 G.711 to G.729 Transcoding

The following chart plots CPU and memory utilization of the server hosting the Asterisk B2BUA as a function of simultaneous calls. Calls to the B2BUA used the G.711 codec and were transcoded by Asterisk to G.729. Memory utilization is minimal and CPU utilization grows linearly with the number of simultaneous calls. At 414 simultaneous calls, the CPU utilization reaches 93%. The original retail price, in early 2007, for this server was \$3000. The license royalty for the G.729 codec, available from Digium, is \$10 per port. The total system cost is \$3,000 for server hardware plus the \$4,000 G.729 codec royalty (\$10 per port * 400 ports) is \$7,000. The total cost per B2BUA port is \$7,000 divided 400 simultaneous calls or \$17.50 per port.



2 Test Bed Diagram

The following diagram illustrates the test network and call scenario. The server hosting the Asterisk B2BUA is a Dell Precision 490 server with two Intel Xeon 5140 dual core, 2.33 GHz, CPUs and 4 GB RAM.



1. The SIPp client sends a SIP Invite to the Asterisk B2BUA.
2. Asterisk sends an OSP AuthorizationRequest to the OSP server. The OSP server returns an OSP AuthorizationResponse with five possible destinations in random order.
3. Asterisk sends a SIP Invite to the destination. Only the SIPp server can complete the call. The other destinations simulate call failures. On average, Asterisk attempts the call to two failure destinations before completing the call to the SIPp server.
4. When the call ends, Asterisk sends a call detail record (CDR) to the OSP server.

2.1 Call Scenarios

This test plan measures Asterisk performance for three different transcoding scenarios:

1. No transcoding. Inbound call to B2BUA is G.711 – Outbound call is G.711. The RTP packets for every call are routed through the Asterisk B2BUA.
2. G.711 to G.729. Inbound call to B2BUA is G.711 alaw – Outbound call is G.729
3. G.729 to G.711. Inbound call to B2BUA is G.729 – Outbound call is G.711 alaw

2.1.1 Call Scenario without Codec Translation

1. SIPp client sends a SIP INVITE to Asterisk B2BUA.
2. Asterisk B2BUA sends an OSP AuthorizationRequest to OSP server.
3. The OSP server returns five destinations in random order (Reject, No Device, No Route, No VoIP, SIPp Server). Four destinations test specific failover/retry scenarios. Only the SIPp server destination can complete the call and the other four destinations are configured to fail. Therefore, around
 - 20% of the calls are completed on the first try.

- 20% of the calls fail on the first attempt and complete on the second attempt.
 - 20% of the calls fail on the first two attempts and complete on the third attempt.
 - 20% of the calls fail on the first three attempts and complete on the fourth attempt.
 - 20% of the calls fail on the first four attempts and complete on the fifth attempt.
4. SIPp server agrees using g711 alaw codec.
 5. SIPp client sends around 3 minute g711 alaw voice stream (including moh_alaw.pcap, around 3 minute long) to Asterisk B2BUA.
 6. Asterisk forwards the voice stream without codec translation to SIPp server.
 7. SIPp client sends a BYE to Asterisk B2BUA.
 8. When the call is finished, Asterisk B2BUA sends OSP UsageIndication messages (Call Detail Records) to the OSP server.

2.1.2 Call Scenario with G711 alaw to G729

1. SIPp client sends a SIP INVITE to Asterisk B2BUA.
2. Asterisk B2BUA sends an OSP AuthorizationRequest to OSP server.
3. The OSP server returns five destinations in random order (Reject, No Device, No Route, No VoIP, SIPp Server). Four destinations test specific failover/retry scenarios. Only the SIPp server destination can complete the call and the other four destinations are configured to fail. Therefore, around
 - 20% of the calls are completed on the first try.
 - 20% of the calls fail on the first attempt and complete on the second attempt.
 - 20% of the calls fail on the first two attempts and complete on the third attempt.
 - 20% of the calls fail on the first three attempts and complete on the fourth attempt.
 - 20% of the calls fail on the first four attempts and complete on the fifth attempt.
4. SIPp server agrees using g729 codec.
5. SIPp client sends around 3 minute g711 alaw voice stream (including moh_alaw.pcap, around 3 minute long) to Asterisk B2BUA
6. Asterisk forwards the voice stream with codec translation (g711 alaw to g729) to SIPp server.
7. SIPp client sends a BYE to Asterisk B2BUA
8. When the call is finished, Asterisk B2BUA sends OSP UsageIndication messages (Call Detail Records) to the OSP server.

2.1.3 Call Scenario with G729 to G711 alaw

1. SIPp client sends a SIP INVITE to Asterisk B2BUA.
2. Asterisk B2BUA sends an OSP AuthorizationRequest to OSP server.
3. The OSP server returns five destinations in random order (Reject, No Device, No Route, No VoIP, SIPp Server). Four destinations test specific failover/retry scenarios. Only the SIPp server destination can complete the call and the other four destinations are configured to fail. Therefore, around

- 20% of the calls are completed on the first try.
 - 20% of the calls fail on the first attempt and complete on the second attempt.
 - 20% of the calls fail on the first two attempts and complete on the third attempt.
 - 20% of the calls fail on the first three attempts and complete on the fourth attempt.
 - 20% of the calls fail on the first four attempts and complete on the fifth attempt.
4. SIPp server agrees using g711 codec.
 5. SIPp client sends around 3 minute g729 voice stream (including moh_g729.pcap, around 3 minute long) to Asterisk B2BUA
 6. Asterisk forwards the voice stream with codec translation (g729 to g711 alaw) to SIPp server.
 7. SIPp client sends a BYE to Asterisk B2BUA
 8. When the call is finished, Asterisk B2BUA sends OSP UsageIndication messages (Call Detail Records) to the OSP server.

2.2 Summary of Test Bed Devices

- **OSP Servers:** t2000/172.16.4.75:1080 OSP Server 3.0.y
- **Asterisk B2BUA:** 1.4.11 is used in this test. Asterisk has been compiled with the OSP Toolkit (sourceforge.net/projects/osp-toolkit) to support the OSP protocol. The MAX_RETRANS in chan_sip.c should be reduced to 1 to generate a 2 sec timeout for no connection destinations. MAX_AUTOMONS in autoservice.c should be increased to 2048 to prevent “Exceeded maximum number of automatic monitoring events. Fix autoservice.c” warning message.
- **Server hosting Asterisk B2BUA:** Dell Precision 490 server with two Intel Xeon 5140 dual core, 2.33 GHz, CPUs and 4 GB RAM. The operating system is CentOS4. Server IP address is 172.16.4.32. The NIC must be 1G NIC.
- **Reject:** fedora2/172.16.4.56:5060. The Asterisk for another test is running on fedora2. It will reject the called numbers we use in this test. The TCCode for this destination should be 1. This TCCode is not a standard mapping from 404.
- **No Device:** An IP address in 172.16.4.x network segment but no device uses this address, such as 172.16.4.100. This is a connection timeout destination. The TCCode for this destination should be 18.
- **No Route:** An IP address out of 172.16.4.x network segment, such as 1.1.1.1. This is a connection timeout destination. The TCCode for this destination should be 18.
- **No VoIP:** A PC in 172.16.4.x network segment without running SIP proxy or gateway, such as 172.16.4.1. This is a connection timeout destination. The TCCode for this destination should be 18.
- **SIPp Clients:** fedora1/172.16.4.58, fedora5/172.16.4.59, fedora8/172.16.4.25 and NexTransit06/172.16.4.28. The port will be automatically selected by SIPp client. A pcap file, moh_alaw.pcap, is used to generate 3 minute call duration. The SIPp client script includes the instruction to use this voice file.

- **SIPp Server:** fedora3/172.16.4.60:5060. The port must be 5060. Asterisk B2BUA does not support other port in recent release. The NIC must be 1G NIC.

Note:

1. In order to simulate product environment,
 - The OSP server and Asterisk B2BUA should be hosted on the different boxes.
 - SIPp and Asterisk B2BUA should be hosted on the different boxes.
 - The OSP server should return all 5 destinations for every OSP AuthorizationRequest.
 - Multiple SIPp clients are used.
2. Since the limitation of Asterisk B2BUA and OSP server, the port of the destinations must be 5060, the default SIP port.
3. Digium codec module is used for the test. Digium g729 module is licensed for max 500 simultaneous calls.

3 Test Bed Configuration

3.1 OSP Server

In order to use multiple OSP Server instances on one host, the ports of the OSP Server instances must be configured.

There are 3 places where the configurable ports must agree.

1. PORT_BASE=N000 (in start_osp_server.sh)
2. portbase=N000 (in xitami/defaults.cfg)
3. port = N443 (in xitami/sslhtt.cfs)

N should be 1, 2, 3, etc.

Only one OSP server instance is used in this test.

3.2 SIPp Client

There is a configurable soft limit should be changed. The stack size should be 4096.

Using “*ulimit -s 4096*” to set this parameter.

3.3 CentOS

There is a configurable soft limit should be changed. The max open file should be at least 20480. Using “*ulimit -n 20480*” to set this parameter.

3.4 Routing Info for OSP Server

1. One Customer:
 - STRESS
2. Seven devices:
 - stress_proxy 172.16.4.32
 - stress_reject 172.16.4.56
 - stress_nodevice 172.16.4.100
 - stress_noroute 1.1.1.1
 - stress_novoip 172.15.4.1

- stress_sippsrc1 172.16.4.58
- stress_sippsrc2 172.16.4.59
- stress_sippsrc3 172.16.4.25
- stress_sippsrc4 172.16.4.28
- stress_sippdst 172.16.4.60

Device Properties: All devices, except for stress_sippsrc1/2/3/4 should be configured with the following properties: OSP version 2.1.1, may terminate, is online, and has enrolled. The device stress_sippsrc1/2/3/4 should have the following properties: OSP version 2.1.1, may not terminate, is online, and has enrolled.

3. Six destinations:
 - dst_stress: with 5 same weight devices: stress_reject, stress_nodevice, stress_noroute, stress_novoip, and stress_sippdst. This destination is for returning destinations in random order.
 - dst_stress_reject: This destination is for returning destinations in certain order.
 - dst_stress_nodevice: This destination is for returning destinations in certain order.
 - dst_stress_noroute: This destination is for returning destinations in certain order.
 - dst_stress_novoip: This destination is for returning destinations in certain order.
 - dst_stress_sipp: This destination is for returning destinations in certain order.
4. Five route plans:
 - route_sipp: for all numbers return only 1 dst_stress_sipp. This route plan is used for troubleshooting purpose.
 - route_5_sipp: for all numbers return 5 dst_stress_sipps. This route plan is used for troubleshooting purpose.
 - route_sip_reject: for all numbers return 4 dst_stress_rejects and 1 dst_stress_sipp in random order. This route plan is used for troubleshooting purpose.
 - route_order: for all numbers return the destinations in dst_stress_reject, dst_stress_nodevice, dst_stress_noroute, dst_stress_novoip, and dst_stress_sipp order. This route plan is used for troubleshooting purpose.
 - route_random: for all numbers return the destination dst_stress. This route plan is used for the real tests.
5. Five products:
 - For number 111111111, use route plan route_sipp. This number is used for troubleshooting purpose.
 - For number 222222222, use route plan route_5_sipp. This number is used for troubleshooting purpose.
 - For number 333333333, use route plan route_sipp_reject. This number is used for troubleshooting purpose.
 - For number 999999999, use route plan route_order. This number is used for troubleshooting purpose.
 - For all other numbers, use route plan route_random. These numbers are used for the real tests.
6. Dial plan 1 is used for customer STRESS for all numbers.

