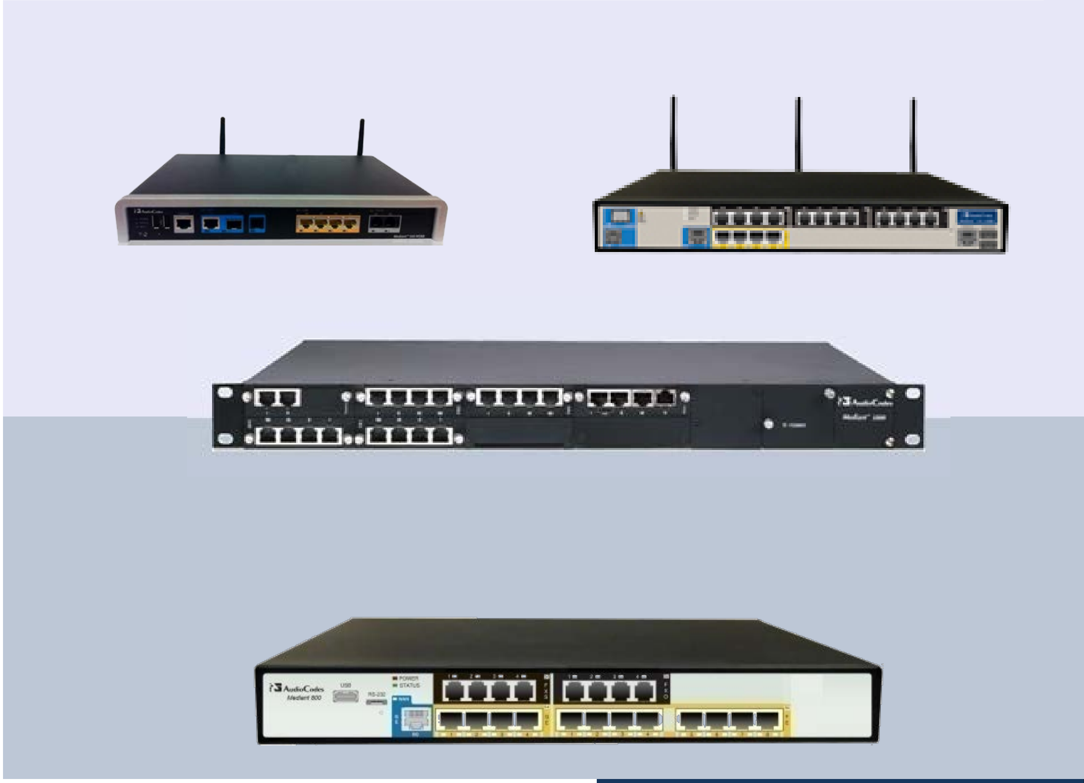
Multi-Service Business Routers

Mediant™ Series

VoIP Analog & Digital Media Gateways

Configuration Note
Capturing Traffic on MSBR



Version 1.0

April 2013

Document # LTRT-40304

Table of Contents

List of Figures

List of Tables

Notice

This document describes capturing traffic on the Mediant 500 MSBR, Mediant 800 MSBR, Mediant 850 MSBR and Mediant 1000B MSBR.

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee the accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document and other documents can be viewed by registered customers at http://www.audiocodes.com/downloads.

This document is subject to change without notice.

Date Published: April-08-2013

Trademarks

AudioCodes, AC, AudioCoded, Ardito, CTI2, CTI², CTI Squared, HD VoIP, HD VoIP Sounds Better, InTouch, IPmedia, Mediant, MediaPack, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, TrunkPack, VMAS, VoicePacketizer, VoIPerfect, VoIPerfectHD, What's Inside Matters, Your Gateway To VoIP and 3GX are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are generally provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For technical support for products purchased directly from AudioCodes, or for customers subscribed to AudioCodes Customer Technical Support (ACTS), contact support@audiocodes.com.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our Web site at http://www.audiocodes.com/downloads.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Note: Throughout this manual, unless otherwise specified, the term *device* refers to the Mediant 500 MSBR, Mediant 800 MSBR, Mediant 850 MSBR and Mediant 1000B MSBR.

