Oracle® Diameter Signaling Controller

Essentials Guide

Release D-SCz2.2.0

Formerly Net-Net Diameter Director

September, 2015

ORACLE®

Table of Contents

Table of Contents iii

About This Guide. vi

Overview vi
Related Documentation vi
Release Features vii
Release Caveats viii
Technical Assistance ix
Revision History ix

1 Diameter Network Architecture. 1

Network Architecture 1

2 Diameter Director Configuration. 3

Diameter Director Elements 3
Diameter Director Interface. 3
Diameter Director Agent 6
Capabilities Exchange Messaging 10
Message Rate Constraints 14
Upstream Congestion Control 18
ToS Field Marking 21
Global Timers 22
Stream Control Transfer Protocol Overview. 22
Incompatible SCTP Association Signaling Workaround 40

3 Routing Diameter Messages 43

Routing. 43

Loop Detection 46
Grouped AVP Routing 47
Dynamic Routing 47
Policy Rejection 48
Diameter Director Group Recursive Routing 51
AVP-Based Local Routing Tables 52
Session Statefulness 56
Subscriber Statefulness 61
Subscriber-only Statefulness 64

4 Load Balancing & Redundancy 67

Diameter Director Groups 67
Active/Active Redundancy 72

5 Diameter Message Manipulations 79

Diameter Message Manipulations 79

6 Security 89

Anonymous Diameter Agent Blocking 89
Natively Securing Network Topology Information 90
IPSec Support 93
Internet Key Exchange (IKEv2) 94
Tunnel Management with the ACLI 113

7 Management Support 115

Palladion Mediation Engine 115
IPFIX 116
Communications Monitor Configuration 116
SNMP 118
SNMP Alarm 121
HDR and SNMP Statistics 121
What is HDR? 122
Configuring HDR via the ACLI 125
Starting and Stopping HDR using the ACLI 130
HDR Groups 131
Supported Commands for KPI Tracking 157

Session and Subscriber Statefulness XML File Maintenance. 160
Active/Active Redundancy Maintenance 160
Net-Net 7000 Hardware Platform Management 161

8 ACLI Reference and Debugging 163

Supporting Configurations. 163
ACLI Configuration Elements. 164
diameter-manipulation 164
diameter-manipulation > diameter-manip-rule 165
diameter-manipulation > diameter-manip-rule > avp-header-rule 166
diameter-director-config 167
diameter-director-interface 168
diameter-director-interface > diameter-director-applications 170
diameter-director-interface > diameter-director-ports 171
diameter-director-agent 171
diameter-director-agent > diameter-director-applications 173
diameter-director-constraints 174
diameter-director-constraints > message-rate-constraints 175
diameter-director-constraints > application-constraints 176
diameter-director-constraints > application-constraints > application-message-constraints . . 176
diameter-director-group. 178
diameter-director-group > diameter-director-applications 179
diameter-director-group > recursive-routing. 179
diameter-director-policy. 180
diameter-director-policy > policy-attribute 180
diameter-director-policy > policy-attributes > sub-avps. 182
Logging. 184
Net-Net Diameter Director Show Commands 184
SCTP Troubleshooting 197
Configuration Changes that Cause Diameter Connection Disconnects. 199

About This Guide

Overview

Diameter Signaling Controllers, or DSCs, are network elements that control Diameter signaling, enabling the seamless communication and control of policy information between network elements within EPC or IMS networks and across EPC network borders.

The Net-Net™ Diameter Director provides control for both the MME / SGSN to HSS and PCRF to PCRF signaling messages at this data border between Home PLMNs, Visited PLMNs, and the wholesale carriers that (optionally) connect the PLMNs together.

Audience

This guide is written for network administrators and those who configure network devices. It provides information related to the features, installation, start-up, operation, and maintenance of the Net-Net Diameter Director Essentials Guide. Only experienced and authorized personnel should perform installation, configuration, and maintenance tasks.

Who is Acme Packet?

Acme Packet, the leader in session delivery network solutions, enables the trusted, first-class delivery of next-generation voice, data and unified communications services and applications across IP networks. Our Net-Net product family fulfills demanding security, service assurance and regulatory requirements in service provider, enterprise and contact center networks. Based in Bedford, Massachusetts, Acme Packet designs and manufactures its products in the USA. For more information, visit www.acmepacket.com.

Related Documentation

The following table lists the members that comprise the documentation set for this release. Some of this documentation is common to multiple software versions. Use the S-Cx6.3 versions of this documentation:

Document Name	Document Description
Net-Net 17350 Hardware Installation Guide (400-0192-10)	Contains information about the components and installation of the Net-Net 17350 system.
Net-Net 7000 Hardware Installation Guide (400-0191-10)	Contains information about the components and installation of the Net-Net 7000 system.
Net-Net 4000 ACLI Configuration Guide (400-0061-00)	Contains information about the administration and software configuration of the Net-Net SBC.

Document Name	Document Description
Net-Net 4000 ACLI Reference Guide (400-0062-00)	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Net-Net 4000 Maintenance and Troubleshooting Guide (400-0063-00)	Contains information about Net-Net SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
Net-Net 4000 MIB Reference Guide (400-0010-00)	Contains information about Management Information Base (MIBs), Acme Packet's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Net-Net 4000 Accounting Guide (400-0015-00)	Contains information about the Net-Net SBC's accounting support, including details about RADIUS accounting.
Net-Net 4000 HDR Resource Guide (400-0141-00)	Contains information about the Net-Net SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Net-Net 4000 Administrative Security Essentials (400-0132-00)	Contains information about the Net-Net SBC's support for its Administrative Security license.
TECH NOTE: Net-Net ESD-VME and SD-VME Installation Guide (590-0014-03)	Contains information about installing Net-Net software on virtual machines.

Release Features

This section provides a table listing new functionality, as described in this guide, in Net-Net OS Release D-Cz2.2.0.

Feature	Description
Subscriber-only Caching	Allows the Net-Net DD to maintain state for application traffic that does not establish sessions, such as s6a.
Enhanced DNS Statistics	Provides additional detail about interaction between the Net-Net DD and a DNS infrastructure.
Routing via LRT	Allows the user to configure the Net-Net DD to route using a local routing table, composed of manual routes.

Feature	Description
Palladion Probe Support	Allows the Net-Net DD to act as a probe, providing traffic information to the Palladion mediation engine.
Upstream Congestion Control	Allows the Net-Net DD to detect congestion at agents and manage traffic rates to those agents.
On-demand Peering	Allows the Net-Net DD to peer with agents only when it needs to pass traffic to them.
DNS lookup via NAPTR and SRV records	Enhances the ability of the Net-Net DD to establish connectivity to agents via the DNS infrastructure.

Release Caveats

The following sections list caveats related to this release of the Net-Net Diameter Director.

IPSec-Related Caveats

- `outbound-fine-grained-sa-match` configuration
  - Issue - The kernel supporting IPSec on Net-Net 7000 and Net-Net 17000 devices only supports 1-1 SA matches.
  - Resolution - Do not use `outbound-fine-grained-sa-match` within your IPSec configuration on the Net-Net 7000 and Net-Net 17000
- Ping in transport mode
  - Issue - Ping is not supported across IPSec connections in transport mode.
  - Resolution - Do not ping from the Net-Net DD across IPSec connections in transport mode. Instead, ping from devices related to the connection.

System-Related Caveat

- SIP threads
  - Issue - SIP redundancy threads configuration can generate `acmePipeDev` errors at the console.
  - Resolution - If you see the error listed above at the console, ensure the SIP threads configuration in the `system-config > options` is set to `sip-threads=1`.

Performance-Related Caveat

- MTU size in conjunction with encryption
  - Issue - Encryption can cause packets to exceed the accepted MTU size.
  - Resolution - If you find your traffic is exceeding MTU size, change the default signaling-MTU on the applicable network interface. For example, if the expected MTU is 1500, configure the applicable network interface for an MTU between 1200 and 1300. This provides the headroom needed to accomodate the encryption.

Technical Assistance

If you need technical assistance with Acme Packet products, you can obtain it on-line by going to support.acmepacket.com. With your customer identification number and password, you can access Acme Packet's on-line resources 24 hours a day. If you do not have the information required to access the site, send an email to tac@acmepacket.com requesting a login.

In the event that you are experiencing a critical service outage and require live assistance, contact the Acme Packet Technical Assistance Center emergency hotline:

- From the United States, Canada, and Mexico call: 1 866 226 3758
- From all other locations, call: +1 781 756 6920

Please note that a valid support/service contract with Acme Packet is required to obtain technical assistance.

Customer Questions, Comments, or Suggestions

Acme Packet is committed to providing our customers with reliable documentation. If you have any questions, comments, or suggestions regarding our documentation, please contact your Acme Packet customer support representative directly or email support@acmepacket.com.

Contact Us

Acme Packet
100 Crosby Drive
Bedford, MA 01730
USA
t 781 328 4400
f 781 425 5077
http://www.acmepacket.com

Revision History

This section contains a revision history for this document.

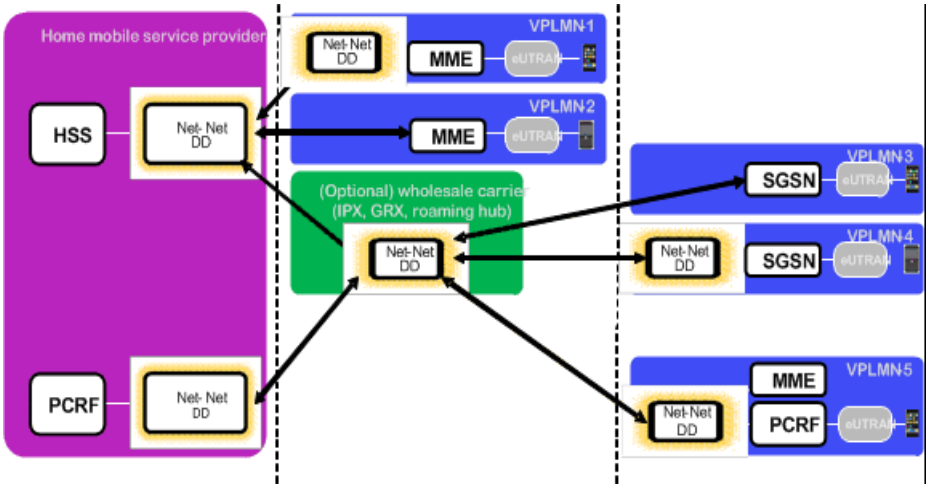
Date	Revision Number	Description
Sept 4, 2013	Revision 1.00	• Initial Release
Nov 22, 2013	Revision 1.10	• Adds diameter-manipulation rule naming restrictions.
May 30, 2014	Revision 1.20	• Clarifies that diameter director groups should be used for recursive routing implementations as opposed to LRTs. • Adds information indiating LRT usage as a next-policy.
September 1, 2015	Revision 1.30	• Corrects example configuration for AVP-Based Local Routing Tables.

1 Diameter Network Architecture

Network Architecture

The Net-Net Diameter Director participates at a border point between Public Land Mobile Networks (PLMNs, or Mobile Service Providers) which is created within the LTE data deployments when roaming.

When a roaming subscriber attempts to use the VPLMN (Visited PLMN/Visited MSP) Diameter signaling messages are exchanged between the Mobility Management Entity (MME) or SGSN (Serving GPRS Support Node) and Home Subscriber Server (HSS) in order to provide authentication and subscription authorization information about the roaming subscriber. When policy control dips are required to authorize data services, the visited PCRF (Policy and Charging Rules Function) contacts the home PCRF via Diameter signaling messages. The high level topology and relationship of these logical elements that are used to exchange signaling messages is shown below.



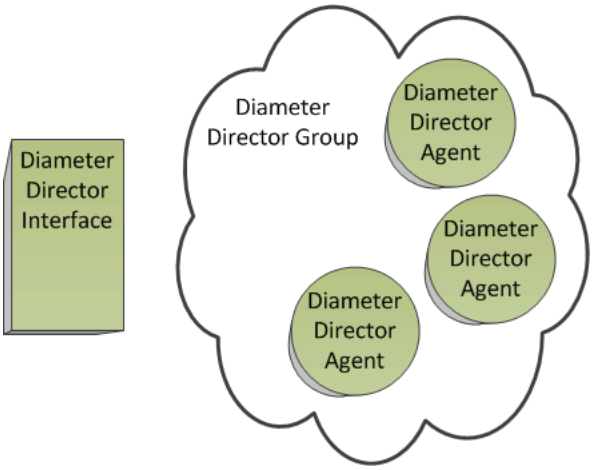
The Net-Net Diameter Director provides control for both the MME / SGSN to HSS and PCRF to PCRF signaling messages at this data border between Home PLMNs, Visited PLMNs, and the wholesale carriers that (optionally) connect the PLMNs together.

2 Diameter Director Configuration

Diameter Director Elements

As the Net-Net Diameter Director sits in a network, the two primary elements that must be configured on the system are the Diameter Director Interface and the Diameter Director Agent. In addition, a Diameter Director Group can be created as a virtual collection of Diameter Director Agents (see the "Diameter Director Groups"section).

The Diameter Director Interface is the Diameter application interface that runs on the Net-Net Diameter Director, while the Diameter Director Agent is the representation of remote Diameter agent.



Diameter Director Interface

The Diameter Director Interface is the Diameter application interface that runs on the Net-Net Diameter Director. Since there can only be one Diameter Director Interface active in a realm, it is defined by that realm, which is configured in the *realm-id* parameter.

When a message is received on a Diameter Director Interface, the Net-Net Diameter Director then determines how and where to route it. Each Diameter Director Interface is configured with a root Diameter Director Policy that sets the starting point of the routing process. See the"Routing Diameter Messages"chapter.

Supported Applications

Each Diameter Director Interface may be configured with a set of Diameter Application IDs and Vendor IDs. These values are used and sent in a capability exchange with external Diameter agents to determine the range of Diameter applications that each network element supports (See: "Capabilities Exchange Messaging").

The Application IDs configured at the Diameter Director Interface level are the least specific Application ID for the Diameter Director Interface to Diameter Director Agent connection. Thus if an Application ID is configured for either the Diameter Director

Group or Diameter Director Agent, they take precedence over the Application IDs configured on the Diameter Director Interface.

In addition, a list of supported vendor IDs can be configured in the root Diameter Director Interface. These configured values are included in the CER/CEA negotiation as Supported-Vendor-ID AVPs.

Origin Host AVP

The Origin-Host AVP (AVP Code 264) is present in all Diameter messages. This AVP identifies the endpoint that originated the Diameter message. When acting as a relay, the Net-Net Diameter Director will not modify this AVP.

The default Net-Net Diameter Director behavior sets the Origin-Host AVP as <Diameter director interface IP address>.<realm-id> in all outgoing messages. The value in the Origin-Host AVP is effectively ignored.

The Origin Host AVP contents can be configured by setting the **origin host identifier** parameter in the *diameter director interface* configuration element. The **origin host format** parameter indicates how to format the Origin Host AVP and accepts the following enumerated values:

- None—The default behavior is retained which sets the Origin-Host AVP as <interface-ip-addr>.<realm-id> in all outgoing messages.
- Identifier—The Origin-Host AVP in all outgoing message will be set to the value of the **origin-host-identifier** parameter.
- Identifier-with-realm—The Origin-Host AVP in all outgoing messages will be set to <origin-host-identifier>.<realm-id>.
- If the **origin-realm** parameter is configured, the Origin-Host AVP is created as: <origin-host-identifier>.<origin-realm>

Diameter Director Ports

Each Diameter Director Interface requires at least one Diameter Director Port subelement. The Diameter Director Port defines the actual IP address/port and transport protocol where a Diameter agent may connect to on a Net-Net Diameter Director and where messages are sent and received. To define this, the *address*, *port* and *transport protocol* are configured. No two Diameter director ports subelements may use the same IP address. In addition, admission control settings can be defined for each Diameter Director Port (see:"Anonymous Diameter Agent Blocking").

SCTP

The Net-Net Diameter Director supports SCTP as the transport protocol between a Diameter Director Interface and a Diameter agent. Set the *transport protocol* parameter to **sctp** within a *diameter director ports* subelement to use SCTP. The corresponding Diameter Director Agent must also have its *transport protocol* parameter set to **sctp** as well.

IPv6

The Net-Net Diameter Director supports IPv6. Please refer to the Net-Net 4000 ACLI Configuration Guide, System Configuration chapter, About Your Net-Net 3800/4500 and IPv6 section onward to learn about baseline IPv6 Support on the Net-Net SBC product, which is similarly applicable here.

Licensing

The Net-Net Diameter Director requires a “Diameter Director” license to unlock all relevent configuration elements, statistics. and let the Diameter Director task begin running. Please ensure that you have this license installed on your system.

> **Note:** If the license was installed after the Net-Net Diameter Director runtime image was loaded, you will need to reboot the system. The Diameter Director license must be installed in the system prior to boot time.

ACLI Instructions

To configure a Diameter Director Interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-interface a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-interface
   ACMEPACKET(diameter-director-interface)#
   ```

4. **state**—Set this parameter to **enabled** to activate this Diameter Director Interface.
5. **realm-id**—Enter the realm name where this Diameter Director Interface exists.
6. **description**—Enter a description of this Diameter Director Interface enclosed in quotes.
7. **routing-policy**—Enter the *name* of the root routing policy to use when a message is received on this Diameter Director Interface.

To configure the diameter director ports subelement:

8. Type **diameter-director-ports** and press <Enter> The system prompt changes to let you know that you can begin configuring individual parameters.
9. **address**—Enter an IP address in the realm where this Diameter Director Interface can send and receive messages.
10. **port**—You can override the default port (3868) by setting this to a value of your choosing.
11. **transport-protocol**—Set this parameter to either TCP or SCTP to indicate the transport-protocol. This Diameter director port subelement will now only use the indicated transport protocol.
12. **allow-anonymous**—See "Anonymous Diameter Agent Blocking".
13. Type **done** to save your work and continue. You may also add additional applications or Diameter director ports instances.
14. Type **exit**.

To configure Supported Vendor IDs and Application IDs and their Vendor IDs:

15. **supported-vendor-id**—Enter the Vendor ID values to include in the CER/CEA exchange.
16. Type **diameter-director-application** and press <Enter> The system prompt changes to let you know that you can begin configuring individual parameters.
17. **application-id**—Enter an Application ID that this Diameter Director Interface supports. (Enter 0xFFFFFFFF for relay).

18. **application-type**—Set this to **accounting** or **authentication** depending on the application type.
19. **vendor-id**—Enter the Vendor ID for this application ID. (Enter 0 for relay).
20. Type **done** to save your work and continue. You may also add additional applications.
21. Type **exit**.

To configure how the Origin Host AVP appears in CER/CEA negotiation:

22. **origin-host-identifier**—Enter a unique identifier to use in the origin host AVP to override the Net-Net Diameter Director's default method of creating an Origin Host AVP.
23. **origin-host-format**—Leave this parameter empty or set it to **identifier** or **identifier-with-realm** to indicate how to create the Origin Host AVP.
24. Save your work using the ACLI **done** command.

Diameter Director Agent

The Diameter Director Agent is the representation of a remote Diameter agent. For all Diameter agents that may connect to the Net-Net Diameter Director, a Diameter Director Agent must be created. Diameter Director Agents must be in the same realm as a Diameter Director Interface to communicate. If there is no Diameter Director Interface that a Diameter Director Agent can point to, no connection will be made.

Like a Diameter Director Interface, a Diameter Director Agent is configured with supported Application IDs to assert in a CER/CEA negotiation. The difference is that the group of Application IDs configured on an Diameter Director Agent supersede those configured on a Diameter Director Interface and Diameter Director Group. To inherit the Application IDs configured on the corresponding Diameter Director Interface or Diameter Director Group, do not configure a *diameter director application* subelement on a Diameter Director Agent.

A Diameter Director Agent is configured with a hostname that is used as a reference within the Net-Net Diameter Director, and primarily in the Diameter Director Group configuration element or a *next-hop* in a Policy Attribute in a Diameter Director Policy.

The *ip-address*, *port*, *realm-id* and *transport protocol* are standard configurations for an external agent. To use an FQDN via DNS for reaching a Diameter Director Agent, you may configure the *hostname* parameter and leave the *ip-address* parameter blank (supporting DNS services must also be configured - see DNS for Diameter Director Agent Hostname Resolution (163)). The *transport protocol* must match the t*ransport protocol* located in the *diameter director ports* subelement on the Diameter Director Interface where it connects. Available transport protocols for Diameter messaging are TCP and SCTP.

Connection Mode

While each possible Diameter Director Agent must be created as a *diameter director agent* configuration element, the *connection-mode* determines whether the Net-Net Diameter Director connects first to the Diameter Director Agent with the **outbound** value or waits for an incoming connection from the agent with the **inbound** value. Any time a Diameter connection is lost between the Net-Net Diameter Director and a Diameter agent with a *connection-mode* of **outgoing**, the Net-Net Diameter Director begins its process of reconnecting with the agent. See "Anonymous Diameter Agent Blocking"for additional information on the security aspects of this feature.

Inbound-dynamic-ip Mode

Some Diameter Director Agents, specifically those who obtain IP addresses from the Dynamic Host Configuration Protocol (DHCP), can go briefly out of service when their IP address leases expire. After lease expiration, DHCP clients may, or may not, obtain their previous IP address from the DHCP server.

Instead of relearning the routes for these Diameter Director Agents when they return to service with a possibly different IP address, the Net-Net Diameter Director provides the ability to establish that a known Diameter Director Agent is coming back online and to reuse the connection and learned routes associated with the prior agent instance.

This capability is specified by the **inbound-dynamic-ip** connection mode. With this mode enabled, the IP address provided by an inbound agent during the capabilities exchange is not considered definitive for admission control.

With inbound-dynamic-ip connection mode enabled, the Net-Net Diameter Director first attempts to match the incoming IP address with an existing agent. If the match fails, which would happen after the DHCP assignment of a new IP address, the Diameter Director checks for the existence of a Diameter Director Agent with an **inbound-dynamic-ip** connection mode on the corresponding network interface. If so, the Diameter Director proceeds directly to the capabilities exchange.

During the exchange, the Diameter Director attempts to match the orig-hostname contained in the CER message with hostname of a Diameter Director Agent with an **inbound-dynamic-ip** connection mode in the same network interface or realms. If a match is found, the connection and previously learned routes are retained and the capabilities exchange is successfully concluded.

On-Demand Peering

You can reduce the overhead required to maintain connectivity with specific agents by configuring the Net-Net Diameter Director to connect to the agent only when needed. To do this, you configure the agent's connection mode to `outbound-on-demand`. This mode contrasts with `outbound` mode in that `outbound-on-demand` does not attempt to reconnect upon disconnection.

Watchdog Messages

All Diameter Director Interfaces reply to Device Watchdog Requests (DWR) issued to them with a Device Watchdog Answers (DWA).

To configure the Net-Net Diameter Director to initiate DWRs toward a Diameter agent, the *watchdog-timer* parameter is configured in the *diameter director agent* configuration element. When set to a non-0 value, this parameter controls how often to send DWRs to a Diameter agent. The Net-Net Diameter Director's internal timer is reset every time it receives a Diameter message. If the watchdog timer expires due to inactivity on a connection, a DWR is sent to the remote agent. If no DWA or other message is received by the Net-Net Diameter Director before the *watchdog-timer* value expires, the socket is torn down.

See "Diameter Heartbeat (DWR/DWA)".

DNS Lookups to Resolve Agent Addresses

The Net-Net Diameter Director is capable of issuing both NAPTR and SRV lookup requests for resolution of diameter-director-agent locations and connectivity establishment. This functionality is compliant with RFC3588. This allows you to exclude ip address and protocol configuration on the Net-Net Diameter Director for an agent that

may change address or allow access via multiple protocols. Resolution of both IPv4 and IPv6 addressing is supported.

The DNS server can reside in any Net-Net Diameter Director realm. If the desired DNS server resides in a realm other than the one the `diameter-director-agent` resides, you configure the agent with its `dns-realm` parameter to specify the correct realm.

When the Net-Net Diameter Director needs to send requests to an agent, it may or may not need to issue NAPTR and SRV lookup requests. As such, it follows the procedure below to determine how to reach an agent:

1. Check to see if the agent has a configured ip-address.

   If there is an address, it proceeds to step 9.
   If not, it begins resolution procedures via DNS at step 2.

2. The Net-Net Diameter Director checks to ensure that the hostname is a resolvable FQDN.

   If the hostname is resolvable, it proceeds to step 3.
   If not, it triggers an alarm and terminates the process.

3. The Net-Net Diameter Director checks for a configured protocol.

   If there is a configured protocol, it proceeds to step 4.
   If not, it proceeds to step 5.

4. The Net-Net Diameter Director checks for a configured port.

   If there is a configured port, it proceeds to step 8.
   If not, it proceeds to step 7.

5. The Net-Net Diameter Director checks that the interface supports multiple protocols.

   If the interface supports multiple protocols, it proceeds to step 6.
   If not, it returns to step 4.

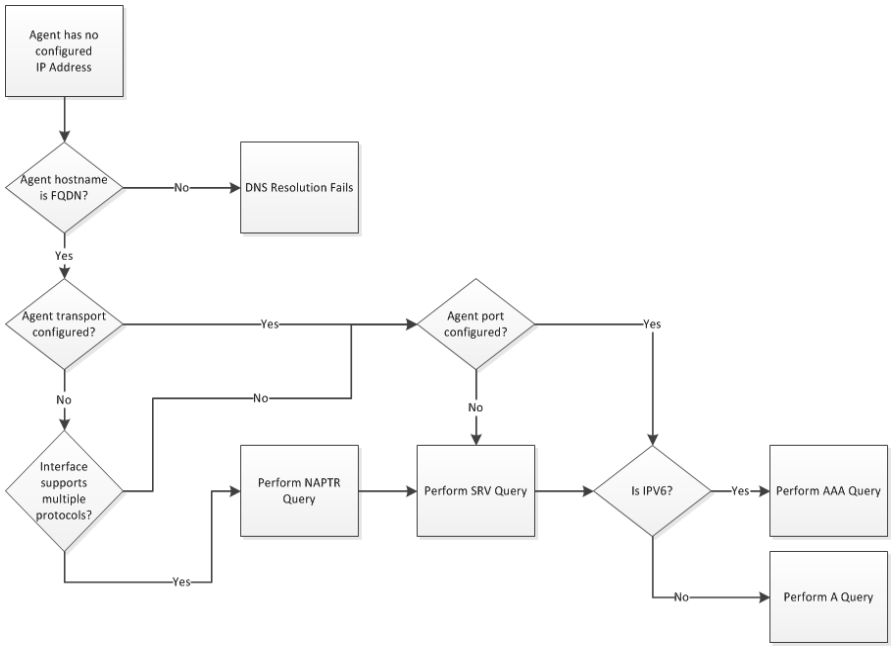
6. The Net-Net Diameter Director forwards a NAPTR query to the DNS server.

7. The Net-Net Diameter Director forwards an SRV query to the DNS server, having obtained protocol, port(s) used and port priority information via NAPTR or the port via the agent's configuration.

8. The Net-Net Diameter Director forwards either an A or an AAA record query, depending on whether v4 or v6 format is required, to the DNS server and obtains the correct ip address.

9. Using the address provided by the DNS server, the Net-Net Diameter Director forwards the messaging to the agent.

The diagram below depicts this process.



The `show diameter-director agent` command displays the active transport protocol and IP address for each configured agent. If the transport protocol or address change after a DNS query, the system includes the updated information in this commands output.

ACLI Instructions

To configure a Diameter Director Agent:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-agent a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-agent
   ACMEPACKET(diameter-director-agent)#
   ```

4. **state**—Set this parameter to **enabled** to activate this Diameter Director Agent.
5. **realm-id**—Enter the realm name where this Diameter Director Agent exists.
6. **description**—Enter a description of this Diameter Director Agent enclosed in quotes.
7. **hostname**—Hostname of the remote agent that the Net-Net Diameter Director is connecting to. The hostname can be in FQDN-style or IP Address format. This must be configured so that other configuration elements can reference this diameter-director-agent instance. For the Diameter Director Agent to connect to a FQDN configured here via DNS, leave the ip-address parameter empty. This is the key field.
8. **ip-address**—IP address of this Diameter Director Agent to which the Net-Net DD initiates socket connections. Leave empty to utilize DNS to contact the agent.

9. **port**—You can override the default port (3868) by setting this to a value of your choosing. You can leave this field empty, obtaining port number via NAPTR resolution.
10. **transport-protocol**—Set to TCP or SCTP to indicate the transport-protocol or retain the default of empty. When empty you can obtain protocol via NAPTR resolution.
11. `dns-realm`—Realm to which this agent issues SRV and NAPTR resolution requests if the target DNS server is not in the same realm as the agent. If the target DNS server is in the same realm as the agent, you can leave this field empty. The system only uses this field if the IP address is empty. This function also requires a resolvable diameter-director-agent hostname.
12. **connection-mode**—Retain the default value of outgoing or set it to **incoming** for the Net-Net Diameter Director to wait for this agent to initiate the Diameter connection. This can also be set to **inbound-dynamic-ip** when reconnecting agents are expected to use different IP addresses than before disconnection.

    In addition, you can reduce the overhead required to maintain connectivity with specific agents by configuring the Net-Net Diameter Director to connect to the agent only when needed. To do this, you configure the agent's connection mode to `outbound-on-demand`. This mode contrasts with `outbound` mode in that it does not attempt to reconnect upon disconnection.**watchdog-timer**—Retain the default value of 30 seconds or change it to another value in seconds.
13. Type **done** to save your work and continue.

To configure Supported Applications on this Diameter Director Agent:

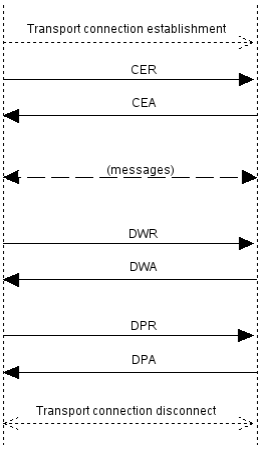
14. Type **diameter-director-application** and press <Enter> The system prompt changes to let you know that you can begin configuring individual parameters.
15. **application-id**—Enter an Application ID that this Diameter Director Agent supports. (Enter 0xFFFFFFFF for relay).
16. **application-type**—Set this to **accounting** or **authentication** depending on the application type.
17. **vendor-id**—Enter the Vendor ID for this application ID. (Enter 0 for relay).
18. Type **done** to save your work. You may also add additional applications.

Capabilities Exchange Messaging

The establishment of a new Diameter connection between the Net-Net Diameter Director and other Diameter agents is called the Diameter Capability Exchange as per RFC 3588. Capability exchanges facilitate the discovery of a peer agent's identity and its capabilities (protocol version number, supported Diameter applications, security mechanisms, etc.).

Functionally, there are two Capability Exchange scenarios. The first scenario is called the *Ingress Capabilities Exchange* and is when a first-hop Diameter agent discovers the Net-Net Diameter Director by sending it a CER message, and the Net-Net Diameter Director responds with a CEA.

The second scenario is called the *Egress Capabilities Exchange* and is when the Net-Net Diameter Director sends a CER message to its peer and expects a CEA in response.



Ingress Capabilities Exchange

A remote Diameter peer initiates a connection to the Net-Net Diameter Director's Diameter Director Interface listening socket in the Ingress Capabilities exchange. Once the connection is established, the Net-Net Diameter Director starts a 30 second timer while it waits to receive the CER message from the peer. If the timer expires before the CER is received, the Net-Net Diameter Director closes the transport layer connection.

If the Net-Net Diameter Director receives the CER before the 30 seconds elapses, it begins processing the message. If the received message is not a CER, then the Net-Net Diameter Director closes the connection. Upon receiving a valid CER, the Net-Net Diameter Director starts its work by extracting the Application IDs.

The Net-Net Diameter Director compares the CER's Application IDs to those configured on the *Diameter Director Interface > diameter director applications* sub element where the message was received.

If no common application values are found, then the Net-Net Diameter Director sends a CEA message to the peer with Result-Code AVP set to DIAMETER_NO_COMMON_APPLICATION and the connection is closed.

When the Net-Net Diameter Director confirms that the peer’s CER includes one or more common Application IDs, the Diameter connection is established. The Net-Net Diameter Director returns a CEA message including the common application IDs between CER and those configured on the system. The CEA message includes a Result-Code AVP with value of DIAMETER_SUCCESS.

Common Application IDs are maintained with this connection. If a subsequently received message does not include a stored Application ID, the Net-Net Diameter Director replies with a message containing a Result-Code AVP (268) of DIAMETER_APPLICATION_UNSUPPORTED - 3007. The answer message also contains Supported-Vendor-Id AVPs whose values are obtained from the *Diameter Director Interface* configuration.

Egress Capabilities Exchange

In the Egress Capabilities Exchange, the Net-Net Diameter Director initiates the connection to a remote peer defined by the *ip address* and *port* parameters in the *diameter director agent* configuration element.

Once a transport-layer connection has been established, the Net-Net Diameter Director sends a CER to the peer. The CER message contains all of the Application-Id AVPs specified in the *diameter director application* configuration elements. If 3 *diameter director application* configuration elements are configured, then all 3 will be sent to the Diameter agent.

If no *diameter director applications* are configured, the Net-Net Diameter Director uses the applications configured for the associated *diameter director group*. If there is no associated Diameter Director Group or if the Diameter Director Group itself has no applications configured, then the applications configured in the Diameter Director Interface are sent in the CER. The set of applications sent in the CER message will be cached. The CER message also contains Supported-Vendor-Id AVPs whose values are configured in the Diameter Director Interface. If the *supported-vendor-ids* list in the Diameter Director Interface configuration is empty, then no Supported-Vendor-Id AVPs will be sent in the CER.

After sending the initial CER, the Net-Net Diameter Director waits 30 seconds to receive the CEA message. This is known as message receive timer - T1. If T1 expires while waiting for the CEA, the Net-Net Diameter Director closes the connection to the Diameter agent.

If a message is received before T1 timer expires, the Net-Net Diameter Director begins parsing and processing the Diameter message. As per RFC 3588, any Diameter message other than a CEA is received during the capability exchange, the Net-Net Diameter Director closes the connection; any other Diameter message at this point is unacceptable.

Diameter Heartbeat (DWR/DWA)

Device-Watchdog-Request (DWR) and Device-Watchdog-Answer (DWA) messages are used to detect transport failures at the application layer between the Net-Net Diameter Director and a Diameter agent. The request/answer message pair forms a heartbeat mechanism that can indicate if the answering side is not reachable.

Diameter Director Interfaces always respond to a DWR by replying with a DWA message. Diameter Director Interfaces do not initiate DWRs (logically, only agents configured as Diameter Director Agents can initiate DWRs). The Origin-Host AVP (264) and the Origin-Realm AVP (296) in the DWA issued from the Net-Net Diameter Director will be values supplied in the Origin-Host AVP (264) and Origin-Realm AVP (296) in the CEA message.

The Net-Net Diameter Director establishes a DWR/DWA heartbeat between itself and configured Diameter Director Agents. The heartbeat will timeout if the Net-Net Diameter Director does not receive either a Diameter request from the Diameter Director Agent or a DWA in the timeout period. The default value is 30 seconds as suggested by Authentication, Authorization and Accounting (AAA) Transport Profile RFC 2539. The timeout value can also be configured by setting the *watchdog timer* parameter in the *diameter director agent*. Setting the watchdog timer to 0 disables the heartbeat mechanism.

If the watchdog timer expires, the Diameter Director Agent sends DWR to the remote agent. If no DWA is received or other message is received on that connection from the remote agent inside the watchdog timer value, the socket is torn down and a CER is sent toward the agent to reestablish the connection.

Upon detection of a transport failure during the capabilities exchange process, DWRs will not be sent to an alternate peer.

Disconnect Peer Messaging

The Net-Net Diameter Director supports Disconnect Peer Messaging DPR alerts to disconnect down to the transport layer. When the Net-Net Diameter Director receives a DPR, it considers the request and the type of socket where the message was received.

If the DPR was received on a forked socket; a listening socket defined by the *diameter director interface*, the Net-Net Diameter Director replies with a success code in the DPA and closes the socket. The Diameter Director Interface no longer responds to or forwards messages on this socket. A new Diameter connection must be created again with the specific peer that sent the DPR.

If the DPR was received on a Net-Net Diameter Director-initiated connection to a configured Diameter Director Agent, the Net-Net Diameter Director considers the Disconnect-Cause-AVP (273) value. If it is set to REBOOTING (0) or BUSY (1), the Net-Net Diameter Director sends a DPA indicating success and sets the connection out of service. The Net-Net Diameter Director then waits for the peer to tear down the socket. After successful socket teardown, the Net-Net Diameter Director attempts to re-establish a connection to the peer.

If the Disconnect-Cause-AVP (273) is set to DO_NOT_WANT_TO_TALK_TO_YOU (2), the Net-Net Diameter Director sends a DPA indicating success and sets the connection out of service. The Net-Net Diameter Director will then wait for the peer to tear down the socket as defined in section 5.4 of RFC 3588 Diameter Base Protocol. Upon successful teardown of the socket under these conditions the Net-Net Diameter Director will not try and re-establish a connection to this peer.

Message Type Verification

The Net-Net Diameter Director accepts all known and unknown requests, regardless of the Application suite they belong to. The Net-Net Diameter Director attempts to proxy the messages based on the next hop that the Net-Net Diameter Director routing engine determines. Request messages are not filtered (or rejected) if they are not supported by the message's Application ID.

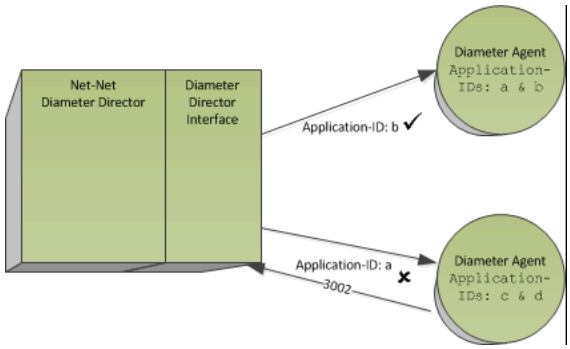
Relay Mode

The Net-Net Diameter Director can operate a Diameter Director Interface in relay mode. This means that it is open to accept and proxy messages of any Application ID value. To run a Diameter Director Interface in relay mode, set the *application-id* value to **0xffffffff** and the *vendor-id* to **0**. Relay mode is the only mode that will take a hex value for the application ID.

If a Diameter agent connects to a Diameter Director Interface running in Relay mode and the agent advertises that it can support 2 Application IDs in it’s CER/CEA, the Diameter Director Interface will set that connection to use the two Application IDs advertised. After the CER/CEA negotiation, the Net-Net Diameter Director will still accept Diameter messages of any Application type. It will only send messages of the 2 Application IDs to that agent.

If a different Diameter agent connects to the same Diameter Director Interface and advertises 2 additional application IDs, the Net-Net Diameter Director will accept both

since it is configured as a relay. Again, all outbound messages to that Diameter interface must match either of the two advertised Application IDs.



Message Rate Constraints

You can limit the number of Diameter messages received by and sent from the Net-Net Diameter Director in order to provide highly granular traffic shaping.

Diameter message rate constraints are configured in messages per seconds. This feature is configured in the *diameter director constraints* configuration element. A unique constraints profile can then be applied to Diameter Director Interfaces and Diameter Director Agents.

Message Rate Shaping

Rate shaping is configured through two types of parameters: rate window and message count. The rate window specifies the length of time in seconds in which message rate constraints are counted. The message count specifies the maximum number of messages that may be sent or received in the configured rate window. Once the message counter value has been reached before the current window ends, the constraint is considered exceeded.

There are two types of constraints, burst and sustain rates. The burst rate is meant to throttle sudden burst of Diameter message sent and/or received by a Diameter Director Interface or Diameter Director Agent within the configured burst window time. The sustained rate is meant to maintain traffic to and from a Diameter Director Interface or Diameter Director Agent at the configured rate within an extended, configured sustained window. As such, the sustain rate window value must be higher than or equal to the burst rate window.

Message Rejection

When a message constraint applied to either a Diameter Director Agent or Diameter Director Interface is exceeded, that element is set to Constraints Exceeded status.

The Net-Net Diameter Director can then take one of two actions: gracefully reject Diameter message with an error message returned to the sender or silently drop the Diameter message.

When set to **REJECT** messages upon constraints being exceeded, the default behavior, the Net-Net Diameter Director sends an error message to the network element that initiated that message. You can configure the numeric value returned to the message's orginator in a Result-Code AVP by configuring the *result code* parameter, the default being 3004.

When set to **DROP** messages upon constraints being exceeded, the Net-Net Diameter Director drops all messages to or from the element in the Constraints Exceeded state without notifying any external element.

Some Diameter agents, upon receiving an error message signifying a reject, attempt to retransmit the message that was rejected, thus creating more traffic and exacerbating the existing overload condition. Setting the constraints exceeded action to **DROP** mitigates this problem. Further, dropping messages ultimately triggers a timeout and provides a longer duration of time before the in-bound or out-bound agent attempts to re-transmit the message.

Dropped and Rejected messages are counted and displayed in the **show diameter-director agent**, **show diameter-director interface**, and **show diameter-director errors** show commands.

Once the constraints are exceeded, the Net-Net Diameter Director begins counting the **time to resume** period. This period represents the length of time while new messages are being rejected or dropped. After the **time to resume** counter expires, the Net-Net Diameter Director is set to an in-service state and accepts messages again. Message counts begin and are applied to the existing burst and sustain windows. Rate windows continue starting and stopping irrespective of **time to resume** period expirations.

Global Constraints

Global message rate constraints are configured in the *diameter director constraints* configuration element. They are applied to a Diameter Director Agent or Diameter Director Interface's **constraint name** parameter.

Bursty Traffic Throttling

You can create separate inbound and outbound maximum burst rates. First configure a **burst rate window** in seconds. Then configure the **max inbound burst rate** and **max outbound burst rate** parameters for the traffic you wish to constraint. In addition you can set an overall maximum burst rate with the **max burst rate** parameter.

Regardless of the inbound or outbound burst rates' headroom before exceeding the message constraints, if the total in/out traffic exceeds the **max burst rate** (when configured), the Diameter Director Agent or Diameter Director Interface will be set to Constraints Exceeded status and thus taken out of service.

Sustained Traffic Throttling

You can create separate inbound and outbound maximum sustained traffic rates. First configure a **sustain rate window** in seconds. Then configure the **max inbound sustain rate** and **max outbound sustain rate** parameters. In addition you can set an overall sustain rate with the **max sustain rate** parameter.

Regardless of the inbound or outbound sustained traffic rates' headroom before exceeding the message constraints, if the total in/out traffic exceeds the **max sustained rate** (when configured), the Diameter Director Agent or Diameter Director Interface will be set to Constraints Exceeded status and thus taken out of service.

Per Message Constraints

The Net-Net Diameter Director can also enforce message rate constraints on specific Diameter messages. Within the *diameter director constraints* configuration element is the *message rate constraints* sub element. Here you configure the message type with the **command** parameter. Then you can apply inbound and outbound burst and sustained constraints with the **max inbound burst rate**, **max outbound burst rate**, **max inbound sustain rate**, and **max outbound sustain rate** parameters. These values are computed

within the burst and sustained rates configured in the parent *diameter director constraints* configuration element.

Regardless of any configured message-specific constraint, exceeding an overall constraint, configured at the top Net-Net Diameter Director constraints level puts the Net-Net Diameter Director element into a constraints exceeded state.

Valid message types you can individually constrain are:

other	notify	server-assignment
update-location	credit-control	location-info
cancel-location	auth-auth	multimedia-auth
authentication-information	re-auth	registration-termination
insert-subscriber-data	session-termination	push-profile
delete-subscriber-data	abort-session	profile-update
purge-ue	accounting	subscribe-notification
reset	user-authorization	push-notification

The "other" message type is used to count the messages not captured by the supported interfaces. If a new, unsupported interface has new messages "bar", "abc", "pqr", the Net-Net Diameter Director counts all of the new messages as "other".

ACLI Instructions

Global Constraints

To configure global Diameter Director Constraints:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-constraints a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-constraints
   ACMEPACKET(diameter-director-constraints)#
   ```

4. **name**—Enter a name for this diameter-director-constraints configuration element. You will reference this value from a Diameter Director Interface or Diameter Director Agent.
5. **action**—Enter the action to take when this constraint profile sets the referencing element to Constraints Exceeded. Setting this parameter to **REJECT** sends an error message as a reply to network element that initiated that message. Setting this parameter to **DROP** silently drops the message without any error response.
6. **state**—Set this parameter to enabled to enable this configuration element.
7. **max-burst-rate**—Enter the maximum number of messages that can pass through the system in the burst rate window before setting the element to Constraints Exceeded.
8. **max-inbound-burst-rate**—Enter the maximum number of inbound messages received by the referencing element within the burst rate window before setting the element to Constraints Exceeded.
9. **max-outbound-burst-rate**—Enter the maximum number of outbound messages forwarded from the referencing element within the burst rate window before setting the element to Constraints Exceeded.

10. **burst-rate-window**—Enter the number of seconds during which to count messages toward a maximum burst rate.
11. **max-sustain-rate**—Enter the maximum number of messages that can pass through the system in the sustained rate window before setting the element to Constraints Exceeded.
12. **max-inbound-sustain-rate**—Enter the maximum number of inbound messages received by the referencing element within the sustained rate window before setting the element to Constraints Exceeded.
13. **max-outbound-sustain-rate**—Enter the maximum number of outbound messages forwarded from the referencing element within the sustained rate window before setting the element to Constraints Exceeded.
14. **sustain-rate-window**—Enter the number of seconds during which to count messages toward a maximum sustained rate.
15. **time-to-resume**—Enter the number of seconds that the referencing element stays in Constraints Exceeded state and rejects messages before it returns to service.
16. **result-code**—Leave this value at its default of 3004 or enter your own numeric value to return to the originating element in case of a rejected message.
17. Type **done** when finished.

Per-message Constraints

To enter per-message rate constraints:

18. Enter the message-rate-constraints sub element by typing message-rate-constraints and press <Enter>.

```
ACMESYSTEM(diameter-director-constraints)# message-rate-constraints
ACMESYSTEM(message-rate-constraints)#
```

19. **command**—Enter the message type you are entering specific constraints upon. The list of valid commands is listed in the Per Message Constraints (15) section.
20. **max-inbound-burst-rate**—Enter the maximum number of inbound messages at the burst rate for this message type.
21. **max-outbound-burst-rate**—Enter the maximum number of outbound messages at the burst rate for this message type.
22. **max-inbound-sustain-rate**—Enter the maximum number of inbound messages at the sustained rate for this message type.
23. **max-outbound-sustain-rate**—Enter the maximum number of outbound messages at the sustained rate for this message type.
24. Type **done** when finished.

    You may create additional message-rate-constraints configuration elements.

Applying Traffic Constraints

To apply Diameter Director Constraints to an existing Diameter Director Agent:

1. Type exit twice to return to the session router path.

```
ACMESYSTEM(message-rate-constraints)# exit
ACMESYSTEM(diameter-director-constraints)# exit
```

2. Type **diameter-director-interface** and **select** an existing configuration element.