3.5 Asterisk B2BUA

Asterisk B2BUA should be compiled with the OSP Toolkit using default compile time options. Detailed instructions for building Asterisk with the OSP Toolkit are available at http://www.transnexus.com/White%20Papers/Asterisk_V1.4_OSP_Module_User_Guide.pdf.

3.5.1 Chan_sip.c

MAX_RETRANS should be reduced to 1. It will reduce the timeout for no connection destinations (no device, no route and no voip) to 2 sec.

3.5.2 Modules.conf

```
;
; Asterisk configuration file
;
; Module Loader configuration file
;

[modules]
autoload=no

; Resources
load => res_adsi.so
load => res_features.so
load => res_indications.so

; PBX
load => pbx_config.so

; Functions
load => func_callerid.so
load => func_cut.so

; Channels
load => chan_sip.so

; Codecs
load => codec_alaw.so
load => codec_g729a.so      ; Digium g729 codec module
;load => codec_g729.so     ; IPP g729 codec module

; Formats
load => format_pcm.so
load => format_wav_gsm.so
load => format_wav.so

; Applications
load => app_dial.so
load => app_osplookup.so
```

3.5.3 Sip.conf

```
;
; SIP Configuration example for Asterisk
;
; Syntax for specifying a SIP device in extensions.conf is
; SIP/devicename where devicename is defined in a section below.
;
; You may also use
; SIP/username@domain to call any SIP user on the Internet
; (Don't forget to enable DNS SRV records if you want to use this)
;
```

```

; If you define a SIP proxy as a peer below, you may call
; SIP/proxyhostname/user or SIP/user@proxyhostname
; where the proxyhostname is defined in a section below
;
; Useful CLI commands to check peers/users:
; sip show peers      Show all SIP peers (including friends)
; sip show users      Show all SIP users (including friends)
; sip show registry   Show status of hosts we register with
;
; sip debug           Show all SIP messages
;
; reload chan_sip.so  Reload configuration file
;                    Active SIP peers will not be reconfigured
;

[general]
;context=default      ; Default context for incoming calls
;allowguest=no        ; Allow or reject guest calls (default is yes)
allowoverlap=no       ; Disable overlap dialing support. (Default is
yes)
;allowtransfer=no     ; Disable all transfers (unless enabled in peers or
users)
;                    ; Default is enabled
;realm=mydomain.tld   ; Realm for digest authentication
;                    ; defaults to "asterisk". If you set a system name in
;                    ; asterisk.conf, it defaults to that system name
;                    ; Realms MUST be globally unique according to RFC

3261
;                    ; Set this to your host name or domain name
;bindport=5060        ; UDP Port to bind to (SIP standard port is
5060)
;                    ; bindport is the local UDP port that Asterisk will
listen on

bindaddr=0.0.0.0      ; IP address to bind to (0.0.0.0 binds to all)
;srvlookup=yes        ; Enable DNS SRV lookups on outbound calls
;                    ; Note: Asterisk only uses the first host
;                    ; in SRV records
;                    ; Disabling DNS SRV lookups disables the
;                    ; ability to place SIP calls based on domain
;                    ; names to some other SIP users on the Internet

;-----Asterisk/OSP Configuration by Di-Shi Sun-----
context=GeneralProxy ; General Proxy

allowguest=yes
bindport=5060
nat=no
canreinvite=no
srvlookup=no
trustrpid=yes
sendrpid=yes
disallow=all
allow=alaw
allow=g729
;-----Asterisk/OSP Configuration by Di-Shi Sun-----

;domain=mydomain.tld ; Set default domain for this host
;                    ; If configured, Asterisk will only allow
;                    ; INVITE and REFER to non-local domains
;                    ; Use "sip show domains" to list local domains
;pedantic=yes        ; Enable checking of tags in headers,
;                    ; international character conversions in URIs
;                    ; and multiline formatted headers for strict

```

```

; SIP compatibility (defaults to "no")

; See doc/README.tos for a description of these parameters.
;tos_sip=cs3 ; Sets TOS for SIP packets.
;tos_audio=ef ; Sets TOS for RTP audio packets.
;tos_video=af41 ; Sets TOS for RTP video packets.

;maxexpiry=3600 ; Maximum allowed time of incoming
registrations
; and subscriptions (seconds)
;minexpiry=60 ; Minimum length of registrations/subscriptions
(default 60)
;defaultexpiry=120 ; Default length of incoming/outgoing registration
;tlmin=100 ; Minimum roundtrip time for messages to monitored
hosts
; Defaults to 100 ms
;notifymime-type=text/plain ; Allow overriding of mime type in MWI NOTIFY
;checkmwi=10 ; Default time between mailbox checks for peers
;buggy-mwi=no ; Cisco SIP firmware doesn't support the MWI RFC
; fully. Enable this option to not get error messages
; when sending MWI to phones with this bug.
;vmexten=voicemail ; dialplan extension to reach mailbox sets the
; Message-Account in the MWI notify message
; defaults to "asterisk"
;disallow=all ; First disallow all codecs
;allow=ulaw ; Allow codecs in order of preference
;allow=ilbc ; see doc/rtp-packetization for framing options
;
; This option specifies a preference for which music on hold class this channel
; should listen to when put on hold if the music class has not been set on the
; channel with Set(CHANNEL(musicclass)=whatever) in the dialplan, and the peer
; channel putting this one on hold did not suggest a music class.
;
; This option may be specified globally, or on a per-user or per-peer basis.
;
;mohinterpret=default
;
; This option specifies which music on hold class to suggest to the peer
channel
; when this channel places the peer on hold. It may be specified globally or on
; a per-user or per-peer basis.
;
;mohsuggest=default
;
;language=en ; Default language setting for all users/peers
; This may also be set for individual users/peers
;relaxdtmf=yes ; Relax dtmf handling
;trustpid=no ; If Remote-Party-ID should be trusted
;sendrp-id=yes ; If Remote-Party-ID should be sent
;progressinband=never ; If we should generate in-band ringing always
; use 'never' to never use in-band signalling, even
in cases
; where some buggy devices might not render it
; Valid values: yes, no, never Default: never
;useragent=Asterisk PBX ; Allows you to change the user agent string
;promiscredir=no ; If yes, allows 302 or REDIR to non-local SIP
address
; Note that promiscredir when redirects are
made to the
; local system will cause loops since Asterisk
is incapable
; of performing a "hairpin" call.
;usereqphone=no ; If yes, ";user=phone" is added to uri that contains
; a valid phone number

```

```

;dtmfmode = rfc2833          ; Set default dtmfmode for sending DTMF. Default:
rfc2833
                                ; Other options:
                                ; info : SIP INFO messages
                                ; inband : Inband audio (requires 64 kbit codec -alaw,
ulaw)
                                ; auto : Use rfc2833 if offered, inband otherwise

;compactheaders = yes        ; send compact sip headers.
;
;videosupport=yes            ; Turn on support for SIP video. You need to turn
this on
                                ; in the this section to get any video support at all.
                                ; You can turn it off on a per peer basis if the
general
                                ; video support is enabled, but you can't enable it
for
                                ; one peer only without enabling in the general
section.
;maxcallbitrate=384          ; Maximum bitrate for video calls (default 384 kb/s)
                                ; Videosupport and maxcallbitrate is settable
                                ; for peers and users as well
;callevts=no                 ; generate manager events when sip ua
                                ; performs events (e.g. hold)
;alwaysauthreject = yes      ; When an incoming INVITE or REGISTER is to be
rejected,
                                ; for any reason, always reject with '401
Unauthorized'
                                ; instead of letting the requester know whether there
was
                                ; a matching user or peer for their request

;g726nonstandard = yes      ; If the peer negotiates G726-32 audio, use
AAL2 packing
                                ; order instead of RFC3551 packing order (this is
required
                                ; for Sipura and Grandstream ATAs, among others).
This is
                                ; contrary to the RFC3551 specification, the peer
_should_
                                ; be negotiating AAL2-G726-32 instead :-(

;matchexterniplocally = yes  ; Only substitute the externip or externhost
setting if it matches
                                ; your localnet setting. Unless you have some
sort of strange network
                                ; setup you will not need to enable this.

;
; If regcontext is specified, Asterisk will dynamically create and destroy a
; NoOp priority 1 extension for a given peer who registers or unregisters with
; us and have a "regexten=" configuration item.
; Multiple contexts may be specified by separating them with '&'. The
; actual extension is the 'regexten' parameter of the registering peer or its
; name if 'regexten' is not provided. If more than one context is provided,
; the context must be specified within regexten by appending the desired
; context after '@'. More than one regexten may be supplied if they are
; separated by '&'. Patterns may be used in regexten.
;
;regcontext=sipregistrations
;
;----- RTP timers -----
;-----

```

```

; These timers are currently used for both audio and video streams. The RTP
timeouts
; are only applied to the audio channel.
; The settings are settable in the global section as well as per device
;
;rtptimeout=60                ; Terminate call if 60 seconds of no RTP or
RTCP activity
                                ; on the audio channel
                                ; when we're not on hold. This is to be able to
hangup
                                ; a call in the case of a phone disappearing from the
net,
                                ; like a powerloss or grandma tripping over a cable.
;rtpholdtimeout=300          ; Terminate call if 300 seconds of no RTP or RTCP
activity
                                ; on the audio channel
                                ; when we're on hold (must be > rtptimeout)
;rtptimeout=<secs>          ; Send keepalives in the RTP stream to keep
NAT open
                                ; (default is off - zero)
;----- SIP DEBUGGING -----
-----
;sipdebug = yes                ; Turn on SIP debugging by default, from
                                ; the moment the channel loads this configuration
;recordhistory=yes           ; Record SIP history by default
                                ; (see sip history / sip no history)
;dumphistory=yes            ; Dump SIP history at end of SIP dialogue
                                ; SIP history is output to the DEBUG logging channel

;----- STATUS NOTIFICATIONS (SUBSCRIPTIONS) -----
-----
; You can subscribe to the status of extensions with a "hint" priority
; (See extensions.conf.sample for examples)
; chan_sip support two major formats for notifications: dialog-info and SIMPLE
;
; You will get more detailed reports (busy etc) if you have a call limit set
; for a device. When the call limit is filled, we will indicate busy. Note that
; you need at least 2 in order to be able to do attended transfers.
;
; For queues, you will need this level of detail in status reporting,
regardless
; if you use SIP subscriptions. Queues and manager use the same internal
interface
; for reading status information.
;
; Note: Subscriptions does not work if you have a realtime dialplan and use the
; realtime switch.
;
;allowsubscribe=no          ; Disable support for subscriptions. (Default is yes)
;subscribecontext = default ; Set a specific context for SUBSCRIBE
requests
                                ; Useful to limit subscriptions to local extensions
                                ; Settable per peer/user also
;notifyringing = yes        ; Notify subscriptions on RINGING state
(default: no)
;notifyhold = yes          ; Notify subscriptions on HOLD state (default: no)
                                ; Turning on notifyringing and notifyhold will add a
lot
                                ; more database transactions if you are using
realtime.
;limitonpeers = yes        ; Apply call limits on peers only. This will improve
                                ; status notification when you are using type=friend
                                ; Inbound calls, that really apply to the user part

