Avaya Aura™ SBC System Operations and Troubleshooting Guide

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of this documentation unless such modifications, additions, or deletions were performed by Avaya.

End User agree to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Websites referenced within this site or documentation(s) provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on this product. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product, while under warranty, is available to Avaya customers and other parties through the Avaya Support Website: http://www.avaya.com/support. Please note that if you acquired the product from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by the said Avaya reseller and not by Avaya.

Licenses

The software license terms available on the Avaya Website, http://support.avaya.com/licenseinfo/ are applicable to anyone who downloads, uses and/or installs Avaya software, purchased from Avaya Inc., any Avaya affiliate, or an authorized Avaya reseller (as applicable) under a commercial agreement with Avaya or an authorized Avaya reseller. Unless otherwise agreed to by Avaya in writing, Avaya does not extend this license if the software was obtained from anyone other than Avaya, an Avaya affiliate or an Avaya authorized reseller, and Avaya reserves the right to take legal action against you and anyone else using or selling the software without a license. By installing, downloading or using the software, or authorizing others to do so, you, on behalf of yourself and the entity for whom you are installing, downloading or using the software (hereinafter referred to interchangeably as "you" and "end user"), agree to these terms and conditions and create a binding contract between you and Avaya Inc. Or the applicable Avaya affiliate ("Avaya").

Avaya grants End User a license within the scope of the license types described below. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the Documentation or other materials available to End User. "Designated Processor" means a single stand-alone computing device. "Server" means a Designated Processor that hosts a software application to be accessed by multiple users. "Software" means the computer programs in object code, originally licensed by Avaya and ultimately utilized by End User, whether as stand-alone products or pre-installed on Hardware. "Hardware" means the standard hardware originally sold by Avaya and ultimately utilized by End User.

License types

- Designated System(s) License (DS):
  End User may install and use each copy of the Software on only one Designated Processor, unless a different number of Designated Processors is indicated in the Documentation or other materials available to End User. Avaya may require the Designated Processor(s) to be identified by type, serial number, feature key, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.
- Concurrent User License (CU):
  End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A "Unit" means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server.
- Named User License (NU):
  End User may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use the Software on a Server so long as only authorized Named Users access and use the Software. "Named User" means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a "Named User" may be, without limitation, designated by name, corporate function (for example, webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.
- Shrinkwrap License (SR):

Customer may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as “shrinkwrap” or “clickthrough” license accompanying or applicable to the Software (“Shrinkwrap License”). (See Third-party Components for more information).

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation(s) and Product(s) provided by Avaya. All content on this site, the documentation(s) and the product(s) provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil, offense under the applicable law.

Third Party Components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Website: http://support.avaya.com/Copyright.

Preventing toll fraud

“Toll fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya fraud intervention

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support Website:

http://www.support.avaya.com/.

Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to:securityalerts@avaya.com.

Trademarks

Avaya® and Avaya Aura™ are trademarks of Avaya Inc.

The trademarks, logos and service marks (“Marks”) displayed in this site, the documentation(s) and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the documentation(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

All non-Avaya trademarks are the property of their respective owners.

Downloading documents

For the most current versions of documentation, see the Avaya Support Website: http://www.avaya.com/support.

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Website: http://www.avaya.com/support.

Contents

Preface

About Avaya Aura™ SBC documentation

This manual is part of a documentation set for the Avaya Aura™ Session Border Controller (AA-SBC), powered by Acme Packet. The following is a complete list of documents supporting the Avaya Aura™ SBC software.

- *Avaya Aura™ SBC – Using the AA-SBC Management Tools*
- *Avaya Aura™ SBC – System Administration Guide*
- *Avaya Aura™ SBC – Session Services Configuration Guide*
- *Avaya Aura™ SBC – Objects and Properties Reference*
- *Avaya Aura™ SBC – System Operations and Troubleshooting Guide*

About this manual

The *Avaya Aura™ SBC System Operations and Troubleshooting Guide* is a quick reference guide for maintaining and troubleshooting the AA-SBC systems using common and most-frequently used practices. It covers the following subjects:

- Basic instructions for accessing systems running AA-SBC software.
- Tools and procedures for system monitoring.
- System maintenance procedures.

- Troubleshooting, including device registration, call completion and media problems, as well as software faults.

Note: The procedures and practices covered in this guide represent a fraction of the capabilities offered by the AA-SBC software. Troubleshooting more complex or unique problems may require not only the examination of call logs and events generated by AA-SBC, but also that of any third-party devices participating in your SIP network. The procedures and practices covered in this guide simply provide a common starting point for general troubleshooting and analysis that will lead you to the best solution.

Conventions used in this manual

Typographical conventions

Key Convention	Function	Example
KEY NAME	Identifies the name of a key to press.	Type **abc**, then press [ENTER]
CTRL+*x*	Indicates a control key combination.	Press CTRL+C
brackets [ ]	Indicates an optional argument.	`[portNumber]`
braces { }	Indicates a required argument with a choice of values; choose one.	`{enabled | disabled}`
vertical bar |	Separates parameter values. Same as "or."	`{TCP | TLS}`
Monospaced bold	In screen displays, indicates user input.	`config> config vsp`
Monospaced italic	In screen displays, indicates a variable—generic text for which you supply a value.	`config servers> config lcs name`
bold	In text, indicates literal names of commands, actions, objects, or properties.	...set as the secondary directory service (with the **unifier** property)...
bold italic	In text, indicates a variable.	...set the **domain** property of the ***directory*** object.

Acronyms

AA-SBC manuals contain the following industry-standard and product-specific acronyms:

AAA	Authentication, authorization, and accounting
AA-SBC	Avaya Aura™ Session Border Controller
ALI	Automatic location identifier
ANI	Automatic number identification
ANSI	American National Standards Institute
AOR	Address of record
API	Application programming interface

ARP	Address Resolution Protocol
AVERT	Anti-virus emergency response team
B2BUA	Back-to-back user agen
BOOTP	Bootstrap Protocol
CA	Certificate authority
CAP	Client application protocol
CBC	Cipher block chaining
CBN	Call back number
CCS	Converged Communication Server
CDR	Call detail record
CIDR	Classless interdomain routing
CLI	Command line interface
CMOS	Comparison mean opinion score
CNAME	Canonical name record
CNI	Calling number identification
CODEC	Compressor/decompressor or coder/decoder
CPE	Customer-premise equipment
CRL	Certificate revocation list
CSR	Certificate signing request
CSTA	Computer-supported telecommunications applications
CSV	Comma-separated values
DDDS	Dynamic delegation discovery system
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized zone
DN	Distinguished name
DNIS	Dialed number identification service
DNS	Domain name service
DOS	Denial of service
EIM	Enterprise instant messaging
ESD	Electrostatic discharge
ESGW	Emergency services gateway
ESQK	Emergency services query key

ESRN	Emergency services routing number
FQDN	Fully qualified domain name
GUI	Graphical user interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I2	National Emergency Number Association defined VoIP solution
ICAP	Internet Calendar Access Protocol
ICMP	Internet Control Message Protocol
IM	Instant messaging
IP	Internet Protocol
JDBC	Java database connectivity
JMX	Java management extensions
JRE	Java runtime environment
LATA	Local access and transport area
LCS	Live Communications Server
LCR	Least-cost routing
LDAP	Lightweight Directory Access Protocol
LIS	Location information service
MAC	Media access control
MCS	Multimedia Communications Server
MIB	Management information base
MOS	Mean opinion score
MSAG	Master street address guide
MTU	Maximum transmission unit
NAPTR	Naming authority pointer
NAT	Network address translation
NENA	National Emergency Number Association
NIC	Network interface card
NS	Name server
NSE	Named signaling events
NTLM	NT Lan Manager
NTP	Network Time Protocol

OC	Office Communicator
OCI	Open Client Interface
ODBC	Open database connectivity
OTP	Over temperature protection
OVP	Over voltage protection
PBX	Private branch eXchange
PEM	Privacy-enhanced mail
PERL	Practical Extraction and Reporting Language
PING	Packet internet groper
PKCS#12	Public Key Cryptography Standard #12
PKI	Public Key Infrastructure
PSAP	Public safety answering point
PSCP	PuTTY secure copy
PSTN	Public switched telephone network
QOP	Quality of protection
QOS	Quality of service
RADIUS	Remote Authentication Dial-in User Service
RTC	Real-time collaboration
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
RTT	Round-trip time
SATA	Serial ATA
SCSI	Small computer system interface
SDK	Software development kit
SDP	Session Description Protocol
SFTP	Secure Shell File Transfer Protocol
SIMPLE	SIP Instant Messaging and Presence Leveraging Extension
SIP	Session Initiation Protocol
SIPS	Session Initiation Protocol over TLS
SLB	Server load balancing
SMB	Server message block
SNMP	Simple Network Management Protocol

SOA	Server of authority
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRTP	Secure Real-time Transport Protocol
SRV	Server resource
SSH	Secure Shell
SSL	Secure socket layer
SSRC	Synchronization source
STUN	Simple Traversal of UDP over NATs
TCP	Transmission Control Protocol
TDM	Time division multiplexing
TGRP	Trunk group
TLS	Transport Layer Security
TOS	Type of service
TTL	Time to live
UPS	Uninterruptable power supply
US	User agent
UAC	User agent client
UAS	User agent server
UDP	User Datagram Protocol
UID	Unique identifier
URI	Uniform resource identifier
URL	Uniform resource locator
UTC	Universal coordinated time
VoIP	Voice over IP
VLAN	Virtual local area network
VPC	VoIP positioning center
VRRP	Virtual Router Redundancy Protocol
VSP	Virtual system partition
VXID	Virtual router interface ID
WAR	Web application resource
WAV	Waveform audio

WM	Windows Messenger
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

1. AA-SBC operations overview

About this chapter

This chapter provides an overview of the basic instructions for accessing systems and blades running AA-SBC software..

Note: The procedures and practices covered in this guide represent a fraction of the capabilities offered by the e software. Troubleshooting more complex or unique problems may require not only the examination of call logs and events generated by e, but also that of any third-party devices that are participating in your SIP network. The procedures and practices covered in this guide simply provide a common starting point for general troubleshooting and analysis that will lead you to the best solution.

Management operations

AA-SBC provides a command line (CLI) and the NetWeb user interface that allow administrators operations personnel to configure and monitor AA-SBC, as well as invoke various actions upon it. With only a few exceptions, each command available from the CLI also has a AA-SBC Management System equivalent, and vice-versa.

