Cisco ASR 5000 Mobile Video Gateway Administration Guide

Version 14.0

Last Updated: September 28, 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-27214-02

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

Modifications to this product not authorized by Cisco could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco ASR 5000 Mobile Video Gateway Administration Guide

CONTENTS

About this Guide

This preface describes the *Cisco ASR 5000 Mobile Video Gateway Administration Guide*, how it is organized and its document conventions.

The guide describes the Mobile Video Gateway and includes network deployments and interfaces, call flows, feature descriptions, configuration instructions, and CLI commands for monitoring the system. It also contains a sample MVG/P-GW configuration file.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
ℹ	Information Note	Provides information about important features or instructions.
⚠	Caution	Alerts you of potential damage to a program, device, or system.
⚡	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
✋	Electro-Static Discharge (ESD)	Warns you to take proper grounding precautions before handling ESD sensitive components or devices.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **`commands`**	This typeface represents commands that you enter, for example: **`show ip access-list`** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **`command`** *`variable`*	This typeface represents a variable that is part of a command, for example: **`show card`** *`slot_number`* slot_number is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped braces. Required keywords and variables are those components that are required to be entered as part of the command syntax.
[**`keyword`** or *`variable`*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by brackets.

Command Syntax Conventions	Description
|	Some commands support alternative variables. These options are documented within braces or brackets by separating each variable with a vertical bar. These variables can be used in conjunction with required or optional keywords or variables. For example: **`{ nonce | timestamp }`** OR [**`count`** *`number_of_packets`* | **`size`** *`number_of_bytes`*]

Supported Documents and Resources

Related Common Documentation

The most up-to-date information for this product is available in the product Release Notes provided with each product release.

The following common documents are available:

- *Hardware Installation Guide* (hardware dependent)
- *System Administration Guide* (hardware dependent)
- *Command Line Interface Reference* (network dependent)
- *AAA Interface Administration and Reference*
- *Product Overview*
- *Release Change Reference*
- *Statistics and Counters Reference*
- *Thresholding Configuration Guide*

Related Product Documentation

The following product documents are also available and work in conjunction with the Mobile Video Gateway:

- *Packet Data Network Gateway Administration Guide*
- *Gateway GPRS Support Node Administration Guide*
- *Home Agent Administration Guide*
- *Enhanced Charging Services Administration Guide*
- *Traffic Performance Optimization Administration Guide*

Obtaining Documentation

The most current Cisco documentation is available on the following website:

http://www.cisco.com/cisco/web/psa/default.html

Use the following path selections to access the Mobile Video Gateway documentation per software release:

Support > Product Support > Wireless > Additional Products > ASR 5000 Series > Configuration Guides

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.

Chapter 1
Mobile Video Gateway Overview

This chapter contains general overview information about the Cisco® Mobile Video Gateway, including:

- Product Description
- Network Deployments and Interfaces
- Features and Functionality
- How the Mobile Video Gateway Works

Product Description

The Cisco® Mobile Video Gateway is the central component of the Cisco Mobile Videoscape. It employs a number of video optimization techniques that enable mobile operators with 2.5G, 3G, and 4G wireless data networks to enhance the video experience for their subscribers while optimizing the performance of video content transmission through the mobile network.

Platform Requirements

The Mobile Video Gateway software runs on a Cisco ASR 5000 chassis functioning as a mobile gateway, enabling the ASR 5000 to function as an integrated Mobile Video Gateway. In this software release, the Mobile Video Gateway software can be integrated with the Cisco P-GW (Packet Data Network Gateway), the Cisco GGSN (Gateway GPRS Support Node), and the Cisco HA (Home Agent). The Mobile Video Gateway software runs on the StarOS operating system. The chassis can be configured with a variety of components to meet specific network deployment requirements. For additional information, refer to the installation guide for the chassis and/or contact your Cisco account representative.