```

```

; of a friend will now be added to and compared with
; the peer limit instead of applying two call limits,
; one for the peer and one for the user.

;----- T.38 FAX PASSTHROUGH SUPPORT -----
;
; This setting is available in the [general] section as well as in device
; configurations.
; Setting this to yes, enables T.38 fax (UDPTL) passthrough on SIP to SIP calls,
; provided
; both parties have T38 support enabled in their Asterisk configuration
; This has to be enabled in the general section for all devices to work. You
; can then
; disable it on a per device basis.
;
; T.38 faxing only works in SIP to SIP calls, with no local or agent channel
; being used.
;
; t38pt_udptl = yes           ; Default false
;
;----- OUTBOUND SIP REGISTRATIONS -----
; Asterisk can register as a SIP user agent to a SIP proxy (provider)
; Format for the register statement is:
;     register => user[:secret[:authuser]]@host[:port][/extension]
;
; If no extension is given, the 's' extension is used. The extension needs to
; be defined in extensions.conf to be able to accept calls from this SIP proxy
; (provider).
;
; host is either a host name defined in DNS or the name of a section defined
; below.
;
; Examples:
;
;register => 1234:password@mysipprovider.com
;
;     This will pass incoming calls to the 's' extension
;
;
;register => 2345:password@sip_proxy/1234
;
; Register 2345 at sip provider 'sip_proxy'. Calls from this provider
; connect to local extension 1234 in extensions.conf, default context,
; unless you configure a [sip_proxy] section below, and configure a
; context.
; Tip 1: Avoid assigning hostname to a sip.conf section like [provider.com]
; Tip 2: Use separate type=peer and type=user sections for SIP providers
;       (instead of type=friend) if you have calls in both directions

;registertimeout=20           ; retry registration calls every 20 seconds (default)
;registerattempts=10          ; Number of registration attempts before we
; give up
;
;                               ; 0 = continue forever, hammering the other server
;                               ; until it accepts the registration
;                               ; Default is 0 tries, continue forever

;----- NAT SUPPORT -----
; The externip, externhost and localnet settings are used if you use Asterisk
; behind a NAT device to communicate with services on the outside.

;externip = 200.201.202.203    ; Address that we're going to put in outbound
; SIP

```

```

; messages if we're behind a NAT

; The externip and localnet is used
; when registering and communicating with other
proxies
; that we're registered with
;externhost=foo.dyndns.net ; Alternatively you can specify an
; external host, and Asterisk will
; perform DNS queries periodically. Not
; recommended for production
; environments! Use externip instead
;externrefresh=10 ; How often to refresh externhost if
; used
; You may add multiple local networks. A reasonable
; set of defaults are:
;localnet=192.168.0.0/255.255.0.0; All RFC 1918 addresses are local networks
;localnet=10.0.0.0/255.0.0.0 ; Also RFC1918
;localnet=172.16.0.0/12 ; Another RFC1918 with CIDR notation
;localnet=169.254.0.0/255.255.0.0 ;Zero conf local network

; The nat= setting is used when Asterisk is on a public IP, communicating with
; devices hidden behind a NAT device (broadband router). If you have one-way
; audio problems, you usually have problems with your NAT configuration or your
; firewall's support of SIP+RTP ports. You configure Asterisk choice of RTP
; ports for incoming audio in rtp.conf
;
;nat=no ; Global NAT settings (Affects all peers and
users)
; yes = Always ignore info and assume NAT
; no = Use NAT mode only according to RFC3581

(/rport)
; never = Never attempt NAT mode or RFC3581

support
; route = Assume NAT, don't send rport
; (work around more UNIDEN bugs)

;----- MEDIA HANDLING -----
-----
; By default, Asterisk tries to re-invite the audio to an optimal path. If
there's
; no reason for Asterisk to stay in the media path, the media will be
redirected.
; This does not really work with in the case where Asterisk is outside and have
; clients on the inside of a NAT. In that case, you want to set
canreinvite=nonat
;
;canreinvite=yes ; Asterisk by default tries to redirect the
; RTP media stream (audio) to go directly from
; the caller to the callee. Some devices do not
; support this (especially if one of them is behind a
NAT).
; The default setting is YES. If you have all clients
; behind a NAT, or for some other reason wants
Asterisk to
; stay in the audio path, you may want to turn this
off.
; In Asterisk 1.4 this setting also affect direct RTP
; at call setup (a new feature in 1.4 - setting up
the
; call directly between the endpoints instead of
sending
; a re-INVITE).

```

```

;directrtpsetup=yes          ; Enable the new experimental direct RTP setup. This
sets up                      ; the call directly with media peer-2-peer without
re-invites.                  ; Will not work for video and cases where the callee
sends                        ; RTP payloads and fntp headers in the 200 OK that
does not match the          ; callers INVITE.

;canreinvite=nonat          ; An additional option is to allow media path
redirection                  ; (reinvite) but only when the peer where the media
is being                    ; sent is known to not be behind a NAT (as the RTP
core can                    ; determine it based on the apparent IP address the
media                       ; arrives from).

;canreinvite=update         ; Yet a third option... use UPDATE for media path
redirection,                 ; instead of INVITE. This can be combined with
'nonat', as                  ; 'canreinvite=update,nonat'. It implies 'yes'.

;----- REALTIME SUPPORT -----
;
; For additional information on ARA, the Asterisk Realtime Architecture,
; please read realtime.txt and extconfig.txt in the /doc directory of the
; source code.
;
;rtcachefriends=yes        ; Cache realtime friends by adding them to the
internal list                ; just like friends added from the config file only
on a                        ; as-needed basis? (yes|no)

;rtsavesysname=yes         ; Save systemname in realtime database at
registration                 ; Default= no

;rtupdate=yes              ; Send registry updates to database using realtime?
(yes|no)                     ; If set to yes, when a SIP UA registers successfully,
the ip address,              ; the origination port, the registration period, and
the username of              ; the UA will be set to database via realtime.
                             ; If not present, defaults to 'yes'.
;rtautoclear=yes           ; Auto-Expire friends created on the fly on the same
schedule                     ; as if it had just registered? (yes|no|<seconds>)
                             ; If set to yes, when the registration expires, the
friend will                  ; vanish from the configuration until requested again.
If set                       ; to an integer, friends expire within this number of
seconds                      ; instead of the registration interval.

;ignoreregexpire=yes       ; Enabling this setting has two functions:
;

```

```

; For non-realtime peers, when their registration
expires, the
the Asterisk database
existing information
; information will _not_ be removed from memory or
; if you attempt to place a call to the peer, the
; will be used in spite of it having expired
;
; For realtime peers, when the peer is retrieved from
realtime storage,
regardless of whether
realtime peer
reasons), the
storage
; the registration information will be used
; it has expired or not; if it expires while the
; is still in memory (due to caching or other
; information will not be removed from realtime

```

```

;----- SIP DOMAIN SUPPORT -----
-----

```

```

; Incoming INVITE and REFER messages can be matched against a list of 'allowed'
; domains, each of which can direct the call to a specific context if desired.
; By default, all domains are accepted and sent to the default context or the
; context associated with the user/peer placing the call.
; Domains can be specified using:
; domain=<domain>[,<context>]
; Examples:
; domain=myasterisk.dom
; domain=customer.com,customer-context
;
; In addition, all the 'default' domains associated with a server should be
; added if incoming request filtering is desired.
; automain=yes
;
; To disallow requests for domains not serviced by this server:
; allowexternaldomains=no

```

```

;domain=mydomain.tld,mydomain-incoming
; Add domain and configure incoming context
; for external calls to this domain
;domain=1.2.3.4
; Add IP address as local domain
; You can have several "domain" settings
;allowexternalinvites=no
; Disable INVITE and REFER to non-local domains
; Default is yes
;automain=yes
; Turn this on to have Asterisk add local host
; name and local IP to domain list.

; fromdomain=mydomain.tld
; When making outbound SIP INVITES to
; non-peers, use your primary domain "identity"
; for From: headers instead of just your IP
; address. This is to be polite and
; it may be a mandatory requirement for some
; destinations which do not have a prior
; account relationship with your server.

```

```

;----- JITTER BUFFER CONFIGURATION -----
-----

```

```

; jbenable = yes
receiving side of a
jitterbuffer will
the receiving
; Enables the use of a jitterbuffer on the
; SIP channel. Defaults to "no". An enabled
; be used only if the sending side can create and

```

```

accept jitter,
will be used only

; side can not accept jitter. The SIP channel can
; thus a jitterbuffer on the receive SIP side
; if it is forced and enabled.

; jbforce = no
side of a SIP
; Forces the use of a jitterbuffer on the receive
; channel. Defaults to "no".

; jbmaxsize = 200
; Max length of the jitterbuffer in milliseconds.

; jbresyncthreshold = 1000
jitterbuffer is
; Jump in the frame timestamps over which the
; resynchronized. Useful to improve the quality
of the voice, with
; big jumps in/broken timestamps, usually sent
from exotic devices
; and programs. Defaults to 1000.

; jbimpl = fixed
receiving side of a SIP
; Jitterbuffer implementation, used on the
; channel. Two implementations are currently
available - "fixed"
; (with size always equals to jbmaxsize) and
"adaptive" (with
; variable size, actually the new jb of IAX2).
Defaults to fixed.

; jblog = no
; Enables jitterbuffer frame logging. Defaults to
"no".
;-----
-----

[authentication]
; Global credentials for outbound calls, i.e. when a proxy challenges your
; Asterisk server for authentication. These credentials override
; any credentials in peer/register definition if realm is matched.
;
; This way, Asterisk can authenticate for outbound calls to other
; realms. We match realm on the proxy challenge and pick an set of
; credentials from this list
; Syntax:
;   auth = <user>:<secret>@<realm>
;   auth = <user>#<md5secret>@<realm>
; Example:
;auth=mark:topsecret@digium.com
;
; You may also add auth= statements to [peer] definitions
; Peer auth= override all other authentication settings if we match on realm
;-----

; Users and peers have different settings available. Friends have all settings,
; since a friend is both a peer and a user
;
; User config options:          Peer configuration:
; -----
; context                      context
; callingpres                   callingpres
; permit                       permit
; deny                         deny
; secret                       secret
; md5secret                    md5secret
; dtmfmode                     dtmfmode

```

```

; canreinvite          canreinvite
; nat                  nat
; callgroup            callgroup
; pickupgroup          pickupgroup
; language              language
; allow                 allow
; disallow              disallow
; insecure              insecure
; trustpid              trustpid
; progressinband       progressinband
; promiscredirect       promiscredirect
; useclientcode         useclientcode
; accountcode           accountcode
; setvar                 setvar
; callerid              callerid
; amaflags              amaflags
; call-limit            call-limit
; allowoverlap          allowoverlap
; allowsubscribe        allowsubscribe
; allowtransfer         allowtransfer
; subscribecontext      subscribecontext
; videosupport          videosupport
; maxcallbitrate        maxcallbitrate
; rfc2833compensate     mailbox
;                       username
;                       template
;                       fromdomain
;                       regexten
;                       fromuser
;                       host
;                       port
;                       qualify
;                       defaultip
;                       rtptimeout
;                       rtpholdtimeout
;                       sendrpid
;                       outboundproxy
;                       rfc2833compensate

;[sip_proxy]
; For incoming calls only. Example: FWD (Free World Dialup)
; We match on IP address of the proxy for incoming calls
; since we can not match on username (caller id)
;type=peer
;context=from-fwd
;host=fwd.pulver.com

;[sip_proxy-out]
;type=peer                ; we only want to call out, not be
called
;secret=guessit
;username=yourusername    ; Authentication user for outbound
proxies
;fromuser=yourusername    ; Many SIP providers require this!
;fromdomain=provider.sip.domain
;host=box.provider.com
;usereqphone=yes         ; This provider requires ";user=phone" on URI
;call-limit=5            ; permit only 5 simultaneous outgoing calls to
this peer
;outboundproxy=proxy.provider.domain ; send outbound signaling to this proxy,
not directly to the peer
time peers,                ; Call-limits will not be enforced on real-
; since they are not stored in-memory