Commands are available to show the current status of AA-SBC and every point in the configuration hierarchy. Status includes:

- Historical status — Provided through various log files
- User session information — Available from the call log and from accounting files
- Tracing facilities— Available to resolve specific types of call-related problems.

Status information is also exported via syslog, SNMP traps and SNMP polling.

Accessing the system

While each systemor blade (also referred to as a 'box' or 'appliance') in a AA-SBC cluster has its own management IP address, the cluster also has a shared management IP address available over a VRRP interface. This shared cluster management address should be used in all operations that involve changing the cluster configuration or accessing the shared system database. It is particularly important to make configuration changes using the cluster management address, otherwise changes will not be propagated across the cluster and may be lost when a system reboots.

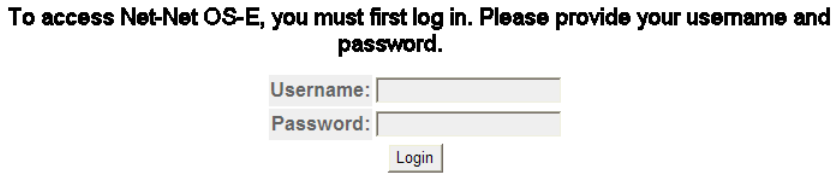
Most normal operations will make use of the cluster management address. Only use the individual box management IP address when there is a need to execute an action specific to that box. Note that the AA-SBC Management System allows most status displays and actions to be applied to a selected system from a drop-down menu.

For CLI access, connect to the management address with an SSH client:

```
login as: root
root@172.30.3.165's password:

username: yogibear
password: ****************

NNOS-E>
```

Notice that there is a Linux-level prompt for a username (normally *root*) and password, followed by an application-level username and password. If the system and SSH clients have been set up to use security certificates, the *root* username and password are not required.

For the AA-SBC Management System, use HTTPS to connect to the management address using a Web browser, such as Internet Explorer.



Notice that Web access only requires the application-level username and password.

In addition to the normal management IP address, the AA-SBC CLI is also available over the console port.

For example:

```
C:\IPMIutils> isolconsole -a -N 10.10.1.165 -U yogibear -P password
isolconsole ver 2.4
Opening connection to node 10.10.1.165 ...
Connected to node  10.10.1.165
pong timeout, after bind complete
-- BMC version 0.52, IPMI version 2.0

[SOL session is running, use '~' to end session.]

NNOS-E>
NNOS-E>~
isolconsole exit via user input
isolconsole: completed successfully
```

Note that physical access to the console port does not require a username or password, while serial-over-LAN does require a username and password.

Using the command line interface (CLI)

The CLI provides three main command categories:

- **Configuration commands** — See the *Net-Net OS-E – System Administration Guide* and the *Net-Net OS-E – Objects and Properties Reference* for details.
- **Status provider (show) command**s — See Chapter 4 of the *Net-Net OS-E – Objects and Properties Reference*.
- **Actions** — See Chapter 3 of the *Net-Net OS-E – Objects and Properties Reference*.

Access to these different categories of command is governed by a user permissions in the system configuration, using the **access->permissions** configuration path. For example, an advanced user is able to change the configuration and carry out actions such as rebooting the system, updating the software, and other administrative actions, Normal users have more restrictive permissions that do not have potentially disruptive impact, such as read-only access to the configuration and access to status provider, but no acces to the actions commands.

For example:

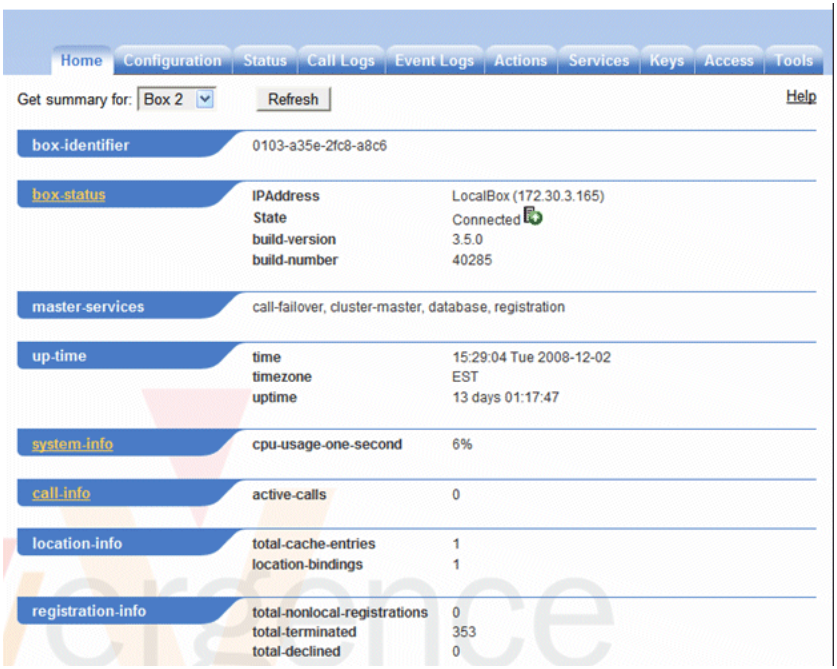
```
config> config access permissions admin
Creating 'permissions admin'
config permissions admin> set ?

 access permission settings
```

```
 cli             permission to access the CLI
 cms             permission for web management
 user-portal     permission to access user portal
 config          permission to access cluster configuration
 status          permission to see status reports
 actions         permission to execute actions
 call-logs       permission to access accounting, user session,
   session, SIP Message logs
 templates       permission to execute web service templates
 troubleshooting permission to execute troubleshooting functionality
 web-services    permission to access the web services interface
 debug           permission to perform debugging operations
 login-attempts  maximum number of failed login attempts

config permissions admin>
```

Using the AA-SBC Management System

The AA-SBC Management System provides access to the same configuration, status and action commands as the CLI, but with some additional capabilities. These additional capabilities support call and event log displays, call histograms, trends, and other graphical features. Refer to the *Net-Net OS-E – Using the NNOS-E Management Tools* for complete information.

The AA-SBC Management System home page displays overall system status, and provides tabs to access the major areas of functionality.



The tab selections are as follows:

Configuration — Main configuration pages: cluster, box, interfaces, protocols and call handling.

Status — Displays state information per status provider, and equivalent to the CLI **show** commands organized by functional area. Additionally, with the status sampling feature enabled, graphical display trends are available with some commands.

Call logs — Displays the files that contains records of SIP calls that have been processed by AA-SBC. Call logs can be customized and filtered using configured criteria.

Event logs — Displays event logs for the cluster and the local system. Each event log is a file of messages that describe AA-SBC activity over a given period. Event logs can be customized and filtered using configured criteria.

Actions — Executes a selected function for immediate processing at the AA-SBC device or cluster. AA-SBC Management System actions are equivalent to the CLI action commands available from the NNOS-E prompt.

Services — Configures logging, external database connections, storage management, periodic tasks, cluster master services and system preferences.

Keys — Allows import and management of TLS certificates and keys from a certificate authority (CA) or other valid encryption source.

Access — Configures users and permissions to multiple functional categories within AA-SBC.

Tools — Provides access to XML schemas, SNMP MIBs, WSDL files and a variety of commonly-used tools for updating software, managing configuration files, licenses, and phone configurations.

2. Monitoring AA-SBC

About this chapter

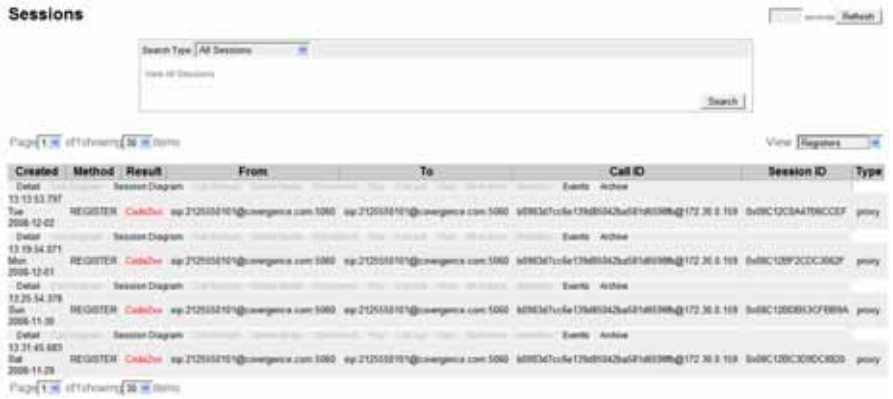
This chapter provides an overview of the tools and procedures for monitoring AA-SBC system hardware and software..

Note: The procedures and practices covered in this guide represent a fraction of the capabilities offered by the AA-SBC software. Troubleshooting more complex or unique problems may require not only the examination of call logs and events generated by AA-SBC, but also that of any third-party devices that are participating in your SIP network. The procedures and practices covered in this guide simply provide a common starting point for general troubleshooting and analysis that will lead you to the best solution.

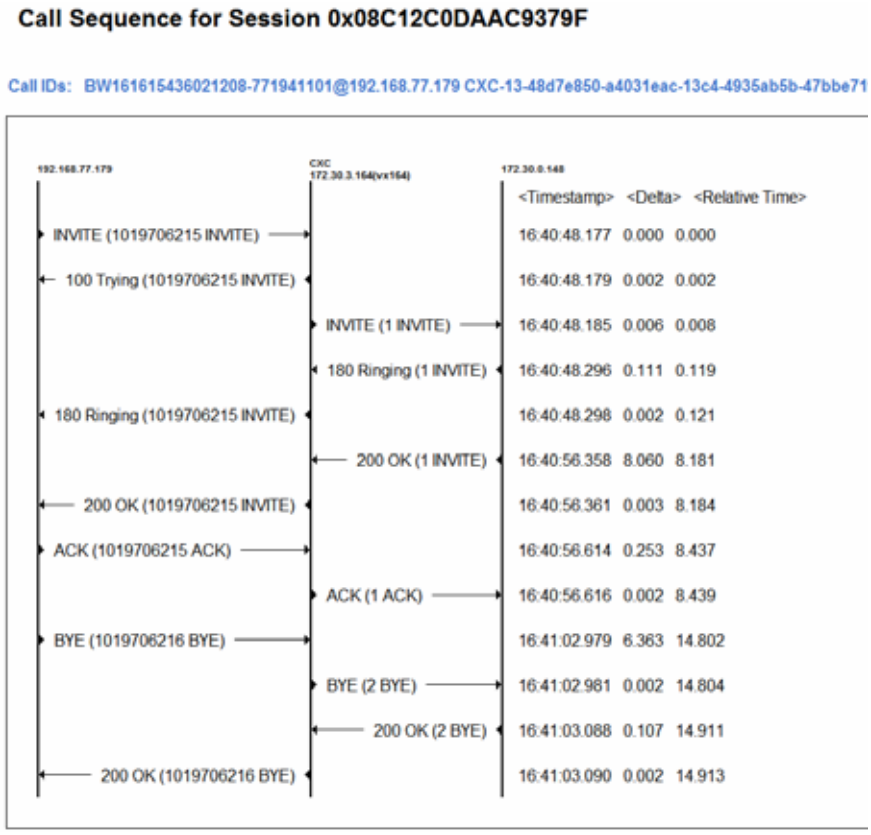
AA-SBC call logs

The call log provides important information when troubleshooting a registration or call problem. It is searchable by calling and called numbers, date and time, or by the call identifier. Registration and call messages are displayed separately in the call log.