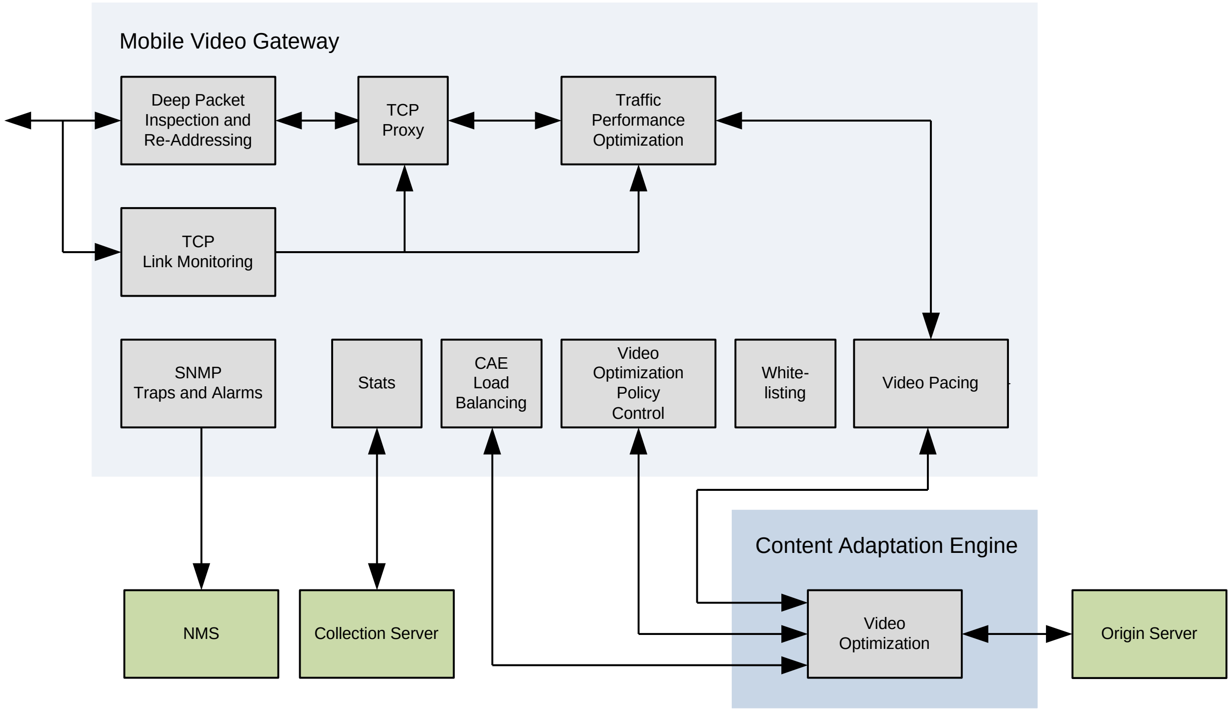
Licenses

The Mobile Video Gateway is a licensed Cisco product. Separate session and feature licenses may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the "Managing License Keys" section of the "Software Management Operations" chapter in the *System Administration Guide*.

Summary of Mobile Video Gateway Features and Functions

The following figure shows the Mobile Video Gateway features and functions.

Figure 1. Mobile Video Gateway Features and Functions



The Mobile Video Gateway features and functions include:

- DPI (Deep Packet Inspection) to identify subscriber requests for video vs. non-video content
- Transparent video re-addressing to the Cisco CAE (Content Adaptation Engine) for retrieval of optimized video content
- CAE load balancing of HTTP video requests among the CAEs in the server cluster
- Video optimization policy control for tiered subscriber services
- Video white-listing, which excludes certain video clips from video optimization
- Video pacing for “just in time” video downloading
- TCP link monitoring
- Dynamically-enabled TCP proxy
- Traffic performance optimization
- N+1 redundancy support
- SNMP traps and alarms (threshold crossing alerts)

- Mobile video statistics
- Bulk statistics for mobile video

The Cisco CAE is an optional component of the Cisco Mobile Videoscape. It runs on the Cisco UCS (Unified Computing System) platform and functions in a UCS server cluster to bring additional video optimization capabilities to the Mobile Videoscape. For information about the features and functions of the Cisco CAE, see the CAE product documentation.

Network Deployments and Interfaces

This section shows the Mobile Video Gateway as it functions in various wireless networks. The section also includes descriptions of its logical network interfaces.