Reader's Notes

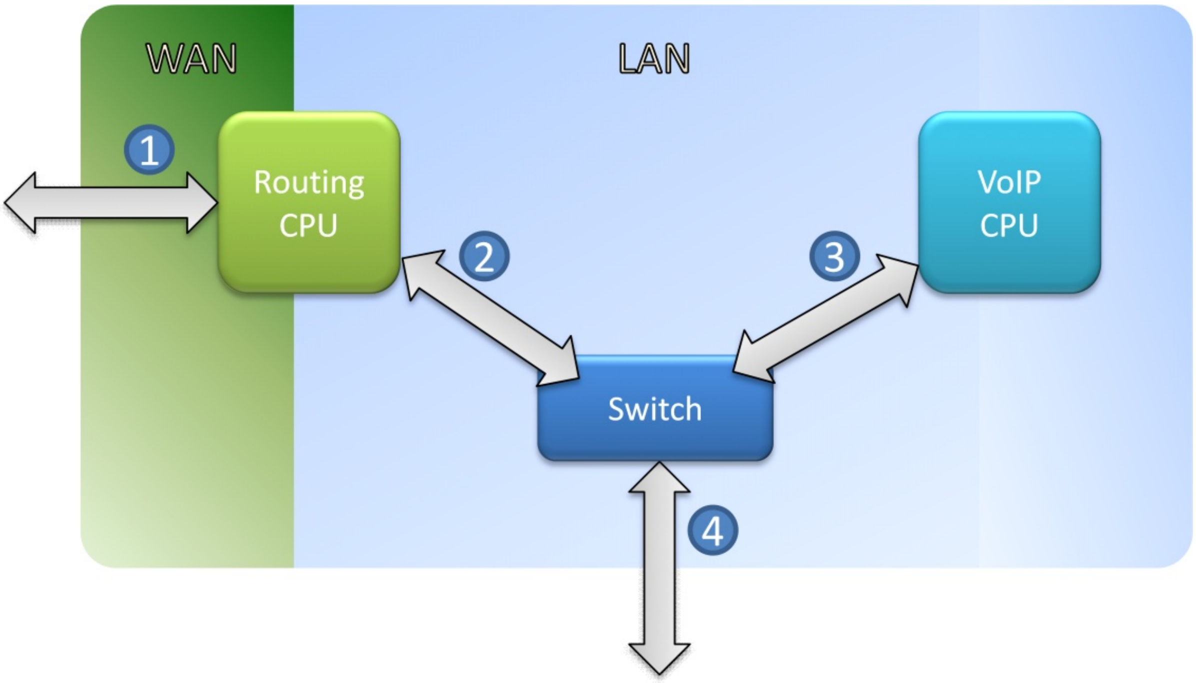
1 Introduction

This Configuration Note is applicable to the Mediant 500 MSBR, Mediant 800 MSBR, Mediant 850 MSBR and Mediant 1000B MSBR (the device, for short). Among its other capabilities, the device is also a router that forwards data between its interfaces.

The device's internal networking architecture is comprised of a routing CPU for data with WAN access, a CPU for VoIP, and a Layer 2 switch that connects between the two (see Figure 1-1 below).

There are several paths a packet takes when transmitted across the device. During the routing of the packet, it can go through one or more changes (e.g., NAT and ALG). It is crucial to track the packet at strategic points when resolving networking problems.

Figure 1-1: Routing Paths



Reader's Notes

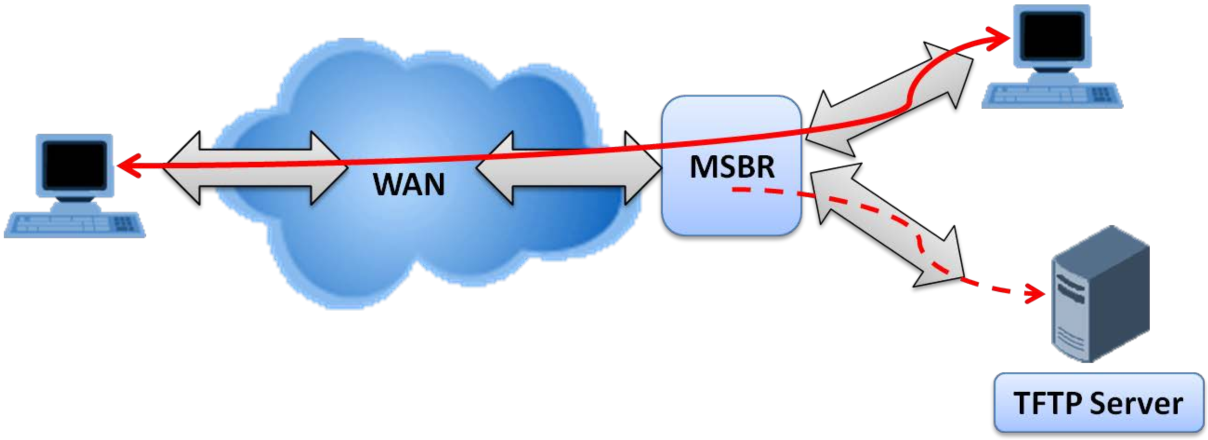
2 Physical Capture

This section describes the CLI commands to capture traffic on points ① and ② (see Figure 1-1).