```
ACMESYSTEM(session-router)# diameter-director-interface
ACMESYSTEM(diameter-director-interface)# select
<realm-id>:
1: realm01
```

```
selection: 1
```

3. **constraint-name**—Enter the name of an existing diameter-director-constraints configuration element.
4. Type **done** when finished.

   A similar process is used for applying a diameter director constraint to a Diameter Director Agent.

Upstream Congestion Control

The Net-Net Diameter Director can detect that upstream devices are congested and throttle traffic to those devices. Upstream devices can provide clues about the extent to which they are congested by sending responses such as DIAMETER_TOO_BUSY, or by simply failing to respond, causing timeouts. When these conditions become evident, the Net-Net Diameter Director can intelligently reduce the transaction rate sent towards the device. It also provides a means of resuming traffic rates when the upstream device is no longer congested.

In addition, the Net-Net Diameter Director can apply differentiated rate-limiting and prioritization based on Application-ID, Command-Code and configurable AVPs for upstream stations in congested state.

You configure how you want the Net-Net Diameter Director to handle congested upstream devices by configuring:

- `diameter-director-constraints`
  - `application-constraints`
  - `application-message-constraints`
- `congestion-control-policy`

You can apply congestion control based on interface and/or agent, with agent configuration taking precedence.

Congestion Detection

You configure detection by setting `congestion-control-policy` elements and applying them to interfaces and/or agents. The Net-Net Diameter Director identifies an element as congested when one of the following two conditions are true:

- The number of error result-codes has exceeded the congestion-window's allow-threshold value.
- The number of transaction-timeouts has exceeded the congestion-window's transaction-timeout-threshold value.

When these conditions apply, the Net-Net Diameter Director marks the element as congested. It maintains the current connection, monitoring status and forwarding transactions based on the congestion policy.

Congestion Action

The Net-Net Diameter Director takes the actions that you configure for congested elements. Actions inlcude:

- drop - The Net-Net Diameter Director does not respond to requests directed towards the congested element.

- reject - The Net-Net Diameter Director sends the configured response to devices that send requests to the congested element. Messages sent include the configured result-code and/or the experimental-result-code.
- constraints - The Net-Net Diameter Director uses the configured congestion-contraints profile to shape the traffic stream.

Note that "reject" is the default action, and is taken if the object is configured for "constraints", but no constraints profile is defined.

Also note that the application name you configure for an application constraint must correspond to an application name configured in the state machine XML.

Congestion Recovery

The congestion policy includes a `time-to-resume` parameter that defines the window within which the Net-Net Diameter Director keeps the affected element labeled as congested. After this timer expires, the Net-Net Diameter Director refers to the current traffic conditions to determine whether the element is still congested. Conditions that indicate the element is no longer congested include:

- The number of error result-codes has fallen back below the congestion-window's allow-threshold value.
- The number of transaction-timeouts has has fallen back below the congestion-window's transaction-timeout-threshold value.

Prioritizing Traffic and Message Constraints

The Net-Net Diameter Director allows you to prioritize traffic for more reliable delivery during congested periods. Prioritization applies when you configure the congestion action to "constraints". Traffic that you can prioritize for this purpose includes:

- Application type
- Message type, based on command code
- Message type, based on AVP

You prioritize traffic via constraint configuration.

Application identification is dependent on the state-machine XML definition. Each type of state is supported, including stateful, stateless, subscriber and subscriber-only stateful, with the application name in the XML matching the XML in your constraint configuration.

To prioritize message types, you extend upon the application constraint configuration with application-message constraints.

Configuring Upstream Congestion Control

This section shows you how to configure upstream congestion control on the Net-Net Diameter Director. Constraint configuration is optional. Steps include:

1. Configure diameter-director-constraints.

   1a. Configure application constraints.

   1b. Configure specific application-message constraints.

2. Configure congestion-control-policy

   2a. Specify the action.

   2b. If the action is constraint, configure the policy to refer to a specific diameter-director-constraint.

3. Apply congestion-control-policy to an interface or diameter-director agent.

ACLI Instructions

To configure the Net-Net Diameter Director with a congestion-control-policy:

1. In Superuser mode, type configure terminal and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **congestion-control-policy** and press <Enter>.

```
ACMEPACKET(session-router)# congestion-control-policy
ACMEPACKET(congestion-control-policy)#
```

4. Type **name,** followed by the name you have chosen, and press <Enter>.

```
ACMEPACKET(congestion-control-policy)#name policy1
```

5. Type **result-codes**, followed by the applicable codes, and press <Enter>.

```
ACMEPACKET(congestion-control-policy)#result-codes (please give
examples)
```

6. Type **result-codes**, followed by the applicable codes, and press <Enter>.

```
ACMEPACKET(congestion-control-policy)#experimental-result-codes
(please give examples)
```

7. Type **allow-threshold**, followed by your setting, and press <Enter>.

```
ACMEPACKET(congestion-control-policy)#allow-threshold 5000
```

8. Type **congestion-action**, followed by your setting, and press <Enter>.

```
ACMEPACKET(congestion-control-policy)#congestion-action constraints
```

Other configuration to this element is optional.

To configure the Net-Net Diameter Director with per-applications constraints:

1. In Superuser mode, type configure terminal and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **congestion-control-policy** and press <Enter>.

```
ACMEPACKET(session-router)# diameter-director-constraints
ACMEPACKET(diameter-director-constraints)#
```

4. Type **application-rate-constraints** and press <Enter>.

```
ACMEPACKET(application-rate-constraints)#
```

5. Type **name,** followed by the name you have chosen, and press <Enter>.

```
ACMEPACKET(application-rate-constraints)#name app-rate-constraints1
```

Other configuration to this element is optional.

To configure the Net-Net Diameter Director with per-message constraints:

1. In Superuser mode, type configure terminal and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type session-router and press <Enter>.

```
ACMEPACKET(configure)# session-router
```

3. Type **congestion-control-policy** and press <Enter>.

```
ACMEPACKET(session-router)# diameter-director-constraints
ACMEPACKET(diameter-director-constraints)#
```

4. Type **application-rate-constraints** to enter this sub-element and press <Enter>.

```
ACMEPACKET(application-rate-constraints)#
```

5. Type **application-message-constraints** to enter this sub-element and press <Enter>.

```
ACMEPACKET(application-message-constraints)#
```

6. Type **name,** followed by the name you have chosen, and press <Enter>.

```
ACMEPACKET(application-message-constraints)#name app-message-
constraints1
```

Other configuration to this element is optional.

ToS Field Marking

You can configure the byte-sized ToS field in an IP header in an ad-hoc fashion to support any and all differentiated services in your network. Upstream devices use these markings to classify traffic in order to determine the priority level of treatment it will receive. This feature manipulates the IP field on egress as traffic is sent via a Diameter Director Interface, Diameter Director Group, or Diameter Director Agent object with higher priority objects overwriting lower prioritied configurations.

Configuration is set with two hexadecimal numerals with the first numeral representing the first nibble of the ToS field and the second numeral representing the second nibble of the ToS field. For example, when the tos-value parameter is configured as 0x7B, the 7 and the B are hexadecimal numerals that correspond to the first and second nibbles in the 8-bit field. Since:

- hexadecimal 7 = binary 0111
- hexadecimal B = binary 1011

Then configuring 0x7B in the tos-value parameter would write the ToS field in an IP packet as: 01111011

ACLI Instructions

To configure a ToS marking on a Diameter Director Interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the session-router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **diameter-director-interface** and **select** an existing configuration element.

```
ACMESYSTEM(session-router)# diameter-director-interface
ACMESYSTEM(diameter-director-interface)# select
<realm-id>:
1: realm01

selection: 1
```

4. **tos-value** — Enter the value to mark the 8-byte ToS field with. This parameter is configured as 2 hexadecimal numerals prepended with `0x`. Valid values are 0x00 - 0xFF

5. Type **done** when finished.

   A similar process is used for applying ToS marking at the Diameter Director Agent or Diameter Director Group level.

Global Timers

Transaction Expiration Timer

You can set a global transaction expiration timer on the Net-Net Diameter Director. This timer expires after the Net-Net Diameter Director does not receive the response to a request it forwarded, within the timer's duration. A configurable transaction timer is useful to network operators to avoid race conditions and timeouts among different supported devices.

As a well-behaved network element, Net-Net Diameter Director follows architectural considerations similar to SIP where the Net-Net Diameter Director's server transaction (that receives the request) timeout is set to a value greater than the Net-Net Diameter Director's client transaction (that forwards the request). To achieve this, the Net-Net Diameter Director's server side transaction timeout is set to 125% of the configured client-side transaction. This ensures that the Net-Net Diameter Director allows the client transactions to expire first with a reasonable amount of time for the server transaction to expire in a non-error condition.

The transaction expiration timer is configured with the **trans-exp-timer** parameter in the *diameter director config* configuration element. The default value is 15 seconds. You may not set this timer to 0 (thereby disabling the timer). Acme packet recommends to use a value of 6 seconds or greater per the smallest value DWR/DWA timer stated in RFC 3588.

Stream Control Transfer Protocol Overview

The Stream Control Transmission Protocol (SCTP) was originally designed by the Signaling Transport (SIGTRAN) group of IETF for Signaling System 7 (SS7) transport over IP-based networks. It is a reliable transport protocol operating on top of an unreliable connectionless service, such as IP. It provides acknowledged,
error-free, non-duplicated transfer of messages through the use of checksums, sequence numbers, and selective retransmission mechanism.

SCTP is designed to allow applications, represented as endpoints, communicate in a reliable manner, and so is similar to TCP. In fact, it has inherited much of its behavior from TCP, such as *association* (an SCTP peer-to-peer connection) setup, congestion control and packet-loss detection algorithms. Data delivery, however, is significantly different. SCTP delivers discrete application messages within multiple logical streams within the context of a single association. This approach to data delivery is more flexible than the single byte-stream used by TCP, as messages can be ordered, unordered or even unreliable within the same association.

Support is compliant with RFC 4960, *Stream Control Transmission Protocol*.

SCTP Packets

SCTP packets consist of a common header and one or more *chunks*, each of which serves a specific purpose.

DATA chunk — carries user data

INIT chunk — initiates an association between SCTP endpoints

INIT ACK chunk — acknowledges association establishment

SACK chunk — acknowledges received DATA chunks and informs the peer endpoint of gaps in the received subsequences of DATA chunks

HEARTBEAT chunk — tests the leachability of an SCTP endpoint

HEARTBEAT ACK chunk — acknowledges reception of a HEARTBEAT chunk

ABORT chunk — forces an immediate close of an association

SHUTDOWN chunk — initiates a graceful close of an association

SHUTDOWN ACK chunk — acknowledges reception of a SHUTDOWN chunk

ERROR chunk — reports various error conditions

COOKIE ECHO chunk — used during the association establishment process

COOKIE ACK chunk — acknowledges reception of a COOKIE ECHO chunk

SHUTDOWN COMPLETE chunk — completes a graceful association close

SCTP Terminology

This section defines some terms commonly found in SCTP standards and documentation.

SCTP Association

is a connection between SCTP endpoints. An SCTP association is uniquely identified by the transport addresses used by the endpoints in the association. An SCTP association can be represented as a pair of SCTP endpoints, for example, assoc = { [IPv4Addr : PORT1], [IPv4Addr1, IPv4Addr2: PORT2]}.

Only one association can be established between any two SCTP endpoints.

SCTP Endpoint

is a sender or receiver of SCTP packets. An SCTP endpoint may have one or more IP address but it always has one and only one SCTP port number. An SCTP endpoint can be represented as a list of SCTP transport addresses with the same port, for example, endpoint = [IPv6Addr, IPv6Addr: PORT].

An SCTP endpoint may have multiple associations.

SCTP Path

is the route taken by the SCTP packets sent by one SCTP endpoint to a specific destination transport address or its peer SCTP endpoint. Sending to different destination transport addresses does not necessarily guarantee separate routes.

SCTP Primary Path

is the default destination source address, the IPv4 or IPv6 address of the association initiator. For retransmissions however, another active path may be selected, if one is available.

SCTP Stream

is a unidirectional logical channel established between two associated SCTP endpoints. SCTP distinguishes different streams of messages within one SCTP association. SCTP makes no correlation between an inbound and outbound stream.

SCTP Transport Address

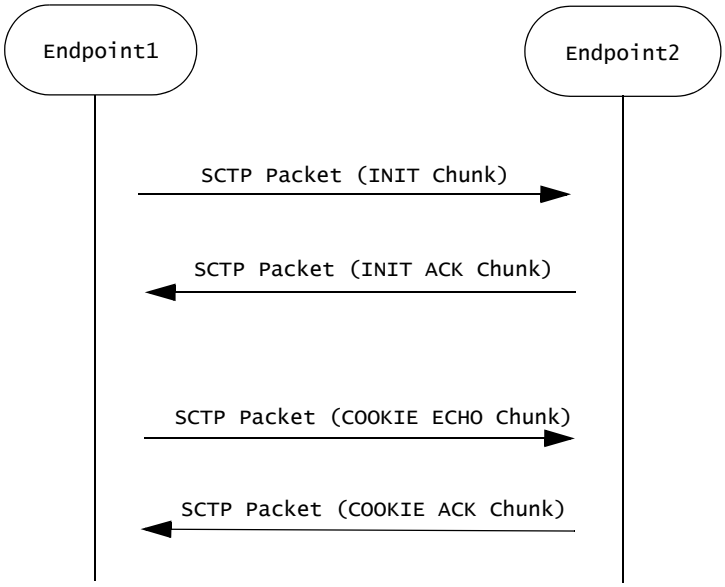
is the combination of an SCTP port and an IP address. For the current release, the IP address portion of an SCTP Transport Address must be a routable, unicast IPv4 or IPv6 address.

An SCTP transport address binds to a single SCTP endpoint.

SCTP Message Flow

Before peer SCTP users (commonly referred to as endpoints) can send data to each other, an association (an SCTP connection) must be established between the endpoints. During the association establishment process a cookie mechanism is employed to provide protection against security attacks. The following figure shows a sample SCTP association establishment message flow.

Endpoint1 initiates the association by sending Endpoint2 an SCTP packet that contains an INIT chunk, which can include one or more IP addresses used by the initiating endpoint. Endpoint2 acknowledges the initiation of an SCTP association with an SCTP packet that contains an INIT_ACK chunk. This chunk can also include one or more IP addresses at used by the responding endpoint.



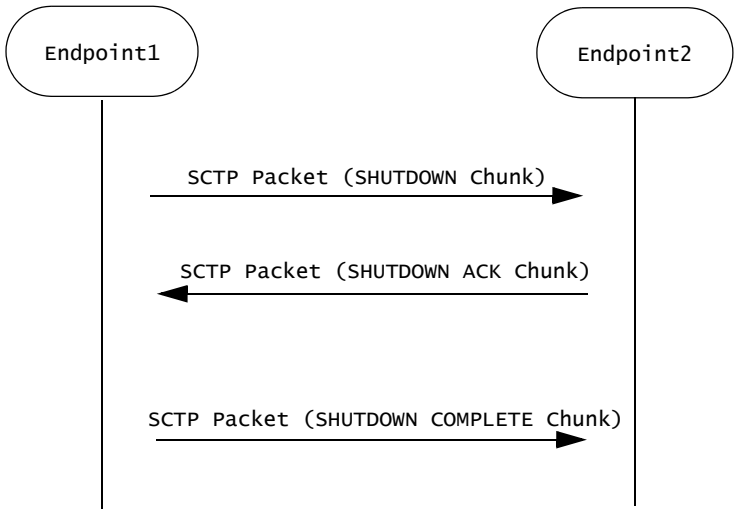
Both the INIT chuck (issued by the initiator) and INIT ACK chunk (issued by the responder) specify the number of outbound streams supported by the association, as well as the maximum inbound streams accepted from the other endpoint.

Association establishment is completed by a COOKIE ECHO/COOKIE ACK exchange that specifies a cookie value used in all subsequent DATA exchanges.

Once an association is successfully established, an SCTP endpoint can send unidirectional data streams using SCTP packets that contain DATA chunks. The recipient endpoint acknowledges with an SCTP packet containing a SACK chunk.

SCTP monitors endpoint reachability by periodically sending SCTP packets that contain HEARTBEAT chunks. The recipient endpoint acknowledges receipt, and confirms availability, with an SCTP packet containing a HEARBEAT ACK chunk.

Either SCTP endpoint can initiate a graceful association close with an SCTP packet that contains a SHUTDOWN chunk. The recipient endpoint acknowledges with an SCTP packet containing a SHUTDOWN ACK chunk. The initiating endpoint concludes the graceful close with an SCTP packet that contains a SHUTDOWN COMPLETE chunk.

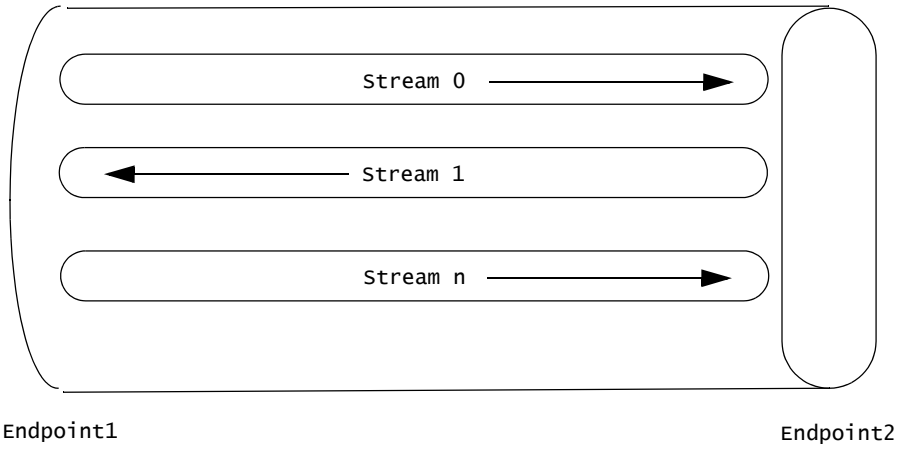


Congestion Control

SCTP congestion control mechanism is similar to that provided by TCP, and includes slow start, congestion avoidance, and fast retransmit. In SCTP, the initial congestion window (*cwnd*) is set to the double of the maximum transmission unit (MTU) while in TCP, it is usually set to one MTU. In SCTP, *cwnd* increases based on the number of acknowledged bytes, rather than the number of acknowledgements in TCP. The larger initial *cwnd* and the more aggressive *cwnd* adjustment provided by SCTP result in a larger average congestion window and, hence, better throughput performance than TCP.

Multi-Streaming

SCTP supports *streams* as depicted in the following figure which depicts an SCTP association that supports three streams,



The multiple stream mechanism is designed to solve the head-of-the-line blocking problem of TCP. Therefore, messages from different multiplexed flows do not block one another.

A stream can be thought of as a sub-layer between the transport layer and the upper layer. SCTP supports multiple logical streams to improve data transmission throughput. As shown in the above figure, SCTP allows multiple unidirectional streams within an association. This multiplexing/de-multiplexing capability is called multi-streaming and it is achieved by introducing a field called *Stream Identifier* contained in every DATA chunk) that is used to differentiate segments in different streams.

Delivery Modes

SCTP supports two delivery modes, *ordered* and *unordered*. Delivery mode is specified by the *U* bit in the DATA chunk header — if the bit is clear (0), ordered delivery is specified; if the bit is set (1), unordered delivery is specified.

Within a stream, an SCTP endpoint must deliver ordered DATA chunks (received with the *U* bit set to 0) to the upper layer protocol according to the order of their *Stream Sequence Number*. Like the *U* bit, the Stream Sequence Number is a field within the DATA chunk header, and serves to identify the chunk's position with the message stream. If DATA chunks arrive out of order of their Stream Sequence Number, the endpoint must delay delivery to the upper layer protocol until they are reordered and complete.

Unordered DATA chunks (received with the *U* bit set to 1) are processed differently. When an SCTP endpoint receives an unordered DATA chunk, it must bypass the ordering mechanism and immediately deliver the data to the upper layer protocol (after reassembly if the user data is fragmented by the sender). As a consequence, the Stream Sequence Number field in an unordered DATA chunk has no significance. The sender can fill it with arbitrary value, but the receiver must ignore any value in field.

When an endpoint receives a DATA chunk with the U flag set to 1, it must bypass the ordering mechanism and immediately deliver the data to the upper layer (after reassembly if the user data is fragmented by the data sender).

Unordered delivery provides an effective way of transmitting out-of-band data in a given stream. Note also, a stream can be used as an unordered stream by simply setting the *U* bit to 1 in all DATA chunks sent through that stream.

Multi-Homing

Call control applications for carrier-grade service require highly reliable communication with no single point of failure. SCTP can assist carriers with its multi-homing capabilities. By providing different paths through the network over separate and diverse means, the goal of no single point of failure is more easily attained.

SCTP built-in support for multi-homed hosts allows a single SCTP association to run across multiple links or paths, hence achieving link/path redundancy. With this capability, and SCTP association can be made to achieve fast failover from one link/path to another with little interruption to the data transfer service.

Multi-homing enables an SCTP host to establish an association with another SCTP host over multiple interfaces identified by different IP addresses. With specific regard to the Acme Packet SBC these IP addresses need not be assigned to the same physical interface, or to the same physical Network Interface Unit.

If the SCTP nodes and the according IP network are configured in such a way that traffic from one node to another travels on physically different paths if different destination IP address are used, associations become tolerant against physical network failures and other problems of that kind.

An endpoint can choose an optimal or suitable path towards a multi-homed destination. This capability increases fault tolerance. When one of the paths fails, SCTP can still choose another path to replace the previous one. Data is always sent over the *primary path* if it is available. If the primary path becomes unreachable, data is migrated to a different, affiliated address — thus providing a level of fault tolerance. Network failures that render one interface of a server unavailable do not necessarily result in service loss. In order to achieve real fault resilient communication between two SCTP endpoints, the maximization of the diversity of the round-trip data paths between the two endpoints is encouraged.

Multi-Homing and Path Diversity

As previously explained, when a peer is multi-homed, SCTP can automatically switch the subsequent data transmission to an alternative address. However, using multi-homed endpoints with SCTP does not automatically guarantee resilient communications. One must also design the intervening network(s) properly.

To achieve fault resilient communication between two SCTP endpoints, one of the keys is to maximize the diversity of the round-trip data paths between the two endpoints. Under an ideal situation, one can make the assumption that every destination address of the peer will result in a different, separate path towards the peer. Whether this can be achieved in practice depends entirely on a combination of factors that include path diversity, multiple connectivity, and the routing protocols that glue the network together. In a normally designed network, the paths may not be diverse, but there may be multiple connectivity between two hosts so that a single link failure will not fail an association.

In an ideal arrangement, if the data transport to one of the destination addresses (which corresponds to one particular path) fails, the data sender can migrate the data traffic to other remaining destination address(es) (that is, other paths) within the SCTP association.

Monitoring, Failure Detection and Recovery

When an SCTP association is established, a single destination address is selected as the primary destination address and all new data is sent to that primary address by default. This means that the behavior of a multi-homed SCTP association when there are no network losses is similar to behavior of a TCP connection. Alternate, or secondary, destination addresses are only used for redundancy purposes, either to retransmit lost packets or when the primary destination address cannot be reached.

A failover to an alternate destination is performed when the SCTP sender cannot elicit an acknowledgement — either a SACK for a DATA chunk, or a HEARTBEAT ACK for a HEARTBEAT chunk — for a configurable consecutive number of transmissions. The SCTP sender maintains an error-counter is maintained for each destination address and if this counter exceeds a threshold (normally six), the address is marked as inactive, and taken out of service. If the primary destination address is marked as inactive, all data is then switched to a secondary address to complete the failover.

If no data has been sent to an address for a specified time, that endpoint is considered to be *idle* and a HEARTBEAT packet is transmitted to it. The endpoint is expected to respond to the HEARTBEAT immediately with a HEARTBEAT ACK. As well as monitoring the status of destination addresses, the HEARTBEAT is used to obtain RTT measurements on idle paths. The primary address becomes active again if it responds to a heartbeat.

The number of events where heartbeats were not acknowledged within a certain time, or retransmission events occurred is counted on a per association basis, and if a certain limit is exceeded, the peer endpoint is considered unreachable, and the association is closed.

The threshold for detecting an endpoint failure and the threshold for detecting a failure of a specific IP addresses of the endpoint are independent of each other. Each parameter can be separately configured by the SCTP user. Careless configuration of these protocol parameters can lead the association onto the dormant state in which all the destination addresses of the peer are found unreachable while the peer still remains in the reachable state. This is because the overall retransmission counter for the peer is still below the set threshold for detecting the peer failure.

ACLI Instructions for Configuring SCTP for DIAMETER Transport

Use the following steps to configure SCTP as the layer 4 transport for a DIAMETER interface.

- create an SCTP-based DIAMETER Director port
- associate network interfaces with existing realms
- set SCTP timers and counters

Configuring an SCTP DIAMETER Director Port

DIAMETER Director ports are created as part of the DIAMETER Director Interface configuration process.

1. From superuser mode, use the following command sequence to access *diameter-director-port* configuration mode.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# session-router
   ACMEPACKET(session-router)# diameter-director-interface
   ACMEPACKET(diameter-director-interface)# diameter-director-ports
   ACMEPACKET(diameter-director-port)#
   ```

2. Use the **address** parameter to provide the IP address of the network interface that supports the DIAMETER Director port.

   This is the primary address of a the local multi-homed SCTP endpoint.

   ```
   ACMEPACKET(diameter-director-port)# address 172.16.10.76
   ACMEPACKET(diameter-director-port)#
   ```

3. Retain the default value, 3868 (the *well-known* DIAMETER port) for the **port** parameter.

   ```
   ACMEPACKET(diameter-director-port)# port 3868
   ACMEPACKET(diameter-director-port)#
   ```

4. Use the **transport-protocol** parameter to identify the layer 4 protocol.

   Supported values are UDP, TCP, TLS, and SCTP.

   Select SCTP.

   ```
   ACMEPACKET(diameter-director-port)# transport-protocol sctp
   ACMEPACKET(diameter-director-port)#
   ```

5. Use the **multi-homed-addrs** parameter to specify one or more local secondary addresses of the SCTP endpoint.

   Multi-homed addresses must be of the same type (IPv4 or IPv6) as that specified by the **address** parameter. Like the address parameter, these addresses identify SD physical interfaces.

   To specify multiple addresses, bracket an address list with parentheses.

   ```
   ACMEPACKET(diameter-director-port)# multi-homed-addrs 182.16.10.76
   ACMEPACKET(diameter-director-port)#
   ```

   To specify multiple addresses, bracket an address list with parentheses.

   ```
   ACMEPACKET(diameter-director-port)# multi-homed-addrs (182.16.10.76
   192.16.10.76 196.15.32.108)
   ACMEPACKET(diameter-director-port)#
   ```

6. Remaining parameters can be safely ignored.

7. Use **done**, **exit**, and **verify-config** to complete configuration of the DIAMETER Director port.

   ```
   ACMEPACKET(diameter-director-port)# done
   ACMEPACKET(diameter-director-interface)# exit
   ACMEPACKET(session-router)# exit
   ACMEPACKET(configure)# exit
   ACMEPACKET# verify-config
   ----------------------------------------------------------------------
   Verification successful! No errors nor warnings in the configuration
   ACMEPACKET#
   ```

Configuring the Realm

After configuring a DIAMETER Director port which identifies primary and secondary multi-homed transport addresses, you list the network interfaces that support these primary and secondary addresses in the realm assigned during DIAMETER Director Interface configuration.

1. From superuser mode, use the following command sequence to access *realm-config* configuration mode.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# media-manager
   ACMEPACKET(media-manager)# realm-config
   ACMEPACKET(realm-config)#
   ```

2. Use the **select** command to access the target realm.

3. Use the **network-interfaces** command to identify the network interfaces that support the SCTP primary and secondary addresses.

   Network interfaces are identified by their name.

   Enter a list of network interface names using parentheses as list brackets. The order of interface names is not significant.

   ```
   ACMEPACKET(realm-config)# network-interfaces (mo1 M10)
   ACMEPACKET(realm-config)#
   ```

4. Use **done**, **exit**, and **verify-config** to complete realm configuration.

   ```
   ACMEPACKET(realm-config)# done
   ACMEPACKET(media-manager)# exit
   ACMEPACKET(configure)# exit
   ```

```
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting SCTP Timers and Counters

Setting SCTP timers and counters is optional. All configurable timers and counters provide default values and most default to recommended values as specified in RFC 4960, *Stream Control Transmission Protocol*.

Management of Retransmission Timer, section 6.3 of RFC 4960 describes the calculation of a Retransmission Timeout (RTO) by the SCTP process. This calculation involves three SCTP protocol parameters: *RTO.Initial*, *RTO.Min*, and *RTO.Max*. *Suggested SCTP Protocol Parameter Values* section 15 of RFC 4960 lists recommended values for these parameters.

The following shows the equivalence of recommended values and ACLI defaults.

RTO.Initial	= 3 seconds	**sctp-rto-initial**	= 3000 ms (default value)
RTO.Min	= 1 second	**sctp-rto-min**	= 1000 ms (default value)
RTO.Max	= 60 seconds	**sctp-rto-max**	= 60000 ms (default value)

Path Heartbeat, section 8.3 of RFC 4960 describes the calculation of a Heartbeat Interval by the SCTP process. This calculation involves the current calculated RTO and a single SCTP protocol parameter — *HB.Interval*.

The following shows the equivalence of recommended the value and ACLI default.

HB.Interval	= 30 seconds	**sctp-hb-interval**	= 3000 ms (default value)

Acknowledgement on Reception of DATA Chunks, section 6.2 of RFC 4960 describes requirements for the timely processing and acknowledgement of DATA chunks. This section requires that received DATA chunks must be acknowledged within 500 milliseconds, and recommends that DATA chunks should be acknowledged with 200 milliseconds. The interval between DATA chunk reception and acknowledgement is specific by the ACLI **sctp-sack-timeout** parameter, which provides a default value of 200 milliseconds and a maximum value of 500 milliseconds.

Transmission of DATA Chunks, section 6.1 of RFC 4960 describes requirements for the transmission of DATA chunks. To avoid network congestion the RFC recommends a limitation on the volume of data transmitted at one time. The limitation is expressed in terms of DATA chunks, not in terms of SCTP packets. The maximum number of DATA chunks that can be transmitted at one time is specified by the ACLI **sctp-max-burst**

parameter, which provides a default value of 4 chunks, the limit recommended by the RFC.

SCTP Network Parameters are not RTC

SCTP configuration parameters within the `network-parameters` element are not real-time configuration (RTC) supported. Reboot your system for changes to these parameter to take effect.

Setting the RTO

An SCTP endpoint uses a retransmission timer to ensure data delivery in the absence of any feedback from its peer. RFC 4960 refers to the timer as *T3-rtx* and to the timer duration as *RTO* (retransmission timeout).

When an endpoint's peer is multi-homed, the endpoint calculates a separate RTO for each IP address affiliated with the peer. The calculation of RTO in SCTP is similar to the way TCP calculates its retransmission timer. RTO fluctuates over time in response to actual network conditions. To calculate the current RTO, an endpoint maintains two state variables per destination IP address — the SRTT (smoothed round-trip time) variable, and the RTTVAR (round-trip time variation) variable.

Use the following procedure to assign values used in RTO calculation.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# system
   ACMEPACKET(system)# network-parameters
   ACMEPACKET(network-parameters)#
   ```

2. Use the **sctp-rto-initial** parameter to assign an initial timer duration.

   Allowable values are integers within the range 0 through 4294967295 that specify the initial duration in milliseconds. In the absence of an explicitly configured integer value, **sctp-rto-initial** defaults to 3000 milliseconds (3 seconds, the recommended default value from RFC 4960).

   As described in Section 6.3 of RFC 4960, the value specified by **sctp-rto-initial** is assigned to the SCTP protocol parameter *RTO.Initial*, which provides a default RTO until actual calculations have derived a fluctuating duration based on network usage. The value specified by the **sctp-rto-initia**l parameter seeds these calculations.

   ```
   ACMEPACKET(network-parameters)# sctp-rto-initial 3000
   ACMEPACKET(network-parameters)#
   ```

3. Use the **sctp-rto-min** and **sctp-rto-max** parameters to assign an RTO floor and ceiling.

   Allowable values are integers within the range 0 through 4294967295 that specify the minimum and maximum durations in milliseconds. In the absence of an explicitly configured integer value, **sctp-rto-min** defaults to 1000 ms (1 second, the recommended default value from RFC 4960), and **sctp-rto-max** defaults to 60000 ms (60 seconds, the recommended default value from RFC 4960.)

   As described in Section 6.3 of RFC 4960, the values specified by **sctp-rto-min** and **sctp-rto-max** are assigned to the SCTP protocol parameters, *RTO.min* and *RTO.max* that limit RTO calculations. If a calculated RTO duration is less than *RTO.min*, the parameter value is used instead of the calculated value; likewise, if a calculated RTO duration is greater than *RTO.max*, the parameter value is used instead of the calculated value.

   ```
   ACMEPACKET(network-parameters)# sctp-rto-min 1000
   ACMEPACKET(network-parameters)# sctp-rto-max 60000
   ```

```
ACMEPACKET(network-parameters)#
```

4. Use **done**, **exit**, and **verify-config** to complete RTO configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting the Heartbeat Interval

Both single-homed and multi-homed SCTP endpoints test the reachability of associates by sending periodic HEARTBEAT chunks to UNCONFIRMED or idle transport addresses.

Use the following procedure to assign values used in Heartbeat Interval calculation.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-hb-interval** parameter to assign an initial Heartbeat Interval duration.

   Allowable values are integers within the range 0 through 4294967295 that specify the initial Heartbeat Interval in milliseconds. In the absence of an explicitly configured integer value, **sctp-hb-interval** defaults to 30000 milliseconds (30 seconds, the recommended default value from RFC 4960).

   As described in Section 8.3 of RFC 4960, the value specified by **sctp-hb-interval** is assigned to the SCTP protocol parameter *HB.Interval*, which provides a default interval until actual calculations have derived a fluctuating interval based on network usage. The value specified by the **sctp-hb-interval** parameter is used during these calculations.

```
ACMEPACKET(network-parameters)# sctp-hb-interval 30000
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete Heartbeat Interval configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting the SACK Delay Timer

An SCTP Selective Acknowledgement (SACK) is sent to the peer endpoint to acknowledge received DATA chunks and to inform the peer endpoint of gaps in the received subsequences of DATA chunks. Section 6.2 of RFC 4960 sets a specific requirement for a SACK Delay timer that specifies the maximum interval between the reception of an SCTP packet containing one or more DATA chunks and the transmission of a SACK to the packet originator.

Use the following procedure to set the SACK Delay timer.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-sack-timeout** parameter to assign a value to the SACK Delay timer.

   Allowable values are integers within the range 0 through 500 which specify the maximum delay (in milliseconds) between reception of a SCTP packet containing one or more Data chunks and the transmission of a SACK to the packet source. The value 0 indicates that a SACK is generated immediately upon DATA chunk reception

   In the absence of an explicitly configured integer value, **sctp-sack-timeout** defaults to 200 ms (the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-sack-timeout 200
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete configuration of the SACK Delay timer.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Limiting DATA Bursts

Section 6.1 of RFC 4960 describes the SCTP protocol parameter, *Max.Burst*, used to limit the number of DATA chunks that are transmitted at one time.

Use the following procedure to assign a value to the SCTP protocol parameter, *Max.Burst*.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-max-burst** parameter to assign a value to *Max.Burst*.

   Allowable values are integers within the range 0 through 4294967295 that specify the maximum number of DATA chunks that will be sent at one time. In the absence

of an explicitly configured integer value, **sctp-max-burst** defaults to 4 (DATA chunks, the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-max-burst 4
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete configuration of DATA burst limitations.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting Endpoint Failure Detection

As described in Monitoring, Failure Detection and Recovery, a single-homed SCTP endpoint maintains a count of the total number of consecutive failed (unacknowledged) retransmissions to its peer. Likewise, a multi-homed SCTP endpoint maintains a series of similar, dedicated counts for all of its destination transport addresses. If the value of these counts exceeds the limit indicated by the SCTP protocol parameter *Association.Max.Retrans*, the endpoint considers the peer unreachable and stops transmitting any additional data to it, causing the association to enter the CLOSED state.

The endpoint resets the counter when (1) a DATA chunk sent to that peer endpoint is acknowledged by a SACK, or (2) a HEARTBEAT ACK is received from the peer endpoint.

Use the following procedure to configure endpoint failure detection.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-assoc-max-retrans** to assign a value to the SCTP protocol parameter *Association.Max.Retrans*.

   Allowable values are integers within the range 0 through 4294967295 which specify the maximum number of transmission requests. In the absence of an explicitly configured integer value, s**ctp-assoc-max-retrans** defaults to 10 (transmission re-tries, the recommended default value from RFC 4960).

```
ACMEPACKET(network-parameters)# sctp-assoc-max-retrans 10
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete endpoint failure detection configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
-------------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Setting Path Failure Detection

As described in Monitoring, Failure Detection and Recovery, when its peer endpoint is multi-homed, an SCTP endpoint maintains a count for each of the peer's destination transport addresses.

Each time the T3-rtx timer expires on any address, or when a HEARTBEAT sent to an idle address is not acknowledged within an RTO, the count for that specific address is incremented. If the value of a specific address count exceeds the SCTP protocol parameter *Path.Max.Retrans*, the endpoint marks that destination transport address as *inactive*.

The endpoint resets the counter when (1) a DATA chunk sent to that peer endpoint is acknowledged by a SACK, or (2) a HEARTBEAT ACK is received from the peer endpoint.

When the primary path is marked *inactive* (due to excessive retransmissions, for instance), the sender can automatically transmit new packets to an alternate destination address if one exists and is *active*. If more than one alternate address is active when the primary path is marked inactive, a single transport address is chosen and used as the new destination transport address.

Use the following procedure to configure path failure detection.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-path-max-retrans** parameter to assign a value to the SCTP protocol parameter *Path.Max.Retrans*.

   Allowable values are integers within the range 0 through 4294967295 that specify the maximum number of RTOs and unacknowledged HEARTBEATS. In the absence of an explicitly configured integer value, **sctp-path-max-retrans** defaults to 5 (RTO and/or HEARTBEAT errors per transport address, the recommended default value from RFC 4960).

   When configuring endpoint and path failure detection, ensure that the value of the **sctp-assoc-max-retrans** parameter is smaller than the sum of the **sctp-path-max-retrans** values for all the remote peer's destination addresses. Otherwise, all the destination addresses can become inactive (unable to receive traffic) while the endpoint still considers the peer endpoint reachable.

```
ACMEPACKET(network-parameters)# sctp-path-max-retrans 5
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete path failure detection configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Specifying the Delivery Mode

As described in Delivery Modes, SCTP support two delivery modes, *ordered* and *unordered*.

1. From superuser mode, use the following command sequence to access *network-parameters* configuration mode.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-parameters
ACMEPACKET(network-parameters)#
```

2. Use the **sctp-send-mode** parameter to select the preferred delivery mode.

   Choose ordered or unordered.

```
ACMEPACKET(network-parameters)# sctp-send-mode unordered
ACMEPACKET(network-parameters)#
```

3. Use **done**, **exit**, and **verify-config** to complete delivery mod configuration.

```
ACMEPACKET(network-parameters)# done
ACMEPACKET(system)# exit
ACMEPACKET(configure)# exit
ACMEPACKET(configure)# exit
ACMEPACKET# verify-config
----------------------------------------------------------------------
Verification successful! No errors nor warnings in the configuration
ACMEPACKET#
```

Example Configurations

The following ACLI command sequences configure two physical interfaces. The first configures a physical interface named *m10*, that will support an SCTP primary address; the second sequence configures an interface named *m01* that will support a secondary SCTP address.

Both sequences show only configuration parameters essential for SCTP operations; other parameters can retain default values, or be assigned other values specific to local network requirements.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# operation-type media
ACMEPACKET(phy-interface)# port 0
ACMEPACKET(phy-interface)# slot 1
ACMEPACKET(phy-interface)# name m10
ACMEPACKET(phy-interface)#
...
...
...
ACMEPACKET(phy-interface)#

ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# phy-interface
ACMEPACKET(phy-interface)# operation-type media
ACMEPACKET(phy-interface)# port 1
ACMEPACKET(phy-interface)# slot 0
ACMEPACKET(phy-interface)# name m01
ACMEPACKET(phy-interface)#
...
...
...
ACMEPACKET(phy-interface)#
```

The next ACLI sequences configure two network interfaces. The first sequence configures a network interface named *m10*, thus associating the network interface with the physical interface of the same name. The ACLI **ip-address** command assigns the IPv4 address 172.16.10.76 to the network interface. In a similar fashion, the second command sequence associates the *m01* network and physical interfaces and assigns an IPv4 address of 182.16.10.76.

Both sequences show only configuration parameters essential for SCTP operations; other parameters can retain default values, or be assigned other values specific to local network requirements.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# name m10
ACMEPACKET(network-interface)# ip-address 172.16.10.76
...
...
...
ACMEPACKET(network-interface)#


ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# network-interface
ACMEPACKET(network-interface)# name m01
ACMEPACKET(network-interface)# ip-address 182.16.10.76
...
...
...
ACMEPACKET(network-interface)#
```

The next sequence configures a DIAMETER Director port for SCTP operations. It specifies the use of SCTP as the transport layer protocol, and assigns the existing network interface address, 172.16.10.76, as the SCTP primary address. Additionally, it identifies three other existing network addresses (182.16.10.76, 192.16.10.76, and 196.15.32.108 as SCTP secondary addresses.

This sequence shows only configuration parameters essential for SCTP operations; other parameters can retain default values.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# diameter-director-interface
ACMEPACKET(diameter-director-interface)# diameter-director-ports
ACMEPACKET(diameter-director-port)# address 172.16.10.76
ACMEPACKET(diameter-director-port)# transport-protocol sctp
ACMEPACKET(diameter-director-port)# multi-homed-addrs (182.16.10.76
192.16.10.76 196.15.32.108)
...
...
...
ACMEPACKET(diameter-director-port)#
```

The next two ACLI sequences configure a realm for SCTP operations. As shown by the first ACLI sequence, a named realm, in this example *core-172*, is assigned to a DIAMETER Director interface during the interface configuration process. The second command sequence accesses the target realm and uses the **network-interfaces** command to associate the named SCTP network interfaces with the realm.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# diameter-director-interface
ACMEPACKET(diameter-director-interface)# realm-id net172
...
...
...
ACMEPACKET(diameter-director-interface)#

ACMEPACKET# configure terminal
ACMEPACKET(configure)# media-manager
ACMEPACKET(media-manager)# realm-config
ACMEPACKET(realm-config)# select
identifier: net172
1. net172 ...

selection: 1
ACMEPACKET(realm-config)# network-interfaces (m01 m10 ...)
...
...
...
ACMEPACKET(ream-config)#
```

Incompatible SCTP Association Signaling Workaround

The Net-Net DD supports RFC5061 allowing the SCTP stack to dynamically add or delete an IP address to or from and SCTP association. RFC 4895 defines a new chunk type which can be used to authenticate SCTP chunks. RFC 5061 mandates use of these authenticated chunks (as defined in RFC 4985) to dynamically modify an SCTP association.

Some devices, however, do not support the AUTH chunk specified by RFC4895. For this reason, the Net-Net D-CZ2.1.0M2 and above allows you to bypass the use of this information, effectively disabling this authentication process, during association change signaling procedures.

This configuration violates RFC5061. Use it only to interoperate with applicable devices.

ACLI Instructions

To disable RFC4895 association change authentication:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **system** and press <Enter> to access the system path.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type **network-parameters a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# network-parameters
```

4. Type **options sctp-asconf-auth-no-check=1 a**nd press <Enter>.
   The default value for this option is “0”, which tells the system to perform the authentication check via the AUTH chunk.

   ```
   ACMEPACKET(network-parameters)# option +sctp-asconf-no-check=1
   ```

5. Save this change to the element with the done command and press <Enter>.

   ```
   ACMEPACKET(network-parameters)# done
   ```

Note that the use of the “+” sign prior to the option ensures that previously configured options stay in effect. Omitting the option tells the sytem to use only the newly configured option.

3 Routing Diameter Messages

Routing

The Net-Net Diameter Director's routing engine routes incoming Diameter messages to external Diameter agents. The routing engine uses Diameter Director Policies as an evaluation device to determine a routable next hop. The next hop is either a configured Diameter Director Agent or Diameter Director Group.

Conventions in this Chapter

Diameter Director Policy—A container for individual Policy Attributes.

Policy Attribute—Individual evaluations that comprise a Diameter Director Policy.

Diameter Message Reception

First, the Net-Net Diameter Director receives a Diameter message on a Diameter Director Interface. Implicitly, the received message defines the following routing criteria:

- Incoming Realm, interface, address, and port
- Application ID in the Diameter header

The Net-Net Diameter Director starts routing evaluation by considering the message with respect to the Diameter Director Policy named in the Diameter Director Interface's *routing policy* parameter.

Routing Criteria

Messages are generally routed to a next hop based on the contents of one or more AVPs. The tests to determine a match work similarly to the Net-Net SBC's HMR functionality.

The Net-Net Diameter Director uses a Diameter Director Policy to evaluate a Diameter message. The Diameter Director Policy is a container for a group of Policy Attributes which are configured as sub-elements. Policy Attributes define the individual comparisons used for routing decisions. The order that Policy Attributes are tested in is determined by the configured *priority* parameter, lower values evaluated first. If two Policy Attributes have the same configured priority, the one which was configured first is evaluated first.

Once the first Policy Attribute has been identified, the routing engine evaluates the message against that Policy Attribute.

The match operation is configured by defining a *match field* and *match value*. The Net-Net Diameter Director evaluates if the *match value* is found in the match field. The *match field* is usually an AVP in a Diameter message configured as `avp=<avp-number>`.

For example, if the match field is configured as

```
avp=264
```

and the match value is configured as

```
.com
```

then the routing engine will make a positive match in the Origin-Host (264) AVP with the string or partial string of ".com" (given that the AVP Type and Comparison type parameter are correctly set: see next section).

Routing Evaluation Data Types

The Net-Net Diameter Director must parse and evaluate the match field correctly to produce an accurate match. Otherwise, the wrong comparison might be made.

AVP Type

The *avp type* of the *match field* is defined so that the Net-Net Diameter Director knows the encoding of the data it is evaluating. AVP types can be found in either the respective RFC or from context. Valid *avp type* values are:

- Octet string
- Diameter URI
- Integer32
- Unsigned Integer 32
- Diameter Enumerated
- Diameter Address
- Grouped

Comparison Type

The *comparison type* is defined so that the Net-Net Diameter Director knows the correct way to determine if the *match value* appears in the *match field*. Valid *comparison types* are:

- Integer—literal number match
- Regex—regular expression match
- Case-sensitive—case-sensitive string match
- Case-insensitive—case-insensitive string match
- Refer-case-sensitive—case sensitive when referring to a previous Diameter Director Policy, defined by $diameter-director-policy
- Refer-case-insensitive—case insensitive when referring to a Diameter Director Policy, defined by $diameter-director-policy
- grouped-avp—matching on data found within a grouped AVP

Reserved Values

A list of five reserved values can be used in the match value parameter. These values are:

- `$avp=<avp-num>`—AVPnumber
- `$appln_id`—application ID of the request being received
- `$cmd_code`—command code
- `$inc_realm`—incoming realm
- `$inc_intf`—incoming interface

Routing Decision Process

The evaluation of match value against match field can produce either a positive match (match value found in match field) or a failed match (match value NOT found in match field).

The Net-Net Diameter Director evaluates each Policy Attribute in turn according to its priority. If all Policy Attributes in a Diameter Director Policy are tested without any making a match, then the Net-Net Diameter Director sends a message to the originating host with DIAMETER_UNABLE_TO_DELIVER-3002 code in a Result-Code AVP (268).

When the routing engine makes its first positive match on a Policy Attribute, it checks if the *next policy* parameter is configured. If it is configured with a valid Diameter Director Policy, the Net-Net Diameter Director starts the evaluation process again using the original Diameter message against the newly referenced Diameter Director Policy.

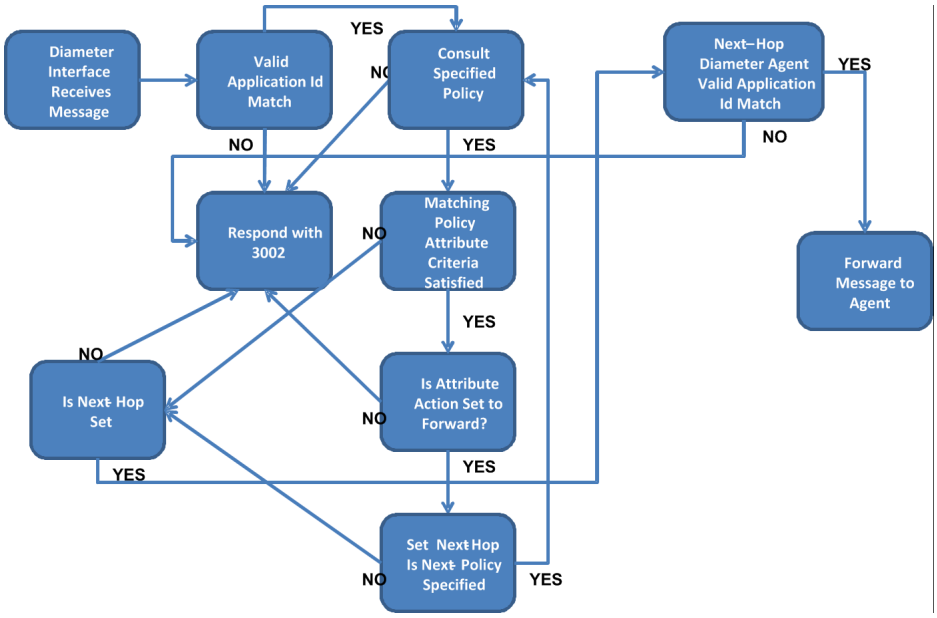
The routing engine continues, following the chain of Policy Attributes within Diameter Director Policies until it makes a match on a Policy Attribute with an empty *next policy* parameter, i.e. a blank value.

If there is no configured *next policy*, the routing engine looks at the *action* parameter. If the *action* is set to reject, then the Net-Net Diameter Director sends a message back to the originating host with DIAMETER_UNABLE_TO_DELIVER-3002 code in a Result-Code AVP (268). If the value is set to *forward*, the Net-Net Diameter Director performs an Application ID check. This check insures that the target next hop had previously stated it could support the ApplicationID found in the Diameter message being routed. This was performed in the CER/CEA transaction. If the Diameter agent can support the ApplicationID, then the Net-Net Diameter Director sends the message to the element defined in the *next hop* parameter.