The following image shows an example log with registration messages:



By selecting an individual session, you can display SIP call diagrams and the message content, as illustrated in the image below.

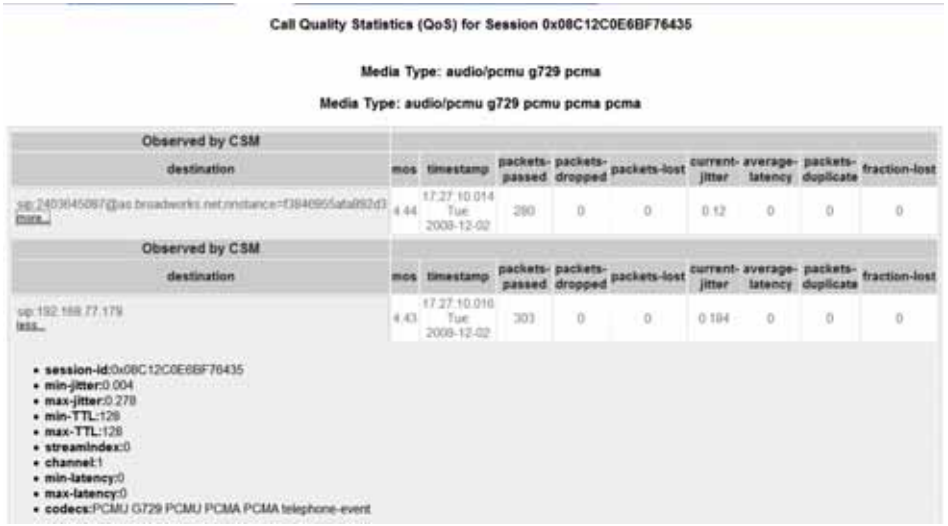


You can display the message contents, or download the message contents in plain text or XML format:

```
Session ID : 0x08C12C0DAAC9379F

----------------------------------------
Timestamp     : 16:40:48.177 2008-12-02
Direction     : RX
Remote IP/Port: 192.168.77.179/5060
Local IP/Port : 172.30.3.164(vx164)/5060
Transport     : UDP
----------------------------------------
```

```
INVITE
   sip:2403645087@172.30.3.164:5060;rinstance=f3846955afa892d3;transp
   ort=udp SIP/2.0
Via:SIP/2.0/UDP
   192.168.77.179;branch=z9hG4bK-BroadWorks.sf2-172.30.3.164V5060-0-1
   019706215-2086082245-1228252575437-
From:<sip:192.168.77.179>;tag=2086082245-1228252575437-
To:"welbourn1
   welbourn1"<sip:2403645087@as.broadworks.net;rinstance=f3846955afa8
   92d3>
Call-ID:BW161615436021208-771941101@192.168.77.179
CSeq:1019706215 INVITE
Contact:<sip:192.168.77.179:5060>
Allow:ACK,BYE,CANCEL,INFO,INVITE,OPTIONS,PRACK,REFER,NOTIFY
Supported:timer
Min-SE:60
Accept:multipart/mixed,application/dtmf-relay,application/
   media_control+xml,application/sdp
Max-Forwards:10
Content-Type:application/sdp
Content-Length:554

v=0
o=BroadWorks 46793 1 IN IP4 172.30.0.159
s=-
c=IN IP4 172.30.0.159
t=0 0
m=audio 3000 RTP/AVP 0 18 96 102 107 104 105 106 97 98 2 99 8 101
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=rtpmap:96 BV16/8000
a=rtpmap:102 BV32/16000
a=rtpmap:107 L16/16000
a=rtpmap:104 PCMU/16000
a=rtpmap:105 PCMA/16000
a=rtpmap:106 L16/8000
a=rtpmap:97 G726-16/8000
a=rtpmap:98 G726-24/8000
a=rtpmap:2 G726-32/8000
a=rtpmap:99 G726-40/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:30
a=silenceSupp:on - - - -

---------------------------------------
```

If calls are anchored and RTP statistics have been enabled in the session configuration, they can be displayed for each call leg when diagnosing call quality problems.



Call Quality Statistics (QoS) for Session 0x08C12C0E6BF76435

Media Type: audio/pcmu g729 pcma

Media Type: audio/pcmu g729 pcmu pcma pcma

Observed by CSM									
destination	mos	timestamp	packets-passed	packets-dropped	packets-lost	current-jitter	average-latency	packets-duplicate	fraction-lost
sip:2403645087@as.broadworks.net;rinstance=f384095afa892d3 more...	4.44	17:27:10.014 Tue 2008-12-02	280	0	0	0.12	0	0	0

Observed by CSM									
destination	mos	timestamp	packets-passed	packets-dropped	packets-lost	current-jitter	average-latency	packets-duplicate	fraction-lost
sip:192.168.77.179 less...	4.43	17:27:10.016 Tue 2008-12-02	303	0	0	0.184	0	0	0

- **session-id**:0x08C12C0E6BF76435
- **min-jitter**:0.004
- **max-jitter**:0.278
- **min-TTL**:128
- **max-TTL**:128
- **streamIndex**:0
- **channel**:1
- **min-latency**:0
- **max-latency**:0
- **codecs**:PCMU G729 PCMU PCMA PCMA telephone-event

Event log monitoring

In many customer environments, AA-SBC is set up to send event logging information to external syslog servers, to the local cluster database, and to local log files on each AA-SBC device.

AA-SBC is typically configured to send events for all system components at either the warning or error level (depending on the component), with critical events sent to syslog servers. Event logs should be checked to ensure that AA-SBC clusters are operating properly.

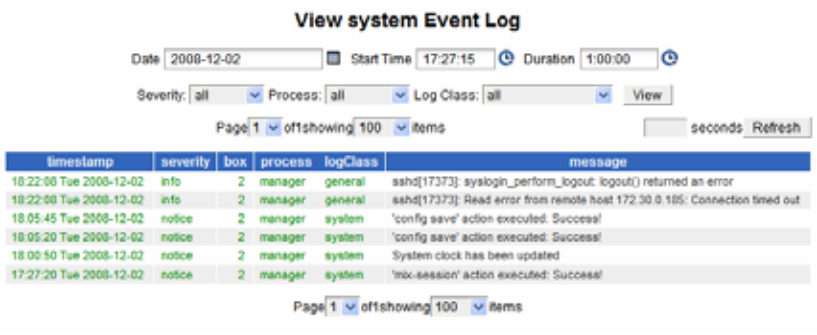
The local cluster database receives error events for all system components, as well as VRRP events logged as notice and higher severities. The event log is accessed from the AA-SBC Management System **Events** tab or from the CLI using the **show event-log** command executed at the cluster master.

Each cluster member has more detailed logging information in its local database log files. These are commonly set up approximately as follows, with one file per event class.

Event class	Content (Level)
all	All (error)
krnlsys	Kernel system (debug)
general + system	General (info), system (info)
db	Database (debug)
access	administrator logins (debug)
dossip	SIP denial-of-service (alert)
dir	Directory (debug)
cms	AA-SBC Management System (CLI and web, including changes to the configuration) (info).
archive	Periodic archiving tasks (info)
sip	Various classes of SIP event (error)

The contents of these files should be viewed from AA-SBC Management System connected to the system on which the file resides. Note that the cluster master will only show its own local files, and not those on other AA-SBC systems.

Thge following image illustrates a sample AA-SBC event log file.



System tracing

Occasionally, it will be necessary to investigate problems under the direction of Technical Support. In some cases, you will need to provide detailed information beyond what is provided in the call and event logs. Use the system **trace** facility to gather this additional information.

System tracing requires CLI access and privileges to access debug-level commands. Debug privilege gives access to various command shells, including the Linux bash shell and various AA-SBC shells. The command to access a shell is **shell** <shellname>, and the command to exit a shell is **exit**. If no shell name is given, then the Linux bash shell is enabled.

The most commonly used AA-SBC shell is the SIP shell, which inherits most of the regular AA-SBC commands and provides additional diagnostic facilities. Here is an example of using the SIP shell to trace SIP traffic for a particular phone number:

```
NNOS-E> shell sip
SIP> trace target trace1.txt
trace trace1.txt> trace * error
trace trace1.txt> trace sip_traffic info
trace trace1.txt> exit
Do you want to save the settings for this target (y or n)? y
Do you want to start tracing to this target (y or n)? n
SIP> trace-filter enabled 9785551234
Start to trace based on user 9785551234
SIP> trace start trace1.txt
```

Leave this trace running until the problem has occurred, at which point turn off tracing:

```
SIP> trace stop trace1.txt
SIP> trace-filter disabled
Disabling filtered tracing
SIP>
```

Note that trace settings for a target file, such as *trace1.txt*, is stored in a file called *trace1.txt.ini* in the /cxc/trace directory.

System monitoring

It is important to regularly monitor AA-SBC clusters for faults and potential capacity and performance issues. If you are using an SNMP monitoring system, this will usually be the first place to examine, followed by the central syslog servers. However, the status of each cluster and individual servers should also be monitored periodically.

The following sections cover measurements of system performance needed to create baselines for a normally operating system, and specific indications of software, hardware and network-related problems.