2.1 Preparing for Traffic Capture

The method captures the traffic as it traverses the router CPU at the lowest level possible (hence, physical capture). It accumulates the information into a .pcap file and sends it to a TFTP server (see Figure 2-1), where you can open the file using Wireshark.

Figure 2-1: Capturing Network Traffic



This section describes how to prepare for capturing traffic using Wireshark and TFTP.

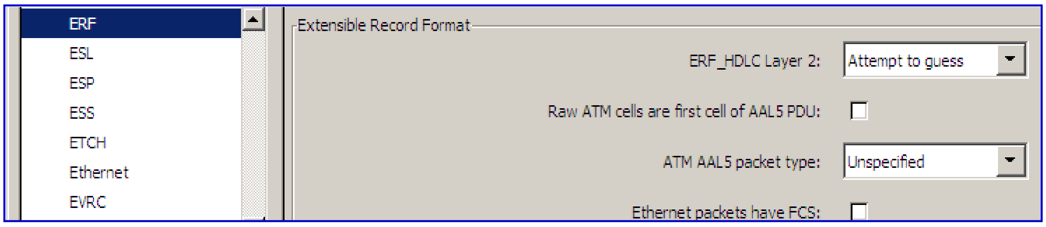
The procedure below describes how to install and configure Wireshark.

2.1.1 Installing and Configuring Wireshark

➢ **To install and configure Wireshark:**

1. Connect a PC to the device and configure the PC's IP address to be in the same subnet as the device. Ping the device to verify connectivity.
2. Download Wireshark (v1.6.0 or later) from http://www.wireshark.org/download.html onto the PC and start it.
3. Access the ‘Preferences - Extensible Record Format’ screen (**Edit** menu > **Preferences** > **Protocols** > **ERF**).

Figure 2-2: Preferences - Extensible Record Format Screen



Extensible Record Format

ERF_HDLC Layer 2: Attempt to guess

Raw ATM cells are first cell of AAL5 PDU: □

ATM AAL5 packet type: Unspecified

Ethernet packets have FCS: □

4. From the ‘ERF_HDLC Layer 2' drop-down list, select **Attempt to guess**.
5. Clear the 'Raw ATM cells are first cell of AAL5 PDU’ check box.
6. From the ‘ATM AAL5 packet type' drop-down list, select **Unspecified**.
7. Clear the ‘Ethernet packets have FCS’ check box.
8. Click **OK**.

Notes:

- It is crucial to follow the above steps carefully. An Extensible Record Format (ERF) header will appear in the beginning of each captured packet. This is an artificial header and is not part of the packets traversing the machine. It is there to allow correct operation of the capture over multiple physical links.
- When viewing the captured file, click the **Time** column to sort the frames according to time.

2.1.2 Activating TFTP

The procedures for activating TFTP are described below.

2.1.2.1 Setting up IP Interface for On-Board TFTP Client

The procedure below describes how to assign a VLAN on the LAN-side of the device, using CLI. An interface with an IP address must be defined on the Routing CPU for the captured file (.pcap file) to be sent to the TFTP server.

➢ **To assign a VLAN to the LAN side using CLI:**

1. At the CLI prompt, enter the following commands:

```
> enable
```

2. Enable data configuration mode:

```
# configure data
```

3. Enable interface configuration mode for VLAN 1:

```
(config-data)# interface VLAN 1
```

4. Configure the IP address for the device’s routing CPU on VLAN 1 (from which the *.pcap* file will be sent to the PC):

```
(conf-if-VLAN 1)# ip address 10.11.99.198 255.255.0.0
(conf-if-VLAN 1)# exit
(config-data)# exit
```

5. Run the **write** command to save the configuration:

```
# write
Writing configuration...done
```

2.1.2.2 Activating TFTP Server

- Activate the TFTP server on your PC.

Note: The sole requirement for the file to be sent to the TFTP server is that the server will be routable from the MSBR. One doesn’t necessarily need to locate the TFTP server on the LAN side, or define VLAN 1, assuming the TFTP server is routable through other interfaces.

2.2 Defining Traffic Capture Interfaces

The procedure below describes how to define the interfaces for capturing traffic using Command Line Interface (CLI) commands.

Note: Defining the interfaces does not start the capturing process.

➢ **To add a new interface to the capture interface list:**

1. Establish a CLI session (e.g., with Telnet).
2. At the CLI prompt, enter the following commands:

```
> enable
# debug capture data physical <physical-interface>
```

where <physical-interface> can be any interface listed in Table 2-1.