The next hop is configured as either a configured Diameter Director Agent or Diameter Director Group; it cannot be configured as an IP address. The *next hop* value must be the Diameter Director Agent or Diameter Director Group's configured *name* parameter. Diameter Director Group groups are configured as `ddg:<diameter-director-group-name>`.

If no next hop is configured, the Net-Net-Net Diameter Director drops the request message and sends a DIAMETER_UNABLE_TO_DELIVER-3002 code in a Result-Code AVP (268) to the requester.

The following diagram represents the routing decision process graphically:



Record Route AVP Creation

When a Diameter message is forwarded to a Diameter agent, the Net-Net Diameter Director inserts the Route-Record AVP (282) into the message. The value in this AVP is the one that the Net-Net Diameter Director inserted into the Origin-Host AVP (264) during the CER/CEA negotiation with that target Diameter agent.

Loop Detection

The Net-Net Diameter Director can detect if the messages it is proxying have entered into routing loops. It performs a routing loop test by checking for its own identity in all Route-Record AVPs (282) in the messeages recieved on a socket. If the Net-Net Diameter Director's identity is found in any of the Record-Route AVP(s) (282) the DSC will not forward that message. Instead, it replies to the message sender with a locally created reply that includes the DIAMETER_LOOP_DETECTED (3005) error code.

Loop detection is enabled by setting the **loop-detection** parameter in the diameter-director-config to enabled. If your network excludes the Record-Route AVP (282) from all messages or you do not wish to use this feature, leave the parameter set to disabled to reduce system load.

ACLI Instructions

To configure network routing loop detection:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-config** and type **select**.

   ```
   ACMESYSTEM(session-router)# diameter-director-config
   ACMESYSTEM(diameter-director-config)# select
   ```

4. **loop-detection**— Leave this at the default of disabled, or set the parameter to enabled for the Net-Net Diameter Director to reject messages that it has classified as indicative of the presence of a routing loop.
5. Type **done** when finished.

Grouped AVP Routing

The Net-Net Diameter Director can route messages based on the data contained in Grouped AVPs. RFC 3588 states that a Grouped AVP is specified as a sequence of AVPs, including their headers and padding. Grouped AVP handling is configured within a Policy Attribute.

The Net-Net Diameter Director is configured to consider a set of AVPs as a Grouped AVP, which model the definition of that grouped AVP. One or more member AVPs of the group are parsed to find if a *match value* is found in the *match field*. All configured *sub avps* must make positive matches to consider the whole Policy Attribute as having made a positive match. That is, the comparison is an AND function for all *sub avps* that comprise the grouped AVP.

To configure a Grouped AVP-handling Policy Attribute, first set the *avp type* to grouped and the *comparison type* to grouped-avp. The next step is to configure sub avps. These act like Policy Attributes, although they can only be configured to match on an AVP. Not all contents of the grouped AVP must be specified as *sub avps*, but all of those that are configured as *sub avps* must match for that Policy Attribute to return a positive match.

Sub avps are configured similarly to Policy Attributes. The *avp code* parameter indicates the member AVP where the match is being tested; you can only match on AVP data in a *sub avp*. The remaining AVP Type, Comparison Type, and *match value* parameters represent the same data as they do in the Policy Attribute configuration.

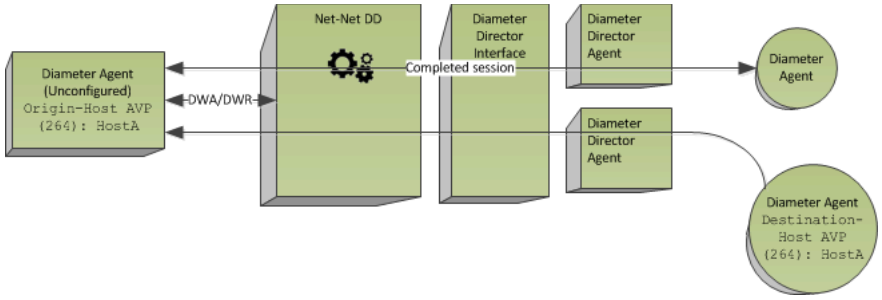
Dynamic Routing

The Net-Net Diameter Director can route a request message from a Diameter Director Agent to a non-configured Diameter agent if that non-configured Diameter agent had already sent a request through the Net-Net Diameter Director.

To contact non-configured agents, the Net-Net Diameter Director caches the non-configured agent's return route for as long as the connection between itself and the endpoint remains up. The Net-Net Diameter Director uses the value of the Origin-Host AVP (264) sent from the non-configured endpoint as the key to reference the route.

Subsequently, when a configured Diameter Director Agent wants to initiate a request to the non-configured agent, the Net-Net Diameter Director can do so without having explicit configuration to support it using the cached route.

Consider the following diagram: Since the route to HostA is already cached, the Net-Net Diameter Director will route any message addressed to Host A when the Diameter agent uses the correct value in the Destination-Host AVP.



When the unconfigured Diameter agent's connection is lost, route entries using that connection are no longer valid and therefore will be removed from all the sockets. The Net-Net Diameter Director then has no route to reach that target. You may enable or disable dynamic routing with the **dynamic-routing** parameter located in the *diameter director config* configuration element.

Policy Rejection

When the Net-Net Diameter Director reaches a Policy Attribute with no next-hop and whose *action* is set to **reject**, a configurable result code can be sent to the originating Diameter agent.

By default, the Net-Net Diameter Director sends an error response with Result-Code AVP set to 3002 (UNABLE_TO_DELIVER) to the requesting Diameter agent. Alternatively the Net-Net Diameter Director can send a configurable Result-Code AVP or Experimental-Result-Code AVP.

To configure the value of the Result-Code AVP included in the error message, set the *reject-result-code* parameter in the Policy Attribute configuration element. To configure the value of the Experimental-Result-Code AVP included in the error message, set the *reject-exp-result-code* and *reject-exp-vendor-id* parameters in the Policy Attribute configuration element.

Setting the *reject result code* parameter to 0 uses the default Result-Code AVP on a policy reject. Setting either the *reject-exp-result-code* or *reject-exp-vendor-id* parameter to 0 does not send an Experimental Result Code AVP on a policy rejection.

ACLI Configuration

Root Diameter Director Policy

To configure the root Diameter Director Policy on a Diameter Director Interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# diameter-director-interface
ACMEPACKET(diameter-director-interface)#
```

4. Type **select** and the number of the pre-configured diameter director interface you want to configure.

```
ACMEPACKET(diameter-director-interface)# select 1
```

5. **routing-policy**—Enter the name of root diameter-director-policy to first be applied to incoming messages on this diameter-director-interface.
6. Save your work using the ACLI **done** command.

Diameter Director Policy

To configure the Diameter Director Policy:

1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type **diameter-director-policy** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# diameter-director-policy
ACMEPACKET(diameter-director-policy)#
```

4. **name**—Enter the name of this Diameter director policy. This name will be referenced from Policy Attributes or from a Diameter Director Interface.
5. **state**—Set this to **enabled** to use this Diameter Director Policy.
6. **policy-attributes**—Type **policy attributes** to continue and enter individual policy attributes.

Policy Attributes

7. **attribute-name**—Enter the name of this policy attribute.
8. **match-value**—Enter the value within the match-field to find and make a positive match on.
9. **action**—Enter either **forward** or **reject** as the action to take after making a positive match on the previously entered match-value.
10. **match-field**—Enter the field within the Diameter message to search for the match-value in. This is required. Valid values are:

    Set this to **avp=<avp-num>** to match on a specific AVP field

    command-code—Used to make match on the Diameter command code

    application-id—Used to make a match on an application id in the Diameter message

    incoming-realm—Used to make a match on the realm where this message was received

    incoming-interface—Used to make a match on the network interface where this message was received

11. **avp-type**—Set this to the data type of the content of the match field. Refer to the Diameter standards document for the encodings of individual AVPs. Valid values are:

    octet-string | integer32 | unsignedint32 | address | time | utfstring | diameterident | diameteruri | enumerated | grouped

12. **comparison-type**—Enter the type of computational comparison the Net-Net Diameter Director uses to test for aDiameter Director Policy match. Valid values are:

    regex | integer | case-sensitive | case-insensitive | refer-case-sensitive | refer-case-insensitive | grouped-avp

13. **priority**—Enter the priority in which to execute this policy attribute with respect to the other policy attributes configured in this Diameter Director Policy. The lowest priority policy attribute is executed first.
14. **reject-result-code**—Enter the value to include in the Result Code AVP when the Net-Net Diameter Director chooses this Policy Attribute with a **reject** action.
15. **reject-exp-result-code**—Enter the value to include in the Experimental Result Code AVP when the Net-Net Diameter Director chooses this Policy Attribute with a **reject** action. This parameter must be configured along with the *reject exp vendor id* parameter.
16. **reject-exp-vendor-id**—Enter the vendor ID to accompany the Experimental Result Code when the Net-Net Diameter Director chooses this Policy Attribute with a **reject** action. This parameter must be configured along with the *reject exp result code* parameter.
17. **next-hop**—Enter the next hop, where to forward the Diameter message when all other policy matching criteria are met. This value is the *hostname* parameter of a Diameter Director Agent. To indicate a Diameter Director Group, use the prefix **ddg:**
18. **next-policy**—Enter the Diameter Director Policy *name* to use against the Diameter message when all other policy matching criteria are met. To indicate the user of a local routing table, use the prefix **lrt:**

You can type **done** or continue to the next section if you are matching on grouped AVP content.

Policy Attributes for Grouped AVP

19. **sub-avp**—Type **sub-avp** to continue and enter individual policy attributes when making a match on content in a grouped AVP. *avp-type* must be set to **grouped** and *comparison-type* must be set to **grouped-avp** to use this feature. See sub-avp subelement that follows.
20. **avp-code**—Enter the grouped AVP code number to make a match within.
21. **avp-type**—Enter the data type of the content of the AVP data the Net-Net Diameter Director is parsing to make a match within the AVP named in the avp-code. Refer to the Diameter standards document for the encodings of individual AVPs. Valid values are:

    octet-string | integer32 | unsignedint32 | address | time | utfstring | diameterident | diameteruri | enumerated

22. **comparison-type**—Enter the type of computational comparison the Net-Net Diameter Director uses to test for a Diameter Director Policy match. Valid values are:

    regex | integer | case-sensitive | case-insensitive | refer-case-sensitive | refer-case-insensitive

23. **match-value**—Enter the explicit value within the AVP defined in the avp-code parameter of this configuration element to find and make a positive match on.
24. Type **done** to save your work.

Diameter Director Group Recursive Routing

When a Diameter Director Group is selected as a *next hop* by the Policy Attribute, the group determines an agent to route the request to based on *strategy*. Upon receiving an

error response from that agent in the group, the response is proxied back to the originator of the request.

Recursive routing is when the Net-Net Diameter Director can resend the Diameter message to the next available Diameter Director Agent in a Diameter Director Group when it receives a defined error message. This behavior is used when the *do recursion* parameter is **enabled** in the recursive routing subelement under the following conditions:

- The error message matches a value configured in the *result codes* parameter in the *recursive routing* subelement.
- The error message matches a value configured in the *exp result codes* parameter in the *recursive routing* subelement (for errors contained in Experimental Result Code AVP).
- A response is not received within the value in milliseconds configured in the *transaction timeout* parameter in the *recursive routing* subelement. This timer is started when a request is sent to an agent.

Further, the recursion to the next agent is ended in the following cases:

- The result code in the response does not match any configured values (*result codes* nor *exp result codes*).
- The *recursion timeout* (milliseconds) parameter in the *recursive routing* subelement expires while performing the recursion before a valid final response is received. This timer starts when the Net-Net Diameter Director sends the request to the first agent in the Diameter Director Group.
- An error response is returned for the last agent unused agent in the Diameter Director Group.
- All remaining agents in the Diameter Director Group are unavailable (Out-Of-Service).

If no more Diameter Director Agent are available to retry sending a message to, then a 3002 (UNABLE_TO_DELIVER) response is sent back to the originating requester.

ACLI Instructions

To configure recursive routing on a Diameter Director Group:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-group** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-group
   ACMEPACKET(diameter-director-group)#
   ```

4. Type **select** and the number of the pre-configured Diameter Director Group you want to configure.

   ```
   ACMEPACKET(diameter-director-group)# select 1
   ```

5. **do-recursion**—Set this parameter to enabled to use recursive routing for Diameter Director Groups.
6. **result-codes**—Enter the Result-Code AVP values, as returned by an agent, which prompts the Net-Net Diameter Director to recurse through this Diameter Director Group. You may enter this as comma separated integers (hyphenation allowed to specify range).

7. **exp-result-codes**—Enter the Experimental-Result-Code AVP values, as returned by an agent, which prompts the Net-Net Diameter Director to recurse through this Diameter Director Group. You may enter this as comma separated integers (hyphenation allowed to specify range).
8. **transaction-timeout**—Enter the time in milliseconds which the Net-Net Diameter Director can take to recurse the current Diameter Director Agent in this Diameter Director Group before failing and trying the next Diameter Director Agent in the group or returning a 3002 to the requesting agent.
9. **recursion-timeout**—Enter the total time in milliseconds which the Net-Net Diameter Director can take to recurse through all Diameter Director Agents in this Diameter Director Group before failing and returning a 3002 to the requesting agent.
10. Save your work using the ACLI **done** command.

AVP-Based Local Routing Tables

The Net-Net Diameter Director supports Local Routing Tables (LRTs), which are XML documents on the Net-Net Diameter Director that contain mappings used for routing. LRTs are crafted by the user, and then transferred from the user's development environment to the Net-Net Diameter Director */code/lrt* directory. After installation and minimal configuration, these LRTs are available for diameter message routing. In addition, Acme Packet's Net-Net Central includes a GUI tool for creating, editing and managing LRTs.

The following information is specific to the AVP-based LRTs. Potential users of this new LRT type should be acquainted with XML, and familiarize themselves with the more generic descriptions of LRTs supplied by the *ACLI Configuration Guide*.

Routing Diameter Messages with LRTs

The Net-Net Diameter Director performs local route table lookups using the AVP or sub-AVP specified in the diameter-director-policy attributes. The local route table, an XML file on the Net-Net Diameter Director, defines the matching AVP. The Net-Net Diameter Director uses the hostname or IP address portion to determine the next hop. If the hostname or IP address matches a configured session agent, the request is sent to that session agent. If the Net-Net Diameter Director does not find a matching session agent for the hostname/IP address, it either performs a DNS query on the hostname to determine its IP address or sends the request directly to the IP address.

When the next hop is defined as a user-parameter lookup key, such as an AVP or sub-AVP, the defined key is used for the local route table lookup. For example, your diameter-director-policy attribute may have a next-hop configuration as follows:

```
next-hop lrt:msisdnRoute;key=701:$0
```

In this case, messages that include the sub-AVP 701 match this policy. The Net-Net Diameter Director locates the route for this key in the local route table and forwards applicable messages to the configured hop.

Multiple (up to 10) next hops per LRT entry are tried in the order in which they appear in the XML file. If the chosen next hop fails because it matches an out-of-service session agent, then the Net-Net Diameter Director tries the next in the ordered list. Recursion to subsequent entries, however, does not take place if the next hop responds with a diameter error response. Use routing configurations with diameter director groups and recursion enabled as a means of implementing recursive routing.

> **Note:** Entering XML comments on the same line as LRT XML data is not currently supported.

Creating an LRT File

An AVP-Based LRT file is a well-formed XML document with a <localRoutes/> *root* element.

<localRoutes/> can contain any number of child <route/> elements.

Each <route/> element contains:

- a required <user/> element that (1) defines the LRT type.

  The following attributes are found within the <user/> element:

  type — This required attribute can be assigned one of two enumerated values (*string*, or *void*); for an AVP LRT.

  string— The characters the Net-Net Diameter Director uses to define a match within the messages to which this route applies.

  void— Use this keywork to preface the initial and each additional route that applies to this AVP.
- a required <next/> element that uses regular expression syntax to specify the routing next hop.
- an optional <description/> element that provides information relevant to the AVP.

When crafting your LRT file, keep the following rules in mind.

1. Set the *type* attribute of the <localRoutes/> root element to *string*.
2. Set the *type* attribute of all <user/> elements to *string*; <user/>.
3. Set the *type* attribute of all <next/> elements to *void*.
4. After completing the LRT file, use FTP or SFTP to install the file in the */code/lrt* directory of the Net-Net Diameter Director.

The following annotated XML sample provides a template to assist users in crafting their own LRT file.

```
<?xml version="1.0" encoding="UTF-8"?>
<localRoutes>
    <route>
        <user type=”string”>161728309711258749311750616367135</user>
        <next type=”void”>Diam_Dir_Agent_Test1</next>
        <next type=”void”>ddg:Group1</next>
        <next type=”void”>ddg:Group2</next>
    </route>
    <route>
        <user type=”string”>oracle.com</user>
        <next type=”void”>Diam_Dir_Agent_Test2</next>
    </route>
</localRoutes>
```

LRT Entry Matching

When searching an LRT for a matching route, the Net-Net Diameter Director can be configured with one of three match modes with the *match mode* parameter in the *local routing config*. These modes are:

- exact—When searching the applicable LRT, the search and table keys must be an exact match.
- best—The longest matching table key in the LRT is the chosen match.
- all—The all mode makes partial matches where the table's key value is a prefix of the lookup key. For example, a lookup in the following table with a key of 1617281 and a prefix length of 7 returns entries 1, and 2. The 'all' mode incurs a performance penalty because it performs multiple searches of the table with continually shortened lookup keys to find all matching entries. This mode also returns any exact matches too.

Entry#	Key	Result
1	16172810123	ddg:main-hss-group
2	16172815000	ddg:backup-hss-group
3	17815551212	ddg:other-hss-group

5. **next-hop**—Enter the next hop, where to forward the Diameter message when all other policy matching criteria are met. To indicate a local routing table, use the prefix **lrt:** and specify the filename.

ACLI Instructions and Examples

This section shows you how to:

- Set up local route configuration
- Specify that a set of local policy attributes needs to use local routing

Setting Up a Local Routing Configuration

The local routing configuration is a new element in the ACLI session-router path. This is where you configure a name for the local route table, the filename you want to give to the database corresponding to this table, and the prefix length (significant digits/bits) to be used for lookup.

To set up a local routing configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter>.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **local-routing-config** and press <Enter>.

   ```
   ACMEPACKET(session-router)# local-routing-config
   ACMEPACKET(local-routing-config)#
   ```

4. **name**—Enter the name (a unique identifier) for the local route table; this name is used for reference in the local policy attributes when to specify that local routing should be used. There is no default for this parameter, and it is required.
5. **file-name**—Enter the name for the file from which the database corresponding to this local route table will be created. You should use the `.gz` format, and the file should be placed in the `/code/lrt/` directory. There is no default for this parameter and it is required.
6. **prefix-length**—Enter the number of significant digits/bits to used for lookup and cache storage. The default value is **0**. The valid range is:
   - Minimum—0

- Maximum—999999999

7. **match-mode**—Set this parameter to either **best**, **all**, or leave it as **exact** which is the default. This indicates to the Net-Net DD how to determine LRT lookup matches.
8. Save and activate your configuration.

Your configured local routing configuration will resemble the following sample.

```
local-routing-config
        name                            imsi-route
        file-name                       imsi-route.gz
        prefix-length                   3
```

Applying the Local Routing Configuration

You apply the local routing configuration by calling it to use in the local policy attributes. You do this by setting a flag in the **next-hop** parameter along with the name of the local routing configuration that you want to use.

To apply the local routing configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter>.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **local-policy** and press <Enter>.

   ```
   ACMEPACKET(session-router)# diameter-director-policy
   ACMEPACKET(diameter-director-policy)#
   ```

4. Type **policy-attributes** and press <Enter>.

   ```
   ACMEPACKET(diameter-director-)# policy-attributes
   ACMEPACKET(local-policy-attributes)#
   ```

5. **next-hop**—In the **next-hop** parameter, type in **lrt:** followed directly by the name of the local routing configuration to be used. The **lrt:** tag tells the Net-Net DD that a local route table will be used.

   ```
   ACMEPACKET(local-policy-attributes)# next-hop lrt:imsi-route
   ```

6. **next-policy**—In the **next-policy** parameter, type in **lrt:** followed directly by the name of the local routing configuration to be used. The **lrt:** tag tells the Net-Net DD that a local route table will be used. Be sure to format your LRT with the flag "`policy`"to indicate the key to be used, such as:

   ```
   <next type="void">policy:myKey</next>
   ```

Save and activate your configuration.

Session Statefulness

Service providers require that Diameter sessions can be bound to the same serving Diameter elements (i.e. a specific PCRF from among many deployed PCRFs) over an extended period for applications including billing and QoS bearer establishment. To accomplish this, the Net-Net Diameter Director provides the ability to conduct session stateful Diameter routing.

Session statefulness contrasts with transaction statefulness. In session stateful Diameter routing, all transactions from an endpoint with a unique Session-ID and other defined attributes are cached in a Session ID State cache and paired with a next hop obtained from a Diameter Director Policy lookup. When a subsequent message with the same Session

ID is received by the Net-Net Diameter Director, the Diameter Director Policy lookup may be skipped and the message can be forwarded to the next hop bound to the session ID in the cache. In this way the Diameter session, identified by a Session ID, maintains its state with respect to a routing destination.

XML File Format

The Net-Net Diameter Director session state machine relies on an XML file to define the decisions that result in maintaining the Session ID State cache. The XML file defines when to start caching a session-id and next-hop together, when to perform a policy lookup again for messages with that session ID, and when to remove the cache entry for that Session ID.

The Session State file should be placed in the /code/ directory. There is no required naming convention for this file. Its format is as follows:

```
<statemachine max-inactivity-time='86400'>
   <application name="GX" id='16777224'>
      <state>session</state>
         <alloc value='272'>
            <avp code='416' type='enumeration'>1</avp>
         </alloc>
         <realloc value='275'></realloc>
         <dealloc value='272'>
            <avp code='416'>3</avp>
         </dealloc>
         <action>
            <cachemiss code='3002'>reject</cachemiss>
         </action>
   </application>
   <application name="Rx" id='16777229'>
      <state>session</state>
         <alloc value='316'></alloc>
         <dealloc value='317'></dealloc>
   </application>
</statemachine>
```

XML Based Trigger Points

The Net-Net Diameter Director uses the XML file to direct the decisions to ultimately create a binding between a received session ID and a Next Hop, obtained nominally through routing policies. The main trigger points in the XML files are:

- alloc —lookup a next hop and create the binding between the Session ID and next hop
- realloc—perform the next hop lookup again and re-cache that destination with the Session ID
- dealloc—remove the entry from the Session ID State cache

Whether to perform these actions is based upon a message's Diameter application-ID and message code. Additional criteria may be also be used for executing a trigger.

statemachine

The statemachine element indicates the start of the session state machine. Use only one of these elements. Here you can specify the max-inactivity-time attribute in seconds. This value indicates how long a session exists after no activity, before it is removed from the cache. The default value, if not configured is 24 hours (86400 seconds). See Session ID State Cache Entry Aging (60) for more information.

application

The application element indicates the Diameter application for which the Net-Net Diameter Director needs to be stateful . The Net-Net Diameter Director first matches a received message's Application ID and Application ID Name. The vendor value is optional, but if specified, must be matched. This combination must be unique in the XML file. If duplicate values are found, they will be skipped.

state

The state tag attribute only uses 'session' as a valid value. This is defined for accommodating any future expansion.

alloc

The alloc attribute value defines which received messages, identified by command-code, for previously matched Application ID, triggers the caching of Session-id and next-hop routing result. Enter the command code's value in the value attribute according to specification for that Application ID already matched upon.

avp

In some instances, the Net-Net Diameter Director triggers alloc and dealloc on receiving the same message/command-code. To further decide what state changing action to take, you can base the alloc/dealloc decision on an AVP using the avp element inside the alloc element. The avp attribute is optional and if not specified, only command-code is used to take action on the binding. When the AVP attribute is used, the code, type and value attributes are mandatory.

Valid types include octet-string, octet-hex, integer32, unsignedint32, address, diameteruri, and enumeration. The previous alloc tag is mandatory and if not specified, the application will not be loaded.

dealloc

The dealloc attribute value defines which received messages, identified by command-code, for previously matched Application ID, and for the same Session-ID trigger the Net-Net Diameter Director to delete the binding in the cache between the session-id and next-hop routing result. Similar to alloc, you must define the numeric command-code value. The dealloc element is mandatory and if not specified the application will not be loaded.

action

The action element specifies what actions to take under certain conditions. You can configure a cachemiss action which indicates whether to reject or proxy the request if no hit is found in the cache for that session-id. This applies only to the messages with command-codes other than alloc element. When setting the cachemiss action to reject, you specify the reject code in the following example:

```
<action>
<cachemiss code='3002'>reject</cachemiss>
</action>
```

If no code is configured, then the default Result Code AVP, 3002 (UNABLE_TO_DELIVER) is sent in case of reject. Action tag is optional and reject is the default action taken on cachemiss condition. If the cachemiss action is proxy, then the Net-Net Diameter Director reverts to being transaction stateful and a Session-ID unaware Diameter Director Policy lookup ensues.

Session ID Statefulness State Machine

This section describes how a received message is evaluated against the XML file.

When a message is first received on a Diameter Director Interface, the Net-Net Diameter Director looks in its Session ID State cache. If the message matches an entry in the cache

(i.e. the process in the following paragraphs has already occurred) with a next hop, it is routed there.

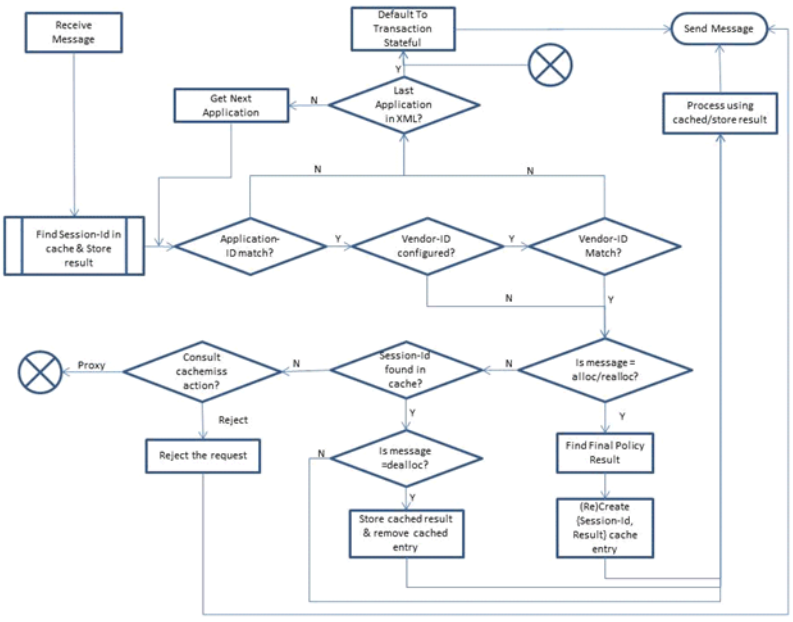
When no cache entry is found, the Net-Net Diameter Director compares the message's Application ID (and optionally vendor) in the XML file. If the application is not found throughout any of the XML file, that message is subject to a one-time Diameter Director Policy lookup (transaction stateful) and forwarded to a next hop based upon those means.

Next, if the message's Application ID is found in the XML file, a Vendor ID match is attempted (if defined). If vendors are defined but there's no exact match, and no other matching Application ID, the message is forwarded in a transaction stateful manner.

With a positive Application ID (and Vendor ID match), the Net-Net Diameter Director evaluates if the XML file specifies an alloc or realloc trigger point. If there's no existing cache entry and an alloc is encountered, the system performs a routing lookup, caches that destination with the Session ID, and then forwards the message to that destination.

Note that for alloc/realloc, first the Net-Net Diameter Director tests for a matching message type, i.e. message code. There must be an initial matching message type to create (alloc) the Session ID State cache entry (if there never is, the cachemiss action trigger point is executed). Subsequent messages, with the same Session ID are forwarded to their cached next hop. It is only upon matching message types defined in the alloc/realloc/dealloc trigger points that the Session ID state cache is modified. Other non-matching message types are forwarded to the cached next hop.

Reallocs redetermine and update the Session ID State Cache with a newly looked up next hop. deallocs forward the message to the next hop, but also remove their Session ID State cache entry. See the following Session State decision flow chart:



Message Rate Constraints

Message rate constraints are applied to incoming requests before linking that request to a session. Therefore, an allocator or deallocator request might be dropped. Thus the

inbound client and/or outbound server should handle the overload response code of 3004 (DIAMETER_TOO_BUSY) as per RFC 3588.

Policy Attribute Next-Hop Determination

When an alloc or realloc initiated lookup fails, the Net-Net Diameter Director returns a 3002 result code to the requesting endpoint, as expected. Also no cache entry is created. All subsequent requests for that session will follow the stateful algorithm from beginning.

If the Net-Net Diameter Director finds a hit on a policy-attribute, the request can either be set to reject or forward. The action, next-hop and reject codes are stored in the cache entry with Session-ID AVP value as the key and applied to all subsequent requests for that session. For example, if the matched policy-attribute is set to REJECT, with a result-code AVP value of 3004 (DIAMETER-TOO-BUSY), then the Session ID State cache entry includes this result. All subsequent requests (not matching a realloc or dealloc) for that session will be 'REJECT'ed with a 3004 result-code. If then a realloc tag initiates a matching policy-attribute set to FORWARD to 'Agent A', then that and all future requests will be 'FORWARD'ed to 'Agent A'.

Recursive Routing Determination

When Diameter Director Group recursive routing is enabled, the Session ID State cache entry is set to an *awaiting final response* state while waiting for a result due to the time used for agent recursion. Additional requests for that session ID are cached while a final next hop is determined.

After the recursion process finds and returns a valid next hop, the Session ID State cache is updated with that next hop. All the pending requests for that session are directly forwarded to that agent. This may create a bust of outbound traffic, which may be rejected due to message rate constraints.

If the Net-Net Diameter Director is unable to find a valid next hop as a result of the Diameter Director Group recursion process, then all current and pending requests are rejected with the appropriate Result-Code AVP value. There is a timer associated with requests sitting in the pending queue. If the timer expires before the final response is received, then the transaction will be deleted and that request will be locally rejected.

Out of Service Cached Next Hop

When an OOS next hop belongs to a recursive-routing group, then the next available agent is selected for forwarding messages, and that second agent is entered into the Session ID State cache for the Session ID. If none of the agents in the recursion list are available then the request will be rejected with a 3002 (UNABLE_TO_DELIVER) result code.

If the OOS agent belongs to a Diameter Director Group with recursive-routing disabled, then the next available agent in the Diameter Director Group is chosen and the request is forwarded there. The cache is also updated to reflect the new next hop. If no in service Diameter agents exist, then the request is rejected with a 3002.

Session ID State Cache and Dynamically Created Routes

The inbound socket for a Diameter client is updated for every message received, which accommodates when multiple clients send requests for using the same session. In such cases, the most recent client is picked to handle the Out of Band (OOB) request.

The Net-Net Diameter Director maintains a Dynamic Route Table (DRT) indexed by the Origin-Host of the active client. When the Net-Net Diameter Director receives an out of band request from the server, the request's Destination-Host is matched to the client's Origin-Host to create the route to the active client.

When Session ID State cache is enabled, a Session is found for the OOB request, but the inbound socket is invalid, then the Dynamic Routing Table is consulted to find the most previous historical egress route (with the same Session ID). The Session ID Cache is updated if the DRT lookup also fails. Then the request is treated as a cachemiss and appropriate action is taken.

Session ID State Cache Entry Aging

The max-inactivity-time in the statemachine XML attribute defines a cache entry's lifetime. Another way to think about this is as the Net-Net Diameter Director's way to enforce a Diameter session length. This ensures that unused resources may be released back to the system. When a new request matching the Session ID received, the inactivity timer is reset. When the timer expires, the cache entry is removed. We recommend that this value be configured to be larger than the server's inactivity timer so that we can still receive an out of band dealloc-initiating request and gracefully terminate the session. The statemachine's internal max-inactivity-time should be used as a last resort.

The Net-Net Diameter Director will not send any request to the client and server upon timer expiry.

ACLI Instructions

To enable Session Statefulness on the Net-Net Diameter Director:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-config
   ACMEPACKET(diameter-director-config)#
   ```

4. Type **select** and press <Enter>.
5. **stateful-policy**—Enter the session statefulness XML file's absolute location. Configuring the file location will enable this feature.

To configure Session ID State Redundancy:

This section describes the parameters that are used to fine tune Session ID Cache redundancy across HA nodes.

6. **redundancy-port**—Leave this at the default port of 1999, or enter your own.
7. **red-max-transactions**—Leave this at the default of 50000, or enter your own number of maximum HA synchronized Session ID Cache transactions. The valid range is:

- Minimum—0
- Maximum—999999999

8. **red-sync-start-time**—Enter the amount of time in milliseconds that the active Net-Net Diameter Director checks to confirm that it is still the active system in the HA node. If the active system is still adequately healthy, this timer will simply reset itself. If for any reason the active has become the standby, it will start to checkpoint with the newly active system when this timer expires. The default is 5000. The valid range is:

- Minimum—0

- Maximum—999999999

9. **red-sync-comp-time**—Enter amount of time in milliseconds that determines how frequently after synchronization the standby Net-Net Diameter Director checkpoints with the activeNet-Net Diameter Director. The first interval occurs after initial synchronizations of the systems; this is the timeout for subsequent synchronization requests. The default is 1000. The valid range is:

- Minimum—0
- Maximum—999999999

10. Save your work using the ACLI **done** command.

Subscriber Statefulness

The Net-Net Diameter Director can operate in either a session or subscriber stateful mode for each unique application specified in received messages. In addition to session statefulness where the Net-Net Diameter Director binds a session ID to a next hop destination, the Net-Net Diameter Director can maintain subscriber statefulness too. Subscriber statefulness binds a Subscriber ID to a next hop. By appropriately formatting AVPs with subscriber information, Diameter requests for the same subscriber may be routed to the same destination Diameter agent that was initially selected by the Net-Net Diameter Director routing algorithm. Subscriber statefulness also supports categories that are used to direct all of one subscriber's messages, even with different application IDs, to the same server.

The subscriber cache works in tandem with the session cache. When the Net-Net Diameter Director receives a message, matching the application element's attributes and with the state element set to subscriber, it then attempts to create a session by matching the conditions listed in the alloc' definition in the XML. After matching the alloc definition, the session is created in the session cache. For this session, the Net-Net Diameter Director then attempts to parse the subscriber key from the message (as per the XML definition). If successful, the new session queries the subscriber cache entry, created and linked to this session, for the routing information (i.e. the next hop). The subscriber cache returns the cached next hop or performs a policy lookup to find and return the next hop. Either way, a subscriber stateful application will learn about the next hop from the subscriber entry. If the session is not subscriber stateful, the session reverts to transaction stateful.

Lookup failures at any point in the state machine are handled similarly to the session cache functionality. All OOS processing which occurs during session stateful operations affects the policy result associated with the subscriber cache.

XML File Format

Using the same XML file required for session statefulness, the Net-Net Diameter Director is also directed how to provide subscriber statefulness. The XML file defines when to start caching a session-id and next-hop together, when to perform a policy lookup again for messages with that session ID, and when to remove the cache entry for that Session ID. In addition, the XML also creates categories, used to route messages of grouped applications to the same Diameter agent.

The Session & Subscriber state file should be placed in the /code/ directory. See the Session Stateful section for more information. There is no required naming convention for this file. The following is an example of an XML that creates subscriber state:

```
<application name='Gx' id='16777224' category='pcrf' >
```

```
    <state>subscriber</state>
    <alloc value='272'>
      <avp code='416' type='enumeration'>1</avp>
    </alloc>
    <dealloc value='272'>
      <avp code='416' type='enumeration'>3</avp>
    </dealloc>
    <action>
      <sessionMiss code='5002'>reject</sessionMiss>
      <subscriberMiss code='5002'>reject</subscriberMiss>
    </action>
    <subscriber>
      <avp code='443' type='grouped'>
        <avp code='450' type='enumeration'>1</avp>
        <avp code='444' type='octet-string' key='true'/>
      </avp>
    </subscriber>
  </application>
```

Defining the Subscriber Cache Entry

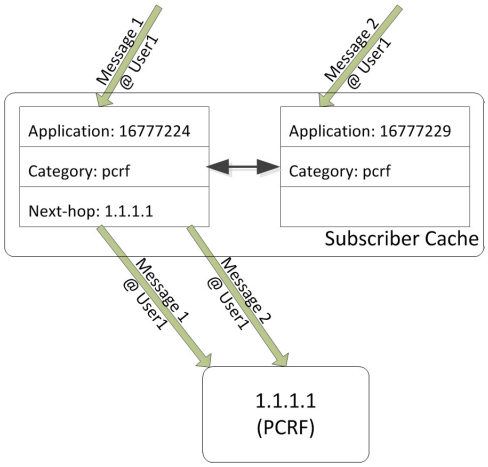
From the session stateful XML file baseline, there are three additions to the XML file that create subscriber statefulness.

1. Within the application attribute, you define the category. Categories are used to route messages of an applications to the same Diameter server.
2. The state element, configured as subscriber, lets you define an application/category association.
3. The subscriber element defines the AVPs in the Diameter message which contain the subscription identification. The subscriber key is used to locate subscriptions in the subscriber cache.

application

The *category* attribute in the application element is required to enable the subscriber state cache. You create a category to encompass multiple applications. By doing so, you can send all Diameter sessions with different applications from one subscriber in the same category to the same next hop (server). For example, if the Gx and Rx applications share the same category name, when a subscriber sends either message type (Gx or Rx), the

messages are routed to the same Diameter Director Policy next-hop, without performing an additional Diameter Director Policy lookup on subsequent messages.



state

Set the state element to 'subscriber' for the Net-Net Diameter Director to create and maintain subscriber state for the alloc and dealloc rules you place next. For a unique application, defined in the XML's application element, the statefulness maybe be configured (and maintained) as subscriber or session. However, an application may only be subscriber stateful if it is internally created as session stateful first.

Subscriber statefulness is managed on a session-by-session basis. For example, if a message triggers a dealloc of user and next-hop binding, the session is removed from the cache, but the subscriber remains cached. When no more sessions remain for a subscriber, and the subscriber cache entry timer reaches 0, that subscriber cache entry is removed.

subscriber

The subscriber element uses the subscriber key used to identify an individual subscriber cache entry. Subsequent messages must match the key created in the subscriber element to be forwarded to the cached next hop; The subscriber cache associates applications with next hop destinations.

In the initial subscribe element you define the message code that begins the cache entry creation or lookup process. Within the subscriber element you configure the matching parameters as AVPs, data type, and AVP value that filter for matching this subscriber. Finally you need to indicate how the key is formed. This is done by adding the key='true' element, where the AVP's data becomes an element of the key. Multiple key creating elements may be used, and the Net-Net Diameter Director concatenates all of them, in listed order as the complete subscriber cache entry key. For example:

```
<subscriber code ='443' >
              <avp code='450' type='enumeration'>0</avp>
              <avp code='444' type='utf8string' key='true'/>
</subscriber>
```

statemachine

For setting a subscriber cache lifetime, with no ongoing sessions, you may configure a subscriber-timeout='xx' attribute in the statemachine element. This value indicates how long a subscriber cache entry exists after no sessions remain for that entry, before it is removed from the cache. The default value, if not configured is 0 seconds.

action

The action element specifies what actions to take under certain conditions. You can configure a subscribermiss action which indicates whether to reject or proxy the request if no hit is found in the cache. This applies only to the messages with command-codes other than alloc element. When setting the subscribermiss action to reject, you specify the reject code in the following example:

```
<action>
<subscribermiss code='3002'>reject</cachemiss>
</action>
```

If no code is configured, then the default Result Code AVP, 3002 (UNABLE_TO_DELIVER) is sent in case of reject. Action tag is optional and reject is the default action taken on subscribermiss condition. If the subscribermiss action is proxy, then the Net-Net Diameter Director reverts to being transaction stateful and a Subscriber unaware Diameter Director Policy lookup ensues.

Subscriber-only Statefulness

Both the session and subscriber state machines depend on the establishment of a session to bind session-ID to a next hop in the system's cache. To enable a state machine for applications that are not session stateful, including s6a, the Net-Net Diameter Director includes a state machine model called subscriber-only statefulness. This state machine allows the Net-Net Diameter Director to capture and store a routing context for an s6a subscriber without using a session-ID.

When the Net-Net Diameter Director receives the first message for the subscriber-only application for a subscriber that is not already cached, it determines the destination for this request based on a policy lookup. It then caches the result of this lookup and the source of the request in the subscriber table by binding the subscriber ID to the next hop. There is no attempt to create a session for these messages. Subsequent messages are simply forwarded based on a match between subscriber ID and the next-hop defined in the subscriber cache. Processing for recursive routing, OOS agents and OOB requests is the same as described for the session stateful state machine.

This feature is primarily implemented so that the Net-Net Diameter Director can mask the origin host (OH), origin realm (OR), destination host (DH) and destination realm (DR) AVPs, similarly to session and subscriber statefulness, and still being able to store a routing context for messages that do not include a session-ID.

Subscriber-only Cache Entry Aging

Given the absense of any session data, it is not possible for the subscriber-only state machine to use the dealloc tag. For this reason, the expiry of the application's `max-inactivity-timer` or the `subscriber-timeout`, configured in the subscriber-only state machine, are the only ways to remove a subscriber-only binding from the subscriber cache.

XML File Format

Subscriber-only stateful mode operates using the same XML state machine file as is used for session and subscriber stateful mode. You configure applications to use the subscriber-only state machine within this file.

Recall that there is no session associated with subscriber-only statefulness, and therefore no session-ID within the messaging. The subscriber-only state, therefore, excludes the following tags:

- `alloc`
- `dealloc`

- realloc
- action > cachemiss

With the exception of the above, the subscriber-only state uses the same elements as the session and subscriber states. The following is an example of the XML that creates subscriber-only state:

```
<statemachine max-inactivity-time='86400'>
  <application name="s6a" id='16777224' category="hss" max-inactivity-
time='43200'>
<state>subscriber-only</state>
         <subscriber code='443'>
            <avp code='450' type='enumeration'>0</avp>
            <avp code='444' type='utf8string'key='true'/avp>
         </subscriber>
         <action>
            <subscriberMiss code='3002'>reject</subscriberMiss>
         </action>
   </application>
</statemachine>
```

subscriber-only state

Set the state element to 'subscriber-only' for the Net-Net Diameter Director to create and maintain subscriber-only state for the specified application.

SubscriberMiss action

The cache-miss action cannot apply to the subscriber-only state machine. A similar action, called subscriber-miss, handles the cases where the system receives a subscriber-only message that does not include enough information to determine the key needed to determine the next hop. Actions include:

reject - The Net-Net Diameter Director replies to the sender with the configured error code.

proxy - The Net-Net Diameter Director handles the message as a transaction stateful proxy and attempts to proxy the message.

4 Load Balancing & Redundancy

Diameter Director Groups

A Diameter Director Group combines several Diameter Director Agents into a single logical entity. Each configured Diameter Director Agent may only belong to one Diameter Director Group.

The Diameter Director Group is defined by the *name* parameter. You can also provide a *description*. All Diameter Director Agents that comprise the Diameter Director Group are added to the *destinations* parameter. The Diameter Director Group can then be a next-hop in a Diameter Director Policy. This is configured by prepending the Diameter Director Group name with a `ddg:` prefix. For example: `ddg:exampleddg`

When the next-hop is a Diameter Director Group and if one of the Diameter Director Agents in the group becomes unavailable, the Net-Net Diameter Director can still communicate with the peer via an alternate Diameter Director Agent. Where the Net-Net Diameter Director communicates with multiple peers, the Diameter Director Group also provides the ability to distribute the traffic among them depending on the message distribution strategy.

Diameter Director Group Strategy

When the Net-Net Diameter Director settles on sending a Diameter message to a *next hop* via a Diameter Director Policy lookup, the *strategy* parameter in the *diameter-director-group* indicates how to choose the discrete Diameter Director Agent in that group. Diameter Director Agent selection is performed on a per-message basis. The two basic strategies for Diameter Director Agent selection are:

- Hunt—The Diameter Director Group selects the Diameter Director Agent in the order they were configured. If the first agent is in service, then it is selected. If the first agent is out of service, then the second or the next available Diameter Director Agent is selected. As long as the first agent remains in service, all incoming requests to the Diameter Director Groupare forwarded there.
- Round Robin—The Diameter Director Group distributes incoming requests among all Diameter Director Agents in the order they were configured. The Diameter Director Agent selection begins with the first configured agent which is in service. Upon receiving the next request, the next Diameter Director Agent is selected and so on. This strategy distributes incoming requests among all Diameter Director Agents uniformly that comprise the Diameter Director Group.

In addition to hunt and round robin, there are other, more advanced Diameter Director Agent selection strategies. They are explained as follows.

Persistent-Hunt Strategy

The persistent hunt strategy is a variation on the hunt Diameter Director Group strategy. When the Net-Net Diameter Director needs to forward a new message into a Diameter Director Group, it continues forwarding messages to the same Diameter Director Agent until it goes Diameter Director Agentout of service. This contrasts with the hunt strategy in which the Net-Net Diameter Director attempts to send every new message to the Diameter Director Agent at the top of the list.

For example, the Net-Net Diameter Director is sending messages to a Diameter Director Group with 2 members and set to persistent hunt. Member 1 goes OOS and the Net-Net Diameter Director starts sending messages to member 2. When member 1 returns to service, the Net-Net Diameter Director continues sending messages to member 2. In standard hunt strategy, when member 1 returns to service, new messages will be directed to member 1.

Rate Load Balance Strategy

Rate load strategy is when the Diameter Director Agent selected from the Diameter Director Group is based on the agent with the lowest relative traffic rate. The Net-Net Diameter Director computes relative traffic rate using message or transaction metrics. Load rate computations only use in-service Diameter Director Agents. If all configured Diameter Director Agents in a Diameter Director Group are configured with the **max sustain rate** parameter configured, then message rates are used for load determination. The chain of configuration for this would be max sustain rate configured in message constraints configuration element and that message constraints object applied to the constraint name parameter in the Diameter Director Agent.

If one or more Diameter Director Agents in a Diameter Director Group is not configured with the **max sustain rate** parameter, then the Net-Net Diameter Director defaults to making its rate load determination based on transaction rate.

Message rate for each Diameter Director Agent is computed according to:

$$\text{message load} = \frac{\text{current sustained message rate}}{\text{max-sustain-rate value}}$$

The Net-Net Diameter Director compares the rate load of all Diameter Director Agents in the group according to the above equation and sends the message to the Diameter Director Agent with the lowest load. In this situation, the Diameter Director Agents with higher capacity (max-sustain-rate) can absorb a higher sustained traffic load, so eventually all the traffic directed to this Diameter Director Group is distributed evenly across the member agents.

Transaction rate is computed similarly. The Net-Net Diameter Director picks the agent with the lowest on-going transaction rate according to:

$$\text{transaction load} = \frac{\text{\# transactions on this agent}}{\text{transactions sent to the whole DD group}}$$

Thus the Net-Net Diameter Director sends messages to the least loaded Diameter Director Agent. Eventually all the traffic directed to this Diameter Director Group is distributed evenly across the member agents.

Recursive Routing and Rate Load Balancing

When recursive-routing is enabled, the Net-Net Diameter Director must first create the recursive list, sorted by load. It then attempts to forward the message to the Diameter Director Agent with the lowest load first.

Diameter Director Group States

A Diameter Director Group's Diameter Director Agent members exist in one of the two states: In Service or Out Of Service. At startup, Diameter Director Agents are considered OOS until the Net-Net Diameter Director establishes a successful Diameter connection with the configured Diameter Director Agent. The Diameter connection is defined as a valid transport layer connection followed by a successful Capability exchange causing the connection to be kept open. Once In Service, the Diameter Director Agent can return to Out of Service if the Diameter connection is lost.

When the Net-Net Diameter Director selects a Diameter Director Agent from a Diameter Director Group as the destination for a Diameter message, if it is Out of Service it is never chosen. Once a Diameter connection is re-established, the Diameter Director Agent is considered In Service again and can participate in the Diameter Director Group selection process.

Diameter Director Group startup and shutdown

When the Net-Net Diameter Director's Diameter Director Group configuration is modified (or removed completely) and then reloaded, the connections from the Net-Net Diameter Director to the Diameter Director Agents in that group are all closed and renegotiated from the CER/CEA forward.

Any application IDs from the Diameter Director Group configuration will be used for negotiation for all Diameter Director Agent, unless the Diameter Director Agent has its own configured application ID.