```

```

;port=80 ; The port number we want to connect to on the
remote side
; Also used as "defaultport" in combination
with "defaultip" settings

;-----
; Definitions of locally connected SIP devices
;
; type = usera device that authenticates to us by "from" field to place calls
; type = peera device we place calls to or that calls us and we match by host
; type = friend two configurations (peer+user) in one
;
; For device names, we recommend using only a-z, numerics (0-9) and underscore
;
; For local phones, type=friend works most of the time
;
; If you have one-way audio, you probably have NAT problems.
; If Asterisk is on a public IP, and the phone is inside of a NAT device
; you will need to configure nat option for those phones.
; Also, turn on qualify=yes to keep the nat session open

;[grandstream1]
;type=friend
;context=from-sip ; Where to start in the dialplan when this phone
calls
;callerid=John Doe <1234> ; Full caller ID, to override the phones config
; on incoming calls to Asterisk
;host=192.168.0.23 ; we have a static but private IP address
; No registration allowed
;nat=no ; there is not NAT between phone and Asterisk
;canreinvite=yes ; allow RTP voice traffic to bypass Asterisk
;dtmfmode=info ; either RFC2833 or INFO for the BudgeTone
;call-limit=1 ; permit only 1 outgoing call and 1 incoming call at
a time
; from the phone to asterisk
; 1 for the explicit peer, 1 for the explicit user,
; remember that a friend equals 1 peer and 1 user in
; memory
; This will affect your subscriptions as well.
; There is no combined call counter for a "friend"
; so there's currently no way in sip.conf to limit
; to one inbound or outbound call per phone. Use
; the group counters in the dial plan for that.
;
;mailbox=1234@default ; mailbox 1234 in voicemail context "default"
;disallow=all ; need to disallow=all before we can use allow=
;allow=ulaw ; Note: In user sections the order of codecs
; listed with allow= does NOT matter!
;allow=alaw
;allow=g723.1 ; Asterisk only supports g723.1 pass-thru!
;allow=g729 ; Pass-thru only unless g729 license obtained
;callingpres=allowed_passed_screen ; Set caller ID presentation
; See README.callingpres for more information

;[xlite1]
; Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
; Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
;type=friend
;regexten=1234 ; When they register, create extension 1234
;callerid="Jane Smith" <5678>
;host=dynamic ; This device needs to register
;nat=yes ; X-Lite is behind a NAT router
;canreinvite=no ; Typically set to NO if behind NAT

```

```

;disallow=all
;allow=gsm           ; GSM consumes far less bandwidth than ulaw
;allow=ulaw
;allow=alaw
;mailbox=1234@default,1233@default      ; Subscribe to status of multiple
mailboxes

;[snom]
;type=friend         ; Friends place calls and receive calls
;context=from-sip    ; Context for incoming calls from this user
;secret=blah
;subscribecontext=localextensions ; Only allow SUBSCRIBE for local extensions
;language=de         ; Use German prompts for this user
;host=dynamic        ; This peer register with us
;dtmfmode=inband    ; Choices are inband, rfc2833, or info
;defaultip=192.168.0.59 ; IP used until peer registers
;mailbox=1234@context,2345 ; Mailbox(-es) for message waiting indicator
;subscribemwi=yes    ; Only send notifications if this phone
;                   ; subscribes for mailbox notification
;vmexten=voicemail   ; dialplan extension to reach mailbox
;                   ; sets the Message-Account in the MWI notify message
;                   ; defaults to global vmexten which defaults to

"asterisk"
;disallow=all
;allow=ulaw          ; dtmfmode=inband only works with ulaw or alaw!

;[polycom]
;type=friend         ; Friends place calls and receive calls
;context=from-sip    ; Context for incoming calls from this user
;secret=blahpoly
;host=dynamic        ; This peer register with us
;dtmfmode=rfc2833    ; Choices are inband, rfc2833, or info
;username=polly      ; Username to use in INVITE until peer
registers
;                   ; Normally you do NOT need to set this parameter

;disallow=all
;allow=ulaw          ; dtmfmode=inband only works with ulaw or alaw!
;progressinband=no   ; Polycom phones don't work properly with "never"

;[pingtel]
;type=friend
;secret=blah
;host=dynamic
;insecure=port       ; Allow matching of peer by IP address without
;                   ; matching port number
;insecure=invite     ; Do not require authentication of incoming INVITES
;insecure=port,invite ; (both)
;qualify=1000        ; Consider it down if it's 1 second to reply
;                   ; Helps with NAT session
;                   ; qualify=yes uses default value
;
; Call group and Pickup group should be in the range from 0 to 63
;
;callgroup=1,3-4     ; We are in caller groups 1,3,4
;pickupgroup=1,3-5   ; We can do call pick-p for call group 1,3,4,5
;defaultip=192.168.0.60 ; IP address to use if peer has not registered
;deny=0.0.0.0/0.0.0.0 ; ACL: Control access to this account based on
IP address
;permit=192.168.0.60/255.255.255.0

;[cisco1]

```

```

;type=friend
;secret=blah
;qualify=200           ; Qualify peer is no more than 200ms away
;nat=yes              ; This phone may be natted
                     ; Send SIP and RTP to the IP address that packet is
                     ; received from instead of trusting SIP headers
;host=dynamic        ; This device registers with us
;canreinvite=no      ; Asterisk by default tries to redirect the
                     ; RTP media stream (audio) to go directly from
                     ; the caller to the callee. Some devices do not
                     ; support this (especially if one of them is
                     ; behind a NAT).
;defaultip=192.168.0.4 ; IP address to use until registration
;username=goran      ; Username to use when calling this device
before registration

;setvar=CUSTID=5678   ; Normally you do NOT need to set this parameter
device               ; Channel variable to be set for all calls from this

;[pre14-asterisk]
;type=friend
;secret=digium
;host=dynamic
;rfc2833compensate=yes ; Compensate for pre-1.4 DTMF transmission
from another Asterisk machine.
                     ; You must have this turned on or DTMF reception will
work improperly.

```

3.5.4 Osp.conf

```

;
; Open Settlement Protocol Sample Configuration File
;
;
; This file contains configuration of providers that
; are used by the OSP subsystem of Asterisk. The section
; "general" is reserved for global options. Each other
; section declares an OSP Provider. The provider "default"
; is used when no provider is otherwise specified.
;
[general]
;
; Should hardware acceleration be enabled? May not be changed
; on a reload.
;
accelerate=no
;
; Defines the token format that Asterisk can validate.
; 0 - signed tokens only
; 1 - unsigned tokens only
; 2 - both signed and unsigned
; The defaults to 0, i.e. the Asterisk can validate signed tokens only.
;
tokenformat=0

[default]
;
; All paths are presumed to be under /var/lib/asterisk/keys unless
; the path begins with '/'
;
; Specify the private keyfile. If unspecified, defaults to the name
; of the section followed by "-privatekey.pem" (e.g. default-privatekey.pem)
;
privatekey=pkey.pem
;

```

```

; Specify the local certificate file.  If unspecified, defaults to
; the name of the section followed by "-localcert.pem"
;
localcert=localcert.pem
;
; Specify one or more Certificate Authority keys.  If none are listed,
; a single one is added with the name "-cacert.pem"
;
cacert=cacert_0.pem
;
; Specific parameters can be tuned as well:
;
; maxconnections: Max number of simultaneous connections to the provider
(default=20)
; retrydelay:      Extra delay between retries (default=0)
; retrylimit:     Max number of retries before giving up (default=2)
; timeout:        Timeout for response in milliseconds (default=500)
;
maxconnections=20
retrydelay=0
retrylimit=2
timeout=500
;
; List all service points for this provider
;
servicepoint=http://172.16.4.75:1080/osp
;
; Set the "source" for requesting authorization
;
;source=foo
source=[172.16.4.32]
;
; Set the authentication policy.
; 0 - NO
; 1 - YES
; 2 - EXCLUSIVE
; Default is 1, validate token but allow no token.
;
authpolicy=1

```

3.5.5 Extensions.conf

```

; extensions.conf - the Asterisk dial plan
;
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;
; This configuration file is reloaded
; - With the "extensions reload" command in the CLI
; - With the "reload" command (that reloads everything) in the CLI
;
; The "General" category is for certain variables.
;
[general]
;
; If static is set to no, or omitted, then the pbx_config will rewrite
; this file when extensions are modified. Remember that all comments
; made in the file will be lost when that happens.
;
; XXX Not yet implemented XXX
;
static=yes
;

```

```

; if static=yes and writeprotect=no, you can save dialplan by
; CLI command 'save dialplan' too
;
writeprotect=no
;
; If autofallthrough is set, then if an extension runs out of
; things to do, it will terminate the call with BUSY, CONGESTION
; or HANGUP depending on Asterisk's best guess. This is the default.
;
; If autofallthrough is not set, then if an extension runs out of
; things to do, Asterisk will wait for a new extension to be dialed
; (this is the original behavior of Asterisk 1.0 and earlier).
;
;autofallthrough=no
;
; If clearglobalvars is set, global variables will be cleared
; and reparsed on an extensions reload, or Asterisk reload.
;
; If clearglobalvars is not set, then global variables will persist
; through reloads, and even if deleted from the extensions.conf or
; one of its included files, will remain set to the previous value.
;
; NOTE: A complication sets in, if you put your global variables into
; the AEL file, instead of the extensions.conf file. With clearglobalvars
; set, a 'reload' will often leave the globals vars cleared, because it
; is not unusual to have extensions.conf (which will have no globals)
; load after the extensions.ael file (where the global vars are stored).
; So, with 'reload' in this particular situation, first the AEL file will
; clear and then set all the global vars, then, later, when the extensions.conf
; file is loaded, the global vars are all cleared, and then not set, because
; they are not stored in the extensions.conf file.
;
clearglobalvars=no
;
; If priorityjumping is set to 'yes', then applications that support
; 'jumping' to a different priority based on the result of their operations
; will do so (this is backwards compatible behavior with pre-1.2 releases
; of Asterisk). Individual applications can also be requested to do this
; by passing a 'j' option in their arguments.
;
;priorityjumping=yes
;
; User context is where entries from users.conf are registered. The
; default value is 'default'
;
;userscontext=default
;
; You can include other config files, use the #include command
; (without the ';'). Note that this is different from the "include" command
; that includes contexts within other contexts. The #include command works
; in all asterisk configuration files.
#include "filename.conf"

; The "Globals" category contains global variables that can be referenced
; in the dialplan with the GLOBAL dialplan function:
; ${GLOBAL(VARIABLE)}
; ${${GLOBAL(VARIABLE)}} or ${text${GLOBAL(VARIABLE)}} or any hybrid
; Unix/Linux environmental variables can be reached with the ENV dialplan
; function: ${ENV(VARIABLE)}
;
[globals]
CONSOLE=Console/dsp ; Console interface for demo
;CONSOLE=Zap/1
;CONSOLE=Phone/phone0