➢ **To remove an interface from the capture interface list:**

- At the CLI prompt, enter the following command:

```
# no debug capture data physical <physical-interface>
```

Depending on the physical links on the device, you can capture on the following interfaces:

Table 2-1: List of Interfaces

Interface	Description	Availability
`eth-lan`	Captures traffic from and to the Routing CPU Ethernet LAN side	Always
`cellular-wan`	Captures traffic from and to the cellular WAN – all logical interfaces	Always
`eth-wan`	Captures traffic from and to the Ethernet WAN – all logical interfaces	Ethernet WAN mezzanine
`shdsl-wan`	Captures traffic from and to the SHDSL WAN – all logical interfaces	SHDSL mezzanine
`shdsl-wan atm cells`	Captures ATM cells from and to the SHDSL mezzanine	SHDSL mezzanine, ATM mode only
`shdsl-wan`	Captures traffic from and to the SHDSL WAN – all logical interfaces	SHDSL mezzanine
`xdsl-wan`	Captures traffic from and to the xDSL WAN – all logical interfaces	ADSL/VDSL mezzanine
`fiber-wan`	Captures traffic from and to the fiber WAN - all logical interfaces	Fiber ports on the Mediant 850 and the Mediant 500

2.3 Capturing Traffic

The procedures for capturing traffic are described below.

2.3.1 Starting and Stopping Traffic Capture

The procedure below describes how to start and stop capturing traffic. The procedure also describes how to view capturing traffic status.

➢ **To start capturing traffic:**

1. Establish a CLI session (e.g. with Telnet).
2. At the CLI prompt, enter the following commands:

```
> enable
# debug capture data physical start
```

➢ **To stop capturing traffic and receive a *.pcap* file on the TFTP server:**

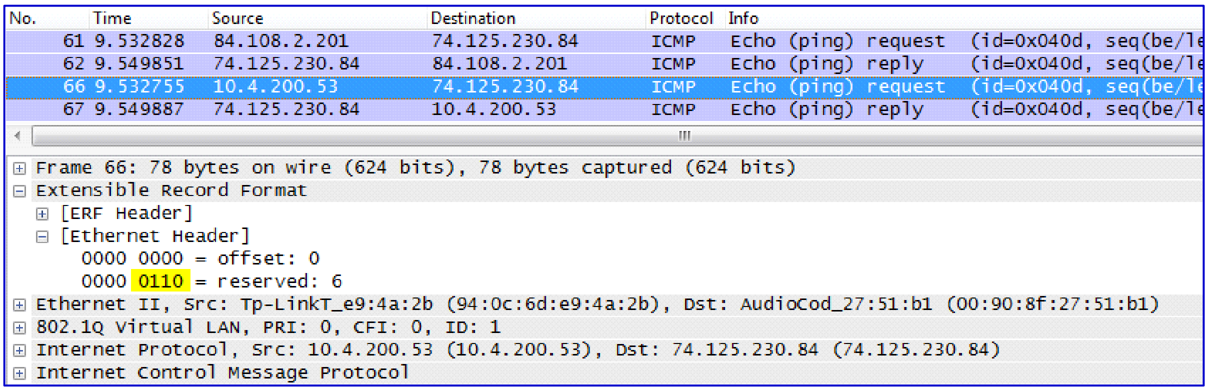
1. Establish a CLI session (e.g. with Telnet).
2. At the CLI prompt, enter the following commands:

```
> enable
# debug capture data physical stop <PC ip address>
```

3. The *debug-capture-data-<date>-<time>.pcap* file is received.
4. When opening the file, remember to click the **Time** column to sort the file by time. Verify that your Wireshark is not configured to show delta-time on this column.

Tip: The device has multiple LAN ports. When inspecting packets that were **received on the LAN side**, you can determine the exact physical switch port it was received on by looking at the *erf.eth.res1* field. For example, in the Wireshark Capture below you can see that the packet was received on the 7th port (as the count starts from 0).

Figure 2-3: Wireshark Capture



```
No.        Time       Source            Destination       Protocol Info
        61 9.532828   84.108.2.201      74.125.230.84     ICMP     Echo (ping) request  (id=0x040d, seq(be/le
        62 9.549851   74.125.230.84     84.108.2.201      ICMP     Echo (ping) reply    (id=0x040d, seq(be/le
        66 9.532755   10.4.200.53       74.125.230.84     ICMP     Echo (ping) request  (id=0x040d, seq(be/le
        67 9.549887   74.125.230.84     10.4.200.53       ICMP     Echo (ping) reply    (id=0x040d, seq(be/le

⊞ Frame 66: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
⊟ Extensible Record Format
  ⊞ [ERF Header]
  ⊟ [Ethernet Header]
      0000 0000 = offset: 0
      0000 0110 = reserved: 6
⊞ Ethernet II, Src: Tp-LinkT_e9:4a:2b (94:0c:6d:e9:4a:2b), Dst: AudioCod_27:51:b1 (00:90:8f:27:51:b1)
⊞ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 1
⊞ Internet Protocol, Src: 10.4.200.53 (10.4.200.53), Dst: 74.125.230.84 (74.125.230.84)
⊞ Internet Control Message Protocol
```

➢ **To view capture status and active rules:**

1. Establish a CLI session (e.g., with Telnet).
2. At the CLI prompt, enter the following commands:

```
> enable
# debug capture data physical show
```

Note: If the TFTP transfer failed for some reason, it is possible to send the last *.pcap* file to the TFTP server again.