To initiate the process of closing and opening connections to Diameter Director Agents, the Net-Net Diameter Director sends DPRs to all affected agents. After the DPA is received, the transport layer connection is reset. If the Net-Net Diameter Director does not receive a DPA within 32 seconds of sending the DPR, the peer is deemed unresponsive and the connection is closed.

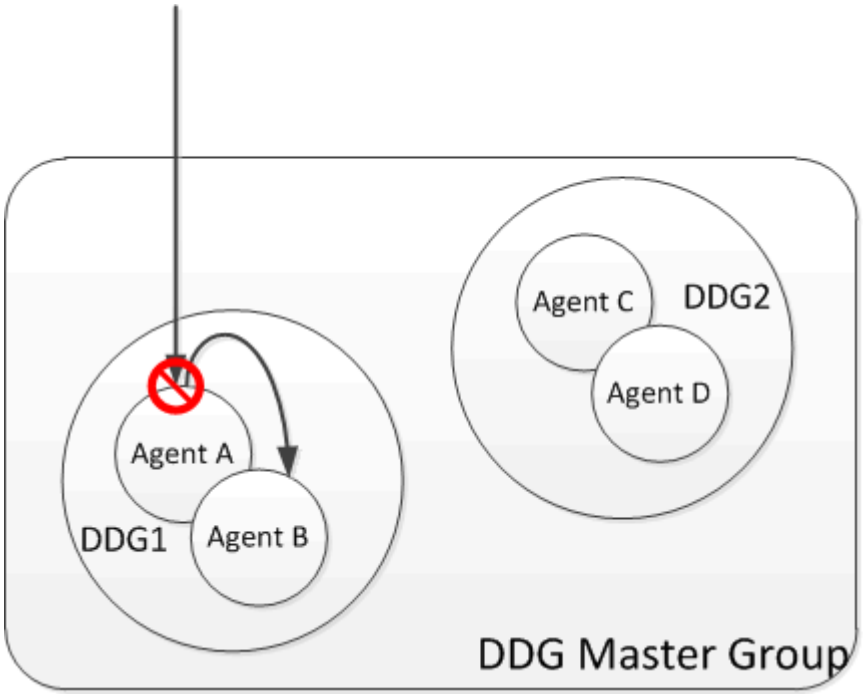
Nested Groups

Nested Diameter Director Groups provide a way, when used in conjunction with the Session Stateful routing, to direct all messages with the same Session ID to a specific server(s) in a failover situation.

A nested Diameter Director Group is created by configuring a Diameter Director Group as a destination within a parent Diameter Director Group. The Net-Net Diameter Director will send messages to members within a subgroup when the initially chosen next-hop agent fails. Failovers are initially confined to the subgroup parent of the agent chosen.

In the following diagram, because Agent A and Agent B are members of the nested subgroup DDG1, when Agent A fails, the Net-Net Diameter Director sends subsequent messages with the same Session ID to Agent B. Without nested Diameter Director

Groups, you could not ensure maintaining state between messages with one Session ID and a specific server or set of servers on Agent A's failure.



You may configure multiple Diameter Director Agents in a nested group. You may also configure multiple nested groups within one master group. The previous example of nested Diameter Director Groups is configured in the ACLI as follows:

```
diameter-director-group
        group-name                      DDG-Master-Group
        state                           enabled
        description
        strategy                        round-robin
        destinations
                                        ddg:ddg1
                                        ddg:ddg2
        […]

diameter-director-group
        group-name                      ddg1
        state                           enabled
        description
        strategy                        round-robin
        destinations
                                        AgentA
                                        AgentB
        […]

diameter-director-group
        group-name                      ddg2
        state                           enabled
        description
        strategy                        round-robin
```

```
destinations
                                    AgentC
                                    AgentD
[…]
```

In the above ACLI configuration, sessions will be initially split between DDG1 and DDG2 in a round robin strategy. It is useful to configure the master Diameter Director Group using round robin so that all initial requests are balanced between the configured nested groups.

When a session is established into a destination in DDG1 and eventually all members of that group become unreachable, only then will the Net-Net Diameter Director start sending Diameter messages to DDG2.

Single Agent Fallback

In some situations, if all the routes into a nested group fail, it may be useful to fallback to a reserved, singleton agent. Thus it is valid to configure a Diameter Director Group(s) and a single Diameter Director Agent.

In single agent fallback situations, it is useful to set the master Diameter Director Group's strategy to hunt so that all messages are forwarded first to the Diameter Director Group, and only when all Diameter Director Agent members of that group are unreachable will messages be forwarded to the fallback agent.

ACLI Instructions

To configure a Diameter Director Group:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-group** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-group
   ACMEPACKET(diameter-director-group)#
   ```

4. **group-name**—Enter the name of this Diameter Director Group. The value entered here will be referenced from a Diameter Director Policy next-hop.
5. **state**—Set this parameter to **enabled** to use this Diameter Director Group.
6. **description**—Enter a description of this Diameter Director Group enclosed in quotes.
7. **strategy**—Set this parameter to either **hunt** or **round-robin** to indicate which way a Diameter Director Agent is chosen for sending messages to.
8. **destinations**—Enter the *hostname* value of a Diameter Director Agent to add that agent to this Diameter Director Group. Multiple values are entered in parenthesis separated by spaces. Individual values are added or removed with a + or - operator. For example:

   ```
   ACMEPACKET(diameter-director-group)#destinations +agent007
   ```

9. Save your work using the ACLI **done** command.

Active/Active Redundancy

The Net-Net Diameter Director offers an Active/Active redundancy mode which enables two or more Net-Net Diameter Directors to share all session and subscriber state information. This architecture enables one Net-Net Diameter Director to assume the other Net-Net Diameter Director's traffic incase it goes out of service. In addition to a Net-Net Diameter Director going OOS, an Active/Active failover can happen when connectivity in one peer's end-to-end session is lost. The remaining Active peer assumes the failed peer's role in routing and forwarding subsequent traffic.

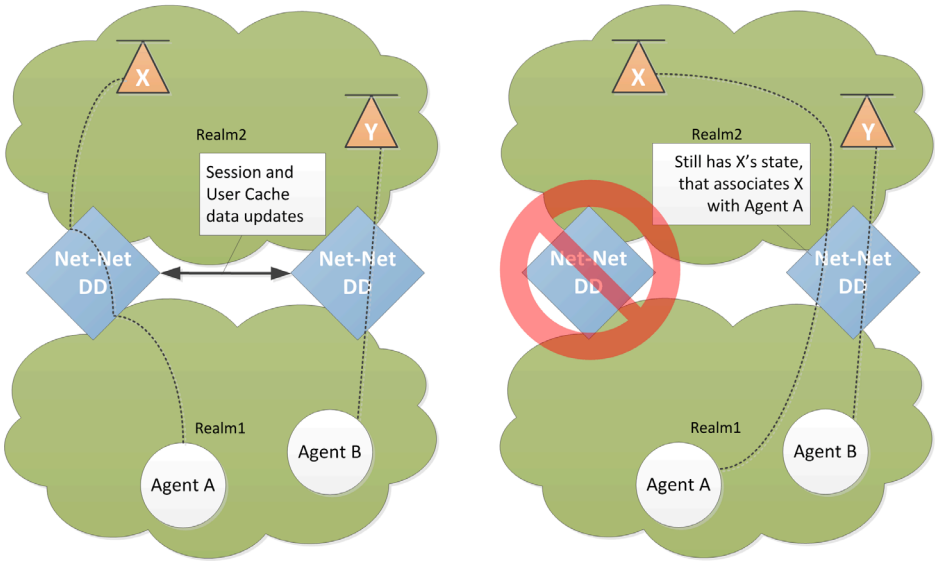
In an Active/Active pair, each Net-Net Diameter Director active system must be able to assume the traffic capacity of the other Net-Net Diameter Director system in the event that traffic is diverted from the other Net-Net Diameter Director active system. The two systems must also be configured to view and participate in the network identically.

The two nodes in the Active/Active redundancy feature both perform all Net-Net Diameter Director functions simultanously. Each node services its own set of traffic as directed to it from the network. Between the Active/Active peers, session and subscriber state information is synchronized: when new sessions are created or session states change, the information will be pushed to the peer.

Example Scenario

In the following diagram, the image on the left represents the baseline scenario. Both Net-Net Diameter Directors are configured with Realm 2 and Realm 2, and Agent A and Agent B with their unique IP addresses. One session is set up from X to Agent A through the "left" Net-Net Diameter Director. A second session is set up from Y to Agent B through the "right" Net-Net Diameter Director.

At some point connectivity along the path from X to Agent A is lost. In an Active/Active deployment, the "right" Net-Net Diameter Director has a copy of X's state information. X must be aware on its own that the left Net-Net Diameter Director (or Agent A) is unreachable and reconnect to the "right" Net-Net Diameter Director. The "right" Net-Net Diameter Director has the state information for X and without performing a Diameter Director Policy lookup, assuming all things are equal, will forward X's messages directly to Agent A.



Synchronization Process

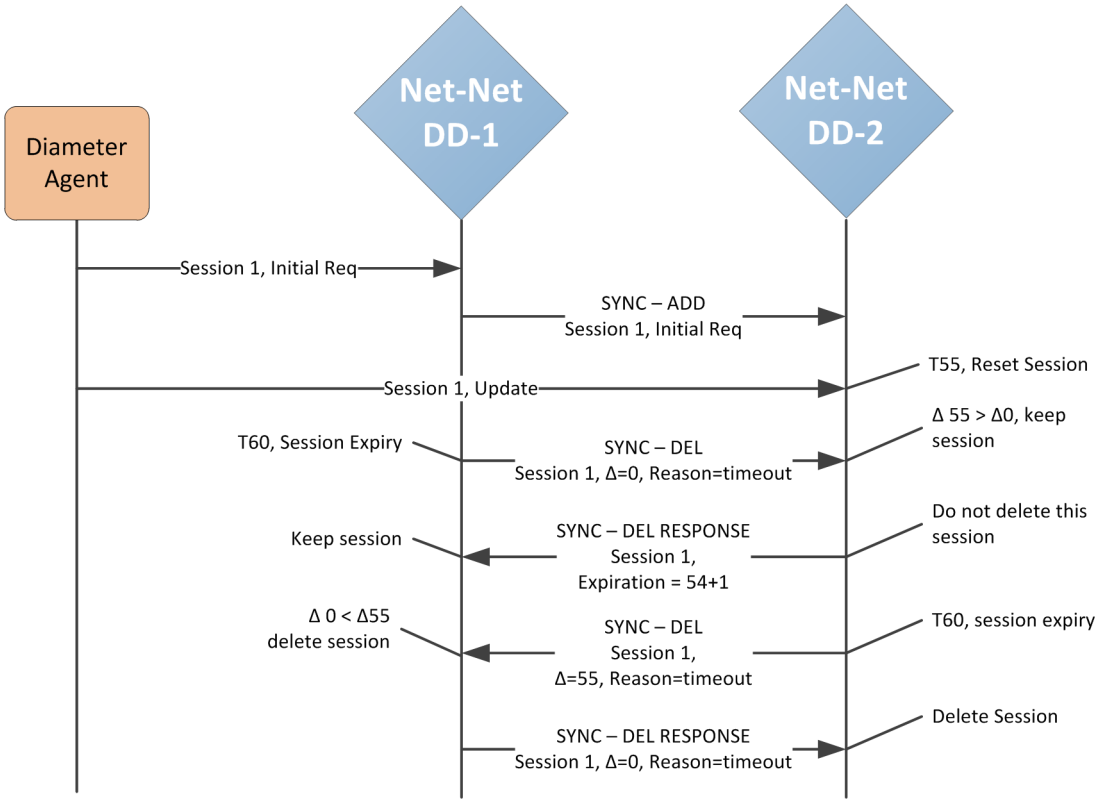
Active/Active nodes perform a software handshake and synchronize all session and subscriber information. After the synchronization process completes, both nodes are considered In-Service by their compliment.

Whenever a new session and/or subscriber is created, deleted, or experiences a state change, an update is pushed to all peers which populates that Net-Net Diameter Director with the state change.

Cache Aging for Expired Sessions

Since the same session exists on both Active/Active peers at the same time, both the peers maintain independent session inactivity timers. The inactivity timer is reset when a new request is received for the session. If all requests for a session are sent to the DD-1, then the inactivity timer will expire sooner on the secondary peer causing the session to be deleted. In addition, if some messages are send to one active peer, and other messages are sent to the other peer, the session timers could become out-of-sync.

In order to resolve this issue, the timers run and expire independently on the two nodes. When an update message is pushed to the peer, it contains a delta value that reflects the time between the session's creation and the last message received. When no message (besides initial creation message) is received, the delta is zero. Refer to the following diagram as the example continues:



When the timer expires on DD-1, it sends a delete request to DD-2. If DD-2's delta is greater than the one included in the delete message, it assumes that it has received an in-session message more recently than the other Active has. In this case, the second active updates the first active with a message indicating to maintain (and not flush) the session.

Along with the do-not-delete response message, DD-2 sends the new expiration time plus 1 second. This allows the timer on the second active to expire before the timer on the first active, thus creating a graceful termination of the session between the systems.

DD-1, upon receiving the delete response maintains the session using the new expiration time. Finally, the expiration timer on tDD-2 expires and a delete message is sent to DD-1, with a delta reflecting the time since the secondary received an in-session message. Since DD-1's delta (0) is lower than the DD-2's delta (55) the session ends up being deleted on both ends.

This logic occurs for session timeouts as noted by the reason header in the delete message. For a different reason header or different circumstance, the usual session deletion procedure is adopted.

Node Failure

Individual nodes in an Active/Active configuration will be set to OOS upon a heartbeat timeout, as seen from the node that remains online. In this case, the system fails to receive 5 consecutive hearbeat messages, and is then considered OOS by the other peer. That peer is considered back in service when heartbeat messages appear again. The **heartbeat interval** parameter determines the number of milliseconds between a peer sending subsequent heartbeat messages. This value x 5 is the length of time the in-service system takes to deterime a peer is OOS.

The consequences of a system seeing a peer as OOS for both failure cases is that state information is not sent to (and from) the OOS system.

Recovery

The Net-Net Diameter Director utilizes a **resume time** parameter wait after a system receives heartbeats, as it exchanges state information, before it considers its peer in-service. This value is also be considered as the Net-Net Diameter Director boots up; it is a safety margin before the Active/Active pair can provide redundant services to the network.

Configuration Guidelines

This section lists system requirements to create an Active/Active redundant configuration.

1. The Active/Active nodes must be connected to each other on an NIU media port. We recommended you dedicate a media port for the Active/Active redundancy traffic.
2. All nodes should run the same version of the software.
3. All nodes must rely upon the same network-centric logical configuration. This means that all instances of the following configuration elements are configured identically. The information synchronized between the peers relies on some of the configuration names being the same.
   - diameter-director-config
   - diameter-director-interface
   - diameter-director-policy
   - diameter-director-agent
   - diameter-director-group
   - diameter-manipulation
   - diameter-director-constraints

   Other network elements like realms, and IP addresses must also be identically configured.

4. All nodes must have unique signaling IP addresses. Each active node should have a unique path to the same set of diameter nodes.
5. All nodes must have enough traffic capacity to assume the traffic processing of its peer in case of a failover.

Active/Active Configuration

Active/Active redundancy is configured across three configuration elements:

session-router > diameter-director-config

session-router > active-active-redundancy

session-router > active-active-redundancy > active-peers

In the diameter director configuration element, you enable this feature with the active redundancy parameter and set the port through which Active/Active peers communicate with the active redundancy port parameter

In the active active redundancy configuation element you set the values that are discussed and determine Node Failure (74) conditions. You also set the realm id in which redundancy data is shared. This is the realm that corresponds to the media ports through which connectivity to the other Active/Active node(s) is achieved.

In the active peers subelement, accessible from the active active redundancy configuration element you define each individual peer with a symbolic name, activity state, IP address, and identification whether the configuration element is this Net-Net Diameter Director or not.

ACLI Instructions

To enable Active/Active Redundancy on the Net-Net Diameter Director:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session-router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-config a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-config
   ACMEPACKET(diameter-director-config)#
   ```

4. Type **select** and press <Enter>.
5. **active-redundancy**—Set this to **enabled** to user this feature.
6. **active-redundancy-port**—Accept the default of port 9010, or enter your own.
7. Type **done** to save your work.
8. Navigate to the active-active-redundancy configuration element:

   ```
   ACMESYSTEM(diameter-director-config)# exit
   ACMESYSTEM(session-router)# exit
   ACMESYSTEM(configure)# system
   ACMESYSTEM(system)# active-active-redundancy
   ACMESYSTEM(active-active-redundancy)#
   ```

9. state—Set this parameter to enabled to use Active/Active redundancy

10. heartbeat-interval—Keep the default value of 1000 or enter your own value (in milliseconds) the Net-Net Diameter Director waits between sending Active/Active peers heartbeat messages
11. resume-time—Keep the default value of 3000 or enter your own value (in milliseconds) the Net-Net Diameter Director waits before setting itself back in service after receving heartbeats, or crossing above the health-threshold.
12. realm-id—Enter the logical realm name which the Active/Active peers reside.
13. Type active-peer **a**nd press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.
14. name—Enter a symbolic name for this Active peer's configuration object. This name does not need to refer to any other specified parameter.
15. state—Set this parameter to enabled to use this peer.
16. address—Enter the IP address of this peer.
17. is-local—Set this parameter to enabled if this configuration object refers to this Net-Net Diameter Director, as part of the Active/Active pair.
18. Type **done** to save your work.
19. Type **exit** and **done** to save your work.
20. Repeat stps 15-20 to configure other Active/Active nodes. Remember that remove nodes should have their is-local parameter set to disabled.

Software Upgrades

When upgrading the software on Active/Active pairs, you must set the pair you are upgrading first to OOS. This stops synchronizing rudundant data to the peers and indicates to the peers that this system is going OOS.

The three high-level steps for upgrading nodes in an Active/Active configuration are:

1. Enter the new bootparamers to reload from a new Net-Net Diameter Director image.
2. Type the **notify ddd active-active-oos** command. This sets the node you are executing the command on to OOS state, and alerts other nodes.
3. Reboot this system, it will load the new boot image and begin the synchronization process with other systems configured as Active/Active nodes.

Failover Cases

There are several events which could prompt the Net-Net Diameter Director in Active/Active Redundancy mode to experience a failover state. The following are some scenarios which could help an external device detect a failover. This section is provided as architectural reference.

Unresponsive Node

When an Active node stops responding to traffic and to DWR/DWAs, the Diameter agent can ascertain that the Active node is now OOS. It should assume that the secondary Active node is capable of handling in-session requests for the sessions served by Active 1 before. The state of Active 1 might still be in-service in terms of redundancy, but it is not responding to the Diameter agent's requests. Once the Active node starts responding to these requests, the Diameter agent can consider it is in-service again and begin sending traffic to that node.

Overloaded Node

When an Active node's traffic constraints are exceeded, it sends a 3004 (DIAMETER_TOO_BUSY) or similar protocol error (as configured by the user) to the

requesting Diameter agent. Such cases are for CPU load limit or message constraints exceeded. An overloaded Active node becomes temporarily unavailable and Diameter agents can re-route their requests to another Active node. The Diameter agent can assume that secondary Active is capable of handling in-session requests for the sessions served by Active 1 before.

Unreachable Node

When a Net-Net Diameter Director returns a 3002 (DIAMETER_UNABLE_TO_DELIVER) protocol error to a requesting Diameter agent, it is indicative that the Net-Net Diameter Director is unable to find an egress route for the message, such as when it loses connection to the destination Diameter agent. In addition, this error might be sent to a Diameter agent when the Net-Net Diameter Director times-out waiting for a response from the destination server. In both cases the requesting agent should re-route its request to the other Active node and expect the message to be successfully routed to the intended destination.

5 Diameter Message Manipulations

Diameter Message Manipulations

The Net-Net Diameter Director can perform manipulations on all grouped and non-grouped AVPs. This is referred to as Diameter Manipulation Rules (DMR). A message manipulation is the ability to search for a predefined string within an AVP and then replace it with another value. This is similar to the Net-Net SBC's header manipulation rules functionality.

Diameter Manipulations can be applied at one of three logical points in the configuration as the message is received or forwarded: Diameter Director Interface, Diameter Director Group, Diameter Director Agent. The Diameter Director Agent has the highest priority, the Diameter Director Group (if configured) has the second highest priority, and Diameter Director Interface has the lowest priority. If more than one manipulation is configured for the trip from Diameter Director Interface to Diameter Director Agent, the manipulation with the highest priority is executed.

A *diameter manipulation* configuration element is defined by a *name* parameter. You can optionally add a *description* field to the diameter manipulation. Within each diameter manipulation you can configure multiple *diam manipulation rule* subelements. The manipulation rule subelements are the configuration where AVPs are identified, searched, and in which the data is replaced.

Manipulation Rule

Creating a manipulation rule is divided into three parts, defining the message type and AVP where the manipulation is performed, defining how the search on the AVP is performed, and defining what to replace the found string with.

You must first define the *name* parameter of the *diam manipulation rule* configuration element. Optionally you can add a d*escr avp code* parameter that is a description of this manipulation rule.

Naming Diameter Manipulations

The Net-Net Diameter Director restricts the way you can name a *diameter-manipulation rule*. Specifically, observe the rules below when naming manipulation elements:

- Character limit - *diameter manipulation rule* names cannot be longer than 24 characters.
- Numeric characters - *diameter manipulation rule* names must not start with a numeric character.
- Special characters - Special characters are not supported within *diameter manipulation rule* names, with the exception of the underscore and hyphen characters.
- Capital letter characters - The Net-Net Diameter Director includes reserved keywords that are named using all-capital letters. To avoid conflicts between keywords and diameter manipulation rules, do not configure *diameter manipulation rule* names using ll capital letters.

Note that, although diameter-manip-rule and avp-header-rule names have the same character-type restrictions, they do not have a character limit.

Message Based Testing

When the Net-Net Diameter Director first receives a message applicable for manipulation, it checks if the message type as **request**, **response**, or **all** as configured in the *msg type* parameter. The Net-Net Diameter Director continues to look at the message command code. Matching values are defined by configuring the *msg cmd code* parameter with a numeric value. You can enter a single value, multiple comma-separated values, or you can leave this parameter blank to indicate all message codes.

AVP Search Value

After the Net-Net Diameter Director has identified the messages where it can look for an AVP, the *avp code* must be defined with a numeric AVP value to be searched. Also the AVP data type is defined so Net-Net Diameter Director knows how to correctly parse the AVP once found. This is configured in the *avp type* parameter with valid values of: octet-string, octet-hex, integer32, unsignedint32, address, utfstring, diameteruri, or enumerated.

The *comparison type* is defined so that the Net-Net Diameter Director knows the correct way to determine if the *match value* appears in the *avp code*. Valid *comparison types* are:

- Case-sensitive—The comparison-type of both case-sensitive and case-insensitive literally compares the value contained in the match-value against the received value.
- Case-insensitive—The comparison-type of both case-sensitive and case-insensitive literally compares the value contained in the match-value against the received value.
- pattern-rule—the match-value is treated as a regular expression.
- boolean—Used when it is necessary to compare the results of two or several manipulation rules with varying logic (e.g. if ($rule1 & ($rule2 | $rule3))). When the comparison-type is set to boolean, the match-value will be evaluated as a boolean expression.

Finally, the match operation is configured by defining a *match value*, which is the string to find. The Net-Net Diameter Director evaluates if the *match value* is found in the *avp code* AVP. You may also leave the *match value* empty for the DMR to be applied on the AVP without testing for a match.

Reserved Keywords

The Net-Net Diameter Director employs certain reserved keywords as a short hand for configuration/message parameters. These keywords are as follows:

HOSTNAME—This keyword refers to the agent hostname this rule is being referenced by. If the rule is applied to a realm/interface then the value of the hostname keyword will be an empty string. If the rule is applied to the group, then the hostname for the agent picked will be used.

ORIGINREALM—This keyword refers to the Origin-Realm value for the configured realm/interface. If the rule is applied to a Diameter Director Agent, then the origin-realm value is derived from the Diameter Director Interface the agent belongs to.

ORIGINHOST—This keyword refers to the Origin-Host value for the configured realm/interface. If the rule is applied to a Diameter Director Agent, then the origin-host value is derived from the Diameter Director Interface the agent belongs to.

Actions on Found Match Value

When the *match-value* is found, the Net-Net Diameter Director references the *action* parameter. This is configured as either **none**, **add**, **delete**, **replace**, **store**, **diameter-manip**, **find-replace-all**, **log** or **group-manip** and indicates the action to perform on the string. If the match-value is not found, the Net-Net Diameter Director continues

processing the message without any AVP manipulation. These actions mean the following:

none None disables a manipulation rule.

add This action inserts the value from the **new value** parameter, creates a new AVP of the specified type and adds it to the list of AVPs at the specified position.The message length and padding are re-computed to account for this newly added AVP.

delete This action removes the specified AVP from the list of AVPs being operated on. The message length and padding will be re-computed to account for this deleted AVP.

replace This action substitutes the existing value with the **new value** parameter. The message length and padding and AVP length and padding will be re-computed to account for changes. This is mostly applicable to variable length AVP types such as octet-string.

store Each manipulation rule has the ability to store the data that was contained in the AVP as a string. This is useful for creating conditional logic to make decisions whether to execute other manipulation rules or to duplicate information within the Diameter message.

Every manipulation rule performs an implicit store operation prior to executing the specified action type. All store operations are based on regular expression patterns configured in the **match value**. The information that is stored in the rule is the resultant of the regular expression applied against the specified string. The **comparison-type** is ignored when the action is set to store as the Net-Net Diameter Director assumes that the **match value** is a regular expression. Conditional logic cannot be used to make a decision whether to perform a store operation or not; storing always occurs. Values stored in a manipulation rule are referred to as “user defined variables”.

diameter-manip When the action is set to **diameter-manip**, the Net-Net Diameter Director executes the diameter-manipulation **name** configured in the **new value**. The diameter-manipulation name in the **new value** must match another diameter-manipulation name exactly (case is sensitive).

diameter-manip action type is primarily to reuse diameter-manipulations that may be common to other use cases. A diameter-manip action should never call back to itself either directly or indirectly through a different diameter-manipulation.

find-replace-all The **find-replace-all** action searches the object's string for the regular expression defined in the match-value and replaces every matching occurrence of that expression with the value supplied in the **new value**. If the regular expression contains sub-groups, a specific sub-group can be specified to be replaced by adding the syntax [[:n:]] at the end of the expression, where n is the sub-group index (zero-based). When the action is find-replace-all, the comparison-type is ignored and the match-value is always treated as a regular expression.

group-manip The **group manip** action is used to manipulate AVPs inside grouped AVPs. For this diameter manipulation, you must set the *avp-type* to **grouped**.

The **group manip** action is similar to the **diameter manip** action in that the Net-Net Diameter Director executes the diameter-manipulation configured in the new value.

There is an important difference between **group manip** and diameter manip. The **diameter-manip** action is context insensitive, meaning when it jumps from one diameter-manipulation to the next diameter-manipulation, it starts looking for the specified AVP from the top of the message.

The **group manip** action is context sensitive, meaning when the processing jumps from one diameter-manipulation to the next diameter-manipulation, it will look for the specified AVP within the grouped AVP. In short, the **group manip** works at an AVP level. All actions are allowed in the subsequent manipulations that are invoked, with the key difference being that those manipulation rules will be applied to the current grouped AVP in the chain. Thus it is possible to apply manipulation to an AVP at any level in the hierarchy.

Consider the following examples:

In order to replace the experimental-result > experimental-result-code AVP value from 5002 to 3002, a **group manip** can be configured as follows:

```
diam-manipulation
        name                            manipExpRslt
        description
        diameter-manipulation-rule
                name                            expRslt
                avp-code                        297
                descr-avp-code
                avp-type                        grouped
                action                          group-manip
                comparison-type                 case-sensitive
                msg-type                        response
                msg-cmd-codes                   316,317,318
                match-value
                new-value                       expRsltCode
        last-modified-by                admin@console
        last-modified-date              2011-09-13 18:50:33
diam-manipulation
        name                            expRsltCode
        description
        diameter-manipulation-rule
                name                            expRsltCode
                avp-code                        298
                descr-avp-code
                avp-type                        unsignedint32
                action                          replace
                comparison-type                 case-sensitive
                msg-type                        response
                msg-cmd-codes                   316,317,318
                match-value                     5002
                new-value                       3002
        last-modified-by                admin@console
        last-modified-date              2011-09-13 18:56:14
```

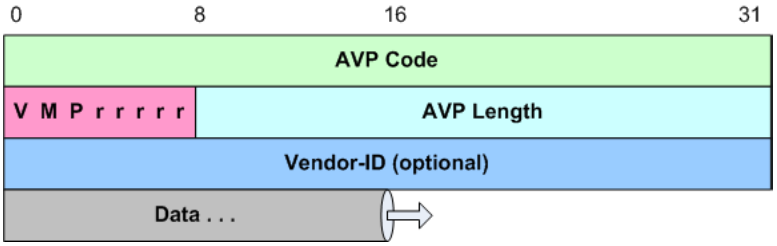
Further, if you want to add a new AVP called AvpD at the following location in the chain of AVPs Message: GrpAvpA > GrpAvpB > GrpAvpC > AvpD, then the manipulation chain would look like this

- diameter-manipulation (name=grpAvpA, action=group-manip, new-value=grpAvpB)
- diameter-manipulation (name=grpAvpB, action=group-manip, new-value=grpAvpC)
- diameter-manipulation (name=grpAvpC, action=group-manip new-value=AvpD)
- diameter-manipulation (name=AvpD action=add new-value="added new value")

Note: using diameter-manip from inside the group-manip chain may have unexpected impact and must be avoided.

AVP Header Manipulation

In addition to manipulating AVPs, you can manipulate the AVP header itself. After performing AVP DMR, the AVP length and padding is recomputed. This is crucial for scenarios where a vendor-id is added or removed from the header. This functionality is configured in the *avp header rules* sub element. The following represents a single AVP's header:



AVP Flag Manipulation

You can manipulate the AVP flags by configuring the **header-type** parameter to **avp-flags**, this invokes operation on the flags byte in the AVP header. AVP flags are 1 byte long, where the first 3 bits represent (1) vendor, (2) must and (3) protected. The last 5 bits are reserved.

The vendor flag is critical to consider here, since it has interdependency with Vendor-Id field in the header shown above. As per RFC 3588, The 'V' bit, known as the Vendor-Specific bit, indicates whether the optional Vendor-ID field is present in the AVP header. When set, the AVP Code belongs to the specific vendor code address space. Please find specific details about the rest of the flags in RFC3588 Section 4.1.

When manipulating AVP flags, a bitwise comparison is performed between the received value and the **match value**. For ease of configuration, the **match value** is configured as a comma-separated enumerated list of `vendor`, `must`, and `protected`. So the **new value** and the **match value** will be used to indicate what bit in the **avp-flag** to operate on. If the **match value** is empty, the configured action is performed without any matching tests. In addition, the new value is configured using the same enumerations. The *AVP header rules* configuration element appears as follows:

```
avp-header-rules
   name         replaceAvpFlags
   header-type  avp-flags
   action       replace
   match-value  must,protected
   new-value must
```

According to the example configuration, the Net-Net Diameter Director makes a positive match when only the must and protected bits are set in the received avp-flags. If only the 'M' bit is set, then the match fails, the Net-Net Diameter Director continues to the next header-rule.

When the match is successful (or if the **match value** is left empty), the configured **action** is performed. Consider all following actions applicable to the avp header rules sub element:

1. none— no action will be performed
2. add—the flags specified in the **new value** are enabled in the header, and state of the existing flags will not be changed.

   If **new value** is empty, the add operation will not be performed.

   If the **new value**=vendor, and the 'V' bit was not originally set, then the 'V' bit is now be set including a vendor-id of 9148 inserted into AVP. 9148 is the Acme Packet vendor-id assigned by IANA. It is expected that a second header-rule will be used to change this to the desired vendor-id.
3. replace—all the received avp-flags will be reset. The values in the **new value** parameter will be set.

   If the **new value** is empty, the replace operation will not be performed.

   If the **new value**=vendor, and the 'V' bit was not originally set, then the 'V' bit will now be set and also a vendor-id of 9148 (Acme Packet's vendor-id) is added to the AVP header. A second header-rule may be used to change this to the desired vendor-id.

   If the **new value** does not contain vendor, and the 'V' bit was originally set, then the 'V' bit will be cleared and the vendor-id will also be set to 0 effectively removing it from the AVP header.
4. delete—all flags configured in **new value**, will be deleted from the AVP header

   If the **new value** is empty, then no flags are deleted.

   If the particular flag is already empty, then it will be skipped. For example, if the **new value**=must and 'M' bit is not set, after applying the DMR the 'M' bit will still be not set.

   If the **new value**=vendor, then the 'V' bit will be cleared (if not cleared already) and the vendor-id is set to 0, effectively removing the vendor-id from the avp-header.

vendor-id Manipulation

You can manipulate the Vendor ID value in the AVP header by configuring the **header-type** parameter to **avp-vendor-id**. This performs the DMR manipulation on the 4-byte vendor-id in the AVP header. AVP vendor id is present in the AVP header only when the 'V' bit is set in the flags. This is important because the DMR application relies upon the bit being set to determine where the data payload begins.

The avp-vendor-id search invokes an unsigned integer comparison between the received value and the **match-value**. If the **match-value** is empty, the configured action is performed without doing any match.

For the case where **match-value** is non-empty, as in the following example, the DMR engine checks whether the 'V' bit is set in the received header. If not, then the vendor id is not present either and the comparison is unsuccessful. If the 'V' bit is set, and the match succeeds, the match is successful. (An unsuccessful match has the DMR proceed to the next header-rule.)

```
avp-header-rules
   name         replaceAvpFlags
   header-type  avp-vendor-id
   action       add
   match-value  9148
   new-value    10415
```

When the match is successful (or if the **match value** is left empty), the configured **action** is performed. Consider all following actions:

1. none—no action will be performed
2. add—a configured vendor-id value in the **new-value** parameter is added to the AVP header and the 'V' bit set to indicate it's presence. If you prefer to set the 'V' bit in an AVP, it is better to do an avp-vendor-id action first and then manipulate the rest of the flags.

   If the **new-value** is empty, the add operation is not performed.

   If a vendor-id already exists in the AVP header, then it is replaced by **new-value**.
3. replace—the existing vendor-id value is replaced with the **new-value**.

   If the **new-value** is empty, the replace operation is not performed.

   If the vendor-id does not exist in the header, then one will be added with the **new-value** and the 'V' bit is set to indicate its presence.
4. delete—the vendor-id will be removed from the AVP header and 'V' bit will be reset to indicate its absence.

   If the **new-value** is empty, then the delete operation will not be performed.

   If the vendor-id does not exist, then the delete operation is not performed.

Multi-instance AVP Manipulation

Some AVPs can appear multiple times within a message. To choose a specific occurrence of an AVP, the **avp code** parameter supports indexing logic. By default, the Net-Net Diameter Director operates on all instances of the specified AVP. However, after configuring an avp-code, you can specify in square brackets, a specific instance of the AVP on which to operate on. The indexing is zero-based. For example,

```
diameter-manipulation-rule
                name                          manip
                avp-code                      293[2]
```

Special characters that refer to non-discrete values are:

- Last occurrence—avp-code[^]
- All—avp-code [*]

The last (^) character is used to refer to the last occurring instance of that AVP. Any [^] refers to the first matching header that matches the specified conditional matching criteria. All [*] is the default behavior, and operates on all headers of the specified-type. For **group manip** action, the AVP index applies to the instance within that grouped AVP.

ACLI Instructions

Diameter Manipulation

To configure a diameter manipulation configuration element:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-manipulation** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-manipulation
   ACMEPACKET(diameter-manipulation)#
   ```

4. **name**—Enter the name of this Diameter manipulation element.
5. **description**—Enter an optional description for this Diameter manipulation.
6. Type **done** and continue.

Manipulation Rule

7. Type **diameter-manip-rules** to continue and enter individual policy attributes. The system prompt changes to let you know that you can begin configuring individual parameters.
8. **name**—Enter the name of this manipulation rule. This value will be referenced in when creating a chain of rules.
9. **descr-avp-code**—Enter a description of the AVP code to manipulate.
10. **msg-cmd-code**—Enter the command code number of the message to execute the manipulation on.
11. **msg-type**—Set this to the type of message this manipulation applies to as **request**, **response**, or **all**.
12. **avp-code**—Enter the AVP by code number where this manipulation applies. You can add a multi instance identifier to the end of the avp code value, enclosed in brackets.
13. **avp-type**—Set this to the data type of the content of the match field. Refer to the Diameter standards document for the encodings of individual AVPs. Valid values are:

    none | octet-string | octet-hex | integer32 | unsignedint32 | address | diameteruri | enumerated | grouped

14. **match-value**—Enter the value within the match-field to find and make a positive match on.
15. **action**—Enter either **none, add, delete, store, diameter-manip, group-manip, find-replace-all**, or **replace** as the action to take after making a positive match on the previously entered match-value.
16. **new-value**—Enter the value that should be added or replaced in the old match-value's position.
17. Type **done** and continue.

AVP Header Manipulation

18. Type **avp-header-rules** to configure AVP header manipulation rules. The system prompt changes to let you know that you can begin configuring individual parameters.
19. **name**—Enter the name of this AVP Header manipulation rule.
20. **header-type**—Set this to either avp-flag or avp-vendor-id depending on which part of the AVP header you are manipulating.

21. **action**—Enter either none, add, delete, or replace as the action to take after making a positive match on the previously entered match-value.
22. **match-value**—Enter the value in the AVP flag field or Vendor ID field to match against.

    When matching in the avp flag field, then match-value is interpreted as comma-separated list of enumerated values <vendor,protected,must>. When matching in the Vendor ID field, then match-value is interpreted as 32 bit unsigned integer <1-4294967295>
23. **new-value**—Enter the new value when the match value is found. The resultant new value is entered as the match value is configured.
24. Type **done** to save your work.

Applying the Manipulation

You can apply a diameter manipulation by name to either the Diameter director agent, Diameter director interface or Diameter director group. All three configuration elements contain the same two parameters: in-manip-id, out-manip-id

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```
2. Type **session-router** and press <Enter> to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```
3. Enter the configuration element where you wish to apply the manipulation. You can configure the Diameter Director Agent, Diameter Director Interface, or Diameter Director Group. This example continues using the Diameter director agent.
4. Type **diameter-director-agent** and press <Enter>.

   ```
   ACMEPACKET(session-router)# diameter-director-agent
   ACMEPACKET(diameter-director-agent)#
   ```
5. Type **select** and then choose the pre-configured Diameter Director Agent you want to configure.

   ```
   ACMEPACKET(diameter-director-agent)# select
   <hostname>:
   1: 172.16.9.201
   2: 172.16.9.1
   3: net192InboundAgent
   4: 172.16.9.2

   selection: 1
   ```

   You may now add a Diameter manipulation to one or both directions of message flows.
6. **in-manip-id**—Enter a name of an existing diameter manipulation to apply as received by the Net-Net Diameter Director on this element.
7. **out-manip-id**—Enter a name of an existing diameter manipulation to apply as forwarded from the Net-Net Diameter Director on this element.
8. Type **done** and continue.

6 Security

Anonymous Diameter Agent Blocking

The Net-Net Diameter Director can prevent anonymous Diameter agents from connecting to specific Diameter Director Interfaces. The Diameter Director Interface contains the *allow anonymous* parameter in the *diameter director ports* subelement. You can configure the Net-Net Diameter Director to allow all connections to the Diameter Director Interface's IP address or allow only Diameter agents that are formally defined as Diameter Director Agents to connect. The *allow-anonymous* parameter defines this behavior by accepting either of the following arguments:

- **all**—Any Diameter agent may connect to this Diameter Director Interface.
- **agents-only**—Only Diameter agents defined as Diameter Director Agents will be allowed by the Net-Net Diameter Director. Any connection attempt from an undefined Diameter peer will be rejected. If an unauthorized Diameter agent connects to the Net-Net Diameter Director, the TCP connection request will be terminated.

In addition to configuring the Diameter Director Interface, Diameter agents have to be pre-defined. This is similar to defining an ACL for known agents. Once a Diameter agent is defined within the Net-Net Diameter Director configuration, it is *known* so that it can connect to a Diameter Director Interface that is set to **agents-only**.

Normally a Diameter Director Agent configuration element defines the behavior that the Net-Net Diameter Director initiates a Diameter connection to that agent. The Diameter Director Agent is automatically considered authorized for connecting to a Diameter Director Interface. But setting the *connection-mode* parameter to **inbound**, the Net-Net Diameter Director waits for the Diameter agent to initiate the connection. This lets agents remain unconnected to the Net-Net Diameter Director, but still remain authorized to connect when needed.

ACLI Instructions

Diameter Director Interface

To configure a Diameter Director Interface to only accept connections from configured Diameter Director Agents:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-interface
   ACMEPACKET(diameter-director-interface)#
   ```

4. Type **select** and the number of the pre-configured Diameter Director Interface you want to configure.

```
ACMEPACKET(diameter-director-interface)# select
<realm-id>:
1: net172 172.16.9.200:3868
2: net192 192.168.9.200:3868

selection: 1
```

Access the diameter director ports subelement:

```
ACMEPACKET(diameter-director-interface)#diameter-director-ports
```

5. **allow-anonymous**—Set this to **agents-only** for the Net-Net Diameter Director to only accept connections from configured Diameter Director Agents.
6. Save your work using the ACLI **done** command.

Diameter Director Agent

To configure a Diameter Director Agent as authorized to connect to this Net-Net Diameter Director, yet let the Net-Net Diameter Director wait for that connection:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the media-related configurations.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **diameter-director-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# diameter-director-agent
   ACMEPACKET(diameter-director-agent)#
   ```

4. Type **select** and the number of the pre-configured Diameter Director Interface you want to configure.

   ```
   ACMEPACKET(diameter-director-agent)# select
   <hostname>:
   1: 172.16.9.201
   2: 172.16.9.1
   3: net192InboundAgent
   4: 172.16.9.2

   selection: 1
   ```

5. **connection-mode**—Set this to **incoming** for the Net-Net Diameter Director to wait for this agent to initiate the connection. Alternatively, leaving this parameter as outgoing configures the Net-Net Diameter Director to start a capability exchange negotiation toward this Diameter Director Agent.
6. Save your work using the ACLI **done** command.

Natively Securing Network Topology Information

In many deployment environments, administrators require that their network topology information be hidden across Diameter elements. The Net-Net Diameter Director provides configuration options to implement this security without requiring discrete diameter manipulation rules. By implementing this processing natively, the Net-Net Diameter Director minimizes the processing and configuration overhead required to secure this information.

There are 6 key AVPs that the Net-Net Diameter Director can secure natively. You secure these AVPs via the `network-topology` parameter and its applicable arguments, as described below:

Topology Masking — Changes origin-host and origin-realm AVPs from the source's values to the values used during capabilities negotiation between the Net-Net Diameter Director and the next-hop diameter agent.

Toplogy Hiding — Changes destination-host and destination-realm AVPs from the source's values to the values used during capabilities negotiation between the Net-Net Diameter Director and the next-hop diameter agent.

Topology Obscuring — Changes session-id to a string encoded by the Net-Net Diameter Director and strips all route-record AVPs from the message.

You can also configure any combination of the above to implement each argument's effects.

You configure this support globally, per diameter-interface, or per diameter-agent. In cases of conflicting configuration, the diameter-agent configuration takes highest precedence, followed by interface, then global. This precedence is not cumulative; that is, any configuration on an agent invalidates the applicable interface's configuration for traffic to that agent.

The system performs the configured function on egress traffic. This provides additional configuration flexibility, giving you the option of performing these functions for either or both Diameter elements involved with the messaging.

Topology Masking

Topology masking is as much a means of establishing compatibility between devices as a security feature. Some devices always expect upstream Diameter traffic to come from a relay. These devices expect the origin-host and origin-realm AVPs in messages to be the same as learned during capabilities negotiation.

In deployments where this security is required or Net-Net Diameter Director needs to appear to be a proxy, use topology masking to change origin-host and origin-realm accordingly.

Topology Hiding

Use topology hiding to further secure networks' topology information from each other. It is common for topology hiding to be implemented in conjunction with topology masking to achieve the resulting cumulative security.

Topology Obscuring

Topology obscuring changes session-id AVPs and removes all record-route AVPs from ingress request and response messages prior to forwarding. When implemented in conjunction with topology masking and hiding, you achieve the maximum level of security.

By definition, topology obscuring is only applicable within the context of stateful applications. The sytem maintains an internal mapping of the original and new session-IDs. During transit, the system inserts the new session-ID in messages from the originator and inserts the original session-ID in messages from the next-hop server back to the originator.

The fomat of the new session ID is as follows:

New `Session-Id = <origin-host>;<timestamp>;<sequence>`

<origin-host> The origin-host value derived from the diameter-director-interface for the egress agent.

<timestamp> The time the socket originally connected, per RFC3588.

<sequence> An incrementing sequence number from 1 to 2^{32} - 1, which is the maximum unsigned 32-bit value. If the sequence reaches the maximum value, the Net-Net Diameter Director generates a new timestamp and re-starts the sequence.

Record-route stripping is a matter of removing these AVPs from requests and responses from the session originator to the next-hop server.

The Net-Net Diameter Director performs topology obscuring only within the context of stateful applications. Any messages associated with a stateless application that transiting a Net-Net Diameter Director element configured for topology obscuring are not affected. Instead, the Net-Net Diameter Director simply logs a message indicating that it forwarded the message without modification.

Network Topology Security and Dynamic Routing

The Net-Net Diameter Director accommodates dynamic routing in paths that include network topology masking, hiding and/or obscuring. Additional user configuration is not required.

Dynamic routing depends on an agent's origin host AVPs to reply to requests. Normally, the Net-Net Diameter Director caches this forwarding information for these agents. Lookups needed to route to these cached endpoints, therefore, fail if the AVP has been changed in transit.

To accommodate this, the Net-Net Diameter Director builds origin host AVPs in such a way that it can recognize messages intended for agents that depend on dynamic routing. The Net-Net Diameter Director creates these AVPs using a pre-specified format that includes a cookie and the AVP. The Net-Net Diameter Director recognizes AVPs in this format and forwards the messages using the route cache.

Securing Topology within Stateless Applications

You must configure stateless applications, such as s6a, as subscriber-only within the Net-Net Diameter Director's state machine to support multi-host topology hiding. If not, an intruder may be able to to determine the number of Diameter elements in the network and extrapolate information about network topology from that information. When a stateless application is operating in subscriber-only mode, the Net-Net Diameter Director is able to generate a cookie and mask the OH, OR, DH and DR as it does for stateful applications. Topology obscuring does not apply to subscriber-only applications because there is no sessionID to mask.

CLI Instructions

To configure an existing diameter-director agent with topology masking, hiding and obscuring security:

1. From superuser mode, use the following command sequence to access a diameter director agent.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# session-router
ACMEPACKET(session-router)# diameter-director-agent
ACMEPACKET(diameter-director-agent)# sel
<hostname>: agent1
```

2. Configure the *network-topology* command, in this case on a diameter agent, with the desired arguments separated by commas.

```
ACMEPACKET(diameter-director-agent)# network-topology masking, hiding, obscuring
ACMEPACKET(diameter-director-agent)# done
```

diameter-director-agent > network-topology

The `network-topology` command specifies the desired level of topology security the system uses

Parameters

`masking`—Enables security of origin-host and origin-realm AVPs in Diameter messages

Default: disabled

`hiding`—Enables security of destination-host and destination-realm AVPs in Diameter messages

Default: disabled

`obscuring`—Secures AVPs in Diameter messages by stripping out record-routes and re-writing session-ids.

Default: disabled

Path

`network-topology` is a command applied against a diameter-director agent, a diameter-drector interface or globally.

IPSec Support

The Net-Net DD offers IPSec for securing signaling, media, and management traffic at the network layer.

Supported Protocols

The Net-Net DD's IPSec implementation supports all required tools for securing Internet communication via the IPSec protocol suite. The following paragraphs list and explain the protocols within the IPSec suite that the Net-Net DD supports. This chapter does not explain how to design and choose the best protocols for your application.

AH vs. ESP

The Net-Net DD supports the two encapsulations that IPSec uses to secure packet flows. Authentication Header (AH) is used to authenticate and validate IP packets. Authentication means that the packet was sent by the source who is assumed to have sent it. Note that AH is incompatible with NAT. Validation means that the recipient is assured that the packet has arrived containing the original, unaltered data as sent.

ESP (Encapsulating Security Payload) provides AH's authentication and validations and extends the feature set by ensuring that the IP packet's contents remain confidential as they travel across the network. Using an encryption algorithm that both peers agree upon, ESP encrypts a full IP packet so that if intercepted, an unauthorized party cannot read the IPSec packet's contents.