```

```

IAXINFO=guest ; IAXtel username/password
;IAXINFO=myuser:mypass
TRUNK=Zap/g2 ; Trunk interface
;
; Note the 'g2' in the TRUNK variable above. It specifies which group (defined
; in zapata.conf) to dial, i.e. group 2, and how to choose a channel to use in
; the specified group. The four possible options are:
;
; g: select the lowest-numbered non-busy Zap channel
; (aka. ascending sequential hunt group).
; G: select the highest-numbered non-busy Zap channel
; (aka. descending sequential hunt group).
; r: use a round-robin search, starting at the next highest channel than last
; time (aka. ascending rotary hunt group).
; R: use a round-robin search, starting at the next lowest channel than last
; time (aka. descending rotary hunt group).
;
TRUNKMSD=1 ; MSD digits to strip (usually 1 or 0)
;TRUNK=IAX2/user:pass@provider

;-----Asterisk/OSP Configuration by Di-Shi Sun-----
[GeneralProxy] ; Proxy
exten => _XXXX.,1,NoOp(OSP-GeneralProxy)
exten => _XXXX.,n,Set(OSPPEERIP=${SIPCHANINFO(peerip)})
exten => _XXXX.,n,Set(OSPINTOKEN=${SIP_HEADER(P-OSP-Auth-Token)})
exten => _XXXX.,n,OSPAuth()
exten => _XXXX.,n,GotoIf($["${OSPAUTHSTATUS}"!="SUCCESS"]?100)
exten => _XXXX.,n,Set(OSPINTIMELIMIT=${OSPINTIMELIMIT})
exten => _XXXX.,n,OSPlookup(${EXTEN})
exten => _XXXX.,n,GotoIf($["${OSPLOOKUPSTATUS}"!="SUCCESS"]?100)
exten => _XXXX.,n(next),Set(CALLERID(number)=${OSPCALLING})
exten => _XXXX.,n,Dial(${OSPTECH}/${OSPDEST},60,oL(${OSPOUTTIMELIMIT}*1000))
exten => _XXXX.,n,NoOp(${DIALSTATUS})
exten => _XXXX.,n,OSPNext(${HANGUPCAUSE})
exten => _XXXX.,n,GotoIf($["${OSPNEXTSTATUS}"!="SUCCESS"]?100)
exten => _XXXX.,n,Goto(next)
exten => _XXXX.,100,Hangup
exten => h,1,NoOp(OSP-GeneralProxy-Hangup)
;-----Asterisk/OSP Configuration by Di-Shi Sun-----

;
; Any category other than "General" and "Globals" represent
; extension contexts, which are collections of extensions.
;
; Extension names may be numbers, letters, or combinations
; thereof. If an extension name is prefixed by a '_'
; character, it is interpreted as a pattern rather than a
; literal. In patterns, some characters have special meanings:
;
; X - any digit from 0-9
; Z - any digit from 1-9
; N - any digit from 2-9
; [1235-9] - any digit in the brackets (in this example, 1,2,3,5,6,7,8,9)
; . - wildcard, matches anything remaining (e.g. _9011. matches
; anything starting with 9011 excluding 9011 itself)
; ! - wildcard, causes the matching process to complete as soon as
; it can unambiguously determine that no other matches are possible
;
; For example the extension _NXXXXXX would match normal 7 digit dialings,
; while _1NXXNXXXXXX would represent an area code plus phone number
; preceded by a one.
;
; Each step of an extension is ordered by priority, which must
; always start with 1 to be considered a valid extension. The priority

```

```

; "next" or "n" means the previous priority plus one, regardless of whether
; the previous priority was associated with the current extension or not.
; The priority "same" or "s" means the same as the previously specified
; priority, again regardless of whether the previous entry was for the
; same extension. Priorities may be immediately followed by a plus sign
; and another integer to add that amount (most useful with 's' or 'n').
; Priorities may then also have an alias, or label, in
; parenthesis after their name which can be used in goto situations
;
; Contexts contain several lines, one for each step of each
; extension, which can take one of two forms as listed below,
; with the first form being preferred. One may include another
; context in the current one as well, optionally with a
; date and time. Included contexts are included in the order
; they are listed.
;
;[context]
;exten => someexten,{priority|label{+|-
;offset}[(alias)],application(arg1,arg2,...)
;exten => someexten,{priority|label{+|-
;offset}[(alias)],application,arg1|arg2...
;
; Timing list for includes is
;
; <time range>|<days of week>|<days of month>|<months>
;
; Note that ranges may be specified to wrap around the ends. Also, minutes are
; fine-grained only down to the closest even minute.
;
;include => daytime|9:00-17:00|mon-fri|*|*
;include => weekend|*|sat-sun|*|*
;include => weeknights|17:02-8:58|mon-fri|*|*
;
; ignorepat can be used to instruct drivers to not cancel dialtone upon
; receipt of a particular pattern. The most commonly used example is
; of course '9' like this:
;
;ignorepat => 9
;
; so that dialtone remains even after dialing a 9.
;
;
; Sample entries for extensions.conf
;
;
[dundi-e164-canonical]
;
; List canonical entries here
;
;exten => 12564286000,1,Macro(stdexten,6000,IAX2/foo)
;exten => _125642860XX,1,Dial(IAX2/otherbox/${EXTEN:7})

[dundi-e164-customers]
;
; If you are an ITSP or Reseller, list your customers here.
;
;exten => _12564286000,1,Dial(SIP/customer1)
;exten => _12564286001,1,Dial(IAX2/customer2)

[dundi-e164-via-pstn]
;
; If you are freely delivering calls to the PSTN, list them here
;

```

```

;exten => _1256428XXXX,1,Dial(Zap/g2/${EXTEN:7}) ; Expose all of 256-428
;exten => _1256325XXXX,1,Dial(Zap/g2/${EXTEN:7}) ; Ditto for 256-325

[dundi-e164-local]
;
; Context to put your dundi IAX2 or SIP user in for
; full access
;
include => dundi-e164-canonical
include => dundi-e164-customers
include => dundi-e164-via-pstn

[dundi-e164-switch]
;
; Just a wrapper for the switch
;
switch => DUNDi/e164

[dundi-e164-lookup]
;
; Locally to lookup, try looking for a local E.164 solution
; then try DUNDi if we don't have one.
;
include => dundi-e164-local
include => dundi-e164-switch
;
; DUNDi can also be implemented as a Macro instead of using
; the Local channel driver.
;
[macro-dundi-e164]
;
; ARG1 is the extension to Dial
;
; Extension "s" is not a wildcard extension that matches "anything".
; In macros, it is the start extension. In most other cases,
; you have to goto "s" to execute that extension.
;
; For wildcard matches, see above - all pattern matches start with
; an underscore.
exten => s,1,Goto(${ARG1},1)
include => dundi-e164-lookup

;
; Here are the entries you need to participate in the IAXTEL
; call routing system. Most IAXTEL numbers begin with 1-700, but
; there are exceptions. For more information, and to sign
; up, please go to www.gnophone.com or www.iaxtel.com
;
[iaxtel700]
exten =>
_91700XXXXXXX,1,Dial(IAX2/${GLOBAL(IAXINFO)}@iaxtel.com/${EXTEN:1}@iaxtel)

;
; The SWITCH statement permits a server to share the dialplan with
; another server. Use with care: Reciprocal switch statements are not
; allowed (e.g. both A -> B and B -> A), and the switched server needs
; to be on-line or else dialing can be severely delayed.
;
[ixaprovider]
;switch => IAX2/user:[key]@myserver/mycontext

[trunkint]
;
; International long distance through trunk

```

```

;
exten => _9011.,1,Macro(dundi-e164,${EXTEN:4})
exten => _9011.,n,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})

[trunkld]
;
; Long distance context accessed through trunk
;
exten => _91NXXNXXXXXX,1,Macro(dundi-e164,${EXTEN:1})
exten => _91NXXNXXXXXX,n,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})

[trunklocal]
;
; Local seven-digit dialing accessed through trunk interface
;
exten => _9NXXXXXX,1,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})

[trunktollfree]
;
; Long distance context accessed through trunk interface
;
exten => _91800NXXXXXX,1,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})
exten => _91888NXXXXXX,1,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})
exten => _91877NXXXXXX,1,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})
exten => _91866NXXXXXX,1,Dial(${GLOBAL(TRUNK)}/${EXTEN:${GLOBAL(TRUNKMSD)}})

[international]
;
; Master context for international long distance
;
ignorepat => 9
include => longdistance
include => trunkint

[longdistance]
;
; Master context for long distance
;
ignorepat => 9
include => local
include => trunkld

[local]
;
; Master context for local, toll-free, and iaxtel calls only
;
ignorepat => 9
include => default
include => trunklocal
include => iaxtel700
include => trunktollfree
include => iaxprovider

;Include parkedcalls (or the context you define in features conf)
;to enable call parking.
include => parkedcalls
;
; You can use an alternative switch type as well, to resolve
; extensions that are not known here, for example with remote
; IAX switching you transparently get access to the remote
; Asterisk PBX
;
; switch => IAX2/user:password@bigserver/local
;

```

```

; An "lswitch" is like a switch but is literal, in that
; variable substitution is not performed at load time
; but is passed to the switch directly (presumably to
; be substituted in the switch routine itself)
;
; lswitch => Loopback/12${EXTEN}@othercontext
;
; An "eswitch" is like a switch but the evaluation of
; variable substitution is performed at runtime before
; being passed to the switch routine.
;
; eswitch => IAX2/context@${CURSERVER}

[macro-trunkdial]
;
; Standard trunk dial macro (hangs up on a dialstatus that should
; terminate call)
;   ${ARG1} - What to dial
;
exten => s,1,Dial(${ARG1})
exten => s,n,Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER,1,Hangup
exten => s-BUSY,1,Hangup
exten => _s-. ,1,NoOp

[macro-stdexten];
;
; Standard extension macro:
;   ${ARG1} - Extension (we could have used ${MACRO_EXTEN} here as well
;   ${ARG2} - Device(s) to ring
;
exten => s,1,Dial(${ARG2},20)                ; Ring the interface, 20 seconds
maximum                                     ;
exten => s,2,Goto(s-${DIALSTATUS},1)         ; Jump based on status
(NOANSWER,BUSY,CHANUNAVAIL,CONGESTION,ANSWER)

exten => s-NOANSWER,1,VoiceMail(${ARG1},u)   ; If unavailable, send to
voiceMail w/ unavail announce
exten => s-NOANSWER,2,Goto(default,s,1)     ; If they press #, return to
start

exten => s-BUSY,1,VoiceMail(${ARG1},b)      ; If busy, send to voiceMail w/
busy announce
exten => s-BUSY,2,Goto(default,s,1)         ; If they press #, return to
start

exten => _s-. ,1,Goto(s-NOANSWER,1)         ; Treat anything else as no
answer

exten => a,1,VoiceMailMain(${ARG1})        ; If they press *, send the user
into VoiceMailMain

[macro-stdPrivacyexten];
;
; Standard extension macro:
;   ${ARG1} - Extension (we could have used ${MACRO_EXTEN} here as well
;   ${ARG2} - Device(s) to ring
;   ${ARG3} - Optional DONTCALL context name to jump to (assumes the s,1
extension-priority)
;   ${ARG4} - Optional TORTURE context name to jump to (assumes the s,1
extension-priority)`
;
exten => s,1,Dial(${ARG2},20|p)             ; Ring the interface, 20 seconds
maximum, call screening