➢ **To send the last *.pcap* file to the TFTP server again, if the TFTP transfer fails:**

1. Establish a CLI session (e.g., with Telnet).
2. At the CLI prompt, enter the following commands:

```
> enable
# debug capture data physical get_last_capture <PC ip address>
```

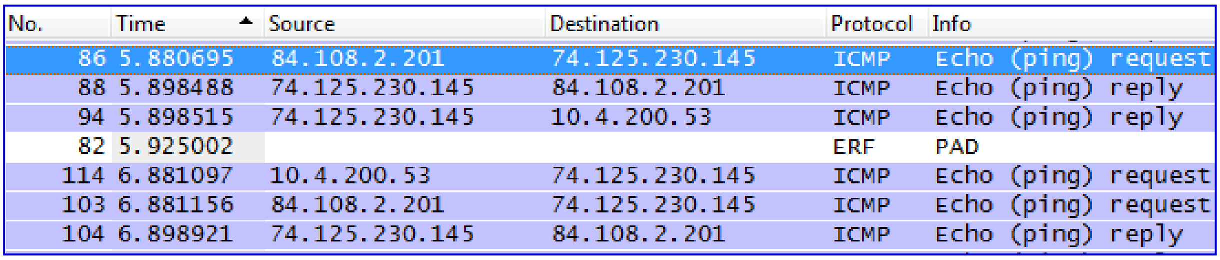
2.3.2 Synchronization with External Events

Sometimes you may wish to know where exactly in the capture an external event occurred, e.g., plugging or unplugging one of the cables, enabling a certain feature, adding a route or initiating a restart of another server. In any of these cases, you can proactively add a bookmark into the file. This may be done by using the following command:

```
> enable
# debug capture data physical insert_pad
```

Using the *"erf.types.type == pad"* filter, a "PAD" packet can later be inspected in the captured file, depending on the time the external event took place.

Figure 2-4: Wireshark Capture – External Events



No.	Time	Source	Destination	Protocol	Info
86	5.880695	84.108.2.201	74.125.230.145	ICMP	Echo (ping) request
88	5.898488	74.125.230.145	84.108.2.201	ICMP	Echo (ping) reply
94	5.898515	74.125.230.145	10.4.200.53	ICMP	Echo (ping) reply
82	5.925002			ERF	PAD
114	6.881097	10.4.200.53	74.125.230.145	ICMP	Echo (ping) request
103	6.881156	84.108.2.201	74.125.230.145	ICMP	Echo (ping) request
104	6.898921	74.125.230.145	84.108.2.201	ICMP	Echo (ping) reply

2.4 Retrieving Captured Data using SSH

On some networks, TFTP is not possible due to routing or security limitations. The MSBR supports a method for transferring the debug capture to the management PC using SFTP (Secure FTP), which is part of the SSH suite. This allows any PC with SSH access (even over the WAN) to retrieve captured traffic without changing security policies.

➢ **To transfer a debug capture using SSH:**

1. Install an SFTP-capable SSH client on your PC, such as WinSCP (http://winscp.net) for Windows or OpenSSH for Linux.
2. Establish an SSH session with the MSBR.
3. Configure and start a debug capture as described under Section 2.3.1.
4. Stop the debug capture without specifying a TFTP IP address, i.e., "debug capture data physical stop".
5. Using SFTP, retrieve the file *debug-capture-data.pcap* from the MSBR.

Example using OpenSSH 5.3 on Linux:

sftp Admin@192.168.1.1:debug-capture-data.pcap file.pcap

2.5 Example of Capturing Traffic

The following example lists the commands for a device that has an SHDSL WAN mezzanine and a cellular WAN modem, and you want to capture traffic on both WAN interfaces and a LAN interface.