Tunnel Mode vs. Transport Mode

In addition to its security encapsulations, the IPSec suite supports two modes: tunnel mode and transport mode. Tunnel mode is used most often for connections between gateways, or between a host and a gateway. Tunnel mode creates a VPN-like path between the two gateways and encapsulates the entire original packet. Transport mode is used to protect end-to-end communications between two hosts providing a secured IP connection and encrypts just the original payload.

Cryptographic Algorithms

IPSec works by using a symmetric key for validation and encryption. Symmetric key algorithms use the same shared secret key for encoding and decoding data on both sides of the IPSec flow. The Net-Net DD's IPSec feature supports the following encryption algorithms:

- DES
- 3DES
- AES128CBC
- AES256CBC
- AES128CTR
- AES256CTR

The Net-Net DD can quickly generate keys for all of the above mentioned algorithms from the CLI. It can additionally support HMAC-SHA1 or HMAC-MD5 keyed-hash message authentication codes.

Internet Key Exchange (IKEv2)

The Net-Net DD provides support for Version 2 of the Internet Key Exchange Protocol (IKEv2) as defined in RFC 4306, *Internet Key Exchange (IKEv2) Protocol*, and for the related Dead Peer Detection (DPD) protocol as defined in RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*.

IKEv2 protocol operations can support either responder or initiator mode, meaning that the IKEv2 protocol instance can receive and respond to tunnel signaling from a remote peer, or can initiate tunnel signaling to a remote peer. In initiator mode, certain IPsec tunnels can be automatically re-established after system restart or boot.

IKEv2 Overview

IKEv2 is used for the generation and exchange of cryptographic material between two IKEv2 peers. Peers use the exchanged material to establish IPsec tunnels.

All IKEv2 messages are request/response pairs. It is the responsibility of the IKEv2 requester to retransmit the request in the absence of a timely response.

IKEv2 has an initial handshake, which usually consists of two request/response pairs. The first request/response pair negotiates cryptographic algorithms and performs a Diffie-Hellman exchange. The second request/response pair (which is encrypted and integrity protected with keys based on the Diffie-Hellman exchange) reveals peer identities and provides for a certificate-based or shared-secret-based integrity check. The initial exchange results in the creation of an IKE Security Association (SA), which is required for the establishment of IPsec tunnels between the remote peers.

After the initial handshake, additional requests can be initiated by either peer and consist of informational messages or requests to establish IPsec tunnels. Informational messages

convey such things as null messages for detecting peer aliveness, or information on the deletion of SAs.

The exchange to establish an IPsec tunnel consists of an optional Diffie-Hellman exchange (if perfect forward secrecy is required), nonces (so that a unique key for the IPsec tunnel is established), and negotiation of traffic selector values which indicate the addresses, ports, and protocol types to be transmitted through the tunnel.

IKEv2 configuration consists of the following steps, some of which are optional.

1. Configure IKEv2 global parameters.
2. Optionally, enable and configure the DPD Protocol.
3. If IKEv2 peer authentication is certificate-based, configure certificate profiles.
4. If configuration payload requests for IP addresses are handled locally, configure one or more local address pools.
5. Configure the interfaces for IKEv2 operations.
6. Configure IKEv2 SAs.
7. Assign the IKEv2 SA to an IPsec Security Policy.
8. Configure IPsec tunnels across the applicable interfaces.

IKEv2 Global Configuration

Use the following procedure to perform IKEv2 global configuration.

1. From superuser mode, use the following command sequence to access *ike-config* configuration mode. While in this mode, you configure global IKEv2 configuration parameters.

   ```
   ragnarok# configure terminal
   ragnarok(configure)# security
   ragnarok(security)# ike
   ragnarok(ike)# ike-config
   ragnarok(ike-config)#
   ```

2. Use the **ike-version** parameter to specify IKEv2.

   ```
   ragnarok(ike-config)# ike-version 2
   ragnarok(ike-config)#
   ```

3. Use the **log-level** parameter to specify the contents of the IKE log.

   Events are listed below in descending order of criticality.

   - emergency (most critical)
     critical
     major
     minor
     warning
     notice
     info (least critical — the default)
     trace (test/debug, not used in production environments)
     debug (test/debug, not used in production environments)
     detail (test/debug, not used in production environments)

   In the absence of an explicitly configured value, the default value of *info* is used.

   ```
   ragnarok(ike-config)# log-level warning
   ragnarok(ike-config)#
   ```

4. Use the optional **udp-port** parameter to specify the port monitored for IKE protocol traffic.

   In the absence of an explicitly configured value, the default port number of *500* is used.

   ```
   ragnarok(ike-config)# udp-port 5000
   ragnarok(ike-config)#
   ```

5. Use the optional **sd-authentication-method** to select the default method used to authenticate the IKEv2 SA.

   Two authentication methods are supported.

   shared-password — (the default) uses a PSK (pre-shared key) to authenticate the remote IKEv2 peer.

   certificate — uses an X.509 certificate to authenticate the remote IKEv2 peer.

   This global default can be over-ridden at the interface level.

   ```
   ragnarok(ike-config)# sd-authentication-method certificate
   ragnarok(ike-config)#
   ```

6. If **sd-authentication-method** is *shared-password*, use the **shared-password** parameter to specify the default PSK required for password-based IKEv2 authentication.

   The PSK is a string of ACSII printable characters no longer than 255 characters (not displayed by the ACLI).

   This global default can be over-ridden at the interface level.

   ```
   ragnarok(ike-config)# shared-password !yetAnotherPaSSword1of87354
   ragnarok(ike-config)#
   ```

7. If **sd-authentication-method** is *certificate*, use the **certificate-profile-id** to identify the default *ike-certificate-profile* configuration element that contains identification and validation credentials required for certificate-based IKEv2 authentication.

   Provide the name of an existing *ike-certificate-profile* configuration element.

   This global default can be over-ridden at the interface level.

   ```
   ragnarok(ike-config)# certificate-profile-id valCred-IKEv2
   ragnarok(ike-config)#
   ```

8. Use the optional **dpd-time-interval** parameter to specify the maximum period of inactivity before the DPD protocol is initiated on a specific endpoint.

   Allowable values are within the range 1 through 999999999 (seconds) with a default of *0*.

   The default value, *0*, disables the DPD protocol; setting this parameter to a non-zero value globally enables the protocol and sets the inactivity timer.

   ```
   ragnarok(ike-config)# dpd-time-interval 20
   ragnarok(ike-config)#
   ```

9. Use the optional **v2-ike-life-seconds** parameter to specify the default lifetime (in seconds) for the IKEv2 SA.

   Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (24 hours).

   This global default can be over-ridden at the interface level.

   ```
   ragnarok(ike-config)# v2-ike-life-seconds 43200
   ragnarok(ike-config)#
   ```

10. Use the optional **v2-ipsec-life-seconds** parameter to specify the default lifetime (in seconds) for the IPsec SA.

    Allowable values are within the range 1 through 999999999 (seconds) with a default of 28800 (8 hours).

    This global default can be over-ridden at the interface level.

    ```
    ragnarok(ike-config)# v2-ipsec-life-seconds 14400
    ragnarok(ike-config)#
    ```

11. Retain the default value for the optional **eap-protocol** parameter.

    The default, and only currently-supported value, *eap-radius-passthru*, specifies the use of a RADIUS server for Extensible Authentication Protocol (EAP) processing. The SG shuttles incoming and outgoing EAP messages between the remote IKEv2 peer and the RADIUS server.

    ```
    ragnarok(ike-config)# eap-protocol eap-radius-passthru
    ragnarok(ike-config)#
    ```

12. Use the optional **eap-bypass-identity** parameter to specify whether or not to bypass the EAP (Extensible Authentication Protocol) identity phase.

    EAP, defined in RFC 3748, *Extensible Authentication Protocol (EAP)*, provides an authentication framework widely used in wired and wireless networks.

    An Identity exchange is optional within the EAP protocol exchange. Therefore, it is possible to omit the Identity exchange entirely, or to use a method-specific identity exchange once a protected channel has been established.

    However, where roaming is supported, it may be necessary to locate the appropriate backend authentication server before the authentication conversation can proceed. The realm portion of the Network Access Identifier (NAI) is typically included within the EAP-Response/Identity to enable the routing of the authentication exchange to the appropriate authentication server. Therefore, while the peer-name portion of the NAI may be omitted in the EAP-Response/Identity where proxies or relays are present, the realm portion may be required.

    Identify bypass is disabled by default — thus requiring an identity exchange.

    ```
    ragnarok(ike-config)# eap-bypass-identity enabled
    ragnarok(ike-config)#
    ```

13. Use the optional **addr-assignment** parameter to specify the default method used to assign addresses in response to an IKEv2 Configuration Payload request.

    The Configuration Payload supports the exchange of configuration information between IKEv2 peers. Typically, a remote IKEv2 peer initiates the exchange by requesting an IP address on the protected network. In response, IKEv2 returns a local address for use by the requesting peer.

    This parameter specifies the source of the returned IP address.

    local — (the default) use local address pool

    radius-only — obtain local address from RADIUS server

    radius-local — try RADIUS server first, then local address pool

    This global default can be over-ridden at the interface level.

    ```
    ragnarok(ike-config)# addr-assignment radius-only
    ragnarok(ike-config)#
    ```

14. Use the **overload-threshold**, **overload-interval**, **overload-action**, **overload-critical-threshold**, and **overload-critical-interval** parameters to configure system response to an overload state.

    Use the optional **overload-threshold** parameter to specify the percentage of CPU usage that triggers an overload state.

    Values are within the range 1 through 100 (percent) with a default of *100*, which effectively disables overload processing.

    ```
    ragnarok(ike-config)# overload-threshold 60
    ragnarok(ike-config)#
    ```

    Use the optional **overload-interval** parameter to specify the interval (in seconds) between CPU load measurements when in the overload state.

    Values are within the range 1 through 60 (seconds) with a default of *1*.

    ```
    ragnarok(ike-config)# overload-interval 3
    ragnarok(ike-config)#
    ```

    Use the optional **overload-action** parameter to specify response to an overload state. The overload state is reached when CPU usage exceeds the percentage threshold specified by the **overload-threshold** parameter.

    By default, no preventive action is taken in response to an overload, You can, however, use this parameter to implement a call rejection algorithm in response to the overload. With the algorithm enabled, the CPU uses the following calculation to reject/drop some number of incoming calls:

    DropRate = (currentLoad - overloadThreshold) / (100 -overloadThreshold)

    Thus, assuming a current CPU load of 70% and an overload threshold of 60%, the Net-Net DD drops 1 of out every 4 incoming calls until the load falls below the threshold value.

    Use **none** to retain default behavior (no action); use **drop-new-connection** to implement call rejection.

    ```
    ragnarok(ike-config)# overload-action drop-new-connection
    ragnarok(ike-config)#
    ```

Use the optional **overload-critical-threshold** parameter to specify the percentage of CPU usage that triggers a critical overload state.

When this threshold is exceeded, the Net-Net DD drops all incoming calls until the load drops below the critical threshold level, at which point it may drop selective calls depending on the value of the **overload-threshold** parameter.

Values are within the range 1 through 100 (percent) with a default of 100, which effectively disables overload processing.

Ensure that this threshold value is greater than the value assigned to **overload-threshold**.

```
ragnarok(ike-config)# overload-critical-threshold 75
ragnarok(ike-config)#
```

Use the optional **overload-critical-interval** parameter to specify the interval (in seconds) between CPU load measurements when in the critical overload state.

Values are within the range 1 through 60 (seconds) with a default of *1*.

```
ragnarok(ike-config)# overload-criticalinterval 2
ragnarok(ike-config)#
```

15. Use the **red-port**, **red-max-trans**, **red-sync-start-time**, and **red-sync-comp-time** parameters to configure redundancy.

    Acme Packet Net-Net DDs can be deployed in pairs to deliver high availability (HA). Two Net-Net DDs operating in this way are called an HA node.

    Two Net-Net DDs work together in an HA node, one in *active* mode and one in *standby* mode.

    - The active Net-Net DD checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it will hand over its role as the active system to the standby Net-Net DD in the node.
    - The standby Net-Net DD is the backup system, which maintains a synchronous configuration with the active node. The standby Net-Net DD monitors the status of the active system so that, if needed, it can assume the active role without the active system having to instruct it to do so.

    Refer to *High Availability Nodes* in the *Net-Net 3000 and 4000 ACLI Configuration Guide* for information on cabling and configuring HA nodes.

    Use the **red-port** parameter to specify the port number monitored for IKEv2 synchronization messages.

    The default value (*0*) effectively disables redundant high-availability configurations. Select a port value other than 0 (for example, 1995) to enable high-availability operations.

    ```
    ragnarok(ike-config)# red-port 1995
    ragnarok(ike-config)#
    ```

    Use the **red-max-trans** parameter to specify the maximum number of retained IKEv2 synchronization messages.

    Values are within the range 0 through 999999999 (messages) with a default of *10000*.

    ```
    ragnarok(ike-config)# red-trans 7500
    ragnarok(ike-config)#
    ```

16. Use the **red-sync-start-time** parameter to specify the interval, in milliseconds, between health checks performed by the active node to confirm that it still retains this role.

If the active role is verified, the timer is reset. If, for any reason, the health check is deficient, the active transitions to the standby role, and the previous standby assumes the active role.

Supported values are integers within the range 0 through 999999999, with a default value of 5000 (5 seconds).

Values are within the range 0 through 999999999 (milliseconds) with a default of *500*.

```
ragnarok(ike-config)# red-sync-start-time 2500
ragnarok(ike-config)#
```

Use the **red-sync-comp-time** parameter to specify the interval between standby initiated probes that confirm the availability of the active node.

Values are within the range 0 through 999999999 (milliseconds) with a default of *500*.

```
ragnarok(ike-config)# red-sync-comp-time 750
ragnarok(ike-config)#
```

17. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv2 global parameters.

DPD Configuration

IKEv2 peers can lose connectivity unexpectedly, perhaps as a result of routing problems, or reboot of one of the peers. Neither IKEv2 nor IPsec offers an efficient and scalable method to respond to connectivity loss. Consequently established SAs can remain in place until their configured lifetimes eventually expire. Such behavior results in mis-management of system resources and the presence of *black holes* where packets are tunneled to oblivion.

With DPD, each peer's state is largely independent of the other's. A peer is free to request proof of connectivity when it needed — there are no mandatory, periodic exchanges as would be required by a detection method based on *keepalive* or *heartbeat* messages. DPD asynchronous exchanges require fewer messages and achieve greater scalability.

If there is ongoing valid IPSec traffic between peers, there is little need to check connectivity. After a period of inactivity, however, connectivity is questionable. Verification of connectivity is only urgently necessary if there is traffic to be sent. For example, if one peer has IPsec traffic to send after the period of idleness, it need to know if its remote peer is still alive. At this point, peer A can initiate the DPD exchange.

If you enabled the DPD protocol with the **dpd-time-interval** parameter, use the following procedure to create a DPD template, an operational set of DPD parameters, that you subsequently assign to the interface one or more IKEv2 interfaces.

This section can be safely ignored if you did not enable DPD.

1. From superuser mode, use the following command sequence to access *dpd-params* configuration mode. While in this mode, you configure DPD templates.

```
ragnarok# configure terminal
ragnarok(configure)# security
ragnarok(security)# ike
ragnarok(ike)# dpd-params
ragnarok(dpd-params)#
```

2. Use the required **name** parameter to provide a unique identifier for this *dpd-params* instance.

   name enables the creation of multiple *dpd-params* instances.

```
ragnarok(dpd-params)# name dpdTemplate-1
ragnarok(dpd-params)#
```

3. Use the **max-loop** parameter to specify the maximum number of DPD peers examined every **dpd-interval**, which value is established during IKE global configuration.

   If CPU workload surpasses the threshold set by **max-cpu-limit**, this value is over-ridden by **load-max-loop**.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of *100*.

```
ragnarok(dpd-params)# max-loop 80
ragnarok(dpd-params)#
```

4. Use the **max-endpoints** parameter to specify the maximum number of simultaneous DPD protocol negotiations supported when the CPU is not under load (as specified by the **max-cpu-limit** property).

   If CPU workload surpasses the threshold set by **max-cpu-limit**, this value is over-ridden by **load-max-endpoints**.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of *25*.

```
ragnarok(dpd-params)# max-endpoints 20
ragnarok(dpd-params)#
```

5. Use the **max-cpu-limit** parameter to specify a threshold value (expressed as a percentage of CPU capacity) at which DPD protocol operations are minimized to conserve CPU resources.

   Allowable values are within the range 0, which effectively disables DPD operations, through 100 (percent) with a default of *60*.

```
ragnarok(dpd-params)# max-cpu-limit 50
ragnarok(dpd-params)#
```

6. Use the **load-max-loop** parameter to specify the maximum number of endpoints examined every **dpd-time-interval** when the CPU is under load, as specified by the **max-cpu-limit** parameter.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of *40*. Ensure that the configured value is less than the value assigned to **max-loop**.

```
ragnarok(dpd-params)# load-max-loop 30
ragnarok(dpd-params)#
```

7. Use the **load-max-endpoints** parameter to specify the maximum number of simultaneous DPD Protocol negotiations supported when the CPU is under load, as specified by the **max-cpu-limit** property.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of *5*. Ensure that the configured value is less than the value assigned to **max-endpoints**.

   ```
   ragnarok(dpd-params)# load-max-endpoints 3
   ragnarok(dpd-params)#
   ```

8. Use **done**, **exit**, and **verify-config** to complete configuration of the DPD template instance.

9. If necessary, repeat Steps 1 through 8 to configure additional DPD templates.

Certificate Profile Configuration

If authentication between IKEv2 peers is certificate based, use the following procedure to create one or more certificate profiles that provide identification and validation credentials for a specific IKEv2 identity.

This section can be safely ignored if authentication is based upon a PSK.

1. From superuser mode, use the following command sequence to access *ike-certificate-profile* configuration mode. While in this mode, you configure certificate profiles.

   ```
   ragnarok# configure terminal
   ragnarok(configure)# security
   ragnarok(security)# ike
   ragnarok(ike)# ike-certificate-profile
   ragnarok(ike-certificate-profile)#
   ```

2. Use the required **identity** parameter to specify the IKEv2 entity that uses the authentication and validation credentials provided by this *ike-certificate-profile* instance.

   Identify the subject of this *ike-certificate-profile* by either an IP address or fully-qualified domain name (FQDN).

   identity enables the creation of multiple *ike-certificate-profile* instances.

   ```
   ragnarok(ike-certificate-profile)# identity jojo.net
   ragnarok(ike-certificate-profile)#
   ```

3. Use the required **end-entity-certificate** parameter to supply the unique name of a *certificate-record* configuration element referencing the identification credential (specifically, an X509.v3 certificate) offered by a local IKEv2 entity to verify its asserted identity.

   ```
   ragnarok(ike-certificate-profile)# end-entity-certificate ACME-1a
   ragnarok(ike-certificate-profile)#
   ```

4. Use the required **trusted-ca-certificates** parameter to compile a list or one or more *certificate-record* configuration elements referencing trusted Certification Authority (CA) certificates used to authenticate a remote IKEv2 peer

   Provide a comma separated list of existing CA **certificate-record** configuration elements.

   ```
   ragnarok(ike-certificate-profile)# trusted-ca-certificates
   verisignClass3-a,verisignClass3-b,baltimore,thawte-a
   ragnarok(ike-certificate-profile)#
   ```

5. Use the optional **verify-depth** parameter to specify the maximum number of chained certificates that will be processed while authenticating the IKEv2 peer.

Provide an integer within the range 1 through 10 (the default).

```
ragnarok(ike-certificate-profile)# verify-depth 10
ragnarok(ike-certificate-profile)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of the *ike-certificate-profile* instance.
7. If necessary (for instance if you require individual certificates for each IPsec tunnel instance, repeat Steps 1 through 6 to configure additional *ike-certificate-profile* instances.

Interface Configuration

Use the following procedure to configure the interface for IKEv2 operations.

1. Obtain the IP address of the interface.

   For media interfaces, use the diameter-director interface address configuration, accessible via the show command below:

```
ragnarok# show configuration diameter-director-interface
diameter-director-interface
        state                           enabled
        realm-id                        access1
        description
        diameter-director-ports
                address                         192.168.0.127
                port                            3868
                multi-home-addrs
                transport-protocol              TCP
                allow-anonymous                 all
```

   For management interfaces, use the following command sequence to access the boot parameters which contain the address.

   Press `Enter` to scroll through the boot parameters.

   The `inet on ethernet (e)` parameter contains the IP address

```
ragnarok# configure terminal
ragnarok(configure)# bootparam

'.' = clear field; '-' = go to previous field; q = quit

bootdevice            : wancom0
processor number:     : 0
host name             : goose
file name             : nnSC620b1.gz
inet on ethernet (e)  : 172.30.55.127
...
...
```

2. From configuration mode, use the following command sequence to access *ike-interface* configuration mode.

```
ragnarok(configure)# security
ragnarok(security)# ike
ragnarok(ipsec)# ike-interface
ragnarok(ike-interface)#
```

3. Use the **address** parameter to specify the address.

```
ragnarok(ike-interface)# address 192.168.0.127
```

```
ragnarok(ike-interface)#
```

4. Use the **realm-id** parameter to specify the realm that contains the IP address assigned to this IKEv2 interface.

```
ragnarok(ike-interface)# realm-id MGMT
ragnarok(ike-interface)#
```

5. Use the **ike-mode** parameter to specify the operational mode, either *responder* (the default) or *initiator*.

```
ragnarok(ike-interface)# ike-mode initiator
ragnarok(ike-interface)#
```

6. Use the optional interface-specific **sd-authentication-method** parameter to select the method used to authenticate the IKEv2 SA.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

   Two authentication methods are supported.

   shared-password — (the default) uses a PSK that is used to calculate a hash over a block of data.

   certificate — uses an X.509 certificate to digitally sign a block of data.

```
ragnarok(ike-interface)# sd-authentication-method shared-password
ragnarok(ike-interface)#
```

7. If **sd-authentication-method** is *shared-password*, use the **shared-password** parameter to specify an interface-specific PSK required for password-based IKEv2 authentication.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

```
ragnarok(ike-interface)# shared-password 123ffGGH65900tnojbt=+
ragnarok(ike-interface)#
```

8. If **sd-authentication-method** is *certificate*, use the **certificate-profile-id** parameter to identify an interface-specific *ike-certificate-profile* instance that contains identification and validation credentials required for certificate-based IKEv2 authentication.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

```
ragnarok(ike-interface)# certificate-profile-id jojo.net
ragnarok(ike-interface)#
```

9. If DPD has been enabled at the global level, use the **dpd-params-name** parameter to assign a DPD template, an operational set of DPD parameters, to the current IKEv2 interface.

   If DPD has not been enabled, this parameter can be safely ignored.

```
ragnarok(ike-interface)# dpd-params-name olivier
ragnarok(ike-interface)#
```

10. Use the optional interface-specific **v2-ike-life-seconds** parameter to specify the lifetime (in seconds) for the IKEv2 SAs supported by this IKEv2 interface.

    By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

    Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (24 hours).

```
ragnarok(ike-interface)# v2-ike-life-seconds 21600
```

```
ragnarok(ike-interface)#
```

11. Use the optional interface-specific **v2-ipsec-life-seconds** parameter to specify the lifetime (in seconds) for the IPsec SAs supported by this IKEv2 interface.

    By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

    Allowable values are within the range 1 through 999999999 (seconds) with a default of 28800 (8 hours).

    ```
    ragnarok(ike-interface)# v2-ipsec-life-seconds 7200
    ragnarok(ike-interface)#
    ```

12. Retain the default value for the optional **eap-protocol** parameter.

    The default, and only currently-supported value, *eap-radius-passthru*, specifies the use of a RADIUS server for Extensible Authentication Protocol (EAP) processing. The SG shuttles incoming and outgoing EAP messages between the remote IKEv2 peer and the RADIUS server.

    ```
    ragnarok(ike-interface)# eap-protocol eap-radius-passthru
    ragnarok(ike-interface)#
    ```

13. Use the optional interface-specific **addr-assignment** parameter to specify the method used to assign addresses in response to an IKEv2 Configuration Payload request.

    The Configuration Payload supports the exchange of configuration information between IKEv2 peers. Typically, an IRAC (IPsec Remote Access Client) initiates the exchange by requesting an IP address on the gateway's protected network. In response, the gateway, referred to as an IRAS (IPsec Remote Access Server), returns a local address for the IRAC's use.

    By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

    Supported values are:

    local — (the default) use local address pool

    radius-only — obtain local address from RADIUS server

    radius-local — try RADIUS server first, then local address pool

    ```
    ragnarok(ike-interface)# addr-assignment local
    ragnarok(ike-interface)#
    ```

14. Use **done**, **exit**, and **verify-config** to complete initial configuration.

Tunnel Origination Parameters Configuration

If you have set the IKEv2 mode to initiator, and want to enable the automatic re-establishment of IPsec tunnels on the interface during system restart or boot, you must next configure a *tunnel-orig-params* configuration element, which contains the information necessary to re-establish IPsec tunnels.

Use the following procedure to configure a *tunnel-orig-params* configuration element.

1. From superuser mode, use the following command sequence to access *tunnel-orig-params* configuration mode. While in this mode, you define remote tunnel endpoints.

    ```
    ragnarok# configure terminal
    ragnarok(configure)# security
    ragnarok(security)# ike
    ragnarok(ike)# tunnel-orig-params
    ragnarok(tunnel-orig-params)#
    ```

2. Use the **name** parameter to identify this instance of the *tunnel-orig-params* configuration element.

```
ragnarok(tunnel-orig-params)# name syslog
ragnarok(tunnel-orig-params)#
```

3. Use the **remote-addr** parameter to identify the remote IKEv2 peer at the remote end of the IPsec tunnel.

```
ragnarok(tunnel-orig-params)# remote-addr 192.168.34.90
ragnarok(tunnel-orig-params)#
```

4. Use the **retry-limit** parameter to specify the maximum number of tunnel initiation attempts.

   Allowable values are within the range 1 through 5, with a default value of 3.

```
ragnarok(tunnel-orig-params)# retry-limit 5
ragnarok(tunnel-orig-params)#
```

5. Use the **retry-time** parameter to specify the interval (in seconds) between tunnel initiation attempts.

   Allowable values are within the range 5 through 60 (seconds), with a default value of 10.

```
ragnarok(tunnel-orig-params)# retry-time 24
ragnarok(tunnel-orig-params)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of this instance of a *tunnel-orig-params* configuration element.

7. If necessary, repeat Steps 1 through 9 to configure additional *tunnel-orig-params* instances.

Use the following procedure, which assigns one or more *tunnel-orig-params* to the interface, to complete configuration.

1. From super mode, use the following command sequence to access the interface.

```
ragnarok# configure terminal
ragnarok(configure)# security
ragnarok(security)# ike
ragnarok(ipsec)# ike-interface
ragnarok(ike-interface)# select
<address>:
172.30.1.150
172.30.1.151
192.168.0.127

selection: 3
ragnarok(ike-interface)#
```

2. Use the **tunnel-orig-name-list** parameter to assign one or more *tunnel-orig-params* instances to the interface.

   Each instance specifies the remote end of a single IPsec tunnel.

   Identify *tunnel-orig-params* instances by name; enclose multiple entries with quotation marks’

```
ragnarok(ike-interface)# tunnel-orig-name-list “syslog FTPserver SNMP-1 SNMP-2 auditLog keyStore”
ragnarok(ike-interface)#
```

3. Use **done**, **exit**, and **verify-config** to complete configuration of the interface.

spi is the security parameter index (SPI) — part of the SA negotiated by the endpoint peers.

Use the **show security ipsec sad <interface> brief** command to display the SPI

IKEv2 Security Association Configuration

Use the following procedure to create an IKEv2 SA that identifies cryptographic material available for IPsec tunnel establishment. You will later assign this IKEv2 SA to an IPsec Security Policy.

1. From superuser mode, use the following command sequence to access *ike-sainfo* configuration mode. While in this mode, you configure global IKEv2 SAs.

   ```
   ragnarok# configure terminal
   ragnarok(configure)# security
   ragnarok(security)# ike
   ragnarok(ike)# ike-sainfo
   ragnarok(ike-sainfo)#
   ```

2. Use the required **name** parameter to provide a unique identifier for this *ike-sainfo* instance.

   name enables the creation of multiple *ike-sainfo* instances.

   ```
   ragnarok(ike-sainfo)# name SA-1
   ragnarok(ike-sainfo)#
   ```

3. Use the **security-protocol** parameter to specify the IPsec security (authentication and encryption) protocols supported by this SA.

   The following security protocols are available.

   Authentication Header (AH) — the default value — as defined by RFC 4302, *IP Authentication Header*, which provides authentication integrity to include the mutual identification of remote peers, non-repudiation of received traffic, detection of data that has been altered in transit, and detection of data that has been replayed, that is copied and then re-injected into the data stream at a later time. Authentication services utilize the authentication algorithm specified by the **auth-algo** parameter.

   Encapsulating Security Payload (ESP) as defined by RFC 4303, *IP Encapsulating Security Payload*, which provides both authentication and privacy services. Privacy services utilize the encryption algorithm specified by the **encryption-algo** parameter.

ESP-AUTH (also RFC 4303-based), which supports ESP's optional authentication.

ESP-NULL (also RFC 4303-based) which proves NULL encryption as described in RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*. This option provides no privacy services, and is not recommended for production environments.

Refer to the following figures for additional details.

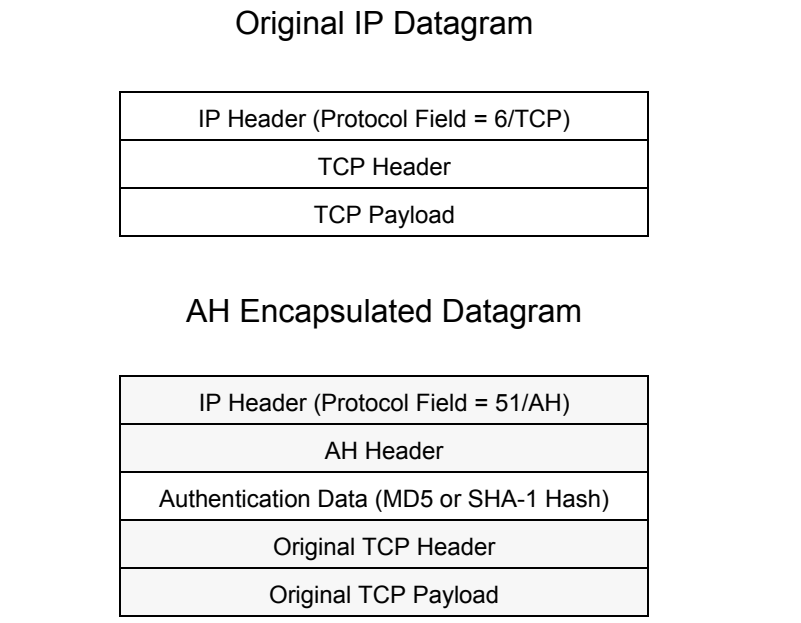


Figure 1: AH Transport Mode

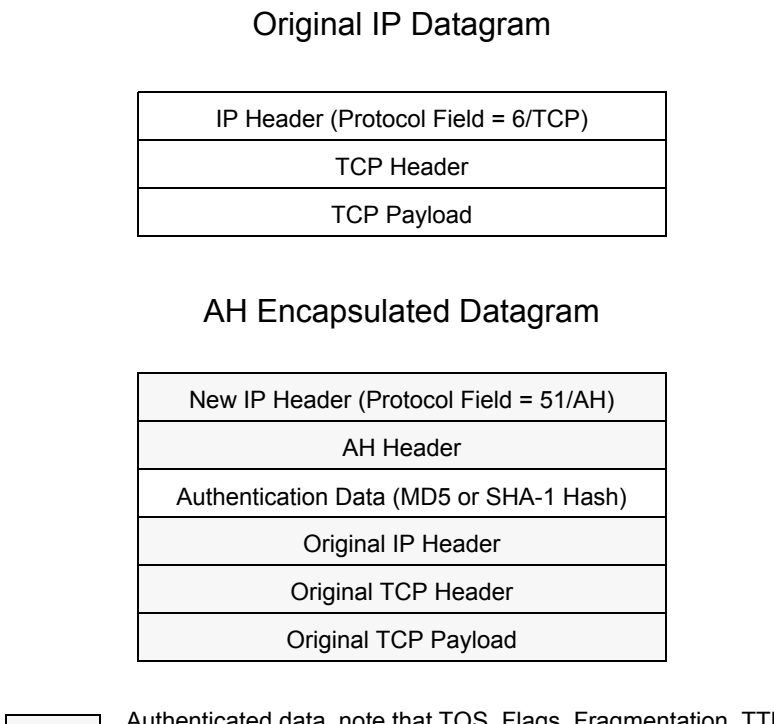


Figure 2: AH Tunnel Mode

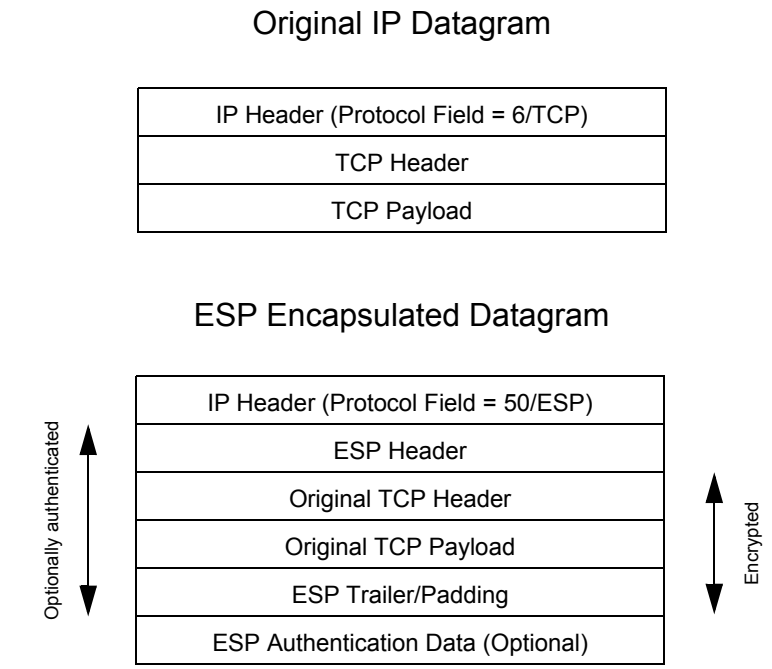


Figure 3: ESP Transport Mode

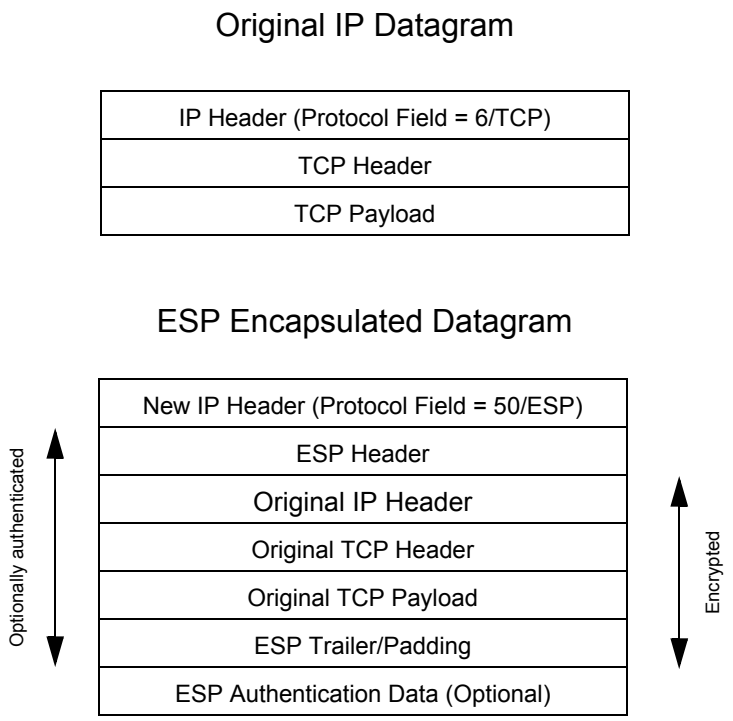


Figure 4: ESP Tunnel Mode

```
ragnarok(ike-sainfo)# security-protocol esp
ragnarok(ike-sainfo)#
```

4. Use the **auth-algo** parameter to specify the authentication algorithms supported by this SA.

   The following authentication protocols are available

   Message Digest Algorithm 5 (md5) — as defined by RFC 1321, *The MD5 Message-Digest Algorithm*.

   Secure Hash Algorithm (sha) — as defined by FIPS PUB 180-1, *Secure Hash Standard*.

   any (the default) — supports both MD5 and SHA-1.

   ```
   ragnarok(ike-sainfo)# auth-algo md5
   ragnarok(ike-sainfo)#
   ```

5. Use the **encryption-algo** parameter to specify the encryption algorithms supported by this SA.

   The following encryption protocols are available

   Triple DES (3des) — as defined by ANSI X.9.52 1998, *Triple Data Encryption Algorithm Modes of Operation*.

   Advanced Encryption Standard (aes) — FIPS PUB 197, *Advanced Encryption Standard*.

   NULL Encryption (null) — as described in RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*. This option provides no privacy services, and is not recommended for production environments.

   any (the default) — supports all listed encryption protocols.

   ```
   ragnarok(ike-sainfo)# encryption-algo aes
   ragnarok(ike-sainfo)#
   ```

6. Use the **ipsec-mode** parameter to specify the IPSec operational mode.

   Transport mode (the default) provides a secure end-to-end connection between two IP hosts. Transport mode encapsulates the IP payload.

   Tunnel mode provides VPN service where entire IP packets are encapsulated within an outer IP envelope and delivered from source (an IP host) to destination (generally a secure gateway) across an untrusted internet.

   Refer to the previous figures for encapsulation details.

   ```
   ragnarok(ike-sainfo)# ipsec-mode tunnel
   ragnarok(ike-sainfo)#
   ```

7. If **ipsec-mode** is *tunnel*, use the required **tunnel-local-addr** parameter to specify the IP address of the local IKEv2 interface that terminates the IPsec tunnel.

   This parameter can safely be ignored if **ipsec-mode** is *transport*.

   ```
   ragnarok(ike-sainfo)# tunnel-local-addr 192.169.204.14
   ragnarok(ike-sainfo)#
   ```

8. If **ipsec-mode** is *tunnel*, use the **tunnel-remote-addr** parameter to specify the IP address of the remote IKEv2 peer that terminates the IPsec tunnel.

   Provide the remote IP address, or use the default wild-card value (*) to match all IP addresses.

   This parameter can safely be ignored if **ipsec-mode** is *transport*.

   ```
   ragnarok(ike-sainfo)# tunnel-remote-addr *
   ragnarok(ike-sainfo)#
   ```

9. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv2 SA.
10. If necessary, repeat Steps 1 through 9 to configure additional IKEv2 SAs.

Security Policy Configuration

Use the following procedure to assign an IKEv2 SA to an existing Security Policy.

From superuser mode, use the following command sequence to access *security-policy* configuration mode. While in this mode, you configure security policies.

```
ragnarok# configure terminal
ragnarok(configure)# security
ragnarok(security)# ipsec
ragnarok(ipsec)# security-policy
ragnarok(security-policy)#
```

11. Use the **ike-sainfo-name** parameter to assign an IKEv2 SA to this Security Policy.

    ```
    ragnarok(security-policy)# ike-sainfo-name SA-1
    ragnarok(security-policy)#
    ```

12. Use **done**, **exit**, and **verify-config** to complete configuration of this Security Policy.

The following sample security policies support IKEv2 over the interface. The first policy (*ikepol*) opens port 500, while the second policy (*poll*) specifies IPsec on all other ports.

```
ragnarok# show running-config security-policy
security-policy
    name                            ikepol
    network-interface               s0p0:0
    priority                        0
    local-ip-addr-match             192.168.0.127
    remote-ip-addr-match            172.30.89.11
    local-port-match                500
    remote-port-match               500
    trans-protocol-match            ALL
    direction                       both
    local-ip-mask                   255.255.255.255
    remote-ip-mask                  255.255.255.255
    action                          allow
    ike-sainfo-name
    outbound-sa-fine-grained-mask
        local-ip-mask               255.255.255.255
        remote-ip-mask              255.255.255.255
        local-port-mask             0
        remote-port-mask            0
        trans-protocol-mask         0
        valid                       enabled
        vlan-mask                   0xFFF
    last-modified-by                admin@console
    last-modified-date              2009-11-11 19:06:32
security-policy
    name                            pol1
    network-interface               s0p0:0
    priority                        1
    local-ip-addr-match             172.30.89.10
    remote-ip-addr-match            172.30.89.11
    local-port-match                0
    remote-port-match               0
    trans-protocol-match            ALL
    direction                       both
    local-ip-mask                   255.255.255.255
    remote-ip-mask                  255.255.255.255
    action                          ipsec
    ike-sainfo-name                 ikesa1
    outbound-sa-fine-grained-mask
        local-ip-mask               255.255.255.255
        remote-ip-mask              255.255.255.255
        local-port-mask             0
        remote-port-mask            0
        trans-protocol-mask         0
        valid                       enabled
        vlan-mask                   0xFFF
    last-modified-by                admin@console
    last-modified-date              2009-11-11 19:07:03
```

Tunnel Management with the ACLI

The ACLI provides commands to re-initiate or to delete a specific IPsec tunnels.

To initiate tunnels:

```
ragnarok# security ike initiate-tunnel <interface-IP-address>
```

Initiates the same sequence for establishing IKEv2 initiator tunnels as occurs during system boot.

To delete a specific tunnel:

```
ragnarok# security ipsec delete tunnel <remote-IP-address> <spi>
```

remote-IP-address is the address of the IKEv2 peer at the remote end of the tunnel

7 Management Support

This chapter describes Palladion support, as well as SNMP and HDR statistics for the Net-Net Diameter Director.

Palladion Mediation Engine

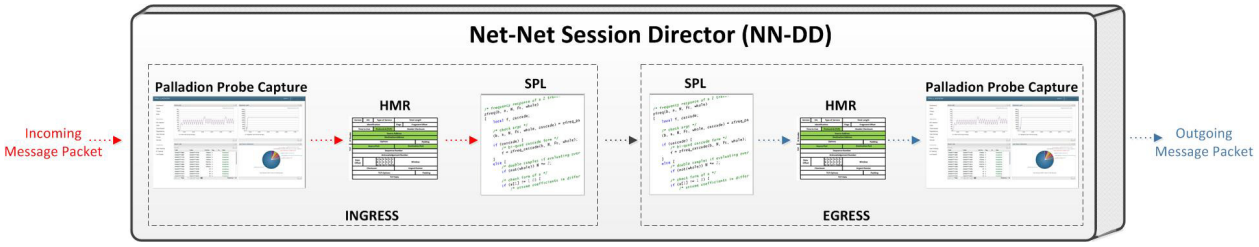
Palladion is Acme Packet's *Communication Experience Manager*.

The manager is powered by the Palladion Mediation Engine, a platform that collects SIP, DNS, ENUM, and other protocol message traffic received from Palladion Probes. The mediation engine stores the traffic in an internal database, and analyzes aggregated data to provide comprehensive, multi-level monitoring, troubleshooting, and interoperability information.

The Net-Net Diameter Director supports an embedded, user-configurable Palladion Communications Monitoring Probe, Version 1. Acting as a Probe, or as an exporter, the Net-Net Diameter Director can:

1. Establish an authenticated, persistent, reliable TCP connection between itself and one or more Palladion Mediation Engines.
2. Use the TCP connection to send a UTC-timestamped, unencrypted copy of a protocol message to the Palladion Engine(s).
3. Accompany the copied message with related data to include: the port/vlan on which the message was sent/received, local and remote IP:port information, and the transport layer protocol.
4. Include information that allows the Palladion ME to correlate messages on which the Net-Net Diameter Director is performing topology hiding, masking and/or obscuring.

The following illustration shows how the Palladion Communications Monitor Probe handles incoming and outgoing monitored data on the Net-Net Diameter Director.



HMR - Header Manipulation Rules
SPL - Session Plug-in Language
SM&T - SIP Monitor and Trace

IPFIX

The Net-Net Diameter Director uses the IPFIX suite of standards to export protocol message traffic and related data to the Palladion Mediation Engine.

- RFC 5101, Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information
- RFC 5102, Information Model for IP Flow Information Export
- RFC 5470, Architecture for IP Flow Information Export
- RFC 5655, Specification of the IP Flow Information Export (IPFIX) File Format
- RFC 5815, Definitions of Managed Objects for IP Flow Information Export

The IPFIX standards describe the use of templates to format the export of specific types of protocol traffic. The Net-Net Diameter Director and the Palladion Mediation Engine share ten pre-defined templates that facilitate protocol message exchange, and subsequent processing and analysis by the Palladion Engine.

The pre-defined templates that apply to the Net-Net Diameter Director include:

- incoming DSC messaging over TCP
- incoming DSC messaging over SCTP
- outgoing DSC messaging over TCP
- outgoing DSC messaging over SCTP
- outgoing DNS over UDP (entire IP and UDP header not included)
- IPFIX handshake (used for connection establishment)

Messaging metadata included specifically for the Net-Net Diameter Director includes diameter-director agent name, interface name, group name and so forth.

Specific metadata TLV data types included to support monitoring of Net-Net Diameter Director traffic include:

```
IPFIX_META_DATA_PAYLOAD_MSG = 0
IPFIX_META_DATA_DD_SESSION_ID = 1
IPFIX_META_DATA_DD_AGENT_HOSTNAME = 2
IPFIX_META_DATA_DD_REALM_ID = 3
IPFIX_META_DATA_DD_ORIGIN_HOST = 4
IPFIX_META_DATA_DD_ORIGIN_REALM = 5
IPFIX_META_DATA_DD_DEST_HOST = 6
IPFIX_META_DATA_DD_DEST_REALM = 7
IPFIX_META_DATA_DD_GROUP_NAME = 8
IPFIX_META_DATA_DD_ROUTE_RECORD = 9
IPFIX_META_DATA_MAX = 10
```

Communications Monitor Configuration

Communications Monitor configuration consists of configuring one or more Net-Net Diameter Director/Palladion exporter/collector pairs. Configuration of the *-config* object, which defines common operations across all interfaces, is not required. Default values can be retained to enable standard service. Note that QoS is not relevant within the

context of Net-Net Diameter Director monitoring and that this setting should retain its default of disabled.

Communication Monitor Probe

Use the following procedure to configure a communication monitoring probe.