Typical problems

The types of problem that may be encountered include the following:

- Performance issues (excessive CPU usage, call volumes, etc)
- Housekeeping problems (disk usage and database maintenance)
- Software faults (crashes, memory allocation failures, resource leaks)
- Configuration errors (which generally result in connectivity problems)
- Hardware faults (box, interface or component failures)
- Network connectivity problems (which may result in loss of server connectivity, registrations, decreased call volumes, etc)

Baselines

It is important to create baselines for AA-SBC system behavior to determine normal vs. abnormal activity. Because no two AA-SBC configurations and their subscriber traffic profiles are the same, you should determine how your systems behave over typical daily and weekly cycles to anticipate how external events may impact calling patterns.

While system failures (whether of software, hardware or network connectivity) will usually be obvious, you should watch for subtle or latent issues, such as resource leaks.

Critical information to monitor includes the following:

- Memory usage

- Disk usage
- Session counts
- Registration counts
- Call counts for upstream servers

Measuring system utilization

The following section covers the status provider commands for checking how system resources are being utilized.

Measuring memory usage

Use the **show system-heap** command to show memory usage by process:

```
NNOS-E> show system-heap
-------------------------------------------------------------------------------
    -
Process  Maximum Size  Current Size  Percent  Lockable  Failures | All Failures
-------------------------------------------------------------------------------
    -
monitor       128.00M         9.99M       7%         0         0 |            0
manager         1.500G        46.99M       3%         0         0 |            0
SIP             2.601G       649.99M      24%         0         0 |            0
media         128.00M        12.99M      10%         0         0 |            0
auth          512.00M        13.99M       2%         0         0 |            0
reg             2.601G        32.99M       1%         0         0 |            0
userdb        128.00M        10.99M       8%         0         0 |            0
-------------------------------------------------------------------------------
    -
```

If the memory usage for a process rises steadily over time, this may be an indication of a memory leak. If so, contact Technical Support.

Measuring disk usage

Use the **show mounts** command to show disk usage:

```
NNOS-E> show mounts

drive    mount-point    drive-name     filesystem     drive-size percent-free
-----    -----------    ----------     ----------     ---------- ------------
system-1 /mnt/backup                                  0          0
system-2 /              /dev/root      reiserfs       11864      7
common   /cxc_common    /dev/sda2      reiserfs       92924      65
data-1   /cxc_common/data1                            0          0
data-2   /cxc_common/data2                           0          0
usb      /mnt/usb                                     0          0
cdrom    /mnt/cdrom                                   0          0
ramdisk  /mnt/ramdisk   rootfs         tmpfs          1024       100
```

Pay special attention to the percentage of free space on the mounted disks. If this number diminishes significantly, delete the obsolete CDR files and recorded media, as these files consume significant disk space.

Measuring session counts

Use the **show active-call-summary** command to count the number of active signaling sessions passing through (or mirrored by) a box:

```
NNOS-E> show active-call-summary

state                          count
-----                          -----
B2B_CONNECTED                  2
```

This corresponding command for media sessions is show media-stream-counts:

```
NNOS-E> show media-stream-counts

client-id       server-id       sessions
---------       ---------       --------
0.0.0.0         0.0.0.0         2
```

If call counts are unexpectedly high, it may be a sign that traffic is being diverted somewhere in the network; if they are unexpectedly low, it is probably an indication of a problem in the network where traffic is not reaching AA-SBC.

Measuring registrations

Use the **show location-summary** command to count the number of registration bindings.

```
NNOS-E> show location-summary

                      total-AORs: 2
                  total-aliases: 2
                 total-bindings: 2
       total-local-registrations: 1603
    total-delegate-registrations: 212
total-disconnected-registrations: 0
        total-aged-registrations: 0
          total-re-registrations: 156729
             total-registrations: 1815
           total-downloaded-AORs: 0
               total-registereds: 2
        total-registered-aliases: 1727
             total-unregistereds: 0
      total-unregistered-aliases: 0
```

```
                    total-tryings: 0
              total-trying-aliases: 0
                 total-in-services: 0
           total-in-service-aliases: 0
              total-out-of-services: 0
      total-out-of-service-aliases: 0
```

Pay special attention to the total-bindings number.

Measuring server traffic

Use the **show sip-server-cac** command to show the numbers of calls and registrations sent to peer servers:

```
NNOS-E> show sip-server-cac -v

                                  peer: broadworks
                                server: broadworks
                             admission: disabled
                              emission: disabled
                         max-bandwidth: unlimited
                        used-bandwidth: 174 kbits-per-second
                   available-bandwidth: unlimited kbits-per-second
       max-number-of-concurrent-calls: 1000
                       connected-calls: 2
                    max-calls-in-setup: 30
                        calls-in-setup: 0
              call-rate-limiting-state: disabled
               call-rate-limiting-rate: 0
          call-rate-limiting-interval: 0
          max-number-of-registrations: 1000
                      registered-aors: 1
        max-registrations-in-progress: 300
            registrations-in-progress: 0
               cluster-used-bandwidth: 174 kbits-per-second
              cluster-connected-calls: 2
               cluster-calls-in-setup: 0
         cluster-wca-next-percentage: 0
    cluster-wca-next-percentage-calls: 3
              cluster-registered-aors: 0
    cluster-registrations-in-progress: 0
      cluster-registration-percentage: 0
cluster-next-percentage-registrations: 2
```

Assuming that all calls are being forwarded to servers, the total number of calls forwarded to all servers for any given system should equal the total number of calls being processed by that system.

Note: If the total number of calls sent to servers is different than that shown by **show active-call-summary** by more than a few calls, this would indicate that the CAC mechanism is not counting calls properly, which you should report to Technical Support.

Problem indicators

The following sections describe places to look for indications of problems.

Check the event logs

Check the events logged to the local database, either from AA-SBC Management System or from the CLI **show event-log** command. Event logs may give indications of connectivity problems, recoverable software errors and incorrectly configured objects and properties.

```
NNOS-E> show event-log

timestamp                severity box        process  class            message
---------                -------- ---        -------  -----            -------
05:47:36 Tue 2008-12-02  error    1         manager  tls            Cert entry
    'enms' could not load a private key from '/cxc/certs/enms.cert'; if this is
    not a server certificate, then this error can be ignored
05:47:36 Tue 2008-12-02  error    1         manager  tls              OpenSSL
    error from SSL_CTX_use_PrivateKey_file()() (returned 0):
05:47:36 Tue 2008-12-02  error    1         manager  tls              OpenSSL
    error 0906d06c:  error:0906D06C:PEM routines:PEM_read_bio:no start line
    (pem_lib.c:647)
05:47:36 Tue 2008-12-02  error    1         manager  tls                ->
    Expecting: ANY PRIVATE KEY
05:47:36 Tue 2008-12-02  error    1         manager  tls              OpenSSL
    error 140b0009:  error:140B0009:SSL
    routines:SSL_CTX_use_PrivateKey_file:PEM lib (ssl_rsa.c:669)
05:47:36 Tue 2008-12-02  error    1         manager  tls              OpenSSL
    error from PEM_read_RSAPrivateKey()() (returned 0):
05:47:36 Tue 2008-12-02  error    1         manager  tls              OpenSSL
    error 0906d06c:  error:0906D06C:PEM routines:PEM_read_bio:no start line
    (pem_lib.c:647)
05:47:36 Tue 2008-12-02  error    1         manager  tls                ->
    Expecting: ANY PRIVATE KEY
```

Also view the various categories of events logged to the file system of individual AA-SBC devices from the AA-SBC Management System.

Check system up-time

Use the **show clock** command to check that a system has not restarted unexpectedly:

```
NNOS-E> show clock

  time: 08:29:24 Thu 2008-12-04
uptime: 0 days 01:52:46
```

The **show boxes** command shows the operational status of all the members of the cluster and their up-time, as known from the perspective of the system from which the command is being run:

```
NNOS-E> show boxes
-------------------------------------------------------------------------------
    -
Box Address      ? Prot  State       Up Time   Connects  Errors  Last Error
-------------------------------------------------------------------------------
    -
Local            O None  Connected   02:08:19         1       0  Unknown
10.1.22.37       A TCP   Connected   02:08:08         1       0  None
10.1.22.38       A TCP   Connected   02:08:08         1       0  None
10.1.22.39       A TCP   Connected   02:08:08         1       0  None
10.1.22.40       A TCP   Connected   02:08:08         1       0  None
10.1.22.41       O TCP   Connected   02:08:07         1       1  None
10.1.22.42       A TCP   Connected   02:08:07         1       0  None
-------------------------------------------------------------------------------
```

Check software version

Following an upgrade, use the show version command to check that all the members of a cluster have the correct software version:

```
NNOS-E> show version

image    version   build         branch    time                      computer
-----    -------   -----         ------    ----                      --------
monitor  3.5.0     40285         b3.5.0    06:36:09 Thu 2008-11-06   AUTO2
manager  3.5.0     40285         b3.5.0    06:38:22 Thu 2008-11-06   AUTO2
SIP      3.5.0     40285         b3.5.0    07:01:36 Thu 2008-11-06   AUTO2
media    3.5.0     40285         b3.5.0    06:38:36 Thu 2008-11-06   AUTO2
reg      3.5.0     40285         b3.5.0    06:36:43 Thu 2008-11-06   AUTO2
web      3.5.0     40285         b3.5.0    06:53:00 Thu 2008-11-06   AUTO2
acct     3.5.0     40285         b3.5.0    06:55:00 Thu 2008-11-06   AUTO2
dos      3.5.0     40285         b3.5.0    06:55:00 Thu 2008-11-06   AUTO2
```

Repeat the command for each cluster member. If the cluster is in an inconsistent state following an upgrade, contact Technical Support.

Check for software faults

Use the **show faults** command to check for crashes:

```
NNOS-E> show faults

   time: 05:04:19 Tue 2008-10-14
   file: manager1-000.txt
address:
 reason: 0 (crash report generated by request from 081ca6b3
    MsgRestartAfterPBFailure + 0x23  message/message_socket.c:87)
 uptime: 0 days 00:09:43
version: 3.5.0
  build: 39523-dev
 branch: b3.5

   time: 19:04:20 Tue 2008-10-14
   file: SIP1-000.txt
address:
 reason: Aborted
 uptime: 0 days 00:00:01
version: 3.5.0
  build: 39710-dev
 branch: b3.5
```

If a crash should be found, collect the crash file (which will be found in the directory /cxc_common/crash), together with the configuration file and any traces and event logs that might pertain to the incident, and send it to Technical Support, noting whether or not the incident was service-affecting.