Its VLAN 1 IP address is **192.168.0.1**, and has a PC with an IP address of **192.168.0.50** connected to it.

```
# debug capture data physical shdsl-wan
Interface shdsl-wan was added to the debug capture rules
      Use start command in order to start the debug capture

# debug capture data physical cellular-wan
Interface cellular-wan was added to the debug capture rules
      Use start command in order to start the debug capture

# debug capture data physical eth-lan
Interface eth-lan add to debug capture rules
      Use start command in order to start the debug capture

# debug capture data physical show
Debug Capture Physical is NOT_ACTIVE
Active Physical Ports are:
        shdsl-wan
        cellular-wan
        eth-lan

# debug capture data physical start
NOTE: Debug capture data will be collected locally, and later
      sent to a PC via TFTP. Please make sure that
      VLAN 1 is defined and the PC is accessible through it.

# debug capture data physical stop 192.168.0.50
Sending capture to TFTP server 192.168.0.50, filename debug-
capture-data-<date>-<time>.pcap
```

Note: The resulting file contains traffic captured on different Layer 2 interfaces (ATM for SHDSL, PPP for cellular, and Ethernet for the LAN side). All can be viewed in a single file.

2.6 Limitations

The following limitations apply to traffic capturing:

- The *pcap* file size is limited to 20 MB, in order to prevent interruptions to the device.
- Because of internal limitations, the capture works properly with LAN and WAN with a total traffic rate of up to 100 Mbps (50 Mbps for download and 50 Mbps for upload). Trying to capture traffic with higher rates may cause loss of some network packets that won`t appear in the capture file.

Reader's Notes

3 Logical Capture

This section describes the CLI commands for capturing traffic on points ①, ② and ③ (see Figure 1-1).

The method is only capable of capturing packets over a single interface at a time. Using this capture method, all packets passing through can therefore be captured, e.g., interface VLAN 1. To see another interface, you must connect via another terminal (for example, a telnet session, of which you can have 5 simultaneously), and run another debug capture interface process.

Advantages of performing logical (rather than physical) capturing:

1. It can echo the captured data to the terminal (as well as save the captured data to a file)
2. It features capture filters which can be combined:
   a. Filter by protocol (TCP, UDP, ARP, etc.)
   b. Filter by host (IP address of a host)
   c. Filter by port (N\A to ICMP and ARP)
3. The capture is available to VOIP interfaces as well
4. IPSec traffic can be captured (before being encrypted)
5. It can capture on serial, for rough debugging, when you don't even have an interface with an IP address.

When starting a capture, the logical interface to use for the capture must be provided.

In addition, protocol and host filters are always applied, but can get a wildcard value ("all" protocols and "any" host).

3.1 Syntax

The logical debug capture is accessible via the root menu of the CLI (privileged mode is required).

There are four possible ways to use the logical debug capture:

- Without port filter:

```
# debug capture <Entity> interface <InterfaceType>
<InterfaceId> proto <ProtocolFilter> host <HostFilter>
```

- With port filter:

```
# debug capture <Entity> interface <InterfaceType>
<InterfaceId> proto <ProtocolFilter> host <HostFilter> port
<PortFilter>
```

- Capturing to file:

```
# debug capture <Entity> interface <InterfaceType>
<InterfaceId> proto <ProtocolFilter> host <HostFilter> port
<PortFilter> tftp-server <tftpServerAddress>
```

- IPSEC capture (data only):

```
# debug capture data interface <InterfaceType> <InterfaceId>
ipsec proto <ProtocolFilter> host <HostFilter> port
<PortFilter> tftp-server <tftpServerAddress>
*This variant enables capturing plaintext traffic which is
encrypted by IPsec on the wire.
*Note:  This variant is supported from version 6.40.012.003
```

Where:

<Entity>	voip or data
<InterfaceType>	interface type (for instance: vlan, cellular, GigabitEthernet)
<InterfaceId>	ID of the interface (15, 0/0, 0/0.10, etc.)
<ProtocolFilter>	all / ip / tcp / udp / arp / icmp
<HostFilter>	Capture only packets to/from this IP address
<PortFilter>	Capture only packets to/from this port (N\A for ICMP and ARP)
<tftpServerAddres>	IP address of a TFTP server to send the captured file to

3.2 Capturing to Screen

When capturing to screen is started, the capturing process is started in the background.

The terminal is connected to the capturing process output, and all captured data is echoed to screen.

To terminate the capture, use CTRL-C.