1. From superuser mode, use the following ACLI sequence to access *comm-monitor* configuration mode. From *comm-monitor* mode, you establish a connection between the Net-Net Diameter Director, acting as a exporter of protocol message traffic and related data, and a Palladion Mediation Engine, acting as an information collector.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# system
   ACMEPACKET(system)# system-config
   ACMEPACKET(system-config)# comm-monitor
   ACMEPACKET(comm-monitor)#
   ```

2. Use the **state** parameter to enable or disable communication monitoring.

   Communication monitoring is disabled by default.

   ```
   ACMEPACKET(comm-monitor)# state enabled
   ACMEPACKET(comm-monitor)#
   ```

3. Use the **sbc-group-id** parameter to assign an integer value to the Net-Net Diameter Director, in its role as an information exporter.

   Retain the default value (0) or assign another integer value.

   ```
   ACMEPACKET(comm-monitor)# sbc-group-id 5
   ACMEPACKET(comm-monitor)#
   ```

4. Use the **network-interface** parameter to identify the network interface whose traffic will be exported to the Palladion Mediation Engine.

   To specify a media interface (the usual case):

   ```
   ACMEPACKET(comm-monitor)# network-interface m01
   ACMEPACKET(comm-monitor)#
   ```

   To specify the *wancom0* management interface (supported, but not generally used):

   ```
   ACMEPACKET(comm-monitor)# network-interface wancom0:0
   ACMEPACKET(comm-monitor)#
   ```

5. Use the **monitor-collector** parameter to move to *monitor-collector* configuration mode.

   While in this mode you identify a Palladion Mediation Engine (a receiver of exported data) by IP address and port number.

   ```
   ACMEPACKET(comm-monitor)# monitor-collector
   ACMEPACKET(monitor-collector)#
   ```

6. Use the **address** and **port** parameters to specify the IP address and port number monitored by a Palladion Mediation Engine for incoming IPFIX traffic.

   Enter an IPv4 address and a port number with the range **1025** through **65535**, with a default value of **4739**.

   ```
   ACMEPACKET(monitor-collector)# address 172.30.101.239
   ACMEPACKET(monitor-collector)# port 4729
   ACMEPACKET(monitor-collector)#
   ```

7. Use **done** and **exit** to return to *comm-monitor* configuration mode.

8. Use **done**, **exit**, and **verify-config** to complete configuration.
9. Repeat Steps 1 through 10 to configure additional as required.

SNMP

ap-dd.mib

This section describes the index functionality used to index message types and return codes. SNMP objects and their OID values for the ap-dd.mib are included with their corresponding HDR objects in the HDR section of this document. A text version of the ap-dd.mib in its entirety is attached at the end of this document.

Message Type Info Table

apDdMsgTypeInfoTable

This MIB table creates mapping between a message type index and a human-readable message type string to identify the type of message being sent or received.

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apDdMsgTypeInfoTable: 1.3.6.1.4.1.9148.3.12.1.2.3		
Object Identifier Name: apDdMsgTypeInfoEntry: 1.3.6.1.4.1.9148.3.12.1.2.3.1		
apDdMsgTypeIndex	apDdMsgTypeInfoEntry: 1.3.6.1.4.1.9148.3.12.1.2.3.1.1	An index identifying a specific message type
apDdMsgTypeMsgName	apDdMsgTypeInfoEntry: 1.3.6.1.4.1.9148.3.12.1.2.3.1.2	A human-readable string identifying a specific message type

The following table lists possible values for the SNMP query.

Message Type Index	Message Type String
1	OTHER
2	Update-Location
3	Cancel-Location
4	Authentication-Information
5	Insert-Subscriber-Data
6	Delete-Subscriber-Data
7	Purge-UE
8	Reset
9	Notify
10	Credit-Control
11	Auth-Auth
12	Re-Auth
13	Session-Termination

Message Type Index	Message Type String
14	Abort-Session
15	Accounting
16	User-Authorization
17	Server-Assignment
18	Location-Info
19	Multimedia-Auth
20	Registration-Termination
21	Push-Profile
22	Profile-Update
23	Subscribe-Notification
24	Push-Notification
25	Device-Watchdog
26	Disconnect-Peer

Note: The "OTHER" message type is used to count the messages not captured by the supported interfaces. If a new, unsupported interface has new messages "bar", "abc", "pqr", the Net-Net Diameter Director counts all of the new messages as "OTHER".

Return Code Info Table apDdReturnCodeInfoTable

This MIB table creates mapping between a return code index and a human-readable return code string to identify the type of return code.

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apDdMsgReturnCodeInfoTable: 1.3.6.1.4.1.9148.3.12.1.2.5		
Object Identifier Name: apDdMsgReturnCodeInfoEntry: 1.3.6.1.4.1.9148.3.12.1.2.5.1		
apDdReturnCodeIndex	apDdMsgReturnCodeInfoEntry: 1.3.6.1.4.1.9148.3.12.1.2.5.1.1	An index identifying a specific return code type
apDdMsgReturnCodeName	apDdMsgReturnCodeEntry: 1.3.6.1.4.1.9148.3.12.1.2.5.1.2	A human-readable string identifying a specific return code type

The following table lists possible values for the SNMP query:

Message Type Index	Message Type String
1	1001 Multi Rnd Auth
2	1xxx Informational
3	2001 Success
4	2002 Limited Success
5	2xxx Success
6	3001 Cmd Unsupported

Message Type Index	Message Type String
7	3002 Unable Deliver
8	3003 Realm Not Srv
9	3004 Too Busy
10	3005 Loop Detected
11	3006 Redirect
12	3007 App Unsupported
13	3008 Bad HDR Bits
14	3009 Bad AVP Bits
15	3010 Unknown Peer
16	3xxx Protocol Error
17	4001 Auth Reject
18	4002 Out Of Space
19	4003 Election Lost
20	4100 User Data Unav
21	4xxx Transient Error
22	5001 AVP Unsupported
23	5002 Unknown Session
24	5003 Auth Reject
25	5004 Bad AVP Value
26	5005 Missing AVP
27	5006 Resource Exceed
28	5007 AVP Conflict
29	5008 AVP Not Allowed
30	5009 AVP Too Many
31	5010 No Common App
32	5011 Bad Version
33	5012 Unable to Comply
34	5013 Bad HDR Bit
35	5014 Bad AVP Length
36	5015 Bad Msg Length
37	5016 Bad AVP BitCombo
38	5017 No Common Sec
39	5xxx Permanent Error

SCTP Path Traps

This section provides information on traps that the Net-Net Diameter Director suppors.

The following table identifies the applicable interface traps.

Trap Name	Description
apDdConnectionFailureTrap: 1.3.6.1.4.1.9148.3.12.2.2.0.1	Generated when a connection failure event occurrs in a Diameter Director Interface.
apDdConnectionFailureClearTrap: 1.3.6.1.4.1.9148.3.12.2.2.0.2	Generated when a failed connection event is restored in a Diameter Director Interface.

Both of these traps return data as shown in the table below, which displays the data items and their possible values. Note that "Display String" equates to the names of the applicable elements configured on the Net-Net DD.

Payload	Values
apDdInterfaceRealmName	Display String
apDdDiameterAgentHostName	Display String
apDdDiameterIPPort	Display String
apDdDiameterAgentOriginHostName	Display String
apDdDiameterAgentOriginRealmName	Display String
apDdTransportType	unknown(0), tcp(1), sctp(2)
apDdInterfaceStatusReason	disconnectByRequest(0), diameterAgentUnreachable(1), transportFailure(2)

SNMP Alarm

If any or all of the tunnels designated by a *tunnel-orig-params* configuration element fail to establish after the first attempt, the Net-Net DD makes **retry-limit** attempts to establish the tunnel(s) with an interval of **retry-time** seconds between each initiation attempt.

If the tunnels fail to establish after the retry limit is reached, the Net-Net DD issues an apSecurityTunnelFailureNotification with a supported value of *initiator-timeout* assigned to the apSecurityFailureCause field.

After issuing the alarm the Net-Net DD makes no further attempts to initiate tunnels until the next reboot or restart.

HDR and SNMP Statistics

The Net-Net Diameter Director presents identical information in both HDR and SNMP format. You may configure the system to output applicable statistics to local (or remote) flat files in HDR format, or you may retrieve the same information, grouped identically, via SNMP.

What is HDR?

Historical data recording (HDR) refers to a group of management features that allow you to configure the Net-Net SD to collect statistics about system operation and function, and then send those records to designated servers. System statistics, defined in detail below, are saved to a comma-separated value (CSV) file, which are then sent to the designated server(s).

Information types are grouped so that you can refer to a set of statistics by simply invoking their group name (For example, the system statistics are in a group called **System**; interface statistics are in a group called **Interface**; etc.). Within each group, there are several metrics available.

Enabling/Disabling HDR

In the system configuration, you can enable HDR by first turning on the system's collection function, then choosing the records you want to capture, and finally setting up server(s) to which you want records sent.

The main collect configuration (found within the main system configuration) allows you to create global settings that:

- Enable or disable HDR at boot time
- Set the sample rate in seconds, or the time between sample individual collections
- Set the time, in seconds, between individual pushes to designated servers (configured in the push receiver configuration accessed via the collect configuration)
- Set the time you want the collect to start and stop; time is entered in year, month, day, hours, minutes, and seconds

You also configure settings for each group of data you want to collect, and the push receiver (server) to which you want data sent.

For more information about configuring HDR on the Net-Net SD, see Configuring HDR (21).

Protocol Use

You can configure HDR to send files using File Transfer Protocol (FTP) or Secure File Transfer Protocol (SFTP) for added security. FTP is the default.

> **Note:** Public key authentication is not available when you choose SFTP. Instead, the Net-Net SD uses password authentication. However, for SFTP to work, it is still required that you load the SFTP's host public key on the Net-Net SD.

About the CSV File

When HDR is enabled, statistical records are forwarded from the Net-Net SD to push servers that send the data (in standard format) to a receiving server for viewing in a comma-separated value (CSV) file on the server. Before pushing a file, the collector creates the directory by group name for which the statistic belongs (for example, *fan*, *sip-client*, *system*, etc.), if the directory does not exist from a previous push.

The collector can push multiple CSV files per directory. Each file is formatted as <*Unix timestam*p>*.csv* (for example, *1302041977.csv*). The first record of each file is a header containing the attribute name. For example, in the "*System*" directory, a file name of "*13020441977.csv*" can contain the header names of CPU Utilization, Memory Utilization, Health Score, Redundancy State, etc. The collector appends a Timestamp heading attribute to the beginning of every record as well. You can open the CSV file for

viewing with any application that reads a CSV file format. For more information about the CSV file, see HDR Data (13).

> **Note:** The records in a CSV file may display differently, depending on the record data included in the file, and the method used to open the file. For more information about the display of record data in a CSV file, see Appendix A, CSV File Data Formats (161).

Collection Interval and Push

In your HDR configuration, you can set parameters that allow you to:

- Select the groups for record collection
- Set the frequency of record collection
- Set the frequency of off-box record collection

After configuring and enabling HDR, the Net-Net SD forwards group records to push servers that send the data to a receiving server. The number of records in a push equals the push interval divided by the sample interval time multiplied by the number of groups, plus one:

push interval ÷ sample interval time x number of groups +1 header record per group = number of records in a push

For example, if you set a push interval time of 60 minutes and a sample interval time of 5 minutes, with 10 groups, the Net-Net SD would send 120 group records plus 10 header records (for a total of 130 records) for each push:

[(60 ÷ 5) x 10] +10 = 130

You can configure an option parameter (disabled by default) that instructs the Net-Net SD to send a trap when data has been successfully pushed. This trap is defined in the `ap-smgmt.mib`. It contains the name of the node that successfully pushed the HDR file to an HDR server, a unique file name for the HDR file that was pushed, and the IP address of the push receiver (configured in the global collection configuration). For more information about the HDR SNMP traps, see the product-specific *Net-Net SD MIB Reference Guide*.

After each push, the Net-Net SD clears (deletes) all records. The Net-Net SD also clears files on system reboot, and after three consecutive push failures.

HDR Group Names

The Net-Net Diameter Director supports multiple group names for statistical queries. The table below summarizes these HDR groups.

Group Name	Description
dd-general	Presents general summary data that is not specific to any interface
dd-session	Presents diameter session statistics
dd-subscriber	Presents diameter subscriber statistics
dd-interface	Presents DD interface data
dd-interface-return-code	Presents data on return code types for each interface
dd-interface-error	Presents DD interface error statistics

Group Name	Description
dd-interface-message-type	Presents interface message type statistics
dd-agent	Presents data on individual Diameter Director agents
dd-agent-error	Presents error data on individual Diameter Director agents
dd-agent-message-type	Presents counts of message types sent to and received from Diameter Director agents
dd-agent-return-code	Presents return code data on individual Diameter Director agents

HDR/SNMP/ACLI Data Classification

The Net-Net Diameter Director maintains statistics for quantifying performance. The metrics used appear across different measures of application transactions. The following sections describe the often used metrics for quantifying time and network connections:

Period/Timescale Statistics

Several HDR statistics detailed in this document rely on window periods described in the table below:

Metric	Description
period-active	Number of active occurrences of the item being counted. For example, the number of active transactions during a period
period-high	Highest value achieved by the period-active statistic during the defined period
period-total	Total count of occurrences of the item during the defined period
lifetime-total	Total number of occurrences of the item taken across all periods. This is a cumulative value. For example, the total number of sockets created since the system booted is a lifetime-total.
lifetime-permax	Highest period-total value across all periods. For example, over 5 periods the period-total values are 10, 11, 1, 30 and 23. The lifetime-permax value in this instance is 30.
lifetime-high	Highest period-high value across all periods. For example, over 5 periods the period-high values are 13, 4, 12, 42 and 17. The lifetime-high value in this instance is 42.

Connection-Oriented Category

This HDR group is used to present data on specific interfaces. The data is presented for four different categories described below:

Category	Description
Client Transactions	Number of outstanding transactions with the SD acting as the client (initiating a request)
Server Transactions	Number of outstanding transactions with the SD acting as the server (responding to a request)

Category	Description
Sockets	Number of stateful connections, such as TCP connections. This value is calculated (egress client sockets) + (ingress forked sockets)
Connections	Number of all active sockets.

Configuring HDR via the ACLI

This section provides procedures for configuring HDR. HDR configuration includes:

- setting parameters to govern sample and push intervals, and start/end times for collection
- setting parameters to support HDR across a high availability (HA) node
- setting group parameters to inform the Net-Net Session Border Controller (SBC), which groups of records to collect, when to start and stop collecting, and how often to sample for a specific group.
- setting push receivers that transport the records forwarded by the Net-Net SD

Note: If you modify the HDR configuration parameters using the ACLI, the changed parameters DO NOT take affect until you reboot the SD.

Enabling HDR Collection

You access the parameters that enable and support HDR using the ACLI **system-config** path.

To enable HDR collection:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **system** and press <Enter>.

   ```
   ACMEPACKET(configure)# system
   ACMEPACKET(system)#
   ```

3. Type **system-config** and press <Enter>.

   ```
   ACMEPACKET(system)# system-config
   ACMEPACKET(system-config)#
   ```

4. Enter **collect** and press <Enter>. From here, you can type a question mark (?) to see individual parameters for the configuration.

   ```
   ACMEPACKET(system-config)# collect
   ACMEPACKET(collect)#
   ```

Setting Global Collection

You access the collection configuration through the ACLI system-configuration menu. Once in the collection configuration, you can establish the global settings for HDR collection.

To set HDR global collection:

1. In Superuser mode, navigate to the “collect” parameter level in the ACLI.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# system
   ```

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)#
```

2. Set global collection parameters as applicable. Parameters include:

   - **sample-interval**—Enter the time in minutes for how often you want the Net-Net SD to sample data records. The default is **5**. The valid range is:
     - Minimum—1
     - Maximum—120
   - **push-interval**—Enter the time in minutes for how often you want the Net-Net SD to send collected records to push receiver(s). The default is **15**. The valid range is:
     - Minimum—1
     - Maximum—120
   - **boot-state**—Set this parameter to **enabled** to start group collection, or to **disabled** to prevent the Net-Net SD from collecting HDR statistics. This parameter does not go into effect until the system is rebooted. You can also use the ACLI request collect start command to start collection; using this command, you can start collection for all groups, or for one specified group. The default is **disabled**. Valid values are:
     - enabled | disabled
   - **start-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SD to start HDR collection. You can enter **now** to set the start-time to the current time, or you can specify a time in the future. If you specify a time, it must be in the format `yyyy-mm-dd-hh:mm:ss`, where: `yyyy` is the year, `mm` is the month, `dd` is the day, `hh` in the hour, `mm` is the minutes, and `ss` is the second (24-hour clock). The default is **now**.
   - **end-time**—Enter the exact date and time (for your local timezone) when you want the Net-Net SD to finish HDR collection. You can enter **never** to set the time to never end, or you can specify an end time in the future. If you specify a time, it must be in the format `yyyy-mm-dd-hh:mm:ss`, where: `yyyy` is the year, `mm` is the month, `dd` is the day, `hh` in the hour, `mm` is the minutes, and `ss` is the second (24-hour clock). The default is **never**.
   - **push-success-trap-state**—Set this parameter to **enabled** if you want the Net-Net SD to send a trap confirming successful data pushes to HDR servers. Default is **disabled**. Valid values are:
     - enabled | disabled

Setting HDR for an HA Node

If you are using the HDR feature on a High Availability (HA) node (or redundant pair of Net-Net SDs), several parameters in the collection configuration must be set for HDR to perform properly.

Acme Packet recommends strongly that you do not change these parameters from their defaults for a normal HA node configuration. Therefore, if you need to change them to support HDR, you should do so with caution.

To set HDR support across an HA node:

1. In Superuser mode, navigate to the "**collect**" parameter level in the ACLI.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
```

```
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)#
```

2. Set HDR collection parameters for an HA node as applicable. Parameters include:

   - **red-collect-state**—Set the state of HA support for the collector function. The default is **disabled**. Valid values are:
     - enabled | disabled
   - **red-max-trans**—Enter the maximum number of HA synchronized transactions to maintain on the active system in the HA node. The default is **1000**. The valid range is:
     - Minimum—0
     - Maximum—999999999
   - **red-sync-start-time**—Enter the amount of time, in milliseconds, that the active Net-Net SD checks to confirm that it is still the active system in the HA node. If the active system is still adequately healthy, this timer resets itself. If for any reason the active has become the standby, it starts to checkpoint with the newly active system when this timer expires. The default is **5000**. The valid range is:
     - Minimum—0
     - Maximum—999999999
   - **red-sync-comp-time**—Enter amount of time, in milliseconds, that determines how frequently after synchronization the standby Net-Net SD checkpoints with the active Net-Net SD. The first interval occurs after initial synchronizations of the systems; this is the timeout for subsequent synchronization requests. The default is **1000**. The valid range is:
     - Minimum—0
     - Maximum—999999999

Setting Multiple Collection Groups

You can configure the Net-Net SD to collect multiple groups of statistics. Collection group settings are accessible through the collection configuration. For specific group names, group statistics, and values, see HDR Groups and Group Statistics (13).

The “sample-interval”, “start-time”, and “end-time” parameters that you set for multiple collection groups override the same parameters set for global collection.

> **Note:** For multiple collection groups, the “sample-interval” value must always be smaller than the global collection parameter value for “push-interval”.

To set multiple collection groups:

1. In Superuser mode, navigate to the “collect” parameter level in the ACLI.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)#
```

2. Access the collection group (**group-settings**) parameters.

```
ACMEPACKET(collect)# group-settings
ACMEPACKET(group-settings)#
```

3. Set the group parameters for multiple collection groups as applicable. Parameters include:

   - **group-name**—Enter the group name corresponding to the records that you want to collect; there are 25 possible groups for which the Net-Net SD can collect data. The **system** group name is the default for this parameter. For additional group names, see HDR Groups and Group Statistics (13).
   - **sample-interval**—Enter the time in minutes for how often you want the Net-Net SD to sample data records for the specified group. The default is **5**. The valid range is:
     - Minimum—1
     - Maximum—120
   - **boot-state**—Set this parameter to **enabled** to start group collection, or to **disabled** to prevent the Net-Net SD from collecting HDR statistics for this group. This parameter does not go into effect until the system is rebooted. You can also use the ACLI request collect start command to start collection; using this command, you can start collection for all groups, or for one specified group. The default is **disabled**. Valid values are:
     - enabled | disabled
   - **start-time**—Enter the exact date and time (local timezone) when you want the Net-Net SD to start collecting records for this group. You can enter **now** to set the start-time to the current time, or specify a time in the future. If you specify a time, it must be in the format `yyyy-mm-dd-hh:mm:ss`, where: `yyyy` is the year, `mm` is the month, `dd` is the day, `hh` in the hour, `mm` is the minutes, and `ss` is the second (24-hour clock). The default is **now**.
   - **end-time**—Enter the exact date and time (local timezone) when you want the Net-Net SD to stop collecting records for this group. You can enter **never** to set the time to never end, or you can specify an end time in the future. If you specify a time, it must be in the format `yyyy-mm-dd-hh:mm:ss`, where: `yyyy` is the year, `mm` is the month, `dd` is the day, `hh` in the hour, `mm` is the minutes, and `ss` is the second (24-hour clock). The default is **never**.

Setting Servers as Push Receivers

You can configure multiple push receivers that represent FTP or SFTP destination servers for which the Net-Net SD pushes records. Push receiver settings are accessible through the collection configuration.

If you configure more than one server, the Net-Net SD sends data to all of the servers. If one server fails, the Net-Net SD generates an SNMP trap. The Net-Net SD makes 3 attempts to send data to the failed server. If the server cannot receive the data, the Net-Net SD clears the data for that server. For example, if there are four servers configured, and the Net-Net SD successfully pushes data to three of them, the Net-Net SD generates a trap indicating the fourth server is down and after 3 attempts to send the data, the data is cleared.

To set servers as push receivers:

1. In Superuser mode, navigate to the "collect" parameter level in the ACLI.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# system
ACMEPACKET(system)# system-config
ACMEPACKET(system-config)# collect
ACMEPACKET(collect)#
```

2. Access the push receiver (**push-receiver**) parameters.

```
ACMEPACKET(collect)# push-receiver
ACMEPACKET(push-receiver)#
```

- **address**—Enter the IP address of the push receiver (server) to which you want records sent. The default for this parameter is **0.0.0.0**.
- **username**—Enter the username that the Net-Net SD uses when it tries to send records to this push server using FTP. There is no default for this parameter.
- **password**—Enter the password (corresponding to the username) that the Net-Net SD uses when it sends records to this push server using FTP. There is no default for this parameter. Enter this password parameter using the following procedure:

  a. Type the parameter name **password**, and then press <Enter>.

     ```
     ACMEPACKET(push-receiver)# password
     ```

  b. Enter the password that the Net-Net SD uses to send records to the push server. The display does not echo the password you enter.

     ```
     Enter password: [enter the password]
     ```

c. Enter the password again to confirm that you entered the password correctly. If the passwords match, the user prompt displays to continue the push server configuration.

```
Enter password again: [enter the password again]
ACMEPACKET(push-receiver)#
```

If the passwords do not match, an error message displays. Repeat Steps **a** through **c** to set the password.

```
Error:  Password mismatch - aborted.
ACMEPACKET(push-receiver)#
```

- **data-store**—Enter the directory on the push receiver where you want collected data placed. There is no default for this parameter.
- **protocol**—Set this parameter to the protocol with which to send HDR collection record files. Default is **FTP**. Valid values are:

FTP | SFTP

Starting and Stopping HDR using the ACLI

For ease-of-use, you can start and stop record collection from Acme Packet's command line interface (ACLI) in Superuser Mode. You can start and stop record collection for the entire HDR process, or you can specify a group name for which you want to start and stop collection.

Starting HDR

To start record collections:

- In Superuser mode, at the root prompt, enter **request collect start all** and press <**Enter**>. The Net-Net SD starts all record collection.

```
ACMEPACKET# request collect start all
```

To start a group-name record collection:

- In Superuser mode, at the root prompt, enter **request collect start <group-name>**, and press <**Enter**>. The Net-Net SD starts collection for that group name only. In the following example, "voltage" record collection is started.

```
ACMEPACKET# request collect start voltage
```

Stopping HDR

To stop all record collections:

- In Superuser mode, at the root prompt, enter **request collect stop all** and press <**Enter**>. The Net-Net SD stops all record collection.

```
ACMEPACKET# request collect stop all
```

To stop a group-name record collection:

- In Superuser mode, at the root prompt, enter **request collect stop <group-name>**, and press <**Enter**>. The Net-Net SD stops collection for that group name only. In the following example, "voltage" record collection is stopped.

```
ACMEPACKET# request collect stop voltage
```

HDR Monitoring

If there are problems with push attempts, the Net-Net DD sends a trap. There is also a trap to clear the alarm once conditions are rectified.

`apSysMgmtCollectPushUnreachableTrap`

`apSysMgmtCollectPushUnreachableClearTrap`

There is also a collector log that provides information related to this features's system process.

HDR Groups

General Statistics

This collection of statistics found in the dd-general HDR group presents system-wide summary data that is not specific to any interface.

- Configure HDR Group name as: `dd-general`
- SNMP MIB: `apDdMIBGeneralObjects`
- OID:`1.3.6.1.4.1.9148.3.12.1.1`

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	Total Number of Diameter Director Interfaces, enabled and disabled.		apDdInterfaceNumber: 1.3.6.1.4.1.9148.3.12.1.1.1.0	Integer32
2	Current Transactions in the current period	10	apDdCurrentTransRate: 1.3.6.1.4.1.9148.3.12.1.1.2.0	Gauge32
3	High Transaction Rate - Maximum "Current Transactions" reached across all 10 second periods	10	apDdHighTransRate: 1.3.6.1.4.1.9148.3.12.1.1.3.0	Gauge32
4	Low Transaction Rate - Minimum "Current Transactions" reached across all 10 second periods	10	apDdLowTransRate: 1.3.6.1.4.1.9148.3.12.1.1.4	Gauge32

Interface Statistics

This collection of statistics found in the dd-interface HDR group presents collected data by configured Diameter Director Interface. The statistics presented here are the same as those available in the show diameter-director interface ACLI command.

- Configure HDR Group name as: `dd-interface`
- SNMP MIB: apDdInterfaceStatsTable
- OID:`1.3.6.1.4.1.9148.3.12.1.2.1`

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	Interface		apDdInterfaceIndex: 1.3.6.1.4.1.9148.3.12.1.2.1.1.1	integer32
2	Realm		apDdInterfaceRealmName: 1.3.6.1.4.1.9148.3.12.1.2.1.1.2	Display String
3	Client Transactions - Period Active	10	apDdClientTransCPActive: 1.3.6.1.4.1.9148.3.12.1.2.1.1.3	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
4	Client Transactions - Period High	10	apDdClientTransCPHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.4	Counter32
5	Client Transactions - Period Total	10	apDdClientTransCPTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.5	Counter32
6	Client Transactions - Lifetime Period Maximum		apDdClientTransLTPerMax: 1.3.6.1.4.1.9148.3.12.1.2.1.1.7	Counter32
7	Client Transactions - Lifetime High		apDdClientTransLTHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.8	Counter32
8	Client Transactions - Lifetime Total		apDdClientTransLTTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.6	Counter32
9	Server Transactions - Period Active	10	apDdServerTransCPActive: 1.3.6.1.4.1.9148.3.12.1.2.1.1.9	Gauge32
10	Server Transactions - Period High	10	apDdServerTransCPHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.10	Counter32
11	Server Transactions - Period Total	10	apDdServerTransCPTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.11	Counter32
12	Server Transactions - Lifetime Period Maximum		apDdServerTransLTPerMax: 1.3.6.1.4.1.9148.3.12.1.2.1.1.13	Counter32
13	Server Transactions - Lifetime High		apDdServerTransLTHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.14	Counter32
14	Server Transactions - Lifetime Total		apDdServerTransLTTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.12	Counter32
15	Sockets - Period Active	10	apDdGenSocketsCPActive: 1.3.6.1.4.1.9148.3.12.1.2.1.1.15	Gauge32
16	Sockets - Period High	10	apDdGenSocketsCPHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.16	Counter32
17	Sockets - Period Total	10	apDdGenSocketsCPTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.17	Counter32
18	Sockets - Lifetime Period Maximum		apDdGenSocketsLTPerMax: 1.3.6.1.4.1.9148.3.12.1.2.1.1.19	Counter32
19	Sockets - Lifetime High		apDdGenSocketsLTHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.20	Counter32
20	Sockets - Lifetime Total		apDdGenSocketsLTTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.18	Counter32
21	Connections - Period Active	10	apDdGenConnectsCPActive: 1.3.6.1.4.1.9148.3.12.1.2.1.1.21	Gauge32
22	Connections - Period High	10	apDdGenConnectsCPHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.22	Counter32
23	Connections - Period Total	10	apDdGenConnectsCPTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.23	Counter32
24	Connections - Lifetime Period Maximum		apDdGenConnectsLTperMax: 1.3.6.1.4.1.9148.3.12.1.2.1.1.25	Counter32
25	Connections - Lifetime High		apDdGenConnectsLTHigh: 1.3.6.1.4.1.9148.3.12.1.2.1.1.26	Counter32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
26	Connections - Lifetime Total		apDdGenConnectsLTTotal: 1.3.6.1.4.1.9148.3.12.1.2.1.1.24	Counter32
27	Interface Status		apDdInterfaceStatus 1.3.6.1.4.1.9148.3.12.1.2.1.1.27	integer32

Interface MIB Name Construction

The SNMP object names are derived by concatenating a prefix of "apDd" with the abbreviated Connection-Oriented Category and the abbreviated Period/Timescale Statistics.

```
SNMP Object Name = apDd<Category Abbreviation><Metric Abbreviation>
```

Error Statistics

This collection of statistics found in the dd-interface-error HDR group presents the Net-Net Diameter Director errors. The statistics presented here are the same as those available in the show diameter-director errors ACLI command.

- Configure HDR Group name as: `dd-interface-error`
- SNMP MIB: apDdErrorStatusTable
- OID:`1.3.6.1.4.1.9148.3.12.1.2.2`

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	Interface number, by Index		apDdInterfaceNumber: 1.3.6.1.4.1.9148.3.12.1.1.1.0	Integer32
2	Total number of No Route Found errors in the current period	10	apDdNoRouteFoundRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.1	Gauge32
3	Total number of routing failures across all periods (cumulative)	10	apDdNoRouteFoundTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.2	Counter32
4	Maximum value of **No Route Found Recent** across all periods (high water mark)	10	apDdNoRouteFoundPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.3	Counter32
5	Total number of malformed messages in the current period	10	apDdMalformedMsgRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.4	Gauge32
6	Total number of malformed messages across all periods (cumulative)	10	apDdMalformedMsgTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.5	Counter32
7	Maximum value of **Malformed Msg Recent** across all periods (high water mark)	10	apDdMalformedMsgPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.6	Counter32
8	Total number of rejected message errors in the current period	10	apDdRejectedMsgRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.7	Gauge32
9	Total number of rejected message errors across all periods (cumulative)	10	apDdRejectedMsgTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.8	Counter32
10	Maximum value of **Rejected Messages** across all periods (high water mark)	10	apDdRejectedMsgPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.9	Counter32
11	Total number of dropped message errors in the current period	10	apDdDroppedMsgRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.10	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
12	Total number of dropped message errors across all periods (cumulative)	10	apDdDroppedMsgTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.11	Counter32
13	Maximum value of **Dropped Messages** across all periods (high water mark)	10	apDdDroppedMsgPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.12	Counter32
14	Total number of inbound constraints errors in the current period	10	apDdInboundConstraintsRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.13	Gauge32
15	Total number of inbound constraints errors across all periods (cumulative)	10	apDdInboundConstraintsTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.14	Counter32
16	Maximum value of **Inbound Constraints** across all periods (high water mark)	10	apDdInboundConstraintsPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.15	Counter32
17	Total number of outbound constraints errors in the current period	10	apDdOutboundConstraintsRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.16	Gauge32
18	Total number of outbound constraints errors across all periods (cumulative)	10	apDdOutboundConstraintsTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.17	Counter32
19	Maximum value of **Outbound Constraints** across all periods (high water mark)	10	apDdOOutboundConstraintsPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.18	Counter32

Interface Message Type Statistics

This collection of statistics found in the dd-interface-message-type HDR group presents connection oriented data per message-type on a per-interface basis. The statistics presented here are the same as those available in the show diameter-director interface <interface-name> <message-type> ACLI command.

A simple way to find out all the row indices of the table is to perform a snmpwalk on apDdMsgTypeServerReqRecent.

For HDR, the maximum number of lines of output per sample interval is given by:

```
#lines/sample = (# interfaces) x (# message types)
```

- Configure HDR Group name as: `dd-interface-message-type`
- SNMP MIB: apDdMsgTypeStatsTable
- OID: `1.3.6.1.4.1.9148.3.12.1.2.4`

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	The interface for which this set of statistics applies		apDdInterfaceIndex: 1.3.6.1.4.1.9148.3.12.1.2.1.1.1	Integer32
2	A unique key used to identify the message type via SNMP queries		apDdMsgTypeIndex: 1.3.6.1.4.1.9148.3.12.1.2.3.1.1	integer32
3	The English language name of the message. The mapping of the message type index to the message name is provided via the ApPdMsgTypeInfoTable.		apDdMsgTypeMsgName: 1.3.6.1.4.1.9148.3.12.1.2.3.1.2	String
4	Total number of server requests in the current period		apDdMsgTypeServerReqRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.3	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
5	Total number of server requests across all periods (cumulative)	10	apDdMsgTypeServerReqTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.4	Counter32
6	Maximum value of **Recent Server Requests** across all periods (high water mark)	10	apDdMsgTypeServerReqPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.5	Counter32
7	Total number of client requests in the current period	10	apDdMsgTypeClientReqRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.6	Gauge32
8	Total number of client requests across all periods (cumulative)	10	apDdMsgTypeClientReqTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.7	Counter32
9	Maximum value of **Recent Client Requests** across all periods (high water mark)	10	apDdMsgTypeClientReqPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.8	Counter32
10	Total number of server retransmissions in the current period	10	apDdMsgTypeServerRetransRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.9	Gauge32
11	Total number of server retransmissions across all periods (cumulative)	10	apDdMsgTypeServerRetransTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.10	Counter32
12	Maximum value of Recent Server Retransmissions across all periods	10	apDdMsgTypeServerRetransPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.11	Counter32
13	Total number of client retransmissions in the current period	10	apDdMsgTypeClientRetransRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.12	Gauge32
14	Total number of client retransmissions across all periods (cumulative)	10	apDdMsgTypeClientRetransTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.13	Counter32
15	Maximum value of **Recent Client Retransmissions** across all periods (high water mark)	10	apDdMsgTypeClientRetransPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.14	Counter32
16	Total number of server Response Retransmissions in the current period	10	apDdMsgTypeServerRespRetransRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.15	Gauge32
17	Total number of server Response Retransmissions across all periods (cumulative)	10	apDdMsgTypeServerRespRetransTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.16	Counter32
18	Maximum value of **Recent Server Response Retransmissions** across all periods (high water mark)	10	apDdMsgTypeServerRespRetransPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.17	Counter32
19	Total number of client Response Retransmissions in the current period	10	apDdMsgTypeClientRespRetransRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.21	Gauge32
20	Total number of client Response Retransmissions across all periods (cumulative)	10	apDdMsgTypeClientRespRetransTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.22	Counter32
21	Maximum value of **Recent Client Response Retransmissions** across all periods (high water mark)	10	apDdMsgTypeClientRespRetransPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.23	Counter32
22	Total number of client timeouts in the current period	10	apDdMsgTypeClientTimeoutRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.27	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
23	Total number of client timeouts across all periods (cumulative)	10	apDdMsgTypeClientTimeoutTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.28	Counter32
24	Maximum value of **Recent Client timeouts** across all periods (high water mark)	10	apDdMsgTypeClientTimeoutPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.29	Counter32
25	Total number of client throttled requests in the current period	10	apDdMsgTypeClientThrottledRecent: 1.3.6.1.4.1.9148.3.12.1.2.4.1.33	Gauge32
26	Total number of dropped client requests across all periods (cumulative)	10	apDdMsgTypeClientThrottledTotal: 1.3.6.1.4.1.9148.3.12.1.2.4.1.34	Counter32
27	Maximum value of **Recent Client Throttled Requests** across all periods (high water mark)	10	apDdMsgTypeClientThrottledPerMax: 1.3.6.1.4.1.9148.3.12.1.2.4.1.35	Counter32
28	The average latency between the sending of the request and the receipt of the acknowledgement of the request		apDdMsgTypeAverageLatency: 1.3.6.1.4.1.9148.3.12.1.2.4.1.36	Gauge32
29	The maximum computed latency between the sending of the request and the receipt of the acknowledgement of the request		apDdMsgTypeMaximumLatency: 1.3.6.1.4.1.9148.3.12.1.2.4.1.37	Gauge32
30	Time period, in seconds, over which the average latency is computed		apDdMsgTypeLatencyWindow: 1.3.6.1.4.1.9148.3.12.1.2.4.1.38	Integer32

Interface Return Code Statistics

This collection of statistics found in the dd-interface-return-code HDR group presents connection oriented data per return-code on a per-interface basis. These statistics supplement the Interface Message Type Statistics found in the previous section.

The maximum number of lines of output per sample interval using HDR is given by:

```
#lines/sample = (# interfaces) x (# message types) x (# return codes)
```

- Configure HDR Group name as: `dd-interface-return-code`
- SNMP MIB: apDdMsgStatsReturnCodeTable
- OID: `1.3.6.1.4.1.9148.3.12.1.2.6`

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	The interface for which this set of statistics applies		apDdInterfaceIndex: 1.3.6.1.4.1.9148.3.12.1.2.1.1.1	Integer32
2	Message Type prior to returned message		apDdMsgTypeIndex: 1.3.6.1.4.1.9148.3.12.1.2.3.1.1	integer32
3	This is a unique key used to identify the message type via SNMP queries		apDdMsgReturnCodeIndex: 1.3.6.1.4.1.9148.3.12.1.2.5.1.1	integer32
4	This is the English language name of the return code. The mapping of the return code index to the return code name is provided via the aApPdMsgTypeInfoTable		apDdMsgReturnCodeName: 1.3.6.1.4.1.9148.3.12.1.2.5.1.2	String
5	Number of replies from the server in the most recent period	10	apDdMsgReturnCodeServerRecent: 1.3.6.1.4.1.9148.3.12.1.2.6.1.3	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
6	Total number of server replies across all periods (cumulative)	10	apDdMsgReturnCodeServerTotal: 1.3.6.1.4.1.9148.3.12.1.2.6.1.4	Counter32
7	Maximum value of period-total server replies across all periods (high water mark)	10	apDdMsgReturnCodeServerPerMax: 1.3.6.1.4.1.9148.3.12.1.2.6.1.5	Counter32
8	Number of replies from the in the most recent period	10	apDdMsgReturnCodeClientRecent: 1.3.6.1.4.1.9148.3.12.1.2.6.1.6	Gauge32
9	Total number of client replies across all periods (cumulative)	10	apDdMsgReturnCodeClientTotal: 1.3.6.1.4.1.9148.3.12.1.2.6.1.7	Counter32
10	Maximum value of period-total client replies across all periods (high water mark)	10	apDdMsgTypeClientReqPerMax: 1.3.6.1.4.1.9148.3.12.1.2.6.1.8	Counter32

Per-Agent KPI

The following groups record individual statistics for each configured Diameter Director Agent. All status changes, errors and messages tied to a particular Diameter Director Agent are recorded to a unique statistics list.

Agent Data

This collection of statistics found in the dd-agent HDR group presents Diameter Director Agent data.

- Configure HDR Group name as: `dd-agent`
- SNMP MIB: apDdAgentStatsTable
- OID: `1.3.6.1.4.1.9148.3.12.1.2.7`

HDR Position	Statistic Recorded	Description	SNMP Object	SNMP Data Type
1	Agent	The index ID of the agent for this set of statistics. This index will be persistent across reboots.	apDdAgentIndex 1.3.6.1.4.1.9148.3.12.1.2.7.1.1 Used as the key in an SNMP query to identify the agent.	integer32 (non-zero)
2	Realm	The hostname of the diameter-director agent.	apDdAgentRealmName 1.3.6.1.4.1.9148.3.12.1.2.7.1.2	DisplayString
3	Period Active Client Trans	Number of active client transactions in the current period	apDdClientTransCPActive 1.3.6.1.4.1.9148.3.12.1.2.7.1.3	Gauge32
4	Period High Client Trans	Maximum value of “Period Active Client Trans” in the current period	apDdClientTransCPHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.4	Counter32
5	Period Total Client Trans	Total number of client transactions in the current period	apDdClientTransCPTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.5	Counter32
6	PerMax Client Trans	Maximum value of “Period Total Client Trans” across all periods	apDdClientTransLTPerMax 1.3.6.1.4.1.9148.3.12.1.2.7.1.7	Counter32
7	Lifetime High Client Trans	Maximum value of “Period High Client Trans” across all periods	apDdClientTransLTHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.8	Counter32
8	Lifetime Total Client Trans	Total number of client transactions across all periods	apDdClientTransLTTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.6	Counter32

HDR Position	Statistic Recorded	Description	SNMP Object	SNMP Data Type
9	Period Active Server Trans	Number of active server transactions in the current period	apDdServerTransCPActive 1.3.6.1.4.1.9148.3.12.1.2.7.1.9	Gauge32
10	Period High Server Trans	Maximum value of “Period Active Server Trans” in the current period	apDdServerTransCPHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.10	Counter32
11	Period Total Server Trans	Total number of server transactions in the current period	apDdServerTransCPTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.11	Counter32
12	PerMax Server Trans	Maximum value of “Period Total Server Trans” across all periods	apDdServerTransLTPerMax 1.3.6.1.4.1.9148.3.12.1.2.7.1.13	Counter32
13	Lifetime High Server Trans	Maximum value of “Period High Server Trans” across all periods	apDdServerTransLTHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.14	Counter32
14	Lifetime Total Server Trans	Total number of server transactions across all periods	apDdServerTransLTTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.12	Counter32
15	Period Active Sockets	Number of active sockets in the current period	apDdGenSocketsCPActive 1.3.6.1.4.1.9148.3.12.1.2.7.1.15	Gauge32
16	Period High Sockets	Maximum value of “Period Active Sockets” in the current period	apDdGenSocketsCPHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.16	Counter32
17	Period Total Sockets	Total number of sockets created in the current period	apDdGenSocketsCPTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.17	Counter32
18	Permax Sockets	Maximum value of “Period Total Sockets” across all periods	apDdGenSocketsLTPerMax 1.3.6.1.4.1.9148.3.12.1.2.7.1.19	Counter32
19	Lifetime High Sockets	Maximum value of “Period High Sockets” across all periods	apDdGenSocketsLTHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.20	Counter32
20	Lifetime Total Sockets	Total number of sockets created across all periods	apDdGenSocketsLTTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.18	Counter32
21	Period Active Connections	Number of active connections in the current period	apDdGenConnectsCPActive 1.3.6.1.4.1.9148.3.12.1.2.7.1.21	Gauge32
22	Period High Connections	Maximum value of “Period Active Connections” in the current period	apDdGenConnectsCPHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.22	Counter32
23	Period Total Connections	Total number of connections created in the current period	apDdGenConnectsCPTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.23	Counter32
24	PerMax Connections	Maximum value of “Period Total Connections” across all periods	apDdGenConnectsLTPerMax 1.3.6.1.4.1.9148.3.12.1.2.7.1.25	Counter32
25	Lifetime High Connections	Maximum value of “Period High Connections” across all periods	apDdGenConnectsLTHigh 1.3.6.1.4.1.9148.3.12.1.2.7.1.26	Counter32
26	Lifetime Total Connections	Total number of connections created across all periods	apDdGenConnectsLTTotal 1.3.6.1.4.1.9148.3.12.1.2.7.1.24	Counter32
27	Diameter Director Agent Status	HDR output/SNMP output • O/0 - out-of-service • I/1 - in-service • C/2 - constrained	apDdAgentStatus 1.3.6.1.4.1.9148.3.12.1.2.7.1.27	integer32

Agent Error Data

The agent error data contains common error conditions.

The SNMP table is indexed by the apDdAgentIndex referenced in the Agent Data (137) and has the objects shown in the table below.

In HDR output, each sample interval produces one line of output for each enabled agent.

This collection of statistics found in the dd-agent-error HDR group presents Diameter Director Agent Error data.

- Configure HDR Group name as: dd-agent-error
- SNMP MIB: `apDdAgentErrorStatusTable`
- OID: `1.3.6.1.4.1.9148.3.12.1.2.8`

HDR Position	Statistic Recorded	Description	SNMP Object	SNMP Data Type
1	Agent	The index ID of the agent for this set of statistics. This index will be persistent across reboots.	apDdAgentIndex 1.3.6.1.4.1.9148.3.12.1.2.7.1.1 Used as the key in an SNMP query to identify the agent.	integer32 (non-zero)
2	No Route Found Recent	Total number of routing failures in the current period	apDdNoRouteFoundRecent 1.3.6.1.4.1.9148.3.12.1.2.2.1.1	Gauge32
3	No Route Found Total	Total number of routing failures across all periods (cumulative)	apDdNoRouteFoundTotal 1.3.6.1.4.1.9148.3.12.1.2.2.1.2	Counter32
4	No Route Found PerMax	Maximum value of “No Route Found Recent” across all periods (high water mark)	apDdNoRouteFoundPerMax 1.3.6.1.4.1.9148.3.12.1.2.2.1.3	Counter32
5	Malformed Msg Recent	Total number of malformed messages in the current period	apDdMalformedMsgRecent 1.3.6.1.4.1.9148.3.12.1.2.2.1.4	Gauge32
6	Malformed Msg Total	Total number of malformed messages across all periods (cumulative)	apDdMalformedMsgTotal 1.3.6.1.4.1.9148.3.12.1.2.2.1.5	Counter32
7	Malformed Msg PerMax	Maximum value of “Malformed Msg Recent” across all periods (high water mark)	apDdMalformedMsgPerMax 1.3.6.1.4.1.9148.3.12.1.2.2.1.6	Counter32
8	Rejected Msg Recent	Total number of rejected message errors in the current period	apDdRejectedMsgRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.7	Gauge32
9	Rejected Msg Total	Total number of rejected message errors across all periods (cumulative)	apDdRejectedMsgTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.8	Counter32
10	Rejected Msg Per Max	Maximum value of **Rejected Messages** across all periods (high water mark)	apDdRejectedMsgPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.9	Counter32
11	Dropped Msg Recent	Total number of dropped message errors in the current period	apDdDroppedMsgRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.10	Gauge32
12	Dropped Msg Total	Total number of dropped message errors across all periods (cumulative)	apDdDroppedMsgTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.11	Counter32

HDR Position	Statistic Recorded	Description	SNMP Object	SNMP Data Type
13	Dropped Msg Per Max	Maximum value of **Dropped Messages** across all periods (high water mark)	apDdDroppedMsgPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.12	Counter32
14	Inbound Constraints Recent	Total number of inbound constraints errors in the current period	apDdInboundConstraintsRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.13	Gauge32
15	Inbound Constraints Total	Total number of inbound constraints errors across all periods (cumulative)	apDdInboundConstraintsTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.14	Counter32
16	Inbound Constraints Per Max	Maximum value of **Inbound Constraints** across all periods (high water mark)	apDdInboundConstraintsPerMax: 1.3.6.1.4.1.9148.3.12.1.2.2.1.15	Counter32
17	Outbound Constraints Recent	Total number of outbound constraints errors in the current period	apDdOutboundConstraintsRecent: 1.3.6.1.4.1.9148.3.12.1.2.2.1.16	Gauge32
18	Outbound Constraints Total	Total number of outbound constraints errors across all periods (cumulative)	apDdOutboundConstraintsTotal: 1.3.6.1.4.1.9148.3.12.1.2.2.1.17	Counter32
19	Outbound Constraints Per Max	Maximum value of **Outbound Constraints** across all periods (high water mark)	apDdOutboundConstraintsPerMax : 1.3.6.1.4.1.9148.3.12.1.2.2.1.18	Counter32

Messages Per Agent

The agent message data contains statistics and counts for Diameter Director Agents with respect to sending and receiving specific Diameter messages.

The SNMP table is indexed by the apDdAgentIndex referenced in the Agent Data (137). The apDdAgentMsgTypeIndex is the second index is in the apDdMsgTypeStatsTable which is the index for the 15 message types.

Between the apDdAgentIndex value and apDdAgentMsgTypeIndex value, you can ascertain if the message type is processed in a Diameter Director Agent or system-wide depending on the value of apDdAgentIndex.

In HDR output, the maximum number of lines of output per sample interval is:

#lines/sample = (# agents) x (# message types)

For example, if 5 agents are enabled, 75 (5x15) lines of output will be produced as part of a sample.

- Configure HDR Group name as: dd-agent-message-type
- SNMP MIB: `apDdMsgTypeStatsTable`

- OID: 1.3.6.1.4.1.9148.3.12.1.2.4

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
1	Agent	The index ID of the agent for this set of statistics. This index will be persistent across reboots.	apDdAgentIndex 1.3.6.1.4.1.9148.3.12.1.2.7.1.1 Used as the key in an SNMP query to identify the agent.	integer32 (non-zero)
2	Message Type Index	This is a unique key used to identify the message type via SNMP queries.	apDdAgentMsgTypeIndex 1.3.6.1.4.1.9148.3.12.1.2.3.1.1	integer32 (non-zero)
3	Message Type Name	This is the English language name of the messagename is in apDdMsgTypeInfoTable.	apDdMsgTypeMsgName 1.3.6.1.4.1.9148.3.12.1.2.3.1.2	Display String
4	Recent Server Requests	Total number of server requests in the current period	apDdMsgTypeServerReqRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.3	Gauge32
5	Total Server Requests	Total number of server requests across all periods	apDdMsgTypeServerReqTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.4	Counter32
6	Permax Server Requests	Maximum value of “Recent Server Requests” across all periods	apDdMsgTypeServerReqPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.5	Counter32
7	Recent Client Requests	Total number of client requests in the current period	apDdMsgTypeClientReqRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.6	Gauge32
8	Total Client Requests	Total number of client requests across all periods	apDdMsgTypeClientReqTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.7	Counter32
9	PerMax Client Requests	Maximum value of “Recent Client Requests” across all periods	apDdMsgTypeClientReqPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.8	Counter32
10	Recent Server Retransmissions	Total number of server retransmissions in the current period	apDdMsgTypeServerRetransRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.9	Gauge32
11	Total Server Retransmissions	Total number of server retransmissions across all periods	apDdMsgTypeServerRetransTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.10	Counter32
12	PerMax Server Retransmissions		apDdMsgTypeServerRetransPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.11	Counter32
13	Recent Client Retransmissions	Total number of client retransmissions in the current period	apDdMsgTypeClientRetransRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.12	Gauge32
14	Total Client Retransmissions	Total number of client retransmissions across all periods	apDdMsgTypeClientRetransTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.13	Counter32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
15	PerMax Client Retransmissions	Maximum value of “Recent Client Retransmissions” across all periods	apDdMsgTypeClientRetransPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.14	Counter32
16	Recent Server Response Retransmissions	Total number of server Response Retransmissions in the current period	apDdMsgTypeServerRespRetransRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.15	Gauge32
17	Total Server Response Retransmissions	Total number of server Response Retransmissions across all periods	apDdMsgTypeServerRespRetransTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.16	Counter32
18	PerMax Server Response Retransmissions	Maximum value of “Recent Server Response Retransmissions across all periods	apDdMsgTypeServerRespRetransPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.17	Counter32
19	Recent Client Response Retransmissions	Total number of client Response Retransmissions in the current period	apDdMsgTypeClientRespRetransRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.21	Gauge32
20	Total Client Response Retransmissions	Total number of client Response Retransmissions across all periods	apDdMsgTypeClientRespRetransTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.22	Counter32
21	PerMax Client Response Retransmissions	Maximum value of “Recent Client Response Retransmissions” across all periods	apDdMsgTypeClientRespRetransPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.23	Counter32
22	Recent Client Timeouts	Total number of client timeouts in the current period	apDdMsgTypeClientTimeoutRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.27	Gauge32
23	Total Client Timeouts	Total number of client timeouts across all periods	apDdMsgTypeClientTimeoutTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.28	Counter32
24	PerMax Client Timeouts	Maximum value of “Recent Client timeouts” across all periods	apDdMsgTypeClientTimeoutPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.29	Counter32
25	Recent Client Throttled Requests	Total number of client throttled requests in the current period	apDdMsgTypeClientThrottledRecent 1.3.6.1.4.1.9148.3.12.1.2.4.1.33	Gauge32
26	Total Client Throttled Requests	Total number of dropped client requests across all periods	apDdMsgTypeClientThrottledTotal 1.3.6.1.4.1.9148.3.12.1.2.4.1.34	Counter32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
27	PerMax Client Throttled Requests	Maximum value of "Recent Client Throttled Requests" across all periods	apDdMsgTypeClientThrottledPerMax 1.3.6.1.4.1.9148.3.12.1.2.4.1.35	Counter32
28	Average Latency	The average latency between the sending of the request and the receipt of the acknowledgement of the request	apDdMsgTypeAverageLatency 1.3.6.1.4.1.9148.3.12.1.2.4.1.36	Gauge32
29	Maximum Latency	The maximum computed latency between the sending of the request and the receipt of the acknowledgement of the request	apDdMsgTypeMaximumLatency 1.3.6.1.4.1.9148.3.12.1.2.4.1.37	Gauge32
30	Average Latency Window	Time period, in seconds, over which the average latency is computed	apDdMsgTypeLatencyWindow 1.3.6.1.4.1.9148.3.12.1.2.4.1.38	Integer32

Agent Return Code Types

The agent return code data contains statistics and counts for Diameter Director Agents with respect to sending and receiving specific Diameter return codes for a given message type.