Use the **show memory-failures** command to check for memory allocation problems:

```
box1> show memory-failures -v

Memory allocation failures:
-------------------------------------------------------------------------------
Process  Address   Failures  Oldest Fail  Newest Fail   Smallest    Largest
-------------------------------------------------------------------------------
manager  08110b6b         8   1251:12:37   1251:12:35    1083552    8668416
         MemInitMallocReplacementHeap + 0x3b  util/mem_heap.c:6541 + 0x26
         Oldest failure occured at:  13:41:41 Tue 2008-11-18
         Newest failure occured at:  13:41:43 Tue 2008-11-18
-------------------------------------------------------------------------------
```

If any memory failures are detected, please collect the output of the above command and send it to Technical Support.

Check hardware status

Use the **show sensor-events** command to check the hardware status:

```
NNOS-E> show sensor-events

timestamp               sensor            type                   description
---------               ------            ----                   -----------
```

```
08:04:37 Mon 2008-06-30  Event Log Disabl0  event-logging-disabled   log area
    reset/cleared
08:05:17 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:05:17 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:05:32 Mon 2008-06-30  Pwr Unit Stat0     power-unit               power off/
    down
08:05:32 Mon 2008-06-30  Button0           button                  power button
    pressed
08:05:36 Mon 2008-06-30  Pwr Unit Stat0     power-unit               power off/
    down
08:05:37 Mon 2008-06-30  Button0           button                  power button
    pressed
08:05:40 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:05:40 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:05:53 Mon 2008-06-30  Drv 1 Pres0        drive-slot               device
    inserted/present
08:05:56 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:05:56 Mon 2008-06-30  Power Redundancy0  power-unit               fully
    redundant
08:06:14 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:06:16 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:06:19 Mon 2008-06-30  POST Error0        system-firmware-progress error
04:06:19 Mon 2008-06-30  POST Error0        system-firmware-progress error
04:07:58 Mon 2008-06-30  System Event0      system-event            OEM system
    boot event
04:08:05 Mon 2008-06-30  ACPI State0        system-acpi-power-state  S0/G0:
    working
04:08:18 Mon 2008-06-30  Button0           button                  reset button
    pressed
04:08:46 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:08:45 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:08:51 Mon 2008-06-30  POST Error0        system-firmware-progress error
04:08:51 Mon 2008-06-30  POST Error0        system-firmware-progress error
04:10:39 Mon 2008-06-30  Processor 1 Stat0  processor                presence
    detected
04:10:39 Mon 2008-06-30  Processor 2 Stat0  processor               thermal trip
04:10:39 Mon 2008-06-30  Processor 2 Stat0  processor                presence
    detected
04:10:42 Mon 2008-06-30  Processor 1 Stat0  processor                presence
    detected
04:10:42 Mon 2008-06-30  Processor 2 Stat0  processor                presence
    detected
04:11:24 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:11:24 Mon 2008-06-30  System Event0      system-event             timestamp
    clock sync
04:12:59 Mon 2008-06-30  System Event0      system-event            OEM system
    boot event
04:13:06 Mon 2008-06-30  ACPI State0        system-acpi-power-state  S0/G0:
    working
14:03:24 Mon 2008-06-30  Physical Scrty0    physical-security        system
    unplugged from LAN
14:03:25 Mon 2008-06-30  Physical Scrty0    physical-security        system
    unplugged from LAN
```

Key items to look for are fan and power supply failures. (In the above output, the system-firmware-progress error is not significant.)

Check interface status

Use the **show interfaces** command to check that each cluster member's interfaces are operational:

```
NNOS-E> show interfaces

interface  name            ip-address         op-state         type
---------  ----            ----------         --------         ----
eth2.122   ms0-heartbeat0  10.1.22.45/24      up               public
eth3.122   ms0-heartbeat1  10.1.22.46/24      down             public
eth4       MartyTemp       10.1.13.252/24     up               public
vx1        cluster-mgmt    10.1.13.20/24      up               public
vx1:1      cluster-cdr     10.1.13.21/24      up               public
vx2.113    ms0-node-mgmt   10.1.13.208/24     up               public
vx2.122    ms0-messaging   10.1.22.36/24      up               public
vx20.123   sip0-ext        10.1.23.68/24      down             public
vx20.123:1 sip1-ext        10.1.23.69/24      down             public
vx20.123:2 sip2-ext        10.1.23.70/24      down             public
vx20.123:3 sip3-ext        10.1.23.71/24      down             public
vx21.123   sp1-sip0-ext    10.1.23.72/24      down             public
vx21.125   sp1-rtp0-ext    10.1.25.100/24     down             public
vx22.124   sp1-sip0-int    10.1.24.84/24      down             public
vx22.126   sp1-rtp0-int    10.1.26.116/24     down             public
vx23.123   sp2-sip0-ext    10.1.23.73/24      down             public
vx23.125   sp2-rtp0-ext    10.1.25.102/24     down             public
vx24.124   sp2-sip0-int    10.1.24.85/24      down             public
vx24.126   sp2-rtp0-int    10.1.26.118/24     down             public
vx3.113    ms1-node-mgmt   10.1.13.202/24     down             public
vx3.122    ms1-messaging   10.1.22.37/24      down             public
vx4.113    sd0-node-mgmt   10.1.13.203/24     down             public
vx4.122    sd0-messaging   10.1.22.38/24      down             public
vx5.113    sd1-node-mgmt   10.1.13.204/24     down             public
vx5.122    sd1-messaging   10.1.22.39/24      down             public
vx50.125   sp1-rtp1-ext    10.1.25.101/24     down             public
vx51.126   sp1-rtp1-int    10.1.26.117/24     down             public
vx52.125   sp2-rtp1-ext    10.1.25.103/24     down             public
vx53.126   sp2-rtp1-int    10.1.26.119/24     down             public
vx6.113    sp0-node-mgmt   10.1.13.205/24     down             public
vx6.122    sp0-messaging   10.1.22.40/24      down             public
vx7.113    sp1-node-mgmt   10.1.13.206/24     down             public
vx7.122    sp1-messaging   10.1.22.41/24      down             public
vx8.113    sp2-node-mgmt   10.1.13.207/24     down             public
vx8.122    sp2-messaging   10.1.22.42/24      down             public
```

Note that if a system does not own the VRRP interface, it will be shown as 'down'.

Check running processes

Use the **show processes** command to check that the software components of each cluster member are running as expected. Enabled subsystems should have a run level of 7, and up-times should be consistent:

```
process  id         condition run-level   starts uptime             fds
-------  --         --------- ---------   ------ ------             ---
monitor  5330       running   7           1      0 days 02:17:31    22
manager  5533       running   7           1      0 days 02:17:31    92
SIP      5801       running   7           1      0 days 02:17:29    125
media    5802       running   7           1      0 days 02:17:29    34
auth     6277       running   7           1      0 days 02:17:22    31
reg      5803       running   7           1      0 days 02:17:29    23
H323     0          idle      init        0      0 days 00:00:00    0
dir      6279       running   7           1      0 days 02:17:22    14
web      6003       running   7           1      0 days 02:17:28    165
WS       6004       running   7           1      0 days 02:17:28    16
acct     6275       running   7           1      0 days 02:17:22    11
dos      6278       running   7           1      0 days 02:17:22    11
SSH      6001       running   none        1      0 days 02:17:28    12
LCR      0          idle      init        0      0 days 00:00:00    0
sampling 0          idle      init        0      0 days 00:00:00    0
userdb   6007       running   7           1      0 days 02:17:28    15
presence 0          idle      init        0      0 days 00:00:00    0
```

Check cluster services

Use the **show master-services** command to verify that cluster-wide master services are running on the expected host. (If not, it is a sign that a failover event occurred.)

```
NNOS-E> show master-services

name               hosted position waiting group host            host-position
----               ------ -------- ------- ----- ----            -------------
3pcc               false  0        false   0     0.0.0.0         0
accounting         true   1        false   3     0.0.0.0         1
authentication     true   1        false   4     0.0.0.0         1
call-failover      true   1        false   8     0.0.0.0         1
cluster-master     true   1        false   1     0.0.0.0         1
database           true   1        false   5     0.0.0.0         1
directory          true   1        false   2     0.0.0.0         1
dos-defense        false  0        false   0     0.0.0.0         0
file-mirror        false  0        false   0     0.0.0.0         0
gateway-routing    false  0        false   0     0.0.0.0         0
least-cost-routing false  0        false   0     0.0.0.0         0
load-balancing     false  0        false   9     10.1.22.38      1
registration       true   1        false   6     0.0.0.0         1
sampling           false  0        false   0     0.0.0.0         0
server-load        true   1        false   7     0.0.0.0         1
```

Check database maintenance status

Use the **show database-maintenance-status** command on the system hosting the database master service to display the current maintenance status of database operations. Use this to determine whether an operation (such as a backup or restore) has finished. It will show 'idle' if it has completed correctly. If a check of the database event log indicates that the system could not execute a database operation, use this command to verify the state of the database.

```
NNOS-E> show database-maintenance-status

  status: idle
   table: registration-stop
 started: 09:59:16 Wed 2009-01-14
finished: 09:59:16 Wed 2009-01-14
  result: Success!
```

Check SIP status

Use the **show sip-summary-by-box** command to verify that SIP traffic is passing through the cluster as expected:

```
NNOS-E> show sip-summary-by-box

                box: 6
     connected-calls: 0
     transient-calls: 0
      used-bandwidth: 0
                 MOS: 0
       call-duration: 0
     attempted-calls: 0
    successful-calls: 0
        failed-calls: 0
      rejected-calls: 0
         rx-messages: 294691
           rx-INVITE: 0
         rx-REGISTER: 176814
              rx-ACK: 0
            rx-CANCEL: 0
            rx-NOTIFY: 0
        rx-SUBSCRIBE: 0
          rx-OPTIONS: 0
          rx-MESSAGE: 0
              rx-1xx: 0
              rx-2xx: 117877
              rx-3xx: 0
              rx-4xx: 0
              rx-5xx: 0
```

```
              rx-6xx: 0
rx-clipped-registers: 0
        rx-dos-drops: 0
  rx-checksum-errors: 0
     rx-parse-errors: 0
rx-queue-full-errors: 0
         tx-messages: 294691
           tx-INVITE: 0
         tx-REGISTER: 117877
              tx-ACK: 0
            tx-CANCEL: 0
            tx-NOTIFY: 0
        tx-SUBSCRIBE: 0
           tx-OPTIONS: 0
           tx-MESSAGE: 0
               tx-1xx: 0
               tx-2xx: 176814
               tx-3xx: 0
               tx-4xx: 0
               tx-5xx: 0
               tx-6xx: 0
  tx-retransmissions: 0
        tx-failures: 0
        box-address: 10.1.22.41
```

Use the **show sip-server-pool** command from boxes running SIP to verify that SIP servers are reachable and responding:

```
NNOS-E> show sip-server-pool

peer-name        server        host            TPT port  box state in   out
---------        ------        ----            --- ----  --- ----- --   ---
ser-test-cluster ser-test-cluster 10.1.24.250  UDP 5060  local up    0    0
mike-ser         mike-ser      172.30.0.226    UDP 5070  local up    0    0
west-cluster     west-SER      194.97.59.170   UDP 5060  local up    0    0
```

3. AA-SBC maintenance

About this chapter

This chapter provides an overview for maintaining AA-SBC..