```

```

; option (or use P for databased call
screening)
exten => s,2,Goto(s- $\{DIALSTATUS\}$ ,1) ; Jump based on status
(NOANSWER,BUSY,CHANUNAVAIL,CONGESTION,ANSWER)

exten => s-NOANSWER,1,VoiceMail( $\{ARG1\}$ ,u) ; If unavailable, send to
voiceMail w/ unavail announce
exten => s-NOANSWER,2,Goto(default,s,1) ; If they press #, return to
start

exten => s-BUSY,1,VoiceMail( $\{ARG1\}$ ,b) ; If busy, send to voiceMail w/
busy announce
exten => s-BUSY,2,Goto(default,s,1) ; If they press #, return to
start

exten => s-DONTCALL,1,Goto( $\{ARG3\}$ ,s,1) ; Callee chose to send this call
to a polite "Don't call again" script.

exten => s-TORTURE,1,Goto( $\{ARG4\}$ ,s,1) ; Callee chose to send this call
to a telemarketer torture script.

exten => _s-.,1,Goto(s-NOANSWER,1) ; Treat anything else as no
answer

exten => a,1,VoiceMailMain( $\{ARG1\}$ ) ; If they press *, send the user
into VoiceMailMain

[macro-page];
;
; Paging macro:
;
; Check to see if SIP device is in use and DO NOT PAGE if they are
;
;  $\{ARG1\}$  - Device to page

exten => s,1,ChanIsAvail( $\{ARG1\}$ |js) ; j is for Jump and s is
for ANY call
exten => s,n,GoToIf( $\{AVAILSTATUS\}$  = "1"?autoanswer:fail)
exten => s,n(autoanswer),Set(_ALERT_INFO="RA") ; This is for the
PolyComs
exten => s,n,SIPAddHeader(Call-Info: Answer-After=0) ; This is for the
Grandstream, Snoms, and Others
exten => s,n,NoOp() ; Add others here and Post on the
Wiki!!!!
exten => s,n,Dial( $\{ARG1\}$ ||)
exten => s,n(fail),Hangup

[demo]
;
; We start with what to do when a call first comes in.
;
exten => s,1,Wait(1) ; Wait a second, just for fun
exten => s,n,Answer ; Answer the line
exten => s,n,Set(TIMEOUT(digit)=5) ; Set Digit Timeout to 5 seconds
exten => s,n,Set(TIMEOUT(response)=10) ; Set Response Timeout to 10 seconds
exten => s,n(restart),Background(demo-congrats) ; Play a congratulatory
message
exten => s,n(instruct),Background(demo-instruct) ; Play some instructions
exten => s,n,WaitExten ; Wait for an extension to be dialed.

exten => 2,1,Background(demo-moreinfo) ; Give some more information.
exten => 2,n,Goto(s,instruct)

```

```

exten => 3,1,Set(LANGUAGE(=fr)           ; Set language to french
exten => 3,n,Goto(s,restart)             ; Start with the congratulations

exten => 1000,1,Goto(default,s,1)
;
; We also create an example user, 1234, who is on the console and has
; voicemail, etc.
;
exten => 1234,1,Playback(transfer,skip)   ; "Please hold while..."
; (but skip if channel is not up)
exten => 1234,n,Macro(stdexten,1234,${GLOBAL(CONSOLE)})

exten => 1235,1,Voicemail(1234,u)         ; Right to voicemail

exten => 1236,1,Dial(Console/dsp)         ; Ring forever
exten => 1236,n,Voicemail(1234,b)        ; Unless busy

;
; # for when they're done with the demo
;
exten => #,1,Playback(demo-thanks)        ; "Thanks for trying the demo"
exten => #,n,Hangup                       ; Hang them up.

;
; A timeout and "invalid extension rule"
;
exten => t,1,Goto(#,1)                   ; If they take too long, give up
exten => i,1,Playback(invalid)           ; "That's not valid, try again"

;
; Create an extension, 500, for dialing the
; Asterisk demo.
;
exten => 500,1,Playback(demo-abouttotry); Let them know what's going on
exten => 500,n,Dial(IAX2/guest@misery.digium.com/s@default) ; Call the Asterisk
demo
exten => 500,n,Playback(demo-nogo)        ; Couldn't connect to the demo site
exten => 500,n,Goto(s,6)                 ; Return to the start over message.

;
; Create an extension, 600, for evaluating echo latency.
;
exten => 600,1,Playback(demo-echotest)    ; Let them know what's going on
exten => 600,n,Echo                       ; Do the echo test
exten => 600,n,Playback(demo-echodone)    ; Let them know it's over
exten => 600,n,Goto(s,6)                 ; Start over

;
; You can use the Macro Page to intercom a individual user
exten => 76245,1,Macro(page,SIP/Grandstream1)
; or if your peernames are the same as extensions
exten => _7XXX,1,Macro(page,SIP/${EXTEN})
;
;
; System Wide Page at extension 7999
;
exten => 7999,1,Set(TIMEOUT(absolute)=60)
exten =>
7999,2,Page(Local/Grandstream1@page&Local/Xlite1@page&Local/1234@page/n|d)

; Give voicemail at extension 8500
;
exten => 8500,1,VoicemailMain
exten => 8500,n,Goto(s,6)

```

```

;
; Here's what a phone entry would look like (IXJ for example)
;
;exten => 1265,1,Dial(Phone/phone0,15)
;exten => 1265,n,Goto(s,5)

;
; The page context calls up the page macro that sets variables needed for
auto-answer
; It is in its own context to make calling it from the Page() application as
simple as
; Local/{peername}@page
;
[page]
exten => _X.,1,Macro(page,SIP/${EXTEN})

;[mainmenu]
;
; Example "main menu" context with submenu
;
;exten => s,1,Answer
;exten => s,n,Background(thanks) ; "Thanks for calling press 1 for sales,
2 for support, ..."
;exten => s,n,WaitExten
;exten => 1,1,Goto(submenu,s,1)
;exten => 2,1,Hangup
;include => default
;
;[submenu]
;exten => s,1,Ringing ; Make them comfortable
with 2 seconds of ringback
;exten => s,n,Wait,2
;exten => s,n,Background(submenuopts) ; "Thanks for calling the sales
department. Press 1 for steve, 2 for..."
;exten => s,n,WaitExten
;exten => 1,1,Goto(default,steve,1)
;exten => 2,1,Goto(default,mark,2)

[default]
;
; By default we include the demo. In a production system, you
; probably don't want to have the demo there.
;
include => demo

;
; An extension like the one below can be used for FWD, Nikotel, sipgate etc.
; Note that you must have a [sipprovider] section in sip.conf
;
;exten => _41X.,1,Dial(SIP/${EXTEN:2}@sipprovider,,r)

; Real extensions would go here. Generally you want real extensions to be
; 4 or 5 digits long (although there is no such requirement) and start with a
; single digit that is fairly large (like 6 or 7) so that you have plenty of
; room to overlap extensions and menu options without conflict. You can alias
; them with names, too, and use global variables

;exten => 6245,hint,SIP/Grandstream1&SIP/Xlitel,Joe Schmoe ; Channel hints for
presence
;exten => 6245,1,Dial(SIP/Grandstream1,20,rt) ; permit transfer
;exten => 6245,n(dial),Dial(${HINT},20,rtT) ; Use hint as listed
;exten => 6245,n,Voicemail(6245,u) ; Voicemail (unavailable)
;exten => 6245,s+1,Hangup ; s+1, same as n
;exten => 6245,dial+101,Voicemail(6245,b) ; Voicemail (busy)

```

```

;exten => 6361,1,Dial(IAX2/JaneDoe,,rm) ; ring without time limit
;exten => 6389,1,Dial(MGCP/aaln/1@192.168.0.14)
;exten => 6390,1,Dial(JINGLE/caller/callee) ; Dial via jingle using labels
;exten => 6391,1,Dial(JINGLE/asterisk@digium.com/mogorman@astjab.org) ;Dial via
jingle using asterisk as the transport and calling mogorman.
;exten => 6394,1,Dial(Local/6275/n) ; this will dial ${MARK}

;exten => 6275,1,Macro(stdexten,6275,${MARK}) ; assuming ${MARK} is something
like Zap/2
;exten => mark,1,Goto(6275|1) ; alias mark to 6275
;exten => 6536,1,Macro(stdexten,6236,${WIL}) ; Ditto for wil
;exten => wil,1,Goto(6236|1)

;If you want to subscribe to the status of a parking space, this is
;how you do it. Subscribe to extension 6600 in sip, and you will see
;the status of the first parking lot with this extensions' help
;exten => 6600,hint,park:701@parkedcalls
;exten => 6600,1,noop
;
; Some other handy things are an extension for checking voicemail via
; voicemailmain
;
;exten => 8500,1,VoicemailMain
;exten => 8500,n,Hangup
;
; Or a conference room (you'll need to edit meetme.conf to enable this room)
;
;exten => 8600,1,Meetme(1234)
;
; Or playing an announcement to the called party, as soon it answers
;
;exten = 8700,1,Dial(${MARK},30,A(/path/to/my/announcemsg))
;
; For more information on applications, just type "show applications" at your
; friendly Asterisk CLI prompt.
;
; 'show application <command>' will show details of how you
; use that particular application in this file, the dial plan.
; 'show functions" will list all dialplan functions
; 'show function <COMMAND>' will show you more information about
; one function. Remember that function names are UPPER CASE.

```

3.6 SIPp

In order to send voice stream to test Asterisk B2BUA RTP feature, SIPp must be compiled by “*make pcapplay*”. A voice file, moh_alaw.pcap, is used in the test. It is in g711 alaw format, around 180 sec long.

3.6.1 Server End

3.6.1.1 With G711 Alaw

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->

```

```

<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp default 'uas' scenario. -->
<!-- -->

<scenario name="Basic UAS responder">
  <!-- By adding rrs="true" (Record Route Sets), the route sets -->
  <!-- are saved and used for following messages sent. Useful to test -->
  <!-- against stateful SIP proxies/B2BUAs. -->
  <recv request="INVITE" crlf="true">
  </recv>

  <!-- The '[last_*]' keyword is replaced automatically by the -->
  <!-- specified header if it was present in the last message received -->
  <!-- (except if it was a retransmission). If the header was not -->
  <!-- present or if no message has been received, the '[last_*]' -->
  <!-- keyword is discarded, and all bytes until the end of the line -->
  <!-- are also discarded. -->
  <!-- -->
  <!-- If the specified header was present several times in the -->
  <!-- message, all occurrences are concatenated (CRLF seperated) -->
  <!-- to be used in place of the '[last_*]' keyword. -->

  <send>
    <![CDATA[

      SIP/2.0 180 Ringing
      [last_Via:]
      [last_From:]
      [last_To:];tag=[pid]SIPpTag01[call_number]
      [last_Call-ID:]
      [last_CSeq:]
      Contact: <sip:[local_ip]:[local_port];transport=[transport]>
      Content-Length: 0

    ]]>
  </send>

  <send retrans="500">
    <![CDATA[

      SIP/2.0 200 OK
      [last_Via:]
      [last_From:]
      [last_To:];tag=[pid]SIPpTag01[call_number]
      [last_Call-ID:]
      [last_CSeq:]
      Contact: <sip:[local_ip]:[local_port];transport=[transport]>
      Content-Type: application/sdp
      Content-Length: [len]

      v=0
      o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[media_ip_type] [media_ip]
      t=0 0
      m=audio [media_port] RTP/AVP 8
      a=rtpmap:8 PCMA/8000

    ]]>
  </send>

```

```

    ]]>
</send>

<recv request="ACK"
      optional="true"
      rtd="true"
      crlf="true">
</recv>

<recv request="BYE">
</recv>

<send>
  <![CDATA[

    SIP/2.0 200 OK
    [last_Via:]
    [last_From:]
    [last_To:]
    [last_Call-ID:]
    [last_CSeq:]
    Contact: <sip:[local_ip]:[local_port];transport=[transport]>
    Content-Length: 0

  ]]>
</send>

<!-- Keep the call open for a while in case the 200 is lost to be      -->
<!-- able to retransmit it if we receive the BYE again.              -->
<pause milliseconds="4000"/>

<!-- definition of the response time repartition table (unit is ms)  -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

<!-- definition of the call length repartition table (unit is ms)   -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>

3.6.1.2 With G729

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or    -->
<!-- modify it under the terms of the GNU General Public License as   -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version.                  -->
<!--                                                                    -->
<!-- This program is distributed in the hope that it will be useful,  -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of   -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the   -->
<!-- GNU General Public License for more details.                      -->
<!--                                                                    -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the                    -->
<!-- Free Software Foundation, Inc.,                                    -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA           -->
<!--                                                                    -->
<!--          Sipp default 'uas' scenario.                             -->
<!--                                                                    -->