For SNMP, the table is indexed with three indices: apDdAgentIndex, apDdMsgTypeIndex and apDdMsgReturnCodeIndex. The value of apDdMsgReturnCodeIndex corresponds to one of the 39 return codes which is reflected in the apDdMsgReturnCodeInfoTable. Performing a snmpwalk on apDdMsgReturnCodeName will reveal the correspondence.

For any combination of triplet (apDdAgentIndex, apDdAgentMsgTypeIndex, apDdMsgReturnCodeIndex), if the table row exists, it means that the type of return message code is processed in the Diameter Director Agent or system-wide.

For HDR, the maximum number of lines of output per sample interval is given by:

#lines/sample = (# agent) x (# message types) x (# return codes)

- Configure HDR Group name as: dd-agent-return-code
- SNMP MIB: `apDdMsgStatsReturnCodeTable`
- OID: `1.3.6.1.4.1.9148.3.12.1.2.6`

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
1	Agent	The Agent for which this set of statistics applies. This index will be persistent across reboots of the SD.	apDdAgentIndex 1.3.6.1.4.1.9148.3.12.1.2.7.1.1 Used as the key in an SNMP query to identify the agent.	integer32 (non-zero)
2	Message Type Index	This is a unique key used to identify the message type via SNMP queries.	apDdMsgTypeIndex 1.3.6.1.4.1.9148.3.12.1.2.3.1.1	integer32 (non-zero)

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
3	Return Code Index	This is a unique key used to identify the message type via SNMP queries.	apDdMsgReturnCodeIndex 1.3.6.1.4.1.9148.3.12.1.2.5.1.1	integer32 (non-zero)
4	Return Code Name	This is the English language name of the return code. The mapping of the return code index to the return code name is provided via the apDdMsgTypeInfoTable.	apDdMsgReturnCodeName 1.3.6.1.4.1.9148.3.12.1.2.5.1.2	Display String
5	Recent Server Replies	Number of replies from the server in the most recent period	apDdMsgReturnCodeServerRecent 1.3.6.1.4.1.9148.3.12.1.2.6.1.3	Gauge32
6	Total Server Replies	Total number of server replies across all periods (cumulative)	apDdMsgReturnCodeServerTotal 1.3.6.1.4.1.9148.3.12.1.2.6.1.4	Counter32
7	Permax Server Replies	This metric is based on a period-total of server replies. It is the maximum value of period-total server replies across all periods (high water mark)	apDdMsgReturnCodeServerPerMax 1.3.6.1.4.1.9148.3.12.1.2.6.1.5	Counter32
8	Recent Client Replies	Number of replies from the in the most recent period	apDdMsgReturnCodeClientRecent 1.3.6.1.4.1.9148.3.12.1.2.6.1.6	Gauge32
9	Total Client Replies	Total number of client replies across all periods (cumulative)	apDdMsgTypeClientReqTotal 1.3.6.1.4.1.9148.3.12.1.2.6.1.7	Counter32
10	PerMax Client Replies	This metric is based on a period-total of client replies. It is the maximum value of period-total client replies across all periods (high water mark)	apDdMsgTypeClientReqPerMax 1.3.6.1.4.1.9148.3.12.1.2.6.1.8	Counter32

Session Statistics

This collection of statistics found in the dd-sessions HDR group presents data collected by the Diameter Director. The statistics presented here are the same as those available in the show sessions ACLI command and can also be obtained by a walk on the apDDMIBGeneralObjects container.

- Configure HDR Group name as: `dd-session`
- SNMP MIB: apDDMIBGeneralObjects
- OID: `1.3.6.1.4.1.9148.3.12.1.1`

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
1	Session Period Active	Number of active DD sessions in this period.	apDdSessionPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.6	Integer32
2	Session Period High	Highest number of DD sessions in this period	apDdSessionPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.7	Integer32
3	Session Period Total	Total Number of the DD session in this period	apDdSessionPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.8	Integer32
4	Session Life Total	Total Number of the DD session in life	apDdSessionLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.9	Integer32
5	Session Life PerMax	PerMax number of the DD session in life	apDdSessionLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.10	Integer32
6	Session Life High	Highest number of DDs session in life	apDdSessionLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.11	Integer32
7	Initial Period Active	Number of Active DD sessions and in Initial state in this period	apDdSessInitPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.12	Integer32
8	Initial Period High	Highest number of DD sessions in Initial state in this period	apDdSessInitPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.13	Integer32
9	Initial Period Total	Total Number of DD sessions in Initial state in this period	apDdSessInitPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.14	Integer32
10	Initial Life Total	Total Number of DD sessions in Initial state in life	apDdSessInitLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.15	Integer32
11	Initial Life PerMax	PerMax number of DD sessions in Initial state in life	apDdSessInitLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.16	Integer32
12	Initial Life High	Highest number of DD sessions in Initial state in life	apDdSessInitLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.17	Integer32
13	Established Period Active	Number of Active DD sessions in Established state in this period	apDdSessEstablishedPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.18	Integer32
14	Established Period High	Highest number of DD sessions in Established state in this period	apDdSessEstablishedPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.19	Integer32
15	Established Period Total	Total number of DD sessions in Established state in this period	apDdSessEstablishedPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.20	Integer32
16	Established Life Total	Total number of DD sessions in Established state in this life	apDdSessEstablishedLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.21	Integer32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
17	Established Life PerMax	PerMax number of DD sessions in Established state in life	apDdSessEstablishedLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.22	Integer32
18	Established Life High	Highest number of DD sessions in Established state in life	apDdSessEstablishedLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.23	Integer32
19	Terminated Period Active	Number of Active DD sessions in Terminated state in this period	apDdSessTerminatedPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.24	Integer32
20	Terminated Period High	Highest number of DD sessions in Terminated state in this period	apDdSessTerminatedPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.25	Integer32
21	Terminated Period Total	Total number of DD sessions in Terminated state in this period	apDdSessTerminatedPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.26	Integer32
22	Terminated Life Total	Total Number of DD sessions in Terminated state in life	apDdSessTerminatedLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.27	Integer32
23	Terminated Life PerMax	PerMax Number of DD sessions in Terminated state in life	apDdSessTerminatedPerMax 1.3.6.1.4.1.9148.3.12.1.1.28	Integer32
24	Terminated Life High	Highest number of DD sessions in Terminated state in life	apDdSessTerminatedLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.29	Integer32
25	Timeout Period Total	Total Number of DD sessions in Timeout state in this period	apDdSessTimeoutPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.30	Integer32
26	Timeout Life Total	Total Number of DD sessions in Timeout state in life	apDdSessTimeoutLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.31	Integer32
27	Timeout Life PerMax	PerMax Number of DD sessions in Timeout state in life	apDdSessTimeoutLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.32	Integer32
28	Errors Period Total	Total Number of DD sessions in Errors state in this period	apDdSessErrorsPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.33	Integer32
29	Errors Life Total	Total Number of DD sessions in Errors state in life	apDdSessErrorsLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.34	Integer32
30	Errors Life PerMax	PerMax Number of DD sessions in Errors state in life	apDdSessErrorsLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.35	Integer32
31	Session Miss Period Total	Total Number of DD sessions in Miss state in this period	apDdSessMissPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.36	Integer32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
32	Session Miss Life Total	Total Number of DD sessions in Miss state in life	apDdSessMissLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.37	Integer32
33	Session Miss Life PerMax	PerMax Number of DD sessions in Miss state in life	apDdSessMissLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.38	Integer32

Subscriber Statistics

This collection of statistics found in the dd-sessions HDR group presents data collected by the Diameter Director. The statistics presented here are the same as those available in the show diameter-director subscribers ACLI command and can also be obtained by a walk on the apDDMIBGeneralObjects container.

- Configure HDR Group name as: `dd-subscriber`
- SNMP MIB: apDDMIBGeneralObjects
- OID: `1.3.6.1.4.1.9148.3.12.1.1`

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
1	Subscriber Period Active	Number of Active DD subscribers in this period.	apDdSubscriberPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.100	Integer32
2	Subscriber Period High	Highest number of DD subscribers in this period	apDdSubscriberPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.101	Integer32
3	Subscriber Period Total	Total number of DD subscribers in this period	apDdSubscriberPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.102	Integer32
4	Subscriber Life Total	Total number of DD subscribers in life	apDdSubscriberLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.103	Integer32
5	Subscriber Life PerMax	PerMax number of DD subscribers in life	apDdSubscriberLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.104	Integer32
6	Subscriber Life High	Highest number of DD subscribers in life	apDdSubscriberLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.105	Integer32
7	Subscribe Period Active	Number of Active DD subscribers in Initial state in this period	apDdSubscribePeriodActive 1.3.6.1.4.1.9148.3.12.1.1.106	Integer32
8	Subscribe Period High	Highest Number of DD subscribers in Initial state in this period	apDdSubscribePeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.107	Integer32
9	Subscribe Period Total	Total Number of DD subscribers in Initial state in this period	apDdSubscribePeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.108	Integer32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
10	Subscribe Life Total	Total Number of DD subscribers in Initial state in life	apDdSubscribeLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.109	Integer32
11	Subscribe Life PerMax	perMax Number of DD subscribers in Initial state in life	apDdSubscribeLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.110	Integer32
12	Subscribe Life High	Highest Number of DD subscribers in Initial state in life	apDdSubscribeLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.111	Integer32
13	Unsubscribe Period Active	Number of Active DD subscribers in Established state in this period	apDdUnSubscribePeriodActive 1.3.6.1.4.1.9148.3.12.1.1.112	Integer32
14	Unsubscribe Period High	Highest number of DD subscribers in Established state in this period	apDdUnSubscribePeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.113	Integer32
15	Unsubscribe Period Total	Total number of DD subscribers in Established state in this period	apDdUnSubscribePeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.114	Integer32
16	Unsubscribe Life Total	Total number of DD subscribers in Established state in life	apDdUnSubscribeLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.115	Integer32
17	Unsubscribe Life PerMax	PerMax number of DD subscribers in Established state in life	apDdUnSubscribeLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.116	Integer32
18	Unsubscribe Life High	Highest number of DD subscribers in Established state in life	apDdUnSubscribeLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.117	Integer32
19	Policy Hit Period Active	Number of Active DD subscribers in Terminated state in this period	apDdPolicyHitPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.118	Integer32
20	Policy Hit Period High	Highest number of DD subscribers in Terminated state in this period	apDdPolicyHitPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.119	Integer32
21	Policy Hit Period Total	Total number of DD subscribers in Terminated state in this period	apDdPolicyHitPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.120	Integer32
22	Policy Hit Life Total	Total number of DD subscribers in Terminated state in life	apDdPolicyHitLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.121	Integer32
23	Policy Hit Life PerMax	PerMax number of DD subscribers in Terminated state in life	apDdPolicyHitLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.122	Integer32
24	Policy Hit Life High	Highest number of DD subscribers in Terminated state in life	apDdPolicyHitLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.123	Integer32

HDR Position	Column Name	Description	SNMP Object	SNMP Data Type
25	Policy Miss Period Active	Number of Active DD subscribers in Timeout state in this period	apDdPolicyMissPeriodActive 1.3.6.1.4.1.9148.3.12.1.1.124	Integer32
26	Policy Miss Period High	Highest number of DD subscribers in Timeout state in this period	apDdPolicyMissPeriodHigh 1.3.6.1.4.1.9148.3.12.1.1.125	Integer32
27	Policy Miss Period Total	Total number of DD subscribers in Timeout state in this period	apDdPolicyMissPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.126	Integer32
28	Policy Miss Life Total	Total number of DD subscribers in Timeout state in life	apDdPolicyMissLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.127	Integer32
29	Policy Miss Life PerMax	PerMax number of DD subscribers in Timeout state in life	apDdPolicyMissLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.128	Integer32
30	Policy Miss Life High	Highest number of DD subscribers in Timeout state in life	apDdPolicyMissLifeHigh 1.3.6.1.4.1.9148.3.12.1.1.129	Integer32
31	Subscriber Miss Period Total	Total number of DD subscribers in Errors state in period	apDdSubscriberMissPeriodTotal 1.3.6.1.4.1.9148.3.12.1.1.130	Integer32
32	Subscriber Miss Life Total	Total number of DD subscribers in Errors state in life	apDdSubscriberMissLifeTotal 1.3.6.1.4.1.9148.3.12.1.1.131	Integer32
33	Subscriber Miss Life PerMax	PerMax number of DD subscribers in Errors state in life	apDdSubscriberMissLifePerMax 1.3.6.1.4.1.9148.3.12.1.1.132	Integer32

System Group Statistics

This collection of statistics found in the system HDR group presents system-wide summary data that is not specific to any interface.

- Configure HDR Group name as: `system`
- SNMP MIB: `apSysMgmtMIBGeneralObjects`

- OID:1.3.6.1.4.1.9148.3.2.1.1

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
CPU Utilization	Percentage of total usage of Session Director's (SD) central processing unit (CPU).	N/A	0% to 100%	apSysCPUUtil: 1.3.6.1.4.1.9148.3.2.1.1.1	gauge
Memory Utilization	Percentage of total memory usage on SD	N/A	0% to 100%	apSysMemoryUtil: 1.3.6.1.4.1.9148.3.2.1.1.2	gauge
Health Score	Percentage of system health with a value of 100% being the healthiest.	N/A	0% to 100%	apSysHealthScore: 1.3.6.1.4.1.9148.3.2.1.1.3	gauge
Redundancy State	For Net-Net high availability (HA), specifies whether this Net-Net SD is active or standby. A standalone system has an active state.	N/A	**active** (1): In active mode. **standby** (2): In standby mode. **unassigned** (3): Not been assigned as "active" or "standby". **recovery** (4): In recovery mode. **outOfService** (5): Currently out of service. Contact your Technical Support representative.	apSysRedundancy: 1.3.6.1.4.1.9148.3.2.1.1.4	integer32
Signaling Sessions	Total number of global, concurrent, active sessions in real time.	N/A	0 to $2^{32} - 1$	apSysGlobalConSess: 1.3.6.1.4.1.9148.3.2.1.1.5	Interger32
Signaling Rate	Total number of calls per second (CPS). This is a real-time value which is the sum of SIP H.323 and Media Gateway Control Protocol (MGCP) calls.	N/A	0 to $2^{32} - 1$	apSysGlobalCPS: 1.3.6.1.4.1.9148.3.2.1.1.6	Interger32
CAM Utilization (NAT)	Percentage of Content Addressable Memory (CAM) usage for Network Address Translation (NAT).	N/A	0% to 100%	apSysNATCapacity: 1.3.6.1.4.1.9148.3.2.1.1.7	gauge
CAM Utilization (ARP)	Percentage of Content Addressable Memory (CAM) usage for Address Resolution Protocol (ARP)	N/A	0% to 100%	apSysARPCapacity: 1.3.6.1.4.1.9148.3.2.1.1.8	gauge
I2C Bus State	Current SD state.	N/A	**Online** (0): Online and processing calls. **Becomingoffline** (1): In the process of going offline. **Offline** (2): Offline and not processing calls. However, other administrative functions are available.	apSysState: 1.3.6.1.4.1.9148.3.2.1.1.9	integer32

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
License Capacity	Percentage of licensed SD sessions currently in progress.	N/A	0% to 100%	apSysLicenseCapacity: 1.3.6.1.4.1.9148.3.2.1.1.10	gauge
Current Cached SIP Local Contact Registrations	Total number of currently cached registered contacts in the SD.	N/A	0 to 2^{32} – 1	apSysSipStatsActiveLocalContacts: 1.3.6.1.4.1.9148.3.2.1.1.11	gauge
Current MGCP Public Endpoint Gateway Registrations	Total number of registered Media Gateway Control Protocol (MGCP) gateway endpoints in the SD.	N/A	0 to 2^{32} – 1	apSysMgcpGWEndpoints: 1.3.6.1.4.1.9148.3.2.1.1.12	gauge
Current H323 Number of Registrations	Total number of H323 registrations in the SD.	N/A	0 to 2^{32} – 1	apSysH323Registration: 1.3.6.1.4.1.9148.3.2.1.1.13	gauge
Application Load Rate	Average Central Processing Unit (CPU) utilization of the Net-Net SD during the current window. The average is computed every 10 seconds unless load-limit is configured in the SipConfig record, in which case it is 5 seconds.	30	0 to 2^{32} – 1	apSysApplicationCPULoadRate: 1.3.6.1.4.1.9148.3.2.1.1.16	period
Current Deny Entries Allocated	The total number of endpoints currently denied	NA	NA	apSysCurrentEndptsDenied .1.3.6.1.4.1.9148.3.2.1.1.26 .0	Integer32

Interface Group Statistics

This collection of statistics found in the interface HDR group presents system-wide summary data that is specific to a given interface.

- Configure HDR Group name as: `interface`
- SNMP MIB: `ifEntry`
- OID:1.3.6.1.2.1.2.2.1

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
Index	Unique value that identifies the interface.	N/A	N/A	ifIndex: 1.3.6.1.2.1.2.2.1.1	InterfaceIndex
Description	String that provides a description of the interface.	N/A	N/A	ifDescr 1.3.6.1.2.1.2.2.1.2	DisplayString

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
Type	Type of interface distinguished according to the Physical/Link Protocol(s).	N/A	N/A	ifType: 1.3.6.1.2.1.2.2.1.3	IANAifType
MTU	Maximum Transmission Unit (MTU) - largest datagram size, in octets (eight-bit bytes), that can be sent/received on the interface specified in octets.	N/A	N/A	ifMtu: 1.3.6.1.2.1.2.2.1.4	Integer32
Speed	Estimate of the current bandwidth, in bits per second, on the interface.	N/A	N/A	ifSpeed: 1.3.6.1.2.1.2.2.1.5	Gauge32
Physical Address	IP Address of the interface at the protocol layer immediately below the network layer in the protocol stack.	N/A	N/A	ifPhysAddress: 1.3.6.1.2.1.2.2.1.6	PhysAddress
Admin Status	Current administrative state of the interface.	N/A	N/A	ifAdminStatus: 1.3.6.1.2.1.2.2.1.7	integer32
Operational Status	Current operational state of the interface.	N/A	**up**(1): Interface is operational and in the UP state. **down**(2): Interface is not operational and in the DOWN state. **testing**(3): Interface is in TESTING state. **unknown**(4): Interface state is UNKNOWN. **dormant**(5): Interface is inactive and in DORMANT state. **notPresent**(6): No interface is present. **lowerLayerDown**(7): Lower layer protocol on the interface is down.	ifOperStatus: 1.3.6.1.2.1.2.2.1.8	integer32
If Last Change	Specifies the sysUpTime (system up time) value with the time the interface entered its current operational state	N/A	0 to $2^{32} - 1$	ifLastChange: 1.3.6.1.2.1.2.2.1.9	timeticks
In Octets	Total number of octets received on the interface.	N/A	0 to $2^{32} - 1$	ifInOctets: 1.3.6.1.2.1.2.2.1.10	Counter32
In Unicast Packets	Number of subnetwork-unicast packets delivered to a higher layer protocol. A unicast packet is a regular IP packet that has a destination IP address.	N/A	0 to $2^{32} - 1$	ifInUcastPkts: 1.3.6.1.2.1.2.2.1.11	Counter32
In Non-Unicast Packets	Number of non-unicast packets (i.e., subnetwork-broadcast or subnetwork-multicast packets) delivered to a higher layer protocol.	N/A	0 to $2^{32} - 1$	ifInNUcastPkts: 1.3.6.1.2.1.2.2.1.12	Counter32

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
In Discards	Number of inbound packets that were discarded even though no errors had been detected. This prevented the packets from being delivered to a higher-layer protocol.	N/A	0 to $2^{32} - 1$	ifInDiscards 1.3.6.1.2.1.2.2.1.13	Counter32
In Errors	Number of inbound packets that contained errors, preventing them from being delivered to a higher-layer protocol.	N/A	0 to $2^{32} - 1$	ifInErrors: 1.3.6.1.2.1.2.2.1.14	Counter32
Out Octets	Total number of octets sent out the interface.	N/A	0 to $2^{32} - 1$	ifOutOctets: 1.3.6.1.2.1.2.2.1.16	Counter32
Out Unicast Packets	Total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including packets that were discarded or not sent.	N/A	0 to $2^{32} - 1$	ifOutUcastPkts: 1.3.6.1.2.1.2.2.1.17	Counter32
Out Non-Unicast Packets	Total number of packets that higher-level protocols requested be transmitted to a non-unicast address (i.e., subnetwork-broadcast or subnetwork-multicast addresses), including packets that were discarded or not sent.	N/A	0 to $2^{32} - 1$	ifOutNUcastPkts: 1.3.6.1.2.1.2.2.1.18	Counter32
Out Discards	Number of outbound packets discarded even though no errors were detected, to prevent the packets from being transmitted.	N/A	0 to $2^{32} - 1$	ifOutDiscards: 1.3.6.1.2.1.2.2.1.19	Counter32
Out Errors	Number of outbound packets that were not transmitted because of errors.	N/A	0 to $2^{32} - 1$	ifOutErrors: 1.3.6.1.2.1.2.2.1.20	Counter32

Space Group Statistics

This collection of statistics found in the space HDR group presents system-wide summary data on resource utilization on the Net-Net DD.

- Configure HDR Group name as: `space`
- SNMP MIB: `apSysStorageSpaceEntry`

- OID:1.3.6.1.4.1.9148.3.2.1.1.23.1

System	Statistic Recorded	Timer Value (sec)	Range		SNMP MIB	SNMP Data Type
Volume Name	Name of the volume used for storage space.	N/A	hard-disk0 hard-disk1 hard-disk2 hard-disk3 /ramdrv /boot/code /boot	Hard disk volume 0 Hard disk volume 1 Hard disk volume 2 Hard disk volume 4 RAM drive Boot code volume Boot volume	apSysVolumeName: 1.3.6.1.4.1.9148.3.2.1.1.23.1.2	DisplayString
Space Used	Total space used on the volume in Megabytes (Mb)	N/A	0 to $2^{32} - 1$		apSysVolumeTotalSpace: 1.3.6.1.4.1.9148.3.2.1.1.23.1.3	gauge
Space Available	Total space available on the volume in Megabytes (Mb)	N/A	0 to $2^{32} - 1$		apSysVolumeAvailSpace: 1.3.6.1.4.1.9148.3.2.1.1.23.1.4	gauge

Network-Util Statistics

This collection of statistics found in the `network-util` HDR group, available within the `ap-env-monitor.mib`, presents system-wide summary data on resource utilization on the Net-Net DD.

- Configure HDR Group name as: `network-util`
- SNMP MIB: `apSysMgmtPhyUtilTableEntry`
- OID:1.3.6.1.4.1.9148.3.2.1.8.1.1

System	Statistic Recorded	Timer Value (sec)	Range	SNMP MIB	SNMP Data Type
Index	An integer that contains the ifIndix of a media port	N/A	0 to $2^{32} - 1$	ifIndex: 1.3.6.1.2.1.2.2.1.1	integer32
RX Utilization	Received (Rx) network utlization of the physical port.	1	0 to 100%	apPhyUtilTableRXUtil: 1.3.6.1.4.1.9148.3.2.1.8.1.1.1	gauge
TX Utilization	Transmitted (Tx) network utlization of the physical port.	1	0 to 100%	apPhyUtilTableTXUtil: 1.3.6.1.4.1.9148.3.2.1.8.1.1.2	gauge

DNS Statistics per Interface

This collection of statistics found in the dd.dns HDR group presents summary data that is specific to an interface.

- Configure HDR Group name as: `dd.dns`
- SNMP MIB: `apDdDnsStatsTable`
- OID:`1.3.6.1.4.1.9148.3.12.1.2.11`

- apDdDnsStatsEntry 1.3.6.1.4.1.9148.3.12.1.2.11.1

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
1	Interface - The index ID of the interface for this set of statistics. This index will be persistent across reboots.		apDdInterfaceIndex: 1.3.6.1.4.1.9148.3.12.1.2.1.1.1	integer32
2	Total number of DNS queries in the current period	NA	apDdDnsQueriesRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.1	Gauge32
3	Total number of Queries across all periods (cumulative)	NA	apDdDnsQueriesTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.2	Counter32
4	Maximum value of “DNS Queries Recent” across all periods (high water mark)	NA	apDdDnsQueriesPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.3	Counter32
5	Total number of Cache Hits in the current period	NA	apDdDnsCacheHitsRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.4	Gauge32
6	Total number of Cache Hits across all periods (cumulative)	NA	apDdDnsCacheHitsTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.5	Counter32
7	Maximum value of “Cache Hits Recent” across all periods (high water mark)	NA	apDdDnsCacheHitsPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.6	Counter32
8	Total number of Cache Misses in the current period	NA	apDdDnsCacheMissessRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.7	Gauge32
9	Total number of Cache Misses across all periods (cumulative)	NA	apDdDnsCacheMissesTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.8	Counter32
10	Maximum value of “Cache Misses Recent” across all periods (high water mark)	NA	apDdDnsCacheHitsPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.9	Counter32
11	Total number of Client Errors in the current period	NA	apDdDnsClientErrorsRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.10	Gauge32
12	Total number of Client Errors across all periods (cumulative)	NA	apDdDnsClientErrorsTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.11	Counter32
13	Maximum value of “Client Errors Recent” across all periods (high water mark)	NA	apDdDnsClientErrorsPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.12	Counter32
14	Total number of Server Errors in the current period	NA	apDdDnsServerErrorsRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.13	Gauge32
15	Total number of Server Errors across all periods (cumulative)	NA	apDdDnsServerErrorsTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.14	Counter32
16	Maximum value of “Server Errors Recent” across all periods (high water mark)	NA	apDdDnsServerErrorsPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.15	Counter32
17	Total number of Responses in the current period	NA	apDdDnsResponsesRecent 1.3.6.1.4.1.9148.3.12.1.2.11.1.16	Gauge32

HDR Position	Statistic Recorded	Timer Value (sec)	SNMP MIB	SNMP Data Type
18	Total number of Responses across all periods (cumulative)	NA	apDdDnsResponsesTotal 1.3.6.1.4.1.9148.3.12.1.2.11.1.17	Counter32
19	Maximum value of “Responses Recent” across all periods (high water mark)	NA	apDdDnsResponsesPerMax 1.3.6.1.4.1.9148.3.12.1.2.11.1.18	Counter32

DNS Statistics per Agent

This collection of statistics found in the dd-agent-dns HDR group presents summary data that is specific to an agent.

- Configure HDR Group name as: `dd-agent-dns`
- SNMP MIB: `apDdAgentDnsStatsTable`
- OID:`1.3.6.1.4.1.9148.3.12.1.2.12`
- apDdAgentDnsStatsEntry 1.3.6.1.4.1.9148.3.12.1.2.12.1

1	Agent - The index ID of the agent for this set of statistics. This index will be persistent across reboots.		apDdAgentIndex 1.3.6.1.4.1.9148.3.12.1.2.7.1.1 Used as the key in an SNMP query to identify the agent.	integer32 (non-zero)
2	Total number of DNS queries in the current period	NA	apDdAgentDnsQueriesRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.1	Gauge32
3	Total number of Queries across all periods (cumulative)	NA	apDdAgentDnsQueriesTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.2	Counter32
4	Maximum value of “DNS Queries Recent” across all periods (high water mark)	NA	apDdAgentDnsQueriesPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.3	Counter32
5	Total number of Cache Hits in the current period	NA	apDdAgentDnsCacheHitsRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.4	Gauge32
6	Total number of Cache Hits across all periods (cumulative)	NA	apDdAgentDnsCacheHitsTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.5	Counter32
7	Maximum value of “Cache Hits Recent” across all periods (high water mark)	NA	apDdAgentDnsCacheHitsPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.6	Counter32
8	Total number of Cache Misses in the current period	NA	apDdAgentDnsCacheMissesRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.7	Gauge32
9	Total number of Cache Misses across all periods (cumulative)	NA	apDdAgentDnsCacheMissesTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.8	Counter32
10	Maximum value of “Cache Misses Recent” across all periods (high water mark)	NA	apDdAgentDnsCacheMissesPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.9	Counter32
11	Total number of Client Errors in the current period	NA	apDdAgentDnsClientErrorsRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.10	Gauge32

12	Total number of Client Errors across all periods (cumulative)	NA	apDdAgentDnsClientErrorsTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.11	Counter32
13	Maximum value of “Client Errors Recent” across all periods (high water mark)	NA	apDdAgentDnsClientErrorsPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.12	Counter32
14	Total number of Server Errors in the current period	NA	apDdAgentDnsServerErrorsRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.13	Gauge32
15	Total number of Server Errors across all periods (cumulative)	NA	apDdAgentDnsServerErrorsTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.14	Counter32
16	Maximum value of “Server Errors Recent” across all periods (high water mark)	NA	apDdAgentDnsServerErrorsPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.15	Counter32
17	Total number of Responses in the current period	NA	apDdAgentDnsResponsesRecent 1.3.6.1.4.1.9148.3.12.1.2.12.1.16	Gauge32
18	Total number of Responses across all periods (cumulative)	NA	apDdAgentDnsResponsesTotal 1.3.6.1.4.1.9148.3.12.1.2.12.1.17	Counter32
19	Maximum value of “Responses Recent” across all periods (high water mark)	NA	apDdAgentDnsResponsesPerMax 1.3.6.1.4.1.9148.3.12.1.2.12.1.18	Counter32

Supported Commands for KPI Tracking

The Net-Net Diameter Director supports maintaining Key Performance Indicators (KPI) for Commands that are supported by the following Diameter interfaces:

Interface	Specification	Application-Id	Vendor-ID
Rx	3GPP TS 29.211	16777229	10415
Gq	3GPP TS 29.209	16777222	10415
Rq	ETSI ES 283.026	16777222	13019 10415 [1]
Gy	IETF RFC 4006	4	0
S6a	3GPP TS29.272	16777251	10415
S6d	3GPP TS29.272	16777251	10415
Gx	3GPP TS 29.212	16777238	10415
Gxx	3GPP TS 29.212	16777266	10415
Cx/Dx	3GPP TS 29.228 and 29.229	16777216	10415
Rf/Ro	3GPP TS 32.299	3	10415
Sh	3GPP TS 29.328 and 29.329	16777217	10415

[1] The vendor identifier assigned by IANA to 3GPP is 10415 and the vendor identifier assigned by IANA to ETSI is 13019. The Vendor-Id header for AVPs imported from TS

129 209 [7] shall be set to 3GPP (19415), while AVPs defined in the present document or imported from TS 183 017 [5] or ES 283 034 [4] shall be set to ETSI (13019)

The list of commands found in the previously listed Interfaces is as follows:

Command Name	Code
Abort-Session	274
Accounting	271
Auth-Auth	265
Authentication-Info	318
Cancel-Location	317
Credit-Control	272
Delete-Subcriber-Data	320
Insert-Subscriber-Data	319
Notify	323
Purge-UE	321
Re-Auth	258
Reset	322
Session-Termination	275
Update-Location	316

Cx/Dx Interface and Command Codes

Interface: Cx/Dx

Specification: 3GPP TS 29.228 and 29.229

Application-ID: 16777216

Vendor-ID: 10415

Valid commands for Cx/Dx interfaces:

Command-Name	Abbreviation	Code
User-Authorization-Request	UAR	300
User-Authorization-Answer	UAA	300
Server-Assignment-Request	SAR	301
Server-Assignment-Answer	SAA	301
Location-Info-Request	LIR	302
Location-Info-Answer	LIA	302
Multimedia-Auth-Request	MAR	303
Multimedia-Auth-Answer	MAA	303
Registration-Termination-Request	RTR	304

Command-Name	Abbreviation	Code
Registration-Termination-Answer	RTA	304
Push-Profile-Request	PPR	305
Push-Profile-Answer	PPA	305

Rf/Ro Interface and Command Codes

Interface: Rf/Ro

Specification: 3GPP TS 29.299

Application-ID: 3

Vendor-ID: 10415

Valid commands for Rf interface:

Command-Name	Abbreviation	Code
Accounting-Request	ACR	271
Accounting-Answer	ACA	271
Capabilities-Exchange-Request	CER	257
Capabilities Exchange Answer	CEA	257

Valid commands for Ro interface:

Command-Name	Abbreviation	Code
Credit-Control-Request	CCR	272
Credit-Control-Answer	CCA	272
Capabilities-Exchange-Request	CER	257
Capabilities Exchange Answer	CEA	257

Sh Interface and Command Codes

Interface: Sh

Specification: 3GPP TS 29.328 and 29.329

Application-ID: 16777217

Vendor-ID: 10415

Valid commands for Sh interface:

Command-Name	Abbreviation	Code
User-Data-Request	UDR	306
User-Data-Answer	UDA	306
Profile-Update-Request	PUR	307

Command-Name	Abbreviation	Code
Profile-Update-Answer	PUA	307
Subscribe-Notifications-Request	SNR	308
Subscribe-Notifications-Answer	SNA	308
Push-Notification-Request	PNR	309
Push-Notification-Answer	PNA	309

Counts of these commands are found in the show diameter-director <message-type> and show diameter-director interface ACLI show commands. They are also available via SNMP and HDR output as described in the Interface Message Type Statistics section of this chapter.

Other Diameter commands not listed above are counted in the Other category found in the aformentioned KPI displays.

Session and Subscriber Statefulness XML File Maintenance

XML File Maintenance

Session ID XML files must be synchronized between redundant nodes to ensure that the standby node contains identical XML files. You can either SFTP the same file to both the active and standby node, or you can use the sync command. The **sync** command is always executed from the active system. It copies the specified file from the active to the standby node placing the copy in the same file location on the standby node. Use the command as follows:

```
ACMESYSTEM#sync dsc <stateful-policy-filename>
```

After copying a new XML file to the Net-Net Diameter Director (and its standby peer), you can reload this newer file from the ACLI using the **refresh dsc** command. For example:

```
ACMEPACKET#refresh dsc <stateful-policy-filename>
```

If the <local-subscriber-table name> is not configured, then the file will not be read. A log message at ERROR level will be added.

- Using the **refresh dsc** command selects the XML by name to refresh. Alternatively, saving and activating the configuration will reload the configuration as well and should be used when configuration parameters have also changed.

Active/Active Redundancy Maintenance

notify ddd

You can set an Active/Active peer in or out of service from the command line.

notify ddd active-active-oos—This command forces the current peer to go Out-Of-Service. This can be used to gracefully put a node OOS and perform a software upgrade etc. If Active/Active redundancy is not enabled , then the command will have no effect.

notify ddd active-active-in-service—This command forces the current peer to be put back into service. This includes, connecting to all the outbound agents, starting the listening sockets and initiating handshake messages with all the peers etc. If Active/Active redundancy is not enabled , then the command will have no effect.

Logging

The Active/Active redundancy log messages are tagged with [AARED] in the log files. A new logfile called `dddaared.log` is used for logging AA redundancy interaction with the peers. When the Diameter Director application is running at debug log level the log file will be turned on by default. Altenatively, `notify ddd debug-aared` can be issued to enable Active/Active Redundancy debug logging.

Net-Net 7000 Hardware Platform Management

Please refer to the Net-Net 7000 Hardware Installation Guide for details about the following:

- SNMP traps specific to the Net-Net 7000
- Information on working with iLo system management support

You set an IP address for your iLo management interface that is separate from the Net-Net DD management port address. Subsequently, the overall system delivers SNMP traps from both of these addresses. Ensure network management staff operate accordingly.

8 ACLI Reference and Debugging

Supporting Configurations

The following configuration elements which are not mentioned in this guide are required for the Net-Net Diameter Director to function. Please refer to the Net-Net 4000 ACLI Configuration Guide for details about configuring all supporting elements.

- network-interface
- phy-interface
- realm-config
- sip-config
- system-config

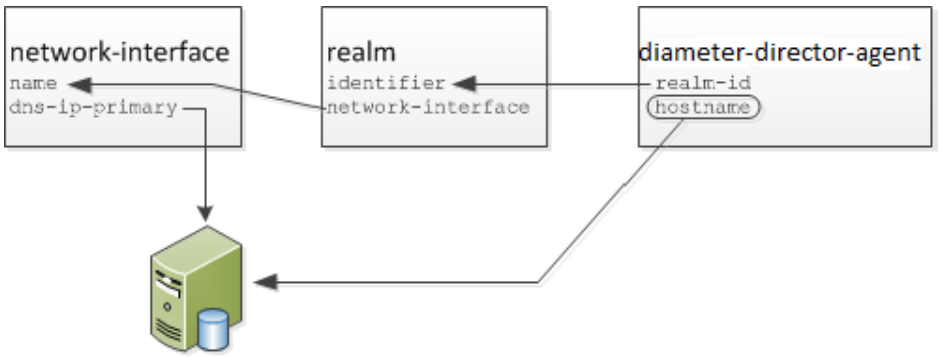
DNS for Diameter Director Agent Hostname Resolution

The Net-Net Diameter Director can use DNS to resolve FQDNs configured for each Diameter Director Agent. Configuring an FQDN for a Diameter Director Agent not only simplifies provisioning, but also eases network scalability and makes the network more resilient to node failure.

To enable the Net-Net Diameter Director to use FQDNs to resolve Diameter Director Agents, you must configure the network interface in which the Diameter Director Agent exists with DNS server information.

ACLI Instructions

Ensure that the **realm**, which is configured in the *diameter-director-agent*'s **realm-id**, exists on the *network-interface* as shown in the following diagram. The *diameter-director-agent*'s **hostname** lookup will be sent to the *network-interface*'s DNS servers.



network-interface

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **system** and press <Enter> to access the system path.

   ```
   ACMEPACKET(configure)# system
   ```

3. Type **network-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(system)# network-interface
```

4. Type **select** and the number of the pre-configured network interface you want to configure.

```
ACMEPACKET(network-interface)# select
<name>:<sub-port-id>:
1: lefty:0    ip=192.168.50.1     gw=192.168.0.1
2: righty:0   ip=192.168.50.5     gw=192.168.0.1