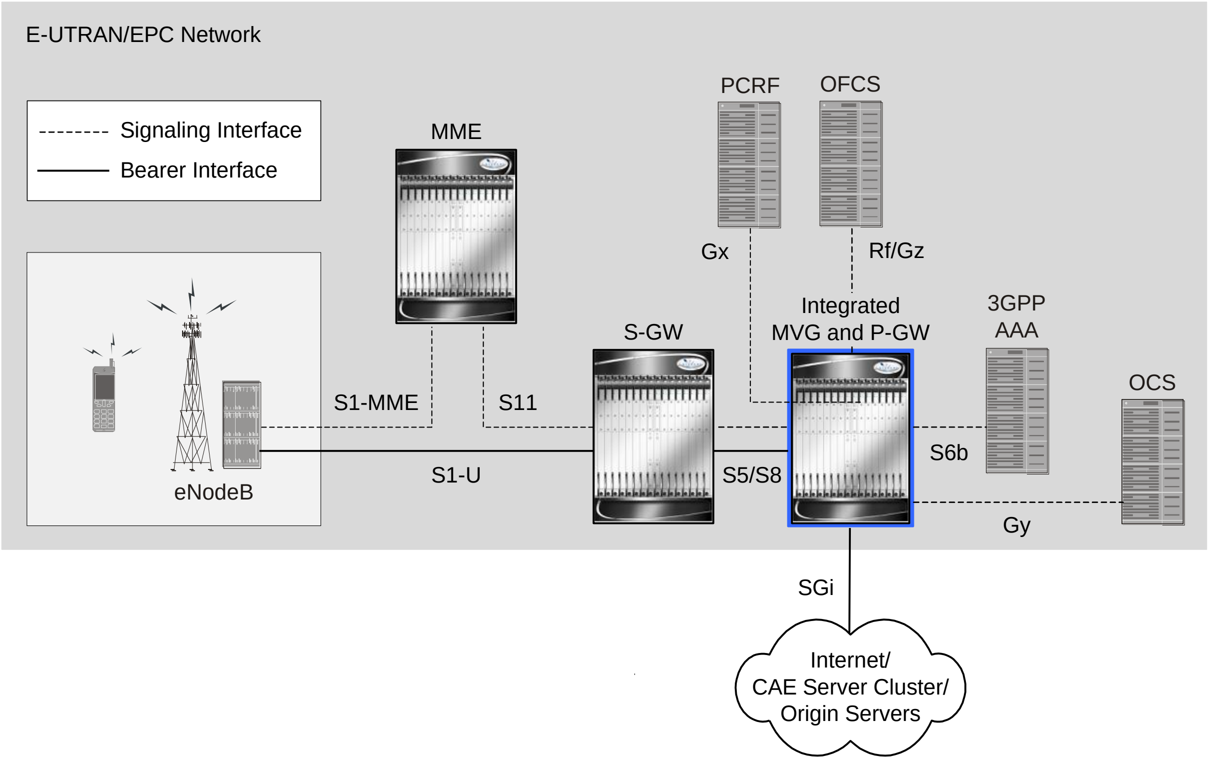
The Mobile Video Gateway in an E-UTRAN/EPC Network

In this software release, the Mobile Video Gateway software can be integrated with the Cisco P-GW in an E-UTRAN/EPC (Evolved UTRAN/Evolved Packet Core) network.

In the EPC (Evolved Packet Core), the Cisco P-GW (Packet Data Network Gateway) is the network node that terminates the SGi interface towards the PDN (Packet Data Network). The P-GW provides connectivity to external PDNs for the subscriber UEs by being the point of exit and entry of traffic for the UEs. A subscriber UE may have simultaneous connectivity with more than one P-GW for accessing multiple PDNs. The P-GW performs policy enforcement, packet filtering for each user, charging support, lawful interception, and packet screening.

The following figure shows the integrated Mobile Video Gateway and P-GW in an E-UTRAN/EPC network.

Figure 2. Mobile Video Gateway in an E-UTRAN/EPC Network



For more information about the Cisco P-GW and its connectivity to related network elements, see the *Packet Data Network Gateway Administration Guide.*

The Mobile Video Gateway in a GPRS/UMTS Network

In this software release, the Mobile Video Gateway software can be integrated with a GGSN (Gateway GPRS Support Node) in a GPRS/UMTS (General Packet Radio Service/Universal Mobile Telecommunications System) network.