<scenario name="Basic UAS responder">

```

```

<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->
<recv request="INVITE" crlf="true">
</recv>

<!-- The '[last_*]' keyword is replaced automatically by the -->
<!-- specified header if it was present in the last message received -->
<!-- (except if it was a retransmission). If the header was not -->
<!-- present or if no message has been received, the '[last_*]' -->
<!-- keyword is discarded, and all bytes until the end of the line -->
<!-- are also discarded. -->
<!-- -->
<!-- If the specified header was present several times in the -->
<!-- message, all occurrences are concatenated (CRLF seperated) -->
<!-- to be used in place of the '[last_*]' keyword. -->

<send>
  <![CDATA[

    SIP/2.0 180 Ringing
    [last_Via:]
    [last_From:]
    [last_To:];tag=[pid]SIPpTag01[call_number]
    [last_Call-ID:]
    [last_CSeq:]
    Contact: <sip:[local_ip]:[local_port];transport=[transport]>
    Content-Length: 0

  ]]>
</send>

<send retrans="500">
  <![CDATA[

    SIP/2.0 200 OK
    [last_Via:]
    [last_From:]
    [last_To:];tag=[pid]SIPpTag01[call_number]
    [last_Call-ID:]
    [last_CSeq:]
    Contact: <sip:[local_ip]:[local_port];transport=[transport]>
    Content-Type: application/sdp
    Content-Length: [len]

    v=0
    o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
    s=-
    c=IN IP[media_ip_type] [media_ip]
    t=0 0
    m=audio [media_port] RTP/AVP 18
    a=rtpmap:18 G729/8000

  ]]>
</send>

<recv request="ACK"
  optional="true"
  rtd="true"
  crlf="true">
</recv>

<recv request="BYE">
</recv>

```

```

<send>
  <![CDATA[

    SIP/2.0 200 OK
    [last_Via:]
    [last_From:]
    [last_To:]
    [last_Call-ID:]
    [last_CSeq:]
    Contact: <sip:[local_ip]:[local_port];transport=[transport]>
    Content-Length: 0

  ]]>
</send>

<!-- Keep the call open for a while in case the 200 is lost to be -->
<!-- able to retransmit it if we receive the BYE again. -->
<pause milliseconds="4000"/>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>

```

3.6.2 Client End

3.6.2.1 With G711 Alaw

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtsp) play -->
<!-- -->

<scenario name="UAC with media">
  <!-- In client mode (sipp placing calls), the Call-ID MUST be -->
  <!-- generated by sipp. To do so, use [call_id] keyword. -->
  <send retrans="500" start_rtd="1">
    <![CDATA[

      INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp
    <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]

```

```

To: sut <sip:[service]@[remote_ip]:[remote_port]>
Call-ID: [call_id]
CSeq: 1 INVITE
Contact: sip:sipp@[local_ip]:[local_port]
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: [len]

v=0
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
s=-
c=IN IP[local_ip_type] [local_ip]
t=0 0
m=audio [auto_media_port] RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11,16

]]>
</send>

<recv response="100" optional="true" rtd="1" start_rtd="2">
</recv>

<recv response="180" optional="true" rtd="2">
</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->
<recv response="200" crlf="true">
</recv>

<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
<send>
  <![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp
<sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 1 ACK
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
  <action>
    <exec play_pcap_audio="moh_alaw.pcap"/>
  </action>
</nop>

<!-- Pause 8 seconds, which is approximately the duration of the -->
<!-- PCAP file -->
<pause milliseconds="180000"/>

```

```

<!-- Play an out of band DTMF '1' -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>
  </action>
</nop>

<pause milliseconds="1000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">
  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp
<sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 BYE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="50, 100, 200, 500, 1100, 2100, 3100, 4100,
5100, 6100, 10000"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>

```

3.6.2.2 With G729

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!-- This program is free software; you can redistribute it and/or -->
<!-- modify it under the terms of the GNU General Public License as -->
<!-- published by the Free Software Foundation; either version 2 of the -->
<!-- License, or (at your option) any later version. -->
<!-- -->
<!-- This program is distributed in the hope that it will be useful, -->
<!-- but WITHOUT ANY WARRANTY; without even the implied warranty of -->
<!-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the -->
<!-- GNU General Public License for more details. -->
<!-- -->
<!-- You should have received a copy of the GNU General Public License -->
<!-- along with this program; if not, write to the -->
<!-- Free Software Foundation, Inc., -->
<!-- 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA -->
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->

```

```

<scenario name="UAC with media">
  <!-- In client mode (sipp placing calls), the Call-ID MUST be -->
  <!-- generated by sipp. To do so, use [call_id] keyword. -->
  <send retrans="500" start_rtd="1">
    <![CDATA[

      INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp
<sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]
      To: sut <sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      CSeq: 1 INVITE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Type: application/sdp
      Content-Length: [len]

      v=0
      o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[local_ip_type] [local_ip]
      t=0 0
      m=audio [auto_media_port] RTP/AVP 18 101
      a=rtpmap:18 G729/8000
      a=rtpmap:101 telephone-event/8000
      a=fmtp:101 0-11,16

    ]]>
  </send>

  <recv response="100" optional="true" rtd="1" start_rtd="2">
  </recv>

  <recv response="180" optional="true" rtd="2">
  </recv>

  <!-- By adding rrs="true" (Record Route Sets), the route sets -->
  <!-- are saved and used for following messages sent. Useful to test -->
  <!-- against stateful SIP proxies/B2BUAs. -->
  <recv response="200" crlf="true">
  </recv>

  <!-- Packet lost can be simulated in any send/recv message by -->
  <!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
  <send>
    <![CDATA[

      ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp
<sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]
      To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
      Call-ID: [call_id]
      CSeq: 1 ACK
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Length: 0

    ]]>
  </send>

```

```

<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
  <action>
    <exec play_pcap_audio="moh_g729.pcap"/>
  </action>
</nop>

<!-- Pause 8 seconds, which is approximately the duration of the -->
<!-- PCAP file -->
<pause milliseconds="180000"/>

<!-- Play an out of band DTMF '1' -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>
  </action>
</nop>

<pause milliseconds="1000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">
  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp
<sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag00[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 BYE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="50, 100, 200, 500, 1100, 2100, 3100, 4100,
5100, 6100, 10000"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>

```

4 SIPp Client Parameters

4.1 Transport Mode

SIPp support multiple transport modes. It can be set by command line, “*-t mode*”, from the client end. The default setting is UDP one socket. To simulate product environments, multiple sockets may be used.

4.2 Call Limit

This parameter is used to control the maximum number of simultaneous calls. It can be set by command line, “*-l limit*”, from the client end. It should be set to a large number, such as 5000, for heavy load.

4.3 Timer Resolution

This parameter can be set by command line, “*-timer_resol ms*”, from the client end. The default timer resolution is 200ms. We want a more precise scheduling, this value is set to 50ms.

4.4 Frequency

The call rate can be set by command line, “*-r rate*” and “*-rp period*”, from the client end. The real call rate is *rate/period* cps. This parameter should be set according to two facts. First, the total SIP traffic load should reach a certain level. For example, CPU usage of Asterisk B2BUA should reach a certain level. Another fact is the capability of the OSP servers. If the OSP servers cannot handle the heavy traffic load, more OSP servers should be used.

4.5 Call Numbers

This parameter is used to stop the test when the total number of the calls reaches this parameter. It can be set by command line, “*-m number*”, from the client end.

4.6 Screen Flush Frequency

This parameter is used to set the flush frequency of displaying test results on the screen. It can be set by command line, “*-f ms*”, from the client end. In order to reduce the overhead, this value should be set to a large number, such as 30000.

5 Scripts for Running the Test

5.1 Data Collection Scripts

Sar is used to collect CPU usage and memory usage data. Sar write script is used to start sar to write test info into a data file. Sar read script is used after test to collect the test info from the data file.

5.1.1 Sar Write Script

```
# Usage: wsar.sh count
# Count is based on 10s internal.
rm -rf ./data.sar
sar -o ./data.sar 10 $1 >/dev/null 2>&1 &
```

The count should be set to a value longer than test duration. For example, the test duration is around 15 minutes, since sar normally is started minutes earlier than the test, the count should be at least 120.

Note: The old data.sar must be erased. Otherwise, sar read script may collect wrong data.

5.1.2 Sar Read Script

```
# Usage: rsar.sh start end
```

```
# Time in hh:mm:ss format
sar -u -r -n DEV -f ./data.sar -s $1 -e $2
```

Since sar records the test data based on local time, to display the test data, the local time must be remembered.

Note: sar on t2000 does not support “-n” options.

5.2 SIPp Scripts

Note:

- SIPp server end should be started before SIPp client is started.
- SIPp client end need root rights to start for pcap feature.

For SIPp client, we need to collect data on the following time intervals:

- Time from INVITE sent until a Trying message received.
- Time from the Trying received to a RINGING message received.

5.2.1 Server End

```
sipp -sf server.xml -nd -i 172.16.4.60
```

Note: The SIPp server end can be run at background using “-bg” command line parameter.

5.2.2 Client End