3.3 Examples

- The example below shows how to capture all traffic passing through interface VLAN 1 of the VOIP entity (CMX)

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto all
host any
21:17:20.237512 00:90:8f:22:81:fe > 00:02:b3:20:00:c3, ethertype
IPv4 (0x0800),
length 56: (tos 0x28, ttl  64, id 42005, offset 0, flags [DF],
proto: TCP (6), l
ength: 42) 192.168.0.2.23 > 192.168.0.25.3730: P, cksum 0x3fb3
(correct), 933017
297:933017299(2) ack 760258666 win 5840
21:17:20.241067 00:90:8f:22:81:fe > 00:90:8f:22:81:ff, ethertype
IPv4 (0x0800),
length 109: (tos 0x0, ttl  64, id 0, offset 0, flags [DF], proto:
UDP (17), leng
th: 95) 192.168.0.2.32768 > 10.4.2.78.514: SYSLOG, length: 67
        Facility user (1), Severity info (6)
        Msg: kernel: [4637146.797000] eth0.1:
dev_set_promiscui[|syslog]

Mediant 800 - MSBG#
```

- The example below shows how to capture all ARP traffic passing through interface VLAN 1 of the VOIP entity (CMX)

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto arp
host any

21:16:46.603259 00:90:8f:22:81:ff > ff:ff:ff:ff:ff:ff, ethertype
ARP (0x0806), l
ength 56: arp who-has 192.168.0.2 tell 192.168.0.1
21:16:46.604173 00:90:8f:22:81:fe > 00:90:8f:22:81:ff, ethertype
ARP (0x0806), l
ength 42: arp reply 192.168.0.2 is-at 00:90:8f:22:81:fe

Mediant 800 - MSBG#
```

- The example below shows how to capture all ICMP traffic passing through interface GigabitEthernet 0/0 (WAN) of the DATA entity (RMX)

```
Mediant 800 - MSBG# debug capture data interface GigabitEthernet
0/0 proto icmp host any

21:17:25.032682 0:13:21:d5:bc:7e 0:90:8f:22:82:0 0800 115:
10.4.2.78 > 10.4.28.5
1: icmp: 10.4.2.78 udp port 514 unreachable (ttl 128, id 22053,
len 101)
21:17:48.709538 0:90:8f:22:82:0 0:e:62:d1:22:f 0800 98: 10.4.28.51
> 10.4.0.1: i
cmp: echo request (DF) (ttl 63, id 0, len 84)
21:17:48.710152 0:e:62:d1:22:f 0:90:8f:22:82:0 0800 98: 10.4.0.1 >
192.168.0.2:
icmp: echo reply (DF) (ttl 255, id 0, len 84)

Mediant 800 - MSBG#
```

- The example below shows how to capture all HTTP (TCP port 80) traffic passing through interface VLAN 1 of the VOIP entity (CMX)

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto tcp
host any port 80

21:26:41.114360 00:02:b3:20:00:c3 > 00:90:8f:22:81:fe, ethertype
IPv4 (0x0800),
length 66: (tos 0x0, ttl 128, id 43660, offset 0, flags [DF],
proto: TCP (6), le
ngth: 52) 192.168.0.25.4103 > 192.168.0.2.80: S, cksum 0x0162
(correct), 1702000
256:1702000256(0) win 65535 <mss 1460,nop,wscale 1,nop,nop,sackOK>
21:26:41.116061 00:90:8f:22:81:fe > 00:02:b3:20:00:c3, ethertype
IPv4 (0x0800),
length 66: (tos 0x28, ttl  64, id 0, offset 0, flags [DF], proto:
TCP (6), lengt
h: 52) 192.168.0.2.80 > 192.168.0.25.4103: S, cksum 0xfb68
(correct), 3412534190
:3412534190(0) ack 1702000257 win 5840 <mss
1460,nop,nop,sackOK,nop,wscale 3>
21:26:41.116706 00:02:b3:20:00:c3 > 00:90:8f:22:81:fe, ethertype
IPv4 (0x0800),
length 60: (tos 0x0, ttl 128, id 43661, offset 0, flags [DF],
proto: TCP (6), le
ngth: 40) 192.168.0.25.4103 > 192.168.0.2.80: ., cksum 0xd306
(correct), ack 1 w
in 32768
Mediant 800 - MSBG#
```

- The example below shows how to capture all ICMP traffic encrypted by IPsec passing through interface GigabitEthernet 0/0 (WAN) of the DATA entity (RMX)

```
Mediant 800 - MSBG# debug capture data interface GigabitEthernet
0/0 ipsec proto icmp host any

21:17:25.032682 0:13:21:d5:bc:7e 0:90:8f:22:82:0 0800 115:
10.4.2.78 > 10.4.28.5
1: icmp: 10.4.2.78 udp port 514 unreachable (ttl 128, id 22053,
len 101)
21:17:48.709538 0:90:8f:22:82:0 0:e:62:d1:22:f 0800 98: 10.4.28.51
> 10.4.0.1: i
cmp: echo request (DF) (ttl 63, id 0, len 84)
21:17:48.710152 0:e:62:d1:22:f 0:90:8f:22:82:0 0800 98: 10.4.0.1 >
192.168.0.2:
icmp: echo reply (DF) (ttl 255, id 0, len 84)