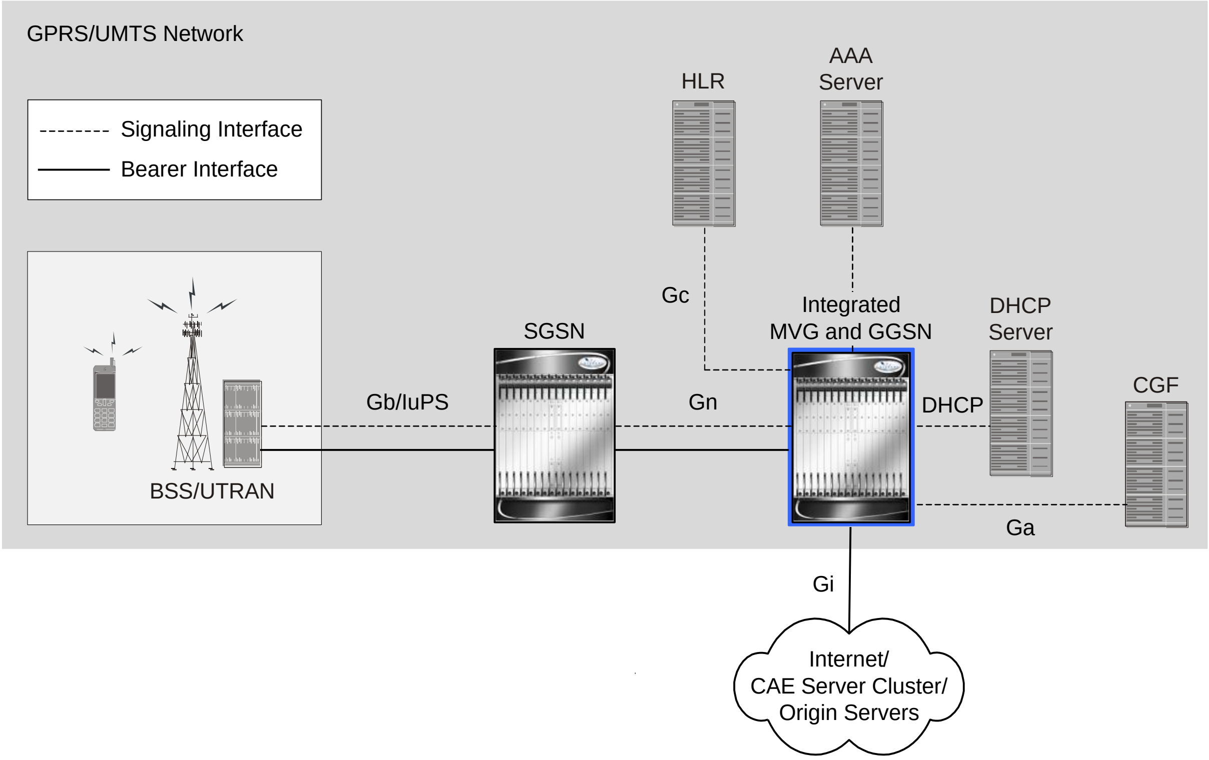
The GGSN works in conjunction with SGSNs (Serving GPRS Support Nodes) in the network to perform the following functions:

- Establish and maintain subscriber IP (Internet Protocol) or PPP (Point-to-Point Protocol) type PDP (Packet Data Protocol) contexts originated by either the MS (Mobile Station) or the network.
- Provide CDRs (Call Detail Records) to the CS (Charging Gateway), also known as the CGF (Charging Gateway Function).
- Route data traffic between the subscriber's MS and a PDN (Packet Data Network) such as the Internet or an intranet.

PDNs are associated with APNs (Access Point Names) configured on the system. Each APN consists of a set of parameters that dictate how subscriber authentication and IP address assignment is to be handled for that APN.

The following figure shows the integrated Mobile Video Gateway and GGSN in a GPRS/UMTS network.