```
sipp -sf client.xml -m 300 -r 1 -rp 2 -l 5000 -s 9876543210 -nd -timer_resol 50 -pause_msg_ign -f 30000 -fd 30 172.16.4.32 -trace_stat
```

Note: “-trace_screen” can be used to collect the test results.

6 Test Procedure

6.1 Start Test

- Restart SIPp server on fedora3
- Start sar write script on fedora3
- Stop OSP Server instance on t2000
- Erase all old CDRs
- Start OSP Server instance
- Start sar write script on t2000
- Restart Asterisk B2BUA on centos4
- Start sar write script on centos4
- Start SIPp client on fedora1/5/8 and NexTransit06 using root account at 0/15/30/45 minute.

6.2 After Test

- Collect statistics data from sressn or client_<pid>_screen.log on fedora1 if “-trace_screen” is set.
- Collect CDRs from all OSP Server instances and calculate the numbers of different CDRs.
- Run CDR pull on NexOSS to generate traffic report.

- Collect test results using sar read script on all test boxes.

7 Test Results

The test results will be displayed on SIPp client end screens. The total number of calls, successful calls, and failed calls will be displayed.

The number of current calls can be obtained from SIPp client statistics files.

The best condition is all calls complete without any unexpected message and without any message loses.

The Ringing message losing with call completing is acceptable when the traffic is heavy.

A low percentage call failing may be acceptable for very heavy traffic conditions.

Note:

1. The numbers of different CDRs can be calculated using the command “*grep -P “\tTCCode\t” *.cdr | wc -l*”
2. The NexOSS report may contain the calls for other tests.
3. OSP Server configuration: 8 instances on T2000
4. There are several data about the Asterisk B2BUA performance:
 - Number of attempts
 - SIPp report
 - OSP Server CDRs
 - NexOSS report

SIPp reported number of completed calls is most important. It means from the source side how many calls completed.

NexOSS report came from OSP Server CDRs. They should be totally same. There may be duplicated CDRs, only 16 CDRs, caused by the re-sending BYE messages out of Asterisk B2BUA absorb timeout.

The relationship of the numbers should be:

- Number of internal CDRs == 5 * number of attempts
- Number of SIPp completed calls <= number of 16 CDRs

There is another reason that number of 16 CDRs may be larger than number of SIPp completed calls except the duplication, the 200 OK for the BYE messages may reach Asterisk B2BUA but did not reach the SIPp client.

7.1 Actual Results

The following pages present raw data collected from test:

7.1.1 Test Results without Codec Translation

Date Test Performed	8/31/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp		
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2		
OpenSER Version	1.4.11	Host Memory	4 GB		
Test Rate (cps)		1	2	3	4
SIPp Traffic Report	Completed	600	1200	1800	2400
	Completed%	100%	100%	100%	100%
	Simultaneous Expected	180	360	540	720
	Simultaneous Actual	184.79	368.71	555.71	740.50
Analysis of OSP Server CDRs					
Internal CDRs	Expected	3000	6000	9000	12000
	Actual	3000	6000	9000	12000
Code 16 CDRs	Expected	600	1200	1800	2400
	Actual	600	1200	1800	2400
SIP 1 CDRs	Expected	~300	~600	~900	~1200
	Actual	181	409	1168	1066
SIP 18 CDRs	Expected	~900	~1800	~2700	~3600
	Actual	880	1489	3491	3810
All Other Error CDRs	Expected	0	0	0	0
	Actual	0	0	0	0
NexOSS Traffic Report					
16 Normal Call Clearing	Expected	600	1200	1800	2400
	Actual	600	1200	1800	2400
SIP 1	Expected	~300	~600	~900	~1200
	Actual	181	409	1168	1066
SIP 18	Expected	~900	~1800	~2700	~3600
	Actual	880	1489	3491	3810
All Other	Expected	0	0	0	0
	Actual	0	0	0	0
System Utilization					
Asterisk Server	CPU idle	95.50%	89.29%	82.03%	71.40%
	Memory	0.70%	1.38%	2.06%	2.65%
	RX/TX (MB/s)	1.97/1.97	3.97/3.97	5.94/5.95	8.00/8.01
SIPp Average Response Time Repartition					
Invite to Trying	< 50 ms	600	1200	1800	2400
	< 100 ms	0	0	0	0
	< 200 ms	0	0	0	0
	< 500 ms	0	0	0	0
	> 500 ms	0	0	0	0
Trying to Ringing	< 0.1 s	75	352	268	613
	< 1.1 s	0	0	0	0
	< 2.1 s	264	323	374	622
	< 3.1 s	0	0	0	0
	< 4.1 s	167	409	357	307
	< 5.1 s	0	0	0	0
	< 6.1 s	94	116	801	858
	> 10 s	0	0	0	0

Date Test Performed	8/31/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp		
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2		
OpenSER Version	1.4.11	Host Memory	4 GB		
Test Rate (cps)		5	6	7	8
SIPp Traffic Report	Completed	3000	3600	4200	4800
	Completed%	100%	100%	100%	100%
	Simultaneous Expected	900	1080	1200	1440
	Simultaneous Actual	924.22	1112.71	1293.35	1474.21
Analysis of OSP Server CDRs					
Internal CDRs	Expected	15000	18000	21000	24000
	Actual	15000	18000	21000	24000
Code 16 CDRs	Expected	3000	3600	4200	4800
	Actual	3000	3600	4200	4800
SIP 1 CDRs	Expected	~1500	~1800	~2100	~2400
	Actual	2059	1992	2141	2028
SIP 18 CDRs	Expected	~4500	~5400	~6300	~7200
	Actual	5719	6794	6915	6792
All Other Error CDRs	Expected	0	0	0	0
	Actual	0	0	0	0
NexOSS Traffic Report					
16 Normal Call Clearing	Expected	3000	3600	4200	4800
	Actual	3000	3600	4200	4800
SIP 1	Expected	~1500	~1800	~2100	~2400
	Actual	2059	1992	2141	2028
SIP 18	Expected	~4500	~5400	~6300	~7200
	Actual	5719	6794	6915	6792
All Other	Expected	0	0	0	0
	Actual	0	0	0	0
System Utilization					
Asterisk Server	CPU idle	60.09%	44.96%	30.62%	14.02%
	Memory	3.24%	4.01%	4.59%	5.31%
	RX/TX (MB/s)	9.88/9.90	11.81/11.83	13.55/13.56	15.47/15.49
SIPp Average Response Time Repartition					
Invite to Trying	< 50 ms	3000	3599	3752	4131
	< 100 ms	0	1	211	213
	< 200 ms	0	0	138	145
	< 500 ms	0	0	74	176
	> 500 ms	0	0	25	135
Trying to Ringing	< 0.1 s	372	647	802	1144
	< 1.1 s	0	0	48	80
	< 2.1 s	603	503	689	1166
	< 3.1 s	0	0	55	147
	< 4.1 s	959	1059	1472	1011
	< 5.1 s	0	0	164	282
	< 6.1 s	1066	1391	871	687
	< 10 s	0	0	99	283
	> 10 s	0	0	0	0

Date Test Performed	9/6/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp		
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2		
OpenSER Version	1.4.11	Host Memory	4 GB		
Test Rate (cps)		8.5			
SIPp Traffic Report	Completed	5100			
	Completed%	100%			
	Simultaneous Expected	1530			
	Simultaneous Actual	1563.42			
Analysis of OSP Server CDRs					
Internal CDRs	Expected	25500			
	Actual	25500			
Code 16 CDRs	Expected	5100			
	Actual	5100			
SIP 1 CDRs	Expected	~2550			
	Actual	2062			
SIP 18 CDRs	Expected	~7650			
	Actual	6667			
All Other Error CDRs	Expected	0			
	Actual	0			
NexOSS Traffic Report					
16 Normal Call Clearing	Expected	5100			
	Actual	5100			
SIP 1	Expected	~2550			
	Actual	2062			
SIP 18	Expected	~7650			
	Actual	6667			
All Other	Expected	0			
	Actual	0			
System Utilization					
Asterisk Server	CPU idle	8.17%			
	Memory	5.48%			
	RX/TX (MB/s)	16.36/16.38			
SIPp Average Response Time Repartition					
Invite to Trying	< 50 ms	4594			
	< 100 ms	253			
	< 200 ms	155			
	< 500 ms	96			
	> 500 ms	2			
Trying to Ringing	< 0.1 s	1308			
	< 1.1 s	69			
	< 2.1 s	1406			
	< 3.1 s	241			
	< 4.1 s	955			
	< 5.1 s	253			
	< 6.1 s	562			
	< 10 s	306			
> 10 s	0				

7.1.2 Test Results with G711 Alaw to Digium G729

Date Test Performed	9/11/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp		
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2		
OpenSER Version	1.4.11	Host Memory	4 GB		
Test Rate (cps)		0.5	1	1.5	2
SIPp Traffic Report	Completed	300	600	900	1200
	Completed%	100%	100%	100%	100%
	Simultaneous Expected	90	180	270	360
	Simultaneous Actual	91.71	183.50	276.64	369.36
Analysis of OSP Server CDRs					
Internal CDRs	Expected	1500	3000	4500	6000
	Actual	1500	3000	4500	6000
Code 16 CDRs	Expected	300	600	900	1200
	Actual	300	600	900	1200
SIP 1 CDRs	Expected	~150	~300	~450	~600
	Actual	110	136	477	710
SIP 18 CDRs	Expected	~450	~900	~1350	~1800
	Actual	277	533	1368	1797
All Other Error CDRs	Expected	0	0	0	0
	Actual	0	0	0	0
NexOSS Traffic Report					
16 Normal Call Clearing	Expected	300	600	900	1200
	Actual	300	600	900	1200
SIP 1	Expected	~150	~300	~450	~600
	Actual	110	136	477	710
SIP 18	Expected	~450	~900	~1350	~1800
	Actual	277	533	1368	1797
All Other	Expected	0	0	0	0
	Actual	0	0	0	0
System Utilization					
Asterisk Server	CPU idle	79.17%	59.35%	37.99%	18.37%
	Memory	0.78%	1.17%	2.01%	2.47%
	RX/TX (MB/s)	0.98/0.36	1.94/0.71	2.97/1.09	3.85/1.41
SIPp Average Response Time Repartition					
Invite to Trying	< 50 ms	300	600	900	1200
	< 100 ms	0	0	0	0
	< 200 ms	0	0	0	0
	< 500 ms	0	0	0	0
	> 500 ms	0	0	0	0
Trying to Ringing	< 0.1 s	145	250	231	346
	< 1.1 s	0	0	0	0
	< 2.1 s	61	256	215	222
	< 3.1 s	0	0	0	0
	< 4.1 s	66	5	209	321
	< 5.1 s	0	0	0	0
	< 6.1 s	28	89	245	311
	> 10 s	0	0	0	0

Date Test Performed	9/11/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp			
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2			
OpenSER Version	1.4.11	Host Memory	4 GB			
Test Rate (cps)		2.25				
SIPp Traffic Report	Completed	1350				
	Completed%	100%				
	Simultaneous Expected	405				
	Simultaneous Actual	414.57				
Analysis of OSP Server CDRs						
Internal CDRs	Expected	6750				
	Actual	6750				
Code 16 CDRs	Expected	1350				
	Actual	1350				
SIP 1 CDRs	Expected	~675				
	Actual	553				
SIP 18 CDRs	Expected	~2025				
	Actual	1915				
All Other Error CDRs	Expected	0				
	Actual	0				
NexOSS Traffic Report						
16 Normal Call Clearing	Expected	1350				
	Actual	1350				
SIP 1	Expected	~675				
	Actual	553				
SIP 18	Expected	~2025				
	Actual	1915				
All Other	Expected	0				
	Actual	0				
System Utilization						
Asterisk Server	CPU idle	6.67%				
	Memory	3.12%				
	RX/TX (MB/s)	4.34/1.59				
SIPp Average Response Time Repartition						
Invite to Trying	< 50 ms	1350				
	< 100 ms	0				
	< 200 ms	0				
	< 500 ms	0				
	> 500 ms	0				
Trying to Ringing	< 0.1 s	408				
	< 1.1 s	0				
	< 2.1 s	299				
	< 3.1 s	0				
	< 4.1 s	313				
	< 5.1 s	0				
	< 6.1 s	330				
	< 10 s	0				
> 10 s	0					

7.1.3 Test Results with Digium G729 to G711 Alaw

Date Test Performed	9/11/07	Host OS	CentOS 4.4 / 2.6.9-42.ELsmp		
Test Performed by	Di-Shi	Host CPU	Intel Xeon 5140 @ 2.33 GHz * 2		
OpenSER Version	1.4.11	Host Memory	4 GB		
Test Rate (cps)		2			
SIPp Traffic Report	Completed	1200			
	Completed%	100%			
	Simultaneous Expected	360			
	Simultaneous Actual	368.86			
Analysis of OSP Server CDRs					
Internal CDRs	Expected	6000			
	Actual	6000			
Code 16 CDRs	Expected	1200			
	Actual	1200			
SIP 1 CDRs	Expected	~600			
	Actual	499			
SIP 18 CDRs	Expected	~1800			
	Actual	1611			
All Other Error CDRs	Expected	0			
	Actual	0			
NexOSS Traffic Report					
16 Normal Call Clearing	Expected	1200			
	Actual	1200			
SIP 1	Expected	~600			
	Actual	499			
SIP 18	Expected	~1800			
	Actual	1611			
All Other	Expected	0			
	Actual	0			
System Utilization					
Asterisk Server	CPU idel	63.28%			
	Memory	2.27			
	RX/TX (MB/s)	1.42/3.95			
SIPp Average Response Time Repartition					
Invite to Trying	< 50 ms	1200			
	< 100 ms	0			
	< 200 ms	0			
	< 500 ms	0			
	> 500 ms	0			
Trying to Ringing	< 0.1 s	333			
	< 1.1 s	0			
	< 2.1 s	278			
	< 3.1 s	0			
	< 4.1 s	434			
	< 5.1 s	0			
	< 6.1 s	155			
	> 10 s	0			