Note: The procedures and practices covered in this guide represent a fraction of the capabilities offered by the AA-SBC software. Troubleshooting more complex or unique problems may require not only the examination of call logs and events generated by AA-SBC, but also that of any third-party devices that are participating in your SIP network. The procedures and practices covered in this guide simply provide a common starting point for general troubleshooting and analysis that will lead you to the best solution.

Backing up system files

Following the commissioning of a system, it is recommended to create a backup image in case of a serious system failure. Use the **restore-stick-create** command to write an image out to a USB drive, and to update an image with its installation-specific files. Backing up system files via the System Platform is the preferred method for AA-SBC.

Installation-specific files that are saved to the USB drive are as follows:

- Configuration files: /cxc/*.cfg, /cxc/*.xml
- Shared secret files: /cxc_common/cxc.pw1, /root/cxc.1
- TLS certificates: /cxc/certs/*
- License files: /cxc/license/*
- SSH keys: /cxc_common/ssh_authorized_keys
- Miscellaneous files: /etc/mactab, /boot/grub/.cxc_options

You should periodically update the USB system image from time to time with any changes that may have been made to these files, particularly configuration files. You can use WinSCP to access remote systems, or you can download the files using the AA-SBC Management System **Tools** page.

Note: Other files, including custom announcements (.WAV files), CDRs, logs and the system database are not saved by the **restore-stick-create** command and should be copied manually, if required.

Managing accounting files

When using local CSV files, call detail records are usually kept in a subdirectory of the /cxc_common directory on the AA-SBC cluster master. New files are created hourly, with the file names in the format,

<configurable prefix>.hourly.2008.11.04.19.25.35

where the file name reflects the time the file was created. These files should be copied off the system periodically using WinSCP or other file transfer mechanism.

If CDRs are being written to a database, these accounting file management considerations do not apply.

Managing log files

By default, each log file is allowed to grow to a maximum size of 10 MB, with five generations of the file. When the maximum number of generations is reached, the first file is emptied for re-writing, and the files are rotated from that point. No manual removal of files should be required.

Log files are stored in the directory /cxc_common/log.

Managing the system database

The information in this section covers the common practices for maintaining the AA-SBC database.

Automatic nightly maintenance

By default, the local system database keeps records for one year, or 365 days. Database maintenance is performed daily at 3 a.m. local time by default. For various reasons, it is possible for the automated database maintenance to fail, and so the results of the maintenance activity should be checked for errors by referring to the syslog, database log file or by using the **show database-maintenance-status** command. A database maintenance failure will appear in the following format:

2008-02-01T02:10:30+13:00[crit] 1:SIP[system] Database table SipMessage has 187506143 unused pointers and requires a VACUUM FULL

2008-02-01T02:16:32+13:00[crit] 1:SIP[system] Database table SpotliteTransportMsg has 7847143 unused pointers and requires a VACUUM FULL

If database maintenance fails, perform the **database vacuum-full database** *tablename*, where *tablename* indicates the table given in the error message. If this fails, contact Technical Support.

Manual preventative maintenance

The normal database maintenance done on a nightly basis performs a purge, vacuum, reindex, and analyze. There is one additional action that should be performed on a regular basis that is not performed automatically called **database vacuum-full**, executed at the prompt.

The normal vacuum process attempts to reclaim any unused space in the database (analogous to a hard drive defragmentation process) without locking any of the tables, as much as is possible without a lock. The database vacuum-full action locks each table one at a time and reclaims all possible disk space. Note that a table lock prevents AA-SBC from writing to the locked table.

It is recommended performing a vacuum-full on a monthly basis by scheduling a maintenance window and running the **database vacuum-full** action.

A maintenance window is recommended because of the need to lock database tables. This can affect the ability of a DOS rule from being triggered and can affect call logs and any other data that is written to the database. This will not affect the ability of AA-SBC to pass SIP and media traffic, accept and delegate registrations, route calls, and perform other directly service-related tasks.

If a site is logging a large volume of data, executing **database vacuum-full** may be needed on a more frequent basis. If the amount of data that is being written to the database is substantial, it may be necessary to reduce the amount of data that is being logged to the database. For example, omitting SIP registration records is one way to reduce database consumption.

4. AA-SBC troubleshooting

About this chapter

This chapter provides information that will help you troubleshoot AA-SBC networks.

Note: The procedures and practices covered in this guide represent a fraction of the capabilities offered by the AA-SBC software. Troubleshooting more complex or unique problems may require not only the examination of call logs and events generated by AA-SBC, but also that of any third-party devices that are participating in your SIP network. The procedures and practices covered in this guide simply provide a common starting point for general troubleshooting and analysis that will lead you to the best solution.

Collecting Diagnostic Data From a Running AA-SBC

The AA-SBC has the ability to collect support data and store it in a single compressed file to be downloaded and forwarded to the support team for analysis. A **collect** action has been created which allows you to collect the information necessary to troubleshoot problems occurring on the AA-SBC.

By default, the AA-SBC collects the following data when the **collect** action is executed.

- Configuration data, including the following:
  - Current running configuration (even if it has not been saved yet)
  - Current /cxc/cxc/cfg configuration file
  - Backup configuration files in /cxc/backup
  - Schema files (*.xsd in /cxc/web)
- Certificate files found in the /cxc/certs directory
- Status data which can be collected in two forms:

- Text files that contain output equivalent to the status show commands
- XML files that contain the same data, but in a structured format that is machine-readable and is used for automated analysis

Status data can be collected in two different ways:

- Default collection, in which a standard, pre-configured list of status classes is collected
- Custom collection, in which status classes not included in the default list can be specified

- Crash files found in the /cxc_common/crash directory
- Log files found in the /cxc_common/log directory
- Directory contents

Enabling and Disabling Default Collection Parameters

Note: Use the **default-collect-settings** property under the instruction of Avaya Aura personnel only.

Using the **services** > **collect** > **default-collect-settings** property, you can enable or disable these default parameters. When one of these properties is set to **disabled**, the corresponding data is not collected.

```
config default-collect-settings>show -v

services
 collect
  default-collect-settings
   config enabled
   certificates enabled
   status enabled
   crash-files enabled
   log-files enabled
```

Under this object you can also edit the list of status classes, databases, and directories from which data is collected.

The **status-class** property specifies additional status classes to be collected. This property is a vector, so you can specify multiple entries. In addition, wildcards can be specified as well as the **-v** property to specify a verbose display in the status text file. For example:

```
config default-collect-settings>set status-class location-bindings-rejected -v
```

```
config default-collect-settings>set status-class system-*
config default-collect-settings>set status-class arena
```

The **database** property specifies the databases you want to collect.

Note: Use the **database** property with caution as it is possible to specify the collection of enormous amounts of data.

The valid databases are:

- log
- spotlite
- status
- dos
- directory
- accounting

This property is a vector, so you can specify multiple entries. For example:

```
config default-collect-settings>set database directory
config default-collect-settings>set database accounting
```

The **directory** property specifies any additional directories to be collected. For example:

```
config default-collect-settings>set directory /cxc_common/data1/dir1
config default-collect-settings>set directory /cxc_common/data1/dir2
```

Note: Use the **directory** property with caution as it is possible to specify the collection of enormous amounts of data.

Customizing Collection Parameters

In addition to the default parameters, you can configure custom collection parameters using the **services** > **collect** > **collect-group** parameter. Once you create a collect-group, you have the ability to disable the default collection parameters, certificates, status, crash-files, and log-files for that collect-group.

The following example shows the AA-SBC configured to collect only data related to accounting, while disabling collection of the other default collection parameters:

```
config collect>config collect-group accounting
```

```
Creating 'collect-group accounting'
config collect-group accounting>set description "Just accounting data"
config collect-group accounting>set certificates disabled
config collect-group accounting>set status disabled
config collect-group accounting>set status-class accounting*
config collect-group accounting>set crash-files disabled
config collect-group accounting>set database accounting
```

To collect this customized data, specify the group name when executing the **collect** action.

```
NNOS-E>collect accounting
```

Managing Collection Output Files

You can specify where the output files will be stored via the **services** > **collect** > **directory** property. The default (/cxc_common/collect) is sufficient in most cases. However, if you are collecting the contents of large databases, this property allows you to specify a mount with more available disk space.

When a new collect file is created, the old files are saved as backups. Older backup files are deleted when the number of backups exceeds the **services** > **collect** > **max-old-files** property.

```
config collect>set directory /cxc_common/collect_2
config collect>set max-old-files 5
```

Collecting Data from a Cluster

By default, the collect action collects data only from the box on which it is executed. Cluster-wide data collection can be specified by adding the cluster parameter to the action.

To collect the default data throughout the cluster, you must specify the default parameter.

```
NNOS-E>collect default cluster
```

To collect custom data from a configured collect-group, specify the collect-group (in this example accounting is used).

```
NNOS-E>collect accounting cluster
```

When cluster-wide data collection is specified, each AA-SBC collects the appropriate data independently and simultaneously. The AA-SBC on which the **collect** action is executed then combines the resulting data into a single file.