Figure 3. Mobile Video Gateway in a GPRS/UMTS Network

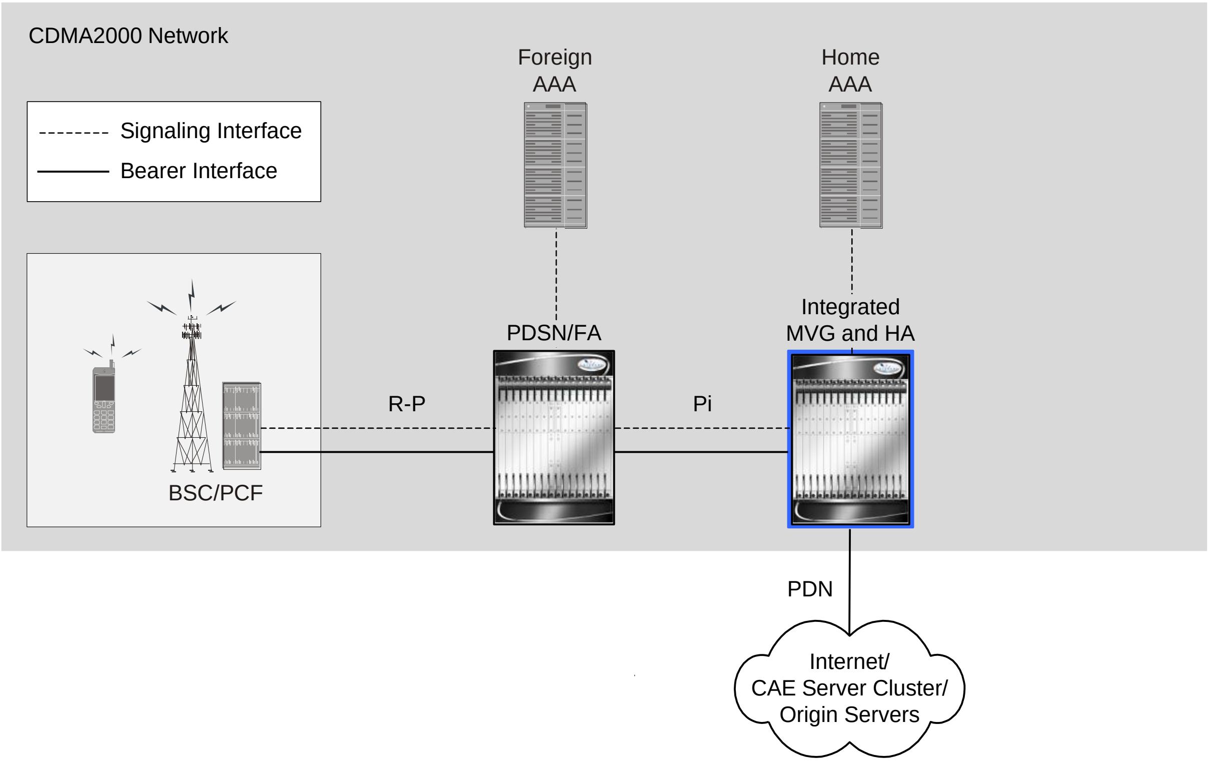


For more information about the Cisco GGSN and its connectivity to related network elements, see the *Gateway GPRS Support Node Administration Guide.*

The Mobile Video Gateway in a CDMA2000 Network

In CDMA2000 networks, the Cisco HA (Home Agent) enables subscribers to be served by their home network even when their mobile devices are not attached to their home network. The Cisco HA performs this function through interaction with the Cisco PDSN/FA (Packet Data Serving Node/Foreign Agent). The PDSN/FA provides the packet processing and redirection to the subscriber's home network via the HA. The following figure shows the integrated Mobile Video Gateway and HA with a PDSN/FA in a CDMA2000 network.

Figure 4. Mobile Video Gateway in a CDMA2000 Network

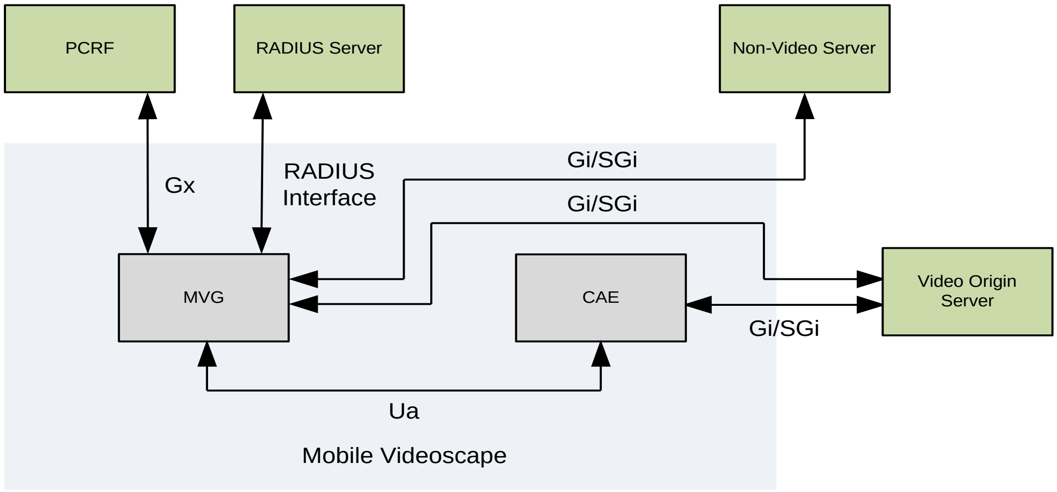


For more information about the Cisco HA and its connectivity to related network elements, see the *Home Agent Administration Guide.*