Mediant 800 - MSBG#
```

- The examples below show how to capture **to file**

  When capturing to file is started, the capturing process is started in the background, saving all captured data directly to a file. The CLI session is temporarily locked during the capture process.

  When starting a logical debug capture to a file, the TFTP server to which to send the captured file must be provided. This server must be accessible from CMX or RMX (depending on which entity the debug capture is going to be executed). If the IP Address of the TFTP server isn't accessible, the captured data will be lost.

  To terminate the capture and send the captured file to the TFTP Server, use CTRL-C.

  When starting a logical debug capture to a file, use the full format of the logical debug capture, including all three filters. It's possible, however, to use wildcard values ("all" and "any") and thus not filter the captured data at all.

Note: The capture file is limited to 2000 captured packets (packet size does not matter). The capture therefore automatically stops once the capture file reaches 2000 packets.

- The example below shows how to capture all traffic passing through interface VLAN 1 of the VOIP entity (CMX) to a file, and when CTRL-C is entered, the captured file is sent to the TFTP server at 192.168.0.25.

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto all
host any port any tftp-server 192.168.0.25
......
Mediant 800 - MSBG#
```

- The example below shows how to capture all ARP traffic passing through interface VLAN 1 of the VOIP entity (CMX) to a file, and when CTRL-C is entered, the captured file is sent to the TFTP server at 192.168.0.25.

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto arp
host any port any tftp-server 192.168.0.25
....................
Mediant 800 - MSBG#
```

- The example below shows how to capture all ICMP traffic passing through interface GigabitEthernet 0/0 (WAN) of the DATA entity (RMX) to a file, and when CTRL-C is entered, the captured file is sent to the TFTP server at 192.168.0.25.

```
Mediant 800 – MSBG# debug capture data interface GigabitEthernet
0/0 proto icmp host any port any tftp-server 192.168.0.25
....................
Mediant 800 - MSBG#
```

- The example below shows how to capture all HTTP (TCP port 80) traffic passing through interface VLAN 1 of the VOIP entity (CMX) to a file, and when CTRL-C is entered, the captured file is sent to the TFTP server at 192.168.0.25.

```
Mediant 800 - MSBG# debug capture voip interface vlan 1 proto tcp
host any port 80 tftp-server 192.168.0.25
....................
Mediant 800 - MSBG#
```

Reader's Notes

4 Port Monitoring

This section describes the CLI commands for capturing traffic on points ②, ③ and ④ (see Figure 1-1).

Port monitoring occurs inside the embedded L2 switch and allows users to monitor traffic routed from each Ethernet LAN port (point ④) to any other single LAN (point ④) or CPU port (points ② and ③).

MSBR products enable monitoring of egress traffic, ingress traffic, or both directions.

4.1 Port Monitoring Configuration

4.1.1 Destination Interface Configuring

The destination interface is the interface to which the monitored data is sent.

Operating the port monitoring is done via the "configure data interface…" CLI menu.

Syntax

The syntax of this command can include the following variations:

```
interface <type> <slot/port>
```

The command’s syntax format is described below:

Arguments	Description
<type>	Destination Interface type FastEthernet/GigabitEthernet
<slot/port>	Destination Interface slot number and port number

Examples

```
Mediant 800 (config)# interface GigabitEthernet 4/3
Mediant 800 (conf-if-GE 4/3)#
```

Note: There can be only one destination port. If you already configured a destination port, you won’t be able to change to another port. Remove the entire port monitoring configuration to be able to configure new port monitoring commands.

4.1.2 Source Interface Configuring

After you choose your destination port you can configure some source ports.
This is done by the command "port-monitor".

Syntax

The syntax of this command can include the following variations:

```
port-monitor <type> <slot/port> <direction>
```

Arguments	Description
<type>	Source Interface type FastEthernet/GigabitEthernet
<slot/port>	Source Interface slot number and port number
<direction>	Monitoring direction Ingress/Egress/Both

Examples

```
Mediant 800(conf-if-GE 4/3)# port-monitor GigabitEthernet 4/1 ingress
Mediant 800(conf-if-GE 4/3)# port-monitor FastEthernet 5/2 egress
Mediant 800(conf-if-GE 4/3)# port-monitor GigabitEthernet 4/4 both-direction
```

4.2 Port Monitoring Status

To see the port monitoring status, use the command "show data port-monitoring"

Syntax

The syntax of this command can include the following variations:

```
show data port-monitoring
```

Examples

```
Mediant 800# show data port-monitor

        Port Monitor Status
        ----------------------------
        Destination port  GigabitEthernet 4/3
        Source      port  GigabitEthernet 4/1     Direction: Ingress
        Source      port  GigabitEthernet 4/2     Direction: Egress
        Source      port  GigabitEthernet 4/4     Direction: Both
```

Configuration Note

www.audiocodes.com