Viewing Status Classes Being Collected

The **show collect-status-classes** action displays which status classes are being collected. When entered with the **default** parameter, the AA-SBC default status classes are listed.

```
NNOS-E>show collect-status-classes default
```

You can also use the **show collect-status-classes** status provider to display status classes defined in custom configurations. The following shows accounting as an example.

```
NNOS-E>show collect-status-classes accounting

Status classes to be collected for 'Accounting':
--------------------------------------------------------------------------------
Source   Status class               Description
--------------------------------------------------------------------------------
config   accounting-recent          calls recently accounted
config   accounting-database        request information for accounting database
connections
config   accounting-files           accounting file information
config   accounting-store           accounting disk storage information
config   accounting-cdr-summary     accounting CDR summary
config   accounting-targets-file-system accounting file-system and
external-file-system targets
config   accounting-targets         accounting targets
```

Collect Log Messages

The log class ‘collect’ has been added. The following messages are logged:

- collect[warning]: Collect action invoked with the following arguments:
- collect[info]: <various progress messages>
- collect[warning]: Collect action succeeded after X seconds; file ‘/cxc_common/collect/collect.tar.gz’ is X bytes

- collect[error]: <various error messages>
- collect[error]: Collect action failed; <error message>

The recommended setting for the ‘collect’ log class is ‘warning’. The ‘info’ setting produces many log messages, all of which will appear in the log file (e.g., /box1/box1.txt).

Device registration problems

Device registration problems generally have the following causes:

- Device has incorrectly configured outbound-proxy IP address or hostname
- Device has incorrectly configured username, domain or password
- Network connectivity problems
- Problem with upstream registration proxy
- Intermediary device (proxy or ALG) changing SIP headers

If multiple subscribers are having problems, there is likely a connectivity problem or a problem with an upstream server. Try the following steps:

1. Use **show sip-server-pool** to determine whether the upstream servers are accessible and responding (see section 0 above).
2. Use **show interfaces** on the Signaling members of the cluster, to verify local connectivity of the SIP-bearing interfaces, both public-side and private-side.
3. Check that the RADIUS servers are operational (if using RADIUS authentication) and reachable from AA-SBC. Use the **auth request** command to send a trial request to the RADIUS server group.
4. Use **show sip-stack** and **show processes** on the signaling members of the cluster to verify that the SIP stack is operational. If there is a problem with the SIP stack, contact Technical Support.

If the problem is specific to an individual subscriber, take the following steps:

1. Verify the configured settings for outbound-proxy.

If the subscriber's device is getting no response at all, it may display a 408 Request Timeout message, which is probably because of an incorrectly configured outbound proxy or a local connectivity problem.

If the subscriber's device does not display a message, check the Call Log for registration messages from the subscriber's phone number. Alternatively, create a simple SIP trace with a filter for the subscriber's phone number and have them reboot their device.

2. Verify the configured username and domain. If either of these are incorrect, the subscriber's device may display a 404 Not Found message. If the device does not display a specific message, check the Call Log or take a trace as described in the previous step.
3. Check for a correctly configured password on the device. If this is incorrect, the subscriber's device may display a 403 Forbidden message. If the device does not display a specific message, check the call log or perform a trace.

Once the device has correctly registered, it should be found in the location cache. Use the following command to verify that a device with the correct Address of Record has registered:

```
show location-cache “aor=sip:<phone number>@<domain>"
```

It is also useful to check the individual address bindings associated with an AOR (there may be more than one):

```
show location-bindings "aor=sip:<phone number>@<domain>"
```

Call completion failures

Once a device is correctly registered, persistent call completion failures are generally due to the following conditions:

- Dial plan routing or number normalization errors
- Upstream server or gateway failures
- Network connectivity problems
- For an on-net call, the called subscriber's device may not be registered

To diagnose upstream server, gateway, or network connectivity failures, perform the following steps:

1. Use **show sip-server-pool** to determine whether the upstream servers are accessible and responding.
2. Use **show interfaces** on the Signaling members of the cluster, to verify local connectivity of the SIP-bearing interfaces, both public-side and private-side.

To diagnose dial-plan and normalization problems, perform the following steps:

1. Use the **call-lookup** command to determine the specific dial plan and route that will be applied to a call. For example:

```
NNOS-E> call-lookup acmepacket.com

Arbiter "Factory Default": apply-method best-match, options 1
Matched route "broadworks" (domain !*acmepacket.com), priority 100,
   best yes
option 0: server broadworks preference -1 bandwidth 2147483647 cap
   1000 rate 0 mos 0 setup-time 0

This call will be forwarded to peer broadworks.  IP 192.168.77.179
   transport UDP port 5060
```

   An unexpected routing decision is most likely the result of an improper configuration.

2. For a more detailed look at the policies that may have been applied that affect call routing, use the following trace (substituting the correct target phone number for the one shown in the example):

```
NNOS-E> shell sip
SIP> trace target policy.txt
trace policy.txt> trace * error
trace policy.txt> trace sip_traffic info
trace policy.txt> trace scale* debug
trace policy.txt> trace policy debug
trace policy.txt> trace rule debug
trace policy.txt> trace cfr debug
trace policy.txt> exit
Do you want to save the settings for this target (y or n)? y
Do you want to start tracing to this target (y or n)? n
SIP> trace-filter enabled 9785551234
Start to trace based on user 9785551234
SIP> trace start policy.txt
```

   Leave this trace running until the required call has been made:

```
SIP> trace stop policy.txt
SIP> trace-filter disabled
```

```
Disabling filtered tracing
SIP>
```

3. Refer to the call log to see the specific normalization that has been applied to a given call. There will be a difference between the SIP Request URI in the received INVITE and the forwarded INVITE. An unexpected transformation of the Request URI is most likely be the result of an improper configuration.

4. For a more detailed look at a normalization problem, perform the following trace, substituting the correct target phone number for the one given in the example:

```
NNOS-E> shell sip
SIP> trace target dial_norm.txt
trace dial_norm.txt> trace * error
trace dial_norm.txt> trace sip_traffic info
trace dial_norm.txt> trace sip_routing debug
trace dial_norm.txt> trace server_arbiter debug
trace dial_norm.txt> trace dial_plan* debug
trace dial_norm.txt> trace registration_plan* debug
trace dial_norm.txt> exit
Do you want to save the settings for this target (y or n)? y
Do you want to start tracing to this target (y or n)? n
SIP> trace-filter enabled 9785551234
Start to trace based on user 9785551234
SIP> trace start dial_norm.txt
```

Leave this trace running until the required call has been made:

```
SIP> trace stop dial_norm.txt
SIP> trace-filter disabled
Disabling filtered tracing
SIP>
```

For on-net call failures where the called party does not respond (typically with a 404 Not Found response), examine the location cache:

1. Use **show location-cache "aor=sip:<number>@<domain>"** or **show location-bindings "aor=sip:<number>@<domain>"** to determine whether the called party's device is registered. If this is the case, it will indicate a mismatch between the registration status as known to an upstream registrar (typically an IP PBX or application server) and AA-SBC.

2. Examine the configuration to determine whether registrations are being forwarded to the registrar with the expected frequency.

Media problems

Occasionally calls will complete but audio, in one or both directions, will not be present. This can be the result of a problem with a VoIP device, but can also be due to problems with network address translation (NAT) and how devices are configured to handle it. When such problems occur, collect the following data sets for further analysis:

1. Ethereal traces for AA-SBC SIP and media ports involved in the call.
2. Run the following set of commands, directing the output to your terminal program's log file:

```
display scrolled
show media-ports-summary
show active-calls -c
show active-calls
show active-session -c
show active-session
show media-stream-addresses
show media-stream-stats
show media-stream-stats
show media-stream srtp
show media-stream-client-sessions -c
show media-stream-client-sessions
show media-stream-server-sessions -c
show media-stream-server-sessions
show kernel-rule -v
show kernel-rule-stats
show kernel-rule-stats
show kernel-rule-stats
show kernel-rule-stats -v
show kernel-rule-stats -v'
show udp-counters
show udp-counters
show tcp-counters
show tcp-counters
show interface-details
show interface-details
shell sip
display scrolled
show locks -v
show locks -v
show locks -v
socket
show pool
```

```
exit
shell media
display scrolled
show locks -v
socket
exit
```

3. Run the following trace by substituting the correct target phone number for the one provided in the example:

```
NNOS-E> shell sip
SIP> trace target issue1.txt
trace issue1.txt> trace * error
trace issue1.txt> trace sip_traffic info
trace issue1.txt> trace scale* debug
trace issue1.txt> trace mstream* debug
trace issue1.txt> trace sdp debug
trace issue1.txt> trace krnl debug
trace issue1.txt> trace krnl_msg debug
trace issue1.txt> trace autonomous_ip debug
trace issue1.txt > exit
Do you want to save the settings for this target (y or n)? y
Do you want to start tracing to this target (y or n)? n
SIP> trace-filter enabled 1115551234
Start to trace based on user 1115551234
SIP> trace start issue1.txt
Leave this trace running until the required call has been made:
SIP> trace stop issue1.txt
SIP> trace-filter disabled
Disabling filtered tracing
SIP> exit
NNOS-E>
```

4. Display the call log associated with the call.

The following commands can be used to inspect the properties of the established session.