Mobile Video Gateway Logical Network Interfaces

The following figure shows the logical network interfaces on the Mobile Video Gateway.

Figure 5. Logical Network Interfaces on the Mobile Video Gateway



The following table provides descriptions of the logical network interfaces on the Mobile Video Gateway. The Mobile Video Gateway also supports the logical network interfaces of the Cisco P-GW and Cisco HA when integrated with those products.

Table 1. Logical Network Interfaces on the Mobile Video Gateway

Interface	Description
PCRF Interface	The Mobile Video Gateway can use the Gx interface to connect to a PCRF (Policy and Charging Rules Function) server to receive subscriber policy information and charging rules.
RADIUS Interface	The Mobile Video Gateway uses a RADIUS interface to exchange signaling messages with the external RADIUS server.
Video Origin Server Interface	The Mobile Video Gateway uses the Gi or SGi interface to connect to the video origin servers in the network. The Mobile Video Gateway also uses the Gi or SGi interface to connect to non-video origin servers.
CAE Interface	The Mobile Video Gateway uses a Cisco-enhanced HTTP interface called the Ua interface to connect to the Cisco CAE. The Cisco CAE is an optional component of the Cisco Mobile Videoscape.

Features and Functionality

The following features and functions are supported on the Mobile Video Gateway:

- Deep Packet Inspection
- Transparent Video Re-addressing
- CAE Load Balancing
- Video Optimization Policy Control
- Video White-listing
- Video Pacing
- TCP Link Monitoring
- Dynamically-enabled TCP Proxy
- Traffic Performance Optimization
- N+1 Stateful Redundancy
- Threshold Crossing Alerts
- Mobile Video Statistics
- Bulk Statistics for Mobile Video

Deep Packet Inspection

The Mobile Video Gateway performs DPI (Deep Packet Inspection) of HTTP traffic to identify video vs. non-video traffic based on configured Active Charging Service rule definitions. An Active Charging Service is a component of the Enhanced Charging Services on the Cisco ASR 5000.

While SPI (Shallow Packet Inspection) examines IP headers (Layer 3) and UDP and TCP headers (Layer 4) for an Active Charging Service, DPI on the Mobile Video Gateway examines URI information (Layer 7) for HTTP message information to identify video vs. non-video content based on configured rules. The following information is used for DPI:

- HTTP Request headers for matching hostnames.
- HTTP Request URLs of the destination websites to identify the video content OSs (Origin Servers).
- HTTP Response headers for matching the content type.

For more information about Enhanced Charging Services on the ASR 5000, see the *Enhanced Charging Services Administration Guide.*

Transparent Video Re-addressing

The Mobile Video Gateway can re-address HTTP video requests intended for video content OSs toward the Cisco CAE for retrieval of optimized video content. The Cisco CAE is an optional component of the Cisco Mobile Videoscape. It functions in a video server cluster to bring additional optimization capabilities to the Mobile Videoscape.

Important: The transparent video re-addressing feature is not fully qualified and is not supported for field deployment. It is available for lab demo/lab trial only.

The transparent video re-addressing feature works in conjunction with the dynamic TCP proxy feature to send video requests to the CAE cluster without using HTTP redirection, so that the re-addressing to the CAEs remains transparent to the video clients on the subscriber UEs.