selection: 1
ACMEPACKET(network-interface)#
```

5. **dns-ip-primary**—Enter the primary DNS server's IP address. You can configure an additional two DNS servers by using the **dns-ip-backup1** and **dns-ip-backup2** parameters.
6. **dns-domain**—Set the default domain name used to populate incomplete hostnames that do not include a domain for use with DNS queries. Entries must follow the Name format.
7. **dns-timeout**—Enter the total time in seconds you want to elapse before a query (and its retransmissions) sent to a DNS server would timeout. The default is 11 seconds.
8. Save your work using the ACLI **done** command.

ACLI Configuration Elements

The following sections describe the Net-Net Diameter Director's unique configuration elements.

diameter-manipulation

The diameter-manipulation configuration element defines the message manipulation object.

Parameters

`name`—Configured name of this diameter manipulation. This is the key field.

Default: empty

Values: 24 character string, no special characters with the exception of the underscore and hyphen characters. Do not start name with numeric character.

`description`—Textual description of this diameter manipulation.

Default: empty

Values: 256 character string

`diameter-manip-rules`—See diameter-manip-rules subelement that follows.

Path

diameter-manipulation is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-manipulation.

diameter-manipulation > diameter-manip-rule

The diameter-manip-rule subelement defines an individual step in creating REGEX-type message manipulation object.

`name`—Configured name of this manipulation rule. This is the key field.

Default: empty

Values: Character string, no special characters with the exception of the underscore characters. Do not start name with numeric character.

`avp-code`—AVP in the Diameter message to be of manipulated by this rule. This parameter must be configured.

Default: 0

Values: Valid AVP code

`descr-avp-code`—Description of AVP code to be manipulated.

Default: empty

Values: 256 character string

`avp-type`—The data type of the content of the field the Net-Net PD is parsing to perform a manipulation on. This parameter must be configured with an enumerated value. Refer to the Diameter standards document for the encodings of individual AVPs.

Default: none

Values: none | octet-string | octet-hex | integer32 | unsignedint32 | address | diameteruri | enumerated

`action`—Type of manipulation action to perform on this AVP.

Default: none

Values: none | add | delete | store | diameter-manip | group-manip | find-replace-all | replace

`comparison-type`—Select the comparison type that the match-value uses.

Default: case-sensitive

Values: case-sensitive | case-insensitive | pattern-rule | boolean

`msg-type`—The message type to which this Diameter manipulation rule applies.

Default: any

Values: any—Both Requests and Reply messages

request—Request messages only

reply— Reply messages only

`msg-cmd-code`—The Diameter message code that this rule applies to. This parameter must be configured or the manipulation can not be applied to any message.

Default: 0

Values: Valid Diameter message code

`match-value`—Enter the exact value to be matched. The action you specify is only performed if the header value matches. The entered value must match the comparison type.

Default: empty

`new-value`—The explicit value for a new element or replacement value for an existing element. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators.

Default: empty

avp-header-rule—See avp-header-rule subelement that follows.

Path

diameter-manip-rule is a subelement under the diameter-manipulation element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-manipulation > diameter-manip-rule.

diameter-manipulation > diameter-manip-rule > avp-header-rule

The avp-header-rule subelement defines how to manipulate an AVP's header.

Parameters

`name`—Configured name of this AVP header rule. This is the key field.

Default: empty

Values: Character string, no special characters with the exception of the underscore characters. Do not start name with numeric character.

`header-type`—Type of AVP header to manipulate, as either the AVP flags or the Vendor ID.

Default: avp-flags

Values: avp-flags | avp-vendor-id

`action`—Type of manipulation action to perform on data range in the AVP header.

Default: none

Values: none | add | delete | replace

`match-value`—Value to be matched in the AVP flags or in the vendor ID bits. When manipulating AVP flags, the enumerated values are used to indicate which flag. When manipulating the vendor ID, an integer is entered.

Default: empty

Values: vendor | must | proxy |

`new-value`—value to replace the match value with. You can enter an expression that includes a combination of absolute values, pre-defined parameters, and operators.

Default: empty

Path

`avp-header-rule` is a subelement under the diameter-manipulation > diameter-manip-rule subelement in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-manipulation > diameter-manip-rule > avp-header-rule.

diameter-director-config

The Diameter director config contains general Net-Net Diameter Director parameters.

Parameters

`state`—Running status of CPU load limiting as enabled or disabled.

Default: enabled

Values: enabled | disabled

`load-limit`—The CPU load threshold at which load limiting is enabled.

Default: 85

Values: 0-100

`load-limit-action`—The action to perform on incoming Diameter messages (excluding CER/CEA and DWR/DWA) when load-limiting is in effect.

Default: reject

Values: reject | drop

`load-limit-result-code`—The result code to return to a requesting Diameter agent in a Result-Code AVP when CPU limiting is in effect and the load limit action is set to reject. This value is entered as a valid result code.

Default: 3004

`stateful-policy`—This parameter is configured with the session state XML file's absolute location (/code/example.xml). Configuring the file location enables session aand subscriber stateful feature.

`trans-exp-timer`—Time in seconds which the Net-Net Diameter Director waits for a response from a Diameter Director Agent before timing out the transaction.

Default: 15

Values: 1-999999999

`redundancy-port`—Port on HA interface over which two Net-Net Diameter Directors communicate and update state information.

Default: 1999

`red-max-transactions`—number of maximum HA synchronized Session ID Cache transactions.

Default: 50000

Values: 0-999999999

`red-sync-start-time`—amount of time in milliseconds that the active Net-Net Diameter Director checks to confirm that it is still the active system in the HA node. If the active system is still adequately healthy, this timer will simply reset itself. If for any reason the active has become the standby, it will start to checkpoint with the newly active system when this timer expires.

Default: 5000

Values: 0-999999999

`red-sync-comp-time`—amount of time in milliseconds that determines how frequently after synchronization the standby Net-Net Diameter Director checkpoints with the active Net-Net Diameter Director. The first interval occurs after initial synchronizations of the systems; this is the timeout for subsequent synchronization requests.

Default: 1000

Values: 0-999999999

dynamic-routing—State of dynamic routing feature as enabled or disabled.

Default: enabled

Values: enabled | disabled

active-redundancy—State of Active/Active redundancy feature as enabled or disabled.

Default: disabled

Values: enabled | disabled

active-redundancy-port—port on the media interface over which Active/Active redundancy information is shared.

Default: 9000

Path

diameter-director-config is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-config.

diameter-director-interface

The diameter-director-interface defines the signaling interface in a realm that can accept Diameter messages to be processed or proxied by the Net-Net Diameter Director.

Parameters

state—Running status of this Diameter Director Interface.

Default: enabled

Values: enabled | disabled

realm-id—The configured realm on the Net-Net PD where this Diameter Director Interface exists. This is the key field, thus only one Diameter Director Interface may exist in each realm.

Default: empty

Values: Existing realm name.

description—Textual description of Diameter Director Interface.

Default: empty

Values: 256 character string

diameter-director-ports—See diameter-director-ports subelement that follows.

supported-vendor-ids—List of 32 bit supported-vendor-ids. These values will be inserted into the Supported-Vendor-Id AVP. Enter the list with spaces to indicate addition vendor IDs.

Use the add-supported-vendor-ids and remove-supported-vendor-ids commands to add or remove ID values, individually from the configured list.

Default: empty

Values: 256 character string

`diameter-director-application`—See diameter-director-application subelement that follows.

`origin-realm`—Enter a unique identifier to be inserted in the Origin-Realm-AVP. Leaving this parameter empty inserts the realm-id configured in this diameter-director-interaface.

Default: empty

`origin-host-identifier`—Enter a unique identifier to use in the origin host AVP to override the Net-Net Diameter Director's default method of creating an Origin Host AVP.

Default: empty

`origin-host-format`—This parameter indicates how to form the origin-host AVP value. Leave this parameter empty or set it to identifier or identifier-with-realm to indicate how to create the Origin Host AVP.

Default: none

Values: none—origin-host AVP set to <ip-addr>.<realm-id>

identifier—origin-host AVP set to <origin-host-identifier>

identifier-with-realm—origin-host AVP set to <origin-host-identifier>.<realm-id> or <origin-host-identifier>.<realm-id> when origin-realm is configured.

`routing-policy`—Name of root Diameter Director Policy to first be applied to incoming messages on this Diameter Director Interface.

Default: empty

Values: 256 character string

`in-manip-id`—Name of diameter-manipulation configuration element to apply to incoming messages from this diameter director interface.

Default: empty

Values: Existing diameter-manipulation rule name

`out-manip-id`—Name of diameter-manipulation configuration element to apply to incoming messages to this diameter director interface.

Default: empty

Values: Existing diameter-manipulation rule name

`constraint-name`—Name of diameter-director-constraint configuration element to apply to messages to this diameter director interface.

Default: empty

Values: Existing diameter-director-constraint name

`tos-value`—The value to write into the TOS field in the IP header.

Default: Empty

Values: Valid TOS value

`network-topology`—Specifies the type of topology hiding performed by the system on this inteface.

Default: Empty

Values: masking, hiding and/or obscuring

`congestion-policy-name`—Constraint policy applied to this Diameter Director Interface.

Default: Empty

Values: Valid congestion control policy name

Path

diameter-director-interface is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-interface.

diameter-director-interface > diameter-director-applications

The diameter-director-applications define the Diameter applications which this interface supports and includes in the CER/CEA capabilities exchange. This configuration element also appears under the diameter-director-groups and diameter-director-agent configuration elements. Increasingly granular remote logical entities, when configured take precedence.

Parameters

`application-id`—The Diameter Application ID value used to advertise applications supported for this interface. This is the key field. Values may be entered in hexadecimal or integer format

Default: empty

Values: 32 bit hexadecimal (0x--------) **OR** 32 bit integer format

`application-type`—Indicates the type of application-id.

Default: authentication

Values: authentication | accounting

`vendor-id`—Vendor Id for the configured application-id.

Default: 0

Values: 32 bit integer

Path

This diameter-director-applications configuration element is a subelement in the diameter-director-interface, in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-interface > diameter-director-applications.

diameter-director-interface > diameter-director-ports

The diameter-director-ports define the IP address where all Diameter messages are sent and received. Only the first configured diameter-director-ports configuration element is used.

Parameters

`address`—This Diameter Director Interface's IP address, which must be in the Diameter Director Interface's realm. This is a key field.

Default: none

Values: IP address in dotted decimal notation

`port`—The initial listening port for incoming Diameter messages. This is a key field.

Default: 3868

Values: Must be > 1023

`transport-protocol`—Transport protocol used for this Diameter Director Interface. This is a key field.

Default: TCP

Values: TCP | SCTP

`allow-anonymous`—Defines the type of diameter-director-agents allowed to connect to the Net-Net Diameter Director over this diameter-director-interface.

Default: all

Values: all—All agents may connect to this Net-Net Diameter Director.

agents-only—Only agents defined as diameter-director-agents may connect to this Net-Net Diameter Director.

Path

This diameter-director-ports configuration element is a subelement in the diameter-director-interface, in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-interface > diameter-director-ports.

diameter-director-agent

The diameter-director-agent defines the configuration necessary for representing a remote Diameter peer.

Parameters

`hostname`—Hostname of the remote agent that the Net-Net Diameter Director is connecting to. The hostname can be in FQDN-style or IP Address format. This must be configured so that other configuration elements can reference this diameter-director-agent instance. For the Diameter Director Agent to connect to a FQDN configured here via DNS, leave the ip-address parameter empty. This is the key field.

Default: empty

`ip-address`—IP address of this Diameter Director Agent which the Net-Net PD initiates a socket connection to.

Default: empty

Values: IP address in dotted decimal notation

`port`—Remote port which the Net-Net Diameter Director initiates a socket connection to on this Diameter Director Agent. You can override the default port (3868) by setting this to a value of your choosing. You can leave this field empty, obtaining port number via NAPTR resolution.

Default: 3868

Values: Valid port number

`transport-protocol`—Transport protocol used for this Diameter Director Agent. Set to TCP or SCTP to indicate the transport-protocol or retain the default of empty. When empty you can obtain protocol via NAPTR resolution.

Default: empty

Values: TCP | SCTP

`dns-realm`—Realm to which this agent issues SRV and NAPTR resolution requests if the target DNS server is not in the same realm as the agent. If the target DNS server is in the same realm as the agent, you can leave this field empty. The system only uses this field if the IP address is empty. This function also requires a resolvable diameter-director-agent hostname.

Default: empty

Values: realm-identifier

`connection-mode`—This parameter indicates if the Net-Net Diameter Director initiates a connection to this Diameter Director Agent when the system starts up, or if it waits for the remoteDiameter Director Agent to initiate the connection on its own.

Default: outbound

Values: outbound—The Net-Net Diameter Director initiates the connection to this Diameter Director Agent upon startup.

inbound—The Net-Net Diameter Director waits for this agent to initiate the connection.

outbound-on-demand—Reduces the overhead required to maintain connectivity with specific agents by setting to this value. The Net-Net Diameter Director connects to the agent only when needed.

inbound-dynamic-ip—The Net-Net Diameter Director tries to match an incoming connection with orig-hostname value in the reconnecting agent's CER message to associate the existing sessions, users, and routes. This is used when the agent is expected to reappear with a different IP address compared to before.

`state`—Running status of this diameter-director-agent.

Default: enabled

Values: enabled | disabled

`description`—Textual description of this Diameter Director agent.

Default: empty

Values: 256 character string

`realm-id`—Realm on Net-Net Diameter Director where this Diameter Director Agent exists.

Default: empty

Values: Existing realm name

`watchdog-timer`—Time in seconds that must elapse between the last DWR or other Diameter message that the Net-Net DD will send another DWR to act as a keep alive.

Default: 30

Values: 0-65535

in-manip-id—Name of diameter-manipulation configuration element to apply to incoming messages from this diameter director agent.

Default: empty

Values: Existing diameter-manipulation rule name

out-manip-id—Name of diameter-manipulation configuration element to apply to incoming messages to this diameter director agent.

Default: empty

Values: Existing diameter-manipulation rule name

constraint-name—Name of diameter-director-constraint configuration element to apply to messages to this diameter director agent.

Default: empty

Values: Existing diameter-director-constraint name

ondemand-max-inactivity—Time in seconds that must elapse before the Net-Net PD will disconnect from an on-demand peer.

Default: 0

Values: 0-65535

diameter-director-applications—See diameter-director-application subelement that follows.

congestion-policy-name—Constraint policy applied to this Diameter Director Agent.

Default: Empty

Values: Valid congestion control policy name

Path

diameter-director-agent is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-agent.

diameter-director-agent > diameter-director-applications

This subelement indicates the Diameter applications the Net-Net Diameter Director advertises when performing it's CER/CEA handshake with this agent. See diameter-director-interface > diameter-director-applications (170).

Path

This diameter-director-applications configuration element is a subelement in the diameter-director-agent configuration element, in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-agent > diameter-director-applications.

diameter-director-constraints

The diameter director constraints configuration element contains the necessary information for the Net-Net DD to create a message constraint profile to apply to a Diameter Director Interface or Diameter Director Agent.

Parameters

`name`—Configured name of this Diameter Director constraints object which is referenced from a Diameter Director Interface or Diameter Director Agent.

`state`—Running status of this Diameter Director constraint object.

Default: enabled

`max-burst-rate`—maximum number of messages that can pass through the system in the burst rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`max-inbound-burst-rate`—maximum number of inbound messages received by the referencing element within the burst rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`max-outbound-burst-rate`—maximum number of outbound messages forwarded from the referencing element within the burst rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`burst-rate-window`—number of seconds during which to count messages toward a maximum burst rate.

Default: 0

Values: 0-999999

`max-sustain-rate`—maximum number of messages that can pass through the system in the sustained rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`max-inbound-sustain-rate`—maximum number of inbound messages received by the referencing element within the sustained rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`max-outbound-sustain-rate`—maximum number of outbound messages forwarded from the referencing element within the sustained rate window before setting the element to Constraints Exceeded.

Default: 0

Values: 0-999999

`sustain-rate-window`—number of seconds during which to count messages toward a maximum sustained rate.

Default: 0

Values: 0-999999

`time-to-resume`—number of seconds that the referencing element stays in Constraints Exceeded state and rejects messages before it returns to service.

Default: 0

Values: 0-999999

`result-code`—numeric value to return to the originating element in case of a rejected message due to constraints exceeded state of element.

Default: 3004

Values: 1000-6000

Path

diameter-director-constraints is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-constraints.

diameter-director-constraints > message-rate-constraints

This subelement is used to configure per-message type constraints.

Parameters

`command`—message type you are entering specific constraints upon.

Default: Update-Location

Values: other | update-location | cancel-location | authentication-information | insert-subscriber-data | delete-subscriber-data | purge-ue | reset | notify | credit-control | auth-auth | re-auth | session-termination | abort-session | accounting | user-authorization | server-assignment | location-info | multimedia-auth | registration-termination | push-profile | profile-update | subscribe-notification | push-notification

`max-inbound-burst-rate`—maximum number of inbound messages at the burst rate for this message type.

Default: 0

Values: 0-999999

`max-outbound-burst-rate`—maximum number of outbound messages at the burst rate for this message type.

Default: 0

Values: 0-999999

`max-inbound-sustain-rate`—maximum number of inbound messages at the sustained rate for this message type.

Default: 0

Values: 0-999999

`max-outbound-sustain-rate`—maximum number of outbound messages at the sustained rate for this message type.

Default: 0

Values: 0-999999

diameter-director-constraints > application-constraints

This subelement is used to configure per-application constraints.

Parameters

Values: application-name | max-inbound-burst-rate | max-outbound-burst-rate | max-inbound-sustain-rate | max-outbound-sustain-rate | application-message-constraints

`application-name`—Name assigned to this profile. This name must correspond to an application name within the state machine XML.

`max-inbound-burst-rate`—maximum number of inbound messages at the burst rate for this application type.

Default: 0

Values: 0-999999

`max-outbound-burst-rate`—maximum number of outbound messages at the burst rate for this application type.

Default: 0

Values: 0-999999

`max-inbound-sustain-rate`—maximum number of inbound messages at the sustained rate for this application type.

Default: 0

Values: 0-999999

`max-outbound-sustain-rate`—maximum number of outbound messages at the sustained rate for this application type.

Default: 0

Values: 0-999999

`application-message-constraints`—Enter this sub-element for applicable configuration.

diameter-director-constraints > application-constraints > application-message-constraints

This subelement is used to configure per-message message constraints for this application. Specific message types are constrained for the application

Parameters

Values: message-type | max-inbound-burst-rate | max-outbound-burst-rate | max-inbound-sustain-rate | max-outbound-sustain-rate

message-type—Specific values are listed in the table below.

Message-type	Meaning	XML Definition	Valid Application States
initial	Message that creates a session	alloc	session subscriber
terminate	Message that terminates a session	dealloc	session subscriber
in-session	Messages received for a cached session after initial and before terminate.	NA	session subscriber
new-subscribers	Messages that create a new subscriber cache context	subscriber	subscriber, subscriber-only
existing-subscribers	Messages that belong to a cached subscriber cache context	subscriber	subscriber subscriber-only

max-inbound-burst-rate—maximum number of inbound messages at the burst rate for this message type.

Default: 0

Values: 0-999999

max-outbound-burst-rate—maximum number of outbound messages at the burst rate for this message type.

Default: 0

Values: 0-999999

max-inbound-sustain-rate—maximum number of inbound messages at the sustained rate for this message type.

Default: 0

Values: 0-999999

max-outbound-sustain-rate—maximum number of outbound messages at the sustained rate for this message type.

Default: 0

Values: 0-999999

diameter-director-group

The Diameter Director Group configuration element contains the necessary information for the Net-Net Diameter Director to create a Diameter Director Group containing a set of PDAs.

Parameters

group-name—Configured name of this Diameter Director Group. This is the key field.

Default: empty

`state`—Running status of this Diameter Director Group.

Default: enabled

Values: enabled | disabled

`description`—Textual description of this Diameter Director Group.

Default: empty

Values: 256 character string

`strategy`—Strategy used to select the next member policy agent to begin and maintain a session with.

Default: hunt

Values: hunt

roundrobin

`destinations`—List of diameter-director-agents that comprise this Diameter Director Group. Configure each Diameter Director Agent using its configured hostname according to the following:

- for single-entry: dest1
- for multi-entry: (dest1 dest2 dest3 dest4)
- for adding a single entry to an existing list: +dest5
- for deleting a single entry from an existing list: -dest5

`in-manip-id`—Name of diameter-manipulation configuration element to apply to incoming messages from this Diameter Director Group.

Default: empty

Values: Existing diameter-manipulation rule name

`out-manip-id`—Name of diameter-manipulation configuration element to apply to outgoing messages to this Diameter Director Group.

Default: empty

Values: Existing diameter-manipulation rule name

`recursive-routing`—See recursive-routing subelement that follows.

`diameter-director-applications`—See diameter-director-applications subelement that follows.

Path

diameter-director-group is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-group.

diameter-director-group > diameter-director-applications

This subelement indicates the Diameter applications the Net-Net Diameter Director advertises when performing it's CER/CEA handshake with any agent in this Diameter Director Group. Configuring this value takes precedence over the diameter-director-applications configured on a Diameter Director Interface. See diameter-director-interface > diameter-director-applications (170).

Path

This diameter-director-applications configuration element is a subelement in the diameter-director-group configuration element, in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-group > diameter-director-applications.

diameter-director-group > recursive-routing

This subelement indicates how the Net-Net Diameter Director shall perform recursive routing through the configured Diameter Director Agents in a Diameter Director Group.

Parameters

`do-recursion`—Status of if this Diameter Director Group can recurse through configured agents upon receipt of a failure message. This is required.

Default: disabled

Values: enabled | disabled

`result-codes`—The Result-Code AVP values, as returned by an agent, which prompt the Net-Net Diameter Director to recurse through this Diameter Director Group. Values can be entered individually, as comma separated integers, or hyphenated to indicate a range of values.

Default: empty

Values: Valid Diameter result codes

`exp-result-codes`—The Experimental-Result-Code AVP values, as returned by an agent, which prompt the Net-Net Diameter Director to recurse through this Diameter Director Group. Values can be entered individually, as comma separated integers, or hyphenated to indicate a range of values.

Default: empty

Values: Valid Diameter experimental result codes

`transaction-timeout`—Time in milliseconds which the Net-Net Diameter Director waits for a response from the current Diameter Director Agent in this Diameter Director Group before failing and returning a 3002 to the requesting agent or recursing to the next agent.

Default: 32000

Values: 0-32000

`recursion-timeout`—Enter the total time in milliseconds which the Net-Net Diameter Director can take to recurse through all Diameter Director Agents in this Diameter Director Group before failing and returning a 3002 to the requesting agent.

Default: 100000

Path

This `recursive-routing` configuration element is a subelement in the diameter-director-group configuration element, in the session-router path. The full path from the topmost ACLI prompt is: `configure terminal > session-router > diameter-director-group > recursive-routing`.

diameter-director-policy

The diameter-director-policy configuration element defines the Net-Net Diameter Director routing policies.

Parameters

`name`—Configured name of this Diameter Director Policy. This is the key field.

Default: empty

`state`—Running status of this diameter-director-policy.

Default: enabled

Values: enabled | disabled

`policy-attribute`—See policy-attributes subelement that follows.

Path

diameter-director-policy is an element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-policy.

diameter-director-policy > policy-attribute

The policy-attribute configuration element defines a policy match based on an incoming Diameter message. Further, the policy attribute defines the action to take after making a positive match on the Diameter message.

Parameters

`attribute-name`—The name of this policy attribute. This is the key field and must be configured.

`match-value`—The explicit value within the match-field to find and make a positive match on.

Default: empty

`action`—Action to take on the message after making a positive match.

Default: none

Values: forward | reject | none

`match-field`—Field within the Diameter message to search for the match-value in. This is required.

Default: empty

Values: Set this to "`avp=[avp-num]`" to match on a specific AVP field

command-code—Used to make match on the Diameter command code

application-id—Used to make a match on an application id in the Diameter message

incoming-realm—Used to make a match on the realm where this message was received

incoming-interface—Used to make a match on the network interface where this message was received

`avp-type`—Set this to the data type of the content of the field the Net-Net Diameter Director is parsing to make a match. Refer to the Diameter standards document for the encodings of individual AVPs.

Default: none

Values: none | octet-string | octet-hex | integer32 | unsignedint32 | address | diameteruri | enumeration | grouped

`sub-avps`—See sub-avps subelement that follows.

`comparison-type`—The type of computational comparison the Net-Net Diameter Director uses to test for a Diameter Director Policy match.

Default: none

Values: regex | integer | case-sensitive | case-insensitive | refer-case-sensitive | refer-case-insensitive

`priority`—Numbered priority to execute this policy attribute when testing for the parent Diameter Director Policy. The lowest priority policy attribute is executed first. When more than one policy attribute has the same numbered priority, the one configured first is executed first.

Default: 0

Values: 0-65535

`reject-result-code`—Result code value to send to requesting Diameter agent when a Policy Attribute action of reject is selected.

Default: 3002

`reject-exp-result-code`—Exeperimental Result code value to send to requesting Diameter agent when a Policy Attribute action of reject is selected. This parameter must be configured along with the reject-exp-vendor-id parameter.

Default: 3002

`reject-exp-vendor-id`—Vendor ID to include with Experimental Result code as sent to requesting Diameter agent when a Policy Attribute action of reject is selected. This parameter must be configured along with the reject-exp-result-code parameter.

Default: 0

`next-hop`—The Diameter Director Agent or Diameter Director Group's configured name to forward the message to when a positive match is made. To indicate a Diameter Director Group, use the prefix **ddg:**

`next-policy`—When a positive match is made on this policy attribute, the Net-Net Diameter Director continues and applies the Diameter Director Policy name configured here for additional matching. When both the `next-hop` and `next-policy` parameters are configured, the Net-Net Diameter Director uses the `next-policy` parameter to perform additional matching. To indicate the user of a local routing table, use the prefix **lrt:**

Path

This policy-attributes configuration element is a subelement in the diameter-director-policy configuration element, in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > diameter-director-policy > policy-attributes.

diameter-director-policy > policy-attributes > sub-avps

This configuration element is used when making a match on grouped AVPs. Note that the avp-type must be set to grouped for the Net-Net Diameter Director to apply the parameters in this section to a message content search.

Parameters

`avp-code`—The grouped AVP code number to make a match within.

Default: empty value

Values: Valid AVP Code

`avp-type`—The data type of the content of the AVP data the Net-Net PD is parsing to make a match in this grouped AVP. Refer to the Diameter standards document for the encodings of individual AVPs.

Default: none

Values: octet-string | integer32 | unsignedint32 | address | time | utfstring | diameterident | diameteruri | enumerated | grouped

`comparison-type`—The type of computational comparison the Net-Net PD uses to test for a Diameter Director Policy match.

Default: none

Values: regex | integer | case-sensitive | case-insensitive | refer-case-sensitive | refer-case-insensitive

`match-value`—The explicit value within the AVP defined in the `avp-code` parameter of this configuration element to find and make a positive match on.

Default: empty

Values: Any value that is represented by the `comparison-type` values.

Path

This `sub-avps` configuration element is a subelement in the diameter-director-policy configuration element, in the session-router path. The full path from the topmost ACLI prompt is: `configure terminal > session-router > diameter-director-policy > policy-attributes > sub-avps`.

congestion-control-policy

The **congestion-control-policy** element defines how ... Multiple congestion policies can be referenced and applied; this configuration element groups policies together and allows them to be referenced by a single identifier.

Syntax

```
congestion-control-policy <name | result-code | experimental-
result-codes | allow-threshold | transaction-timeout-threshold |
congestion-window |congestion-action | time-to-resume |
constraint-name | reject-result-code | reject-exp-vendor-id |
reject-exp-result-code | select | no | show | done | exit>
```

Parameters

`name`—Enter the identifier or name for this congestion control policy. This parameter is required.

`result-code`—Enter the identifier (numeric) of the result code the system receives and with which the system measures the element's level of congestion.

`experiment-result-codes`—Enter the identifier (numeric) of the result code the system receives.

`allow-threshold`—Enter the maximum number of result-code messages the system allows before denoting the element as congested.

`transaction-timeout-threshold`—Enter the time, in milleseconds, that the Net-Net Diameter Director waits for a response from an applicable element before it labels that elements as congested.

Default: 0

Values: 0-4294967296

`congestion-window`—Enter the time, in milleseconds that the Net-Net Diameter Director uses between measurements, from which it evaluates an applicable element's congested state.

Default: 0

Values: 0-999999

`congestion-action`—Enter the behavior the Net-Net Diameter Director exhibits when it finds an applicable element to be congested.

- drop - The Net-Net Diameter Director does not respond to requests directed towards the congested element.
- reject - The Net-Net Diameter Director sends the configured response to devices that send requests to the congested element. Messages sent include the configured result-code and/or the experimental-result-code.
- constraints - The Net-Net Diameter Director uses the configured congestion-contraints profile to shape the traffic stream.

`time-to-resume`—Enter the duration, in seconds, that must elapse before the system evaluates the status of a congested far end device. The device status cannot be changed back to "un-congested" until this timer expires and all thresholds are no longer exceeded.

Default: 0

Values: 0-999999

`constraint-name`—Enter the name of the diameter-director-constraint that this policy applies when the congestion action is set to `constraints`.

`reject-result-code`—Enter the value to include in the Result Code AVP when the Net-Net Diameter Director chooses this congestion control policy with a **reject** action.

`reject-exp-vendor-id`—Enter the vendor ID to accompany the Experimental Result Code when the Net-Net Diameter Director chooses this congestion control policy with a **reject** action.

`reject-exp-result-code`—Enter the value to include in the Experimental Result Code AVP when the Net-Net Diameter Director chooses this congestion control policy with a **reject** action. This parameter must be configured along with the *reject exp vendor id* parameter,

Path

congestion-control-policy is an element under the session-router path. The full path from the topmost ACLI prompt is: **configure terminal > session-router > congestion-control-policy.**

Release	First appearance: DD2.2
RTC Status	Supported
Mode	This is a multiple instance configuration element.

Logging

Process Logging

The Net-Net Diameter Director logs important events from the Diameter Director application to the `log.ddd` log file. Entries in this file show events with different severity levels, including the category of event.

Message Logging

The Net-Net Diameter Director can log a Diameter message trace file to the ddd.log log file. This file reflects the messages that are proxied through the system. Diameter datagrams are decoded from binary to a human-readable ASCII format. You must set the process-log-level to debug in the system-config.

```
ACMEPACKET(system-config)# process-log-level debug
```

Or use the notify command to start the ddd process at debug:

```
ACMEPACKET# notify ddd debug
```

Net-Net Diameter Director Show Commands

show diameter-director errors

The **show diameter-director errors** command displays global error counts for recent and lifetime time frames for the Net-Net Diameter Director.

```
ACMESYSTEM# show diameter-director errors
13:16:54-175
Diameter Director Errors/Events
                          -------- Lifetime --------
                          Recent      Total  PerMax
No Route Found             120        584      70
Malformed Messages           0          0       0
Rejected Messages            0          0       0
Dropped Messages             0          0       0
Inbound Constraints         27         88      27
Inbound Rejected             0         36      10
Inbound Dropped             27         52      27
Outbound Constraints         0          0       0
Outbound Rejected            0          0       0
Outbound Dropped             0          0       0
```

These global error KPIs for inbound and outbound constraints will reflect the total number of messages rejected due to constraints being exceeded, across all the interfaces and/or agent configured to check for constraints.

show diameter-director <message-type>

The **show diameter-director** <message-type> command displays information for the specified message type. These counts are given for the Net-Net PD acting in client and

server role, for recent, total, and period max time frames. The following message-types are supported and can be used as arguments with the show diameter-director command:

- Other
- Update-Location
- Cancel-Location
- Authentication-Information
- Insert-Subscriber-Data
- Delete-Subscriber-Data
- Purge-UE
- Reset
- Notify
- Credit-Control
- Auth-Auth
- Re-Auth
- Session-Termination
- Abort-Session
- Accounting

```
ACMEPACKET> show diameter-director update-location
Update-Location (08:57:06-145)
                  --------- Server --------  --------- Client --------
Message/Event       Recent     Total  PerMax   Recent     Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
Update-Loc Requests      3          3       3        3          3       3
Retransmissions          0          0       0        0          0       0
2002 Limited Success     3          3       3        3          3       3
Response Retrans         0          0       0        0          0       0
Transaction Timeouts     -          -       -        0          0       0
Locally Throttled        -          -       -        0          0       0

Avg Latency=0.000 for 0
Max Latency=0.000
```

show diameter-director status

The **show diameter-director status** displays counts for all client and server transactions that are active and have been completed. Period and Lifetime totals are included. For example:

```
ACMESYSTEM# show diameter-director status
13:17:56-136
Diameter Director Status
               -------- Period -------- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Client Trans         0       0       0          0       0       0
Server Trans         0       0       0          0       0       0
Sockets              1       1       0          1       1       1
Connections          0       0       0          0       0       0
TPS=0.0 Hi=0.0 Lo=0.0
```

Transactions per Second

The **show diameter-director status** command displays the number of transactions per second (TPS), the Net-Net Diameter Director is running at. Transactions Per Second are computed on rolling 10 second windows. A Net-Net Diameter Director transaction starts when the Net-Net Diameter Director receives a request (excluding CER, DPR and DWR) and then ends when the Net-Net Diameter Director sends a response back to the requesting agent.

Thus the Net-Net Diameter Director displays the cumulative number of transaction that occurred in the previous 10-second window, divided by 10 to get the number in seconds.

show diameter-director interface

The show diameter-director interface command displays the diameter-director-interface level statistics. The command is entered as

```
show diameter-director interface [<realm-name>] [<message-type> |
errors | dns | all]
```

Without any arguments, this command displays a summary of the interfaces:

```
ACMEPACKET# show diameter-director interface
19:29:52-5711
Diameter Director Interface summary
DD Interface    Status     ClienTrans     ServerTrans    Connections
net172              I 1              0              0
net192              C 0              0              0
```

With a Diameter Director Interface realm ID, this command displays the diameter-director-interface level statistics for a specific Diameter Director Interface:

```
ACMESYSTEM# show diameter-director interface firstrealm
13:19:27-128
Diameter Director Status
               -------- Period -------- -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Client Trans         0       0       0          0       0       0
Server Trans         0       0       0          0       0       0
Sockets              1       1       0          1       1       1
Connections          0       0       0          0       0       0
TPS=0.0 Hi=0.0 Lo=0.0
```

With a realm ID and the `errors` argument, this command displays the diameter-director-interface errors within the specified realm. The output is identical to the show diameter-director errors (184) output.

With the Diameter Director interface realm ID and the constraints argument, , this command provides constraints statistics:

```
ACMESYSTEM# show diameter-director interface net172 constraints
17:40:07-46
DD Interface net172
                              -- Period -- -------- Lifetime --------
                    Active    High   Total      Total  PerMax    High
Inbound Transactions 0       0       0          0       0       0
  Rate Exceeded         -       -       0          0       0       -
```

```
  Burst Rate            0       0       0          0       0       0
Outbound Transactions 0       0       0          0       0       0
  Rate Exceeded         -       -       0          0       0       -
  Burst Rate            0       0       0          0       0       0
```

With a Diameter Director Interface realm ID, and message-type, this command displays per message statistics for the diameter-director-interface statistics for a specific PDI. The output will appear to be same as "show diameter-director <message-type>". Here is the sample output:

```
ACMESYSTEM# show diameter-director interface net172 update-location
Update-Location (13:58:17-397)
                  --------- Server --------  --------- Client --------
Message/Event       Recent     Total  PerMax   Recent     Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
Update-Loc Requests       0         0      0       1         1      1
Retransmissions           0         0      0       0         0      0
2002 Limited Success      0         0      0       1         1      1
Response Retrans          0         0      0       0         0      0
Transaction Timeouts      -         -      -       0         0      0
Locally Throttled         -         -      -       0         0      0

Avg Latency=0.000 for 0
Max Latency=0.000
BurstRate Incoming=0.0 Outgoing=0.0
```

Burst rate information will only be displayed if the burst rate is configured and/or there is data to display.

With a Diameter Director Interface realm ID and the **all** argument, this command displays all statistics for a specified Diameter Director Interface:

```
ACMEPACKET# show diameter-director interface net172 all
16:47:34-3677
Diameter Director Status
               -------- Period --------  -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Client Trans         2       2       2          2       2       2
Server Trans         0       0       0          0       0       0
Sockets              2       2       2          2       2       2
Connections           1       1       1          1       1       1
16:47:34-3677
Diameter Director Errors/Events             ---- Lifetime ----
                        Recent      Total  PerMax
No Route Found               0          0       0
Malformed Messages           0          0       0
---< NO DATA AVAILABLE >----(Other Methods)
Update-Location (16:47:34-3677)
                  --------- Server --------  --------- Client --------
Message/Event       Recent     Total  PerMax   Recent     Total  PerMax
                    ------  ---------  ------   ------  ---------  ------
Update-Loc Requests       0         0      0       1         1      1
Retransmissions           0         0      0       0         0      0
2002 Limited Success      0         0      0       1         1      1
```

```
Response Retrans          0         0      0       0         0      0
Transaction Timeouts      -         -      -       0         0      0
Locally Throttled         -         -      -       0         0      0

Avg Latency=0.000 for 0
Max Latency=0.000
---< NO DATA AVAILABLE >----(Other Methods)
---< NO DATA AVAILABLE >----(Update-Location)
---< NO DATA AVAILABLE >----(Cancel-Location)
---< NO DATA AVAILABLE >----(Authentication-Information)
---< NO DATA AVAILABLE >----(Insert-Subscriber-Data)
---< NO DATA AVAILABLE >----(Delete-Subscriber-Data)
---< NO DATA AVAILABLE >----(Purge-UE)
---< NO DATA AVAILABLE >----(Reset)
---< NO DATA AVAILABLE >----(Notify)
---< NO DATA AVAILABLE >----(Credit-Control)
---< NO DATA AVAILABLE >----(Auth-Auth)
---< NO DATA AVAILABLE >----(Re-Auth)
---< NO DATA AVAILABLE >----(Session-Termination)
---< NO DATA AVAILABLE >----(Abort-Session)
---< NO DATA AVAILABLE >----(Accounting)
```

With a Diameter Director Interface realm ID and the **dns** argument, this command displays DNS statistics for a specified Diameter Director Interface. The output is identical to the `show diameter-director dns` output.

show diameter-director agent

The **show diameter-director agent** command displays the diameter-director-agent level statistics. The command is entered as

```
show diameter-director agent [<hostname>] [errors | dns]
```

For each Diameter Director Agent, the current status as in (I) or out (O) of service, number of Client transactions, number of Server Transactions, and total number of Connections is displayed. For example:

```
DD Agent  Status Transport      Address       ClienTrans ServerTrans  Conn
client1    O       TCP     168.192.24.101:1325      0         0         0
client2    O       TCP     168.192.24.102:1694      0         0         0
server1    I       SCTP    192.168.24.101:3868      0         0         0
server2    I       SCTP    192.168.24.102:3868      0         0         0
server3    O       SCTP    192.168.24.103:3868      0         0         0
```

The show diameter-director agent command displays the active transport protocol and IP address for each configured agent. If the transport protocol or address change after a DNS query, the system includes the updated information in this commands output.

You can enter a hostname as an argument for statistics on the provided Diameter Director Agent. For example:

```
ACMESYSTEM# show diameter-director agent 192.168.24.100
11:28:39-332764
192.168.24.100  (Out Of Service)
Diameter Director Status
                      -------- Period --------  -------- Lifetime --------
                     Active    High   Total      Total  PerMax    High
Client Trans              0      10    1001       1001    1001      10
Server Trans              0       0       0          0       0       0
Sockets                   1       1       1          1       1       1
Connections               1       1       1          1       1       1
TPS=0.0 Hi=1.1 Lo=0.0
```

With a hostname indicating a specific Diameter Director Agent and the `errors` argument, this command displays the errors for the specified agent. The output is identical to the show diameter-director errors (184) output.

With a hostname indicating a specific Diameter Director Agent and the `dns` argument, this command displays the errors for the specified agent. The output is identical to the `show diameter-director dns` output.

show diameter-director sessions

The show diameter-director sessions command with no arguments presents counts for current sessions and their states, given in period and lifetime windows.

For example:

```
ACMESYSTEM# show diameter-director sessions
15:34:13-154
Diameter Session Status
              -------- Period --------  -------- Lifetime --------
                 Active    High   Total      Total  PerMax    High
Sessions              0       0       0          0       0       0
  Initial             0       0       0          0       0       0
  Established         0       0       0          0       0       0
  Terminated          0       0       0          0       0       0
Sessions Miss         -       -       0          0       0       -
```

You may enter the by-session or by-application argument in the proper position to search by that criteria to obtain the filtered output. The command is entered as:

```
show diameter-director sessions [by-application <application-id> |
bysession <session-id>]
```

For example:

```
ACMESYSTEM# show diameter-director sessions by-application <enter>
Diameter Director Sessions Summary
Application    Key              Cached Entries
Gx             16777625:0       3
S6             16777251:10415   0
```

Entering the by-application argument and the application to search for displays all sessions for a provided application:

```
ACMESYSTEM# show diameter-director sessions by-application Gx active
Application  Key           Session-Id                                  Next-Hop
Gx           16777625:0    test2.acmepacket.com;1316538636;1833637    Agent1
Gx           16777625:0    test3.acmepacket.com;1416538636;1933637    Agent2
Gx           16777625:0    test5.acmepacket.com;3316538636;9833637    Agent1
```

Entering the by-session-id argument with a supplied session ID displays information for that session:

```
ACMESYSTEM# show diameter-director sessions by-session-id test2.acmepacket.com;1316538636;1833637
Application    Key          Session-Id                                  Next-Hop
Gx             16777625:0   test2.acmepacket.com;1316538636;1833637    Agent1
```

show diameter-director subscribers

The show diameter director subscribers command displays information about currently cached subscribers.

The show diameter-director subscribers command with no arguments displays the number of current, cached subscribers per application category. For example:

```
ACMESYSTEM# # show diameter-director subscribers <enter>
15:46:52-195
Diameter Subscriber Status
            ----- Period --------  -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Subscribers          9       9       9          9       9       9
  Subscribe          9       9       9          9       9       9
  UnSubscribe        0       0       0          0       0       0
  Policy Hit         0       0       0          0       0       0
  Policy Miss        0       0       0          0       0       0
Subscribers Miss     -       -       0          0       0       -
```

You may enter the by-category, by-key, or by-application arguments in the proper position to search by that criteria to obtain the filtered output. The command is entered as:

```
show diameter-directory subscribers [all | by-category <category> |
categories | by-key <key> | by-application <application>]
```

By entering the all argument, the ACLI will display all cached subscribers, their Category, and the next-hop destination. For example:

```
ACMESYSTEM# show diameter-director subscribers all
Diameter Director Subscribers
Subscriber             Session(s)                 Next-Hop
-----------------------------------------------------------------
457        Gx:P-GW1.foo.com,132220678;540987     192.168.42.101
458        Gx:P-GW1.foo.com,132220678;540988     192.168.42.101
459        Gx:P-GW1.foo.com,132220678;540989     192.168.42.101
...
```

By entering the by-category argument and a category, the ACLI will display all subscribers in the subscriber cache sorted by category. In addition, the next-hop

destination is printed on the screen. If you enter the category argument at the end of the command, only that category's subscribers are listed. For example:

```
ACMESYSTEM# show diameter-director subscribers by-category
Diameter Director Subscribers Categories
Category: hss
Application      AppID:VendorID       Subscriber Key
-------------------------------------------------------------------------------
s6a              16777251:10415       {443:0}[450:0/int32:1,444:0/string:<key>]

Category: ocs
Application      AppID:VendorID       Subscriber Key
-------------------------------------------------------------------------------
Gx               16777224:0           {443:0}[450:0/int32:1,444:0/string:<key>]
Gy                      4:0           {443:0}[450:0/int32:1,444:0/string:<key>]

Category: pcrf
Application      AppID:VendorID       Subscriber Key
-------------------------------------------------------------------------------
Cx               16777216:10415       {443:0}[450:0/int32:1,444:0/string:<key>]
```

```
ACMESYSTEM# show diameter-director subscribers by-category ocs
15:47:12-5315
Diameter Subscriber Status
            ----- Period --------  -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Subscribers         29      29      29         29      29      29
  Subscribe         29      29      29         29      29      29
  UnSubscribe        0       0       0          0       0       0
  Policy Hit         0       0       0          0       0       0
  Policy Miss       29      29      29         29      29      29
Subscribers Miss     -       -       0          0       0       -
```

By entering the by-application <application> argument, the ACLI will display the statistics for that application. For example:

```
ACMESYSTEM# show diameter-director subscribers by-application Gx
15:47:43-145
Diameter Subscriber Status
            ----- Period --------  -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Subscribers          0       0       0          0       0       0
  Subscribe         37      37      37         37      24      37
  UnSubscribe        0       0       0          0       0       0
  Policy Hit         0       0       0          0       0       0
  Policy Miss        0       0       0          0       0       0
Subscribers Miss     -       -       0          0       0       -
```

Entering the active argument prints out cache entries for the application you are querying. For example:

```
ACMESYSTEM# show diameter-director subscribers by-application Gx
active
Diameter Director Subscribers by application: Gx
```

```
Subscriber            Session(s)                  Next-Hop
-------------------------------------------------------------
457        Gx:P-GW1.foo.com,132220678;540987     192.168.42.101
458        Gx:P-GW1.foo.com,132220678;540988     192.168.42.101
459        Gx:P-GW1.foo.com,132220678;540989     192.168.42.101
460        Gx:P-GW1.foo.com,132220678;540990     192.168.42.101
```

By entering the by-key argument with a subscriber key, the ACLI displays all subscribers, their category, and associated next-hop destination. For example:

```
ACMESYSTEM# show diameter-director subscribers by-key 475
Diameter Director Subscribers, subscriber: 208017562123475
Subscriber       Session(s)                     Next-Hop
-----------------------------------------------------------------
475        Gx:P-GW1.foo.com,132220678;541005     192.168.42.101
```

show diameter-director applications

The show diameter-director applications command displays the applications configured in the applications XML file, displaying those configuratios. The output includes session, subscriber and subscriber-only applications, with data presented being specific to the applications type.

```
AcmePacket# show diameter-director applications
diameter application
        name                    Gx
        id                      16777238
        vendor id               0
        category                pcrf
        application state       subscriber
        actions
                action                  sessionCacheMiss
                session cache state     reject
                code                    3002
        max inactivity time     900
        initiate
                type          alloc
                command code  272
                avp code      416
                avp type      1
                avp value     1
        terminate
                type          dealloc
                command code  272
                avp code      416
                avp type      1
                avp value     3
```

The output for a Subscriber-only application, s6a, is shown below.

```
diameter-application
        name                    s6a
        id                      16777251
        vendor id               0
        category                s6a
```

```
application state       subscriber-only
actions
        action                  subscriberMiss
        session cache state     reject
        code                    5002
max inactivity time     3000
Subscriber
        avp-filter
                avp code                443
                avp flags               0
                avp type                Grouped
                is key avp              disabled
                avp-filter
                        avp code                450
                        avp flags               0
                        avp type                Int
                       is key avp              disabled
                avp-filter
                        avp code                444
                        avp flags               0
                        avp type                String
                       is key avp              enabled
```

The output for a Subscriber application, Rx, is shown below.

```
diameter-application
        name                    Rx
        id                      16777236
        vendor id               0
        category                pcrf
        application state       subscriber
        actions
                action                  sessionCacheMiss
                session cache state     reject
                code                    3002
        max inactivity time     900
        initiate
                type            alloc
                command code    265
                avp code        0
                avp type        0
                avp value
        terminate
                type            dealloc
                command code    275
                avp code        0
                avp type        0
                avp value

        Subscriber
                avp-filter
                        avp code                443
                        avp flags               0
                        avp type                Grouped
```

```
is key avp              disabled
avp-filter
        avp code                450
        avp flags               0
        avp type                Int
       is key avp              disabled
avp-filter
        avp code                444
        avp flags               0
        avp type                String
       is key avp              enabled
```

show diameter-director all

The show diameter-director all command displays all Net-Net Diameter Director statistics.

Resetting Statistics

The **reset ddd** command will reset all Net-Net Diameter Director counters to 0.

```
ACMEPACKET # reset ddd
```

Net-Net Diameter Director statistics will also be reset when executing the **reset all** command.

clear-sess diameter-director sessions

You may selectively delete established sessions using the clear-sess diameter-director sessions command. This command is entered as:

```
clear-sess diameter-director sessions [all | by-agent | by-application
| by-session-id]
```

where you supply the appropriate information considering the argument you choose.

The all argument will clear all active, sessions. For example:

```
ACMESYSTEM# clear-sess diameter-director sessions all
Application      Key              Session-Id        Next-Hop
Gx               16777224:0       .,638472530;1     172.16.9.1
s6a              16777251:10415   .,638472530;2     172.16.9.3
Gx               16777224:0       .,638472530;3     172.16.9.1

Cleared 3 of total 3 sessions
```

The by-agent argument requires a Diameter Director Agent configuration. It will clear all sessions going to that Diameter Director Agent. For example:

```
ACMESYSTEM# # clear-sess diameter-director sessions by-agent
172.16.9.1
Application     Key             Session-Id           Next-Hop
Gx              16777224:0      .,638472530;1        172.16.9.1
Gx              16777224:0      .,638472530;3        172.16.9.1

Cleared 2 of total 3 sessions
```

The by-application argument requires an application name. It will clear all sessions received and cached with that application name. For example:

```
ACMESYSTEM# clear-sess diameter-director sessions by-application s6a
Application    Key               Session-Id          Next-Hop
```

```
s6a             16777251:10415     .,638472530;2        172.16.9.3

Cleared 1 of total 3 sessions
```

The by-session-id argument requires a session ID. It will clear a specific application with the provided session ID. For example:

```
ACMESYSTEM# clear-sess diameter-director sessions by-session-id
.,638472530;2
Application        Key                Session-Id          Next-Hop
s6a                16777251:10415     .,638472530;2       172.16.9.3
```

show diameter-director dynamic-routes

The **show diameter-director dynamic-routes** command displays the entire contents of the dynamic routing table.

```
ACMEPACKET# show diameter-director dynamic-routes

Origin    Realm           Source IP          Client/Peer IP          Agent
Host
Seagull   Acme.Packet.com 168.192.24.70.3868 168.192.24.100:3868  Agent172
Seagull2  Acme.Packet.com 168.192.24.70:3868 168.192.24.101:3868         -
Seagull3  Acme.Packet.com 168.192.24.70:3868 168.192.24.102:3868         -
Seagull4  Acme.Packet.com 168.192.24.70.3868 168.192.24.103.3868  Agent172

...
...
...

ACMEPACKET#
```

Origin Host

lists the DIAMETER identity of the endpoint that originated the received DIAMETER request. The endpoint identity is contained in AVP 264, which is required in every DIAMETER message.

Realm

lists the received packet's ingress realm.

Source IP

lists the local DIAMETER Director IP address of the interface which received the request.

Client/Peer IF

lists the remote IP address which originated the request.

Agent

lists the associated in-bound agent, if any. It is possible for an inbound request to be received from a location not tied to a particular agent. Consequently, not all of the displayed routes may have corresponding agent values.

Use the **show diameter-director dynamic-routes by-route** command to filter the dynamic routing table display by the originating endpoint.

```
ACMEPACKET# show diameter-director dynamic-routes by-route Seagull
```

```
Origin    Realm           Source IP         Client/Peer IP         Agent
Host
Seagull   Acme.Packet.com 168.192.24.70.3868 168.192.24.100:3868  Agent172
```

Use the **show diameter-director dynamic-routes by-agent <agent>** command to filter the dynamic routing table display by the supplied agent.

```
ACMEPACKET# show diameter-director dynamic-routes by-agent Agent172

Origin    Realm           Source IP         Client/Peer IP         Agent
Host
Seagull   Acme.Packet.com 168.192.24.70.3868 168.192.24.100:3868  Agent172
Seagull4  Acme.Packet.com 168.192.24.70.3868 168.192.24.103.3868  Agent172
```

show active-redundancy diameter-director

The **show active-redundancy diameter-director** command displays active-active redundancy stats for Net-Net Diameter Director. All configured peers are displayed including their state, health score, address and last synchronized time. For example:

```
ACMEPACKET# show active-redundancy diameter-director

Diameter-Director Synchronized             true

Active Redundancy Peers:
Peer: Ariel
        State                         Active
        Local Address          169.254.1.1:9090

Peer: Europa
        State                         Active
        Local Address          169.254.1.2:9090
```

By adding the status parameter, the ACLI displays client and server position Active/Active message counts. In addition, the types of messages exchanged between the Active/Active peers are also counted.

The **show active-redundancy diameter-director status** command displays redundancy message statistics as exchanged between Active/Active peers.

Sample status statistics for DSC active-active redundancy will look like:

```
ACMEPACKET# show active-redundancy diameter-director
18:23:33-175
Active-Redundancy Transaction Statistics
                  ---------- Period --------- -------- Lifetime --------
                  Active    High   Total      Total  PerMax    High
Server Trans      0         0      0          0      2         2
Client Trans      0         0      0          0      2         2

Active-Redundancy Messages Statistics
                                ---------- Period --------- -------- Lifetime --------
                                Active    High   Total      Total  PerMax     High
Hello Messages                  0         0         0       0      4          4
Hello Request Sent              0         0         0       0      1          1
Hello Response Received         0         0         0       0      1          1
Hello Requests Received         0         0         0       0      1          1
Hello Response Sent             0         0         0       0      1          1
Heartbeat Messages              0         0         0       0      116        112
Heartbeats sent                 0         0         0       0      29         28
    Heartbeats received         0         0         0       0      29         28
    Heartbeat Responses sent    0         0         0       0      29         28
    Heartbeat Responses received 0        0         0       0      29         28
Sync Messages                   0         0         0       0      2          2
    Sync Request sent           0         0         0       0      1          1
    Sync Messages Received      0         0         0       0      1          1
Query Messages                  0         0         0       0      2          2
    Query Request Sent          0         0         0       0      1          1
Query Response Received         0         0         0       0      1          1
Query Request Received          0         0         0       0      0          0
Query Response Sent             0         0         0       0      0          0


Avg Latency=0.000 for 0
Max Latency=0.000

Last redundant transaction processed: 1000
```

SCTP Troubleshooting

show ip asctp

The ACLI **show ip asctp** command provides active SCTP state information as shown below.

```
ACMEPACKET# show ip connections asctp

A-SCTP Internet Connections
Active ASCTP associations (including servers)
Socket     Proto   Type    Local Address      Foreign Address    State
2b2d1a84   SCTP    1to1    10.1.209.50.8192   10.1.209.47.5050   pri ESTAB
                           10.1.210.50.8192   10.1.210.47.5050   sec
```

```
2b2d238C    SCTP    1to1    2.2.2.2:5060                        LISTEN
                            1.1.1.1:5060

2b2d2730    SCTP    1to1    10.1.210.50:5060                    LISTEN
10.1.209.50:5060
```

show diameter-director subscriber-only

The show diameter-director subscriber-only command with no arguments presents counts for current sessions and their states, given in period and lifetime windows.

For example:

```
ACMESYSTEM# show diameter-director subscriber-only
15:34:13-154
Diameter Subscriber-only Status
-------- Period --------  -------- Lifetime --------
                Active    High   Total      Total  PerMax    High
Subscribers          0       0       0          0       0       0
  Initial            0       0       0          0       0       0
  Established        0       0       0          0       0       0
  Terminated         0       0       0          0       0       0
  Timeout            0       0       0          0       0       0
  Application Miss   -       -       0          0       0       -
```

You may enter the by-application or by-key arguments in the proper position to search by that criteria to obtain the filtered output. The command is entered as:

```
show diameter-director sessions [by-application <application-id> | by-
key <session-id>]
```

For example:

```
ACMESYSTEM# show diameter-director subscriber-only by-application
<enter>
Diameter Director Sessions Summary
Application    Key              Cached Entries
Gx             16777625:0       3
S6             16777251:10415   0
```

Entering the by-application argument and the application to search for displays all sessions for a provided application:

Entering the by-key argument with a supplied key displays information for that key.

show diameter-director dns

The **show diameter-director dns** command displays DNS statistics for lifetime time frames on the Net-Net Diameter Director.

```
Acmepacket# show diameter-director dns
17:26:22-187
Diameter Director DNS
                          -------- Lifetime --------
                          Recent      Total  PerMax
DNS Queries                  0          0       0
```

```
DNS Cache Hits              0          0       0
DNS Cache Misses            0          0       0
DNS Client Errors           0          0       0
DNS Server Errors           0          0       0
DNS Responses               0          0       0
```

Configuration Changes that Cause Diameter Connection Disconnects

The table below lists the parameters to the applicable diameter-director elements that cause diameter connections to disconnect when you change their configuration.

- Diameter-Director-Config
  - State
- Diameter-Director-Agent
  - Hostname
  - IP-address
  - Port
  - Transport-Protocol
  - Connection-Mode
  - State
  - Realm-Id
  - Diameter-Director Applications
- Diameter-Director-Interface
  - State
  - Realm-Id
  - Diameter-Director-Ports
  - Address
  - Port
  - Transport-Protocol
  - Multi-home Addresses
  - Allow-anonymous
  - Diameter-Director Applications
  - Origin-Realm
  - Origin-Host-Identifier
  - Origin-Host-Format
- Diameter-Director-Group
  - Destinations
  - Diameter-Director-Applications