The **show active-calls** identifies the relevant call(s), and **show media-stream-stats** shows the number of packets received and sent for each call leg. These numbers should correlate with each other, and the numbers should increment when the command is run repeatedly.

```
NNOS-E> show active-calls

         session-id: 0x8c12c896f23a54a
```

```
                    from: "Welbourn2"
               <sip:2403645088@acmepacket.com>;tag=5d4cc104
                      to: "2403645087" <sip:2403645087@acmepacket.com>
                   state: B2B_CONNECTED
            previous-hop-ip: 172.27.21.58
            next-hop-domain: 192.168.77.179
                  duration: 549 seconds
          inbound-connection:
         outbound-connection:
                header-value:
             subject-to-CAC: true
                    contact: <sip:2403645088@172.27.21.58:50232>

NNOS-E> show media-stream-stats

session-id         stream   call-leg   address             rx-packets tx-packets
----------         ------   --------   -------             ---------- ----------
0x8c12c896f23a54a  1        1          172.30.3.164:24420  40774      40774
                            2          172.30.3.164:24616  40781      40781
```

The **show media-stream-addresses** displays the various IP and UDP ports used for media streams:

```
NNOS-E> show media-stream-addresses

session-id           stream     call-leg   type          origin       address
----------           ------     --------   ----          ------       -------
0x8c12c896f23a54a    1          1          peer-source   rtp          172.27.21.58:55974
                                           anchor-dest   media-port   172.30.3.164:24420
                                           anchor-source media-port   172.30.3.164:24616
                                           peer-dest     sdp          172.30.3.164:24462
                                2          peer-source   rtp          172.30.3.164:24462
                                           anchor-dest   media-port   172.30.3.164:24616
                                           anchor-source media-port   172.30.3.164:24420
                                             peer-dest     sdp          172.27.21.58:55974
```

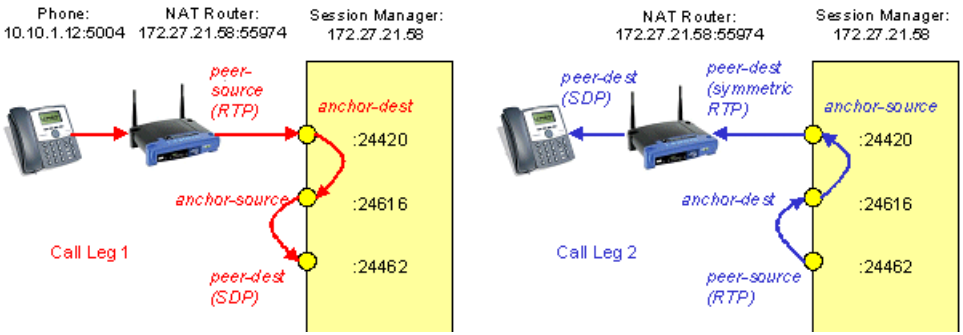
The output displays the following fields:

- *stream* — Identifies the component of the call. For voice-only calls, there is only one stream, whereas video calls have two streams, one for audio (index=1) and one for video (index=2).
- *call leg* — Identifies the two RTP sides of the call, where index=1 represents the outbound leg and index=2 the inbound leg.
- *type* — Indicates the role or the address.
- *origin* — Identifies how the address was determined.

  An origin of RTP indicates the source address of the media stream was used, and SDP indicates that AA-SBC used the address in the SIP dialog. The media-port indicates the ports that AA-SBC has allocated for anchoring the call.

  The other possible value of origin for peer-dest is symmetric-rtp, where the value was determined by using symmetric RTP rather than SDP; and the other possible value of origin for anchor-dest is near-end-nat, where AA-SBC is aware that it is behind a firewall performing network address translation.

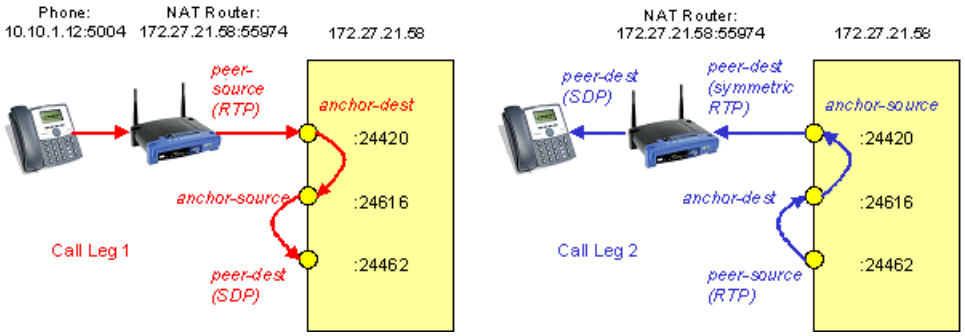
Note that the rows in the display show the progress of the call through AA-SBC, as illustrated in the image below. AA-SBC uses the apparent IP address/port for the source of the RTP (peer-source), allocates a media port (anchor-dest) to receive the RTP, allocates another port from which to send the RTP (anchor-source) and determines where to send the RTP (peer-dest).



If AA-SBC determines that the RTP comes from a different IP address than was signaled in SIP, then it knows that the endpoint is behind a NAT device. In this case it will send the return path media back to the same IP address and port as the outbound path, using symmetric RTP, provided that symmetric RTP has been enabled for media in the session configuration. In these cases the use of RTP is shown as follows:

```
NNOS-E> show media-stream-addresses

session-id          stream     call-leg   type          origin       address
----------          ------     --------   ----          ------       -------
0x8c12c896f23a54a   1          1          peer-source   rtp          172.27.21.58:55974
                                          anchor-dest   media-port   172.30.3.164:24420
                                          anchor-source media-port   172.30.3.164:24616
                                          peer-dest     sdp          172.30.3.164:24462
                               2          peer-source   rtp          172.30.3.164:24462
                                          anchor-dest   media-port   172.30.3.164:24616
                                            anchor-source media-port   172.30.3.164:24420
                                  peer-dest   symmetric-rtp 172.27.21.58:55974
                                            peer-dest     sdp          10.10.1.12:5004
```

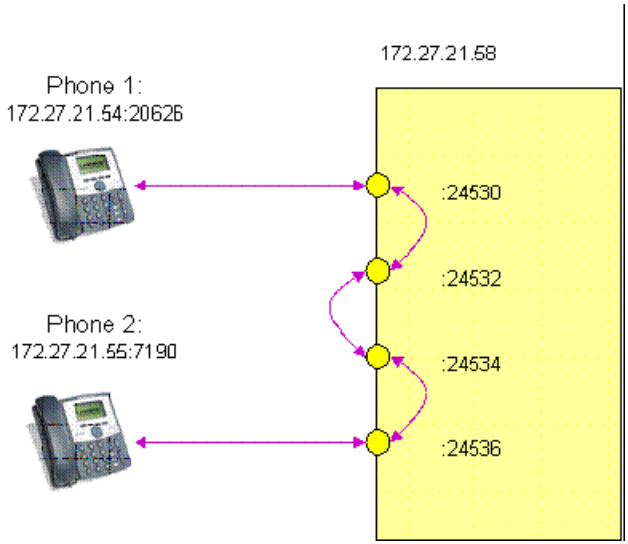
Notice that there are two *peer-dest* entries for the second leg, the first of which (indicating a higher priority) has an *origin* value of *symmetric-rtp*.



If AA-SBC determines that two separate calls are the two halves of the same call (which is the case when the calling and called party are connected to the same AA-SBC device), it will link the media ports for the two calls as shown below in the image below.

```
NNOS-E> show media-stream-addresses

session-id          stream     call-leg   type          origin       address
----------          ------     --------   ----          ------       -------
0x8c12cb789154806   1          1          peer-source   rtp          172.27.21.54:20626
                                          anchor-dest   media-port   172.30.3.164:24530
                                          anchor-source media-port   172.30.3.164:24532
                                          peer-dest     sdp          172.30.3.164:24534
                               2          peer-source   rtp          172.30.3.164:24534
                                          anchor-dest   media-port   172.30.3.164:24532
                                          anchor-source media-port   172.30.3.164:24530
                                          peer-dest     sdp          172.27.21.54:20626
```

```
0x8c12cb789160cdf    1          1          peer-source    rtp          172.30.3.164:24532
                                           anchor-dest    media-port   172.30.3.164:24534
                                           anchor-source  media-port   172.30.3.164:24536
                                           peer-dest      sdp          172.27.21.55:7190
                                2          peer-source    rtp          172.27.21.55:7190
                                           anchor-dest    media-port   172.30.3.164:24536
                                           anchor-source  media-port   172.30.3.164:24534
                                           peer-dest      sdp            172.30.3.164:24532
```



If AA-SBC determines that the two halves of the same call originate behind the same NAT device (because the public IP addresses of the called and calling parties are the same), it will direct the two endpoints to send media to each other directly using their private IP addresses, and will not anchor the media, provided that media anchoring has been set to *auto-anchor* in the session configuration.

In cases where the subscriber is using two layers of NAT, this releasing of the media can cause problems, and it will be necessary to anchor the call.

If AA-SBC is not anchoring media, the addresses displayed in **show media-stream-addresses** will be all zeros:

```
NNOS-E> show media-stream-addresses

session-id        stream    call-leg  type       origin     address
----------        ------    --------  ----       ------     -------
```

```
0x8c12c896f23a54a    1          1          peer-source    rtp          0.0.0.0
                                           anchor-dest    media-port   0.0.0.0
                                           anchor-source  media-port   0.0.0.0
                                           peer-dest      sdp          0.0.0.0
                                2          peer-source    rtp          0.0.0.0
                                           anchor-dest    media-port   0.0.0.0
                                           anchor-source  media-port   0.0.0.0
                                           peer-dest      sdp          0.0.0.0
```

If this is the case, check the configuration to see why the call is not being anchored. If media anchoring is set to *auto-anchor*, then check whether the subscriber is using double NAT, and if so, disable auto-anchoring for this subscriber.

If there are zeros in the *peer-source* line, AA-SBC has not received RTP packets on the indicated call leg. For example:

```
NNOS-E> show media-stream-addresses

session-id           stream     call-leg   type          origin       address
----------           ------     --------   ----          ------       -------
0x8c12c896f23a54a    1          1          peer-source   rtp          172.27.21.58:55974
                                           anchor-dest   media-port   172.30.3.164:24420
                                           anchor-source media-port   172.30.3.164:24616
                                           peer-dest     sdp          172.30.3.164:24462
                                 2          peer-source   rtp    0.0.0.0
                                           anchor-dest   media-port   172.30.3.164:24616
                                            anchor-source media-port   172.30.3.164:24420
                                            peer-dest     sdp          10.10.1.12:5004
```

In this case, AA-SBC expects to receive RTP packets on 172.30.3.164 port 24462, but has not received any packets. This may be due to routing or network issues that are preventing the phone from reaching AA-SBC, or that the phone is behind a NAT device and symmetric RTP is disabled.

If both *peer-source* lines are not all zeros, this indicates that AA-SBC is receiving RTP packets, but one or both phones is not receiving them.

Performance and capacity problems

If calls are being refused during busy periods, check for the following conditions:

1. Call emission control is not causing calls to be refused because of upstream server or link limitations, or that call admission control is not limiting calls due to ingress bandwidth limitations. Use the **show call-admission-control** command.

2. AA-SBC is not running out of media ports. Use the **show media-stream-counts** and **show media-ports-summary** to check on this. If the number of media ports in use is not consistent with the number of active calls, this may be an indication that media ports are not being released properly when calls are disconnected. Contact Technical Support if this is the case.
3. The number of concurrent calls has reached licensed limits. Use **show active-calls-summary** to count the number of active calls and compare this against the limits given in the configuration under **config features** object.

Software failures

If a check for software faults shows that there have been software problems, please consult Technical Support for instructions on how to investigate them and return the necessary diagnostics (event logs, dump files, etc).