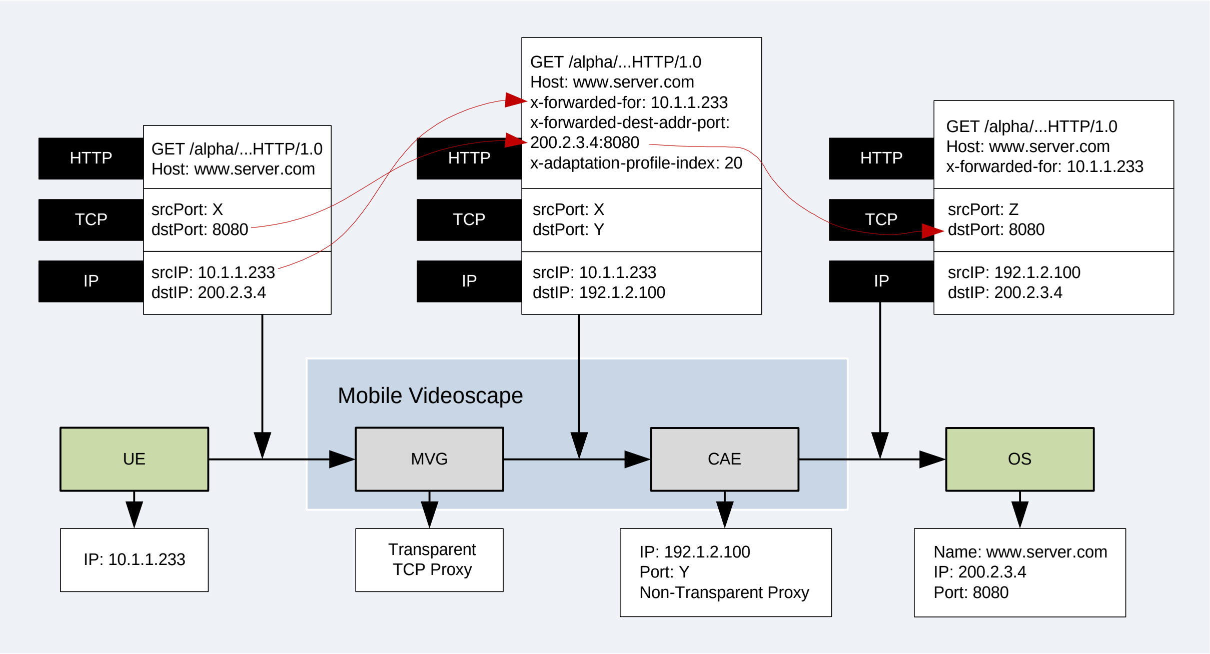
For configuration instructions and a sample configuration, see Chapter 2.

HTTP X-Header Use in Transparent Video Re-addressing

To enable the CAE to reach an OS to retrieve selected video clips for adaptation, the Mobile Video Gateway inserts the Layer 3 destination IP address and Layer 4 destination port number of the OS in a proprietary HTTP x-header in the HTTP video request to the CAE. The CAE uses the information to recreate the Layer 3 and 4 headers to connect to the OS.

The following figure shows how the HTTP x-header is used in transparent video re-addressing to the CAE. In this example, in the original HTTP request from the subscriber UE, the source IP address is 10.1.1.233 and the destination IP address is 200.2.3.4. The destination TCP port is 8080.

Figure 6. HTTP X-Header Use in Transparent Video Re-addressing



Mobile Video Gateway to the CAE

When sending HTTP video requests to the CAE for retrieval of optimized video content, the Mobile Video Gateway inserts the following x-headers:

- **x-forwarded-dest-addr-port:** The IPv4 destination address and TCP port number of the OS.

- **x-adaptation-profile-index:** The index number of the video quality profile for the CAE to use to select the level of video quality for adaptation.

CAE to the OS

When sending HTTP video requests to the OS for video content, the CAE removes the following x-headers:

- **x-forwarded-dest-addr-port:** The IPv4 destination address and TCP port number of the OS.
- **x-adaptation-profile-index:** The index number of the video quality profile for the CAE to use to select the level of video quality for adaptation.

When sending HTTP video requests to the OS for video content, the CAE inserts the following x-header: **x-forwarded-for:** The IPv4 address of the subscriber UE.

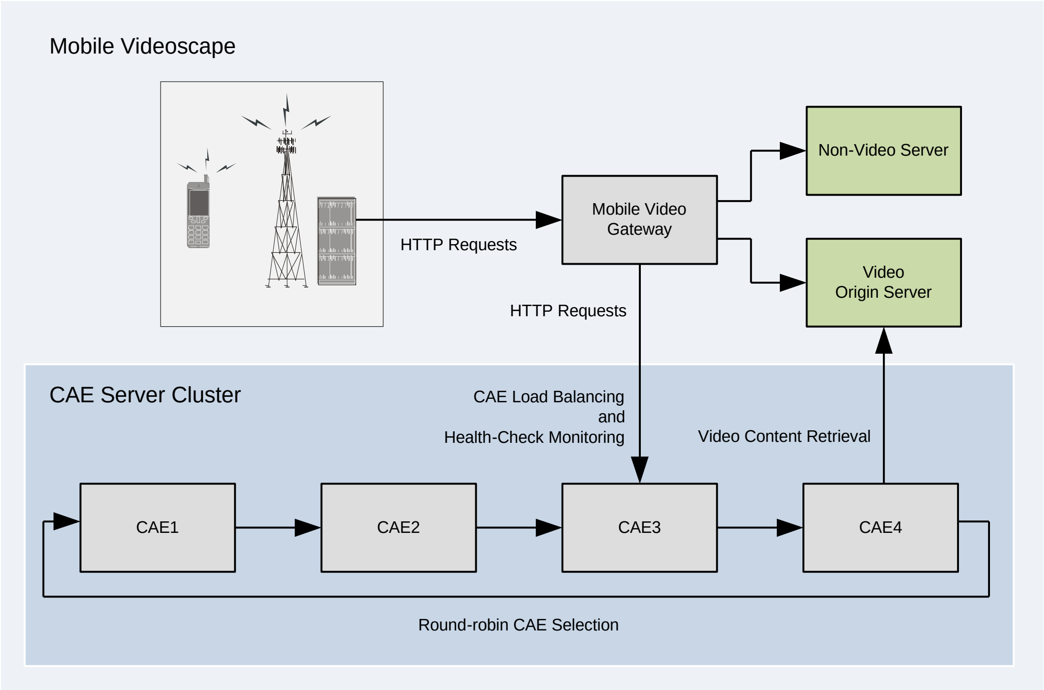
CAE Load Balancing

The optional Cisco CAE runs on the Cisco UCS platform and functions in a UCS server cluster to bring additional optimization capabilities to the Mobile Videoscape. The Mobile Video Gateway interfaces directly with each CAE in the server cluster. The CAE server cluster can serve multiple Mobile Video Gateways simultaneously. In turn, each Mobile Video Gateway is able to support up to 64 CAEs in the server cluster.

Important: The CAE load balancing feature is not fully qualified and is not supported for field deployment. It is available for lab demo/lab trial only.

The following figure shows the CAE in a server cluster.

Figure 7. CAE Server Cluster



The CAE load balancing feature enables the Mobile Video Gateway to distribute HTTP video requests from the subscriber UEs equally among the CAEs in the server cluster.

The CAE load balancing feature is configured and enabled in the context containing the interface to the CAEs, typically the destination context, via system CLI commands. During configuration, each CAE in the server cluster gets defined in a CAE group representing the cluster. Each context on the Mobile Video Gateway can have one and only one CAE group. There can be multiple contexts that contain a CAE group, but there is a system limit of 64 CAEs supported on a Mobile Video Gateway.

In addition to the CAE group configuration above, the CAE load balancing feature gets configured as part of an Active Charging Service, which is a component of the Enhanced Charging Services on the ASR 5000. The feature is configured by creating an Active Charging Service for the Mobile Video Gateway, specifying charging and routing rule definitions, and then creating a charging action for CAE re-addressing, which enables video optimization and CAE load balancing for the CAEs in the CAE group.

For configuration instructions and a sample configuration, see Chapter 2.

CAE Load Balancer Function

When the Mobile Video Gateway identifies a video request during DPI, the CAE load balancer function performs three main operations, as follows:

- It performs CAE load balancing using a round-robin selection of the next available CAE to service the video request.
- It ensures that multiple video flows for a subscriber are serviced by the same CAE once a CAE is selected. This is required for some mobile devices such as the Apple® iPhone®, which can serve video clips using multiple

TCP sessions, such as when an iPhone user skips forward in the middle of playback and the iPhone closes the existing TCP session and starts a new one.

- It maintains health-check monitoring for each of the configured CAEs in the server cluster. If a CAE is currently down, the load balancer function prevents video requests from being sent to the down CAE until is up and available again. All of the CAEs in a CAE group optimize the same video content, so the Mobile Video Gateway can direct the video request to any of the other CAEs until the down CAE is up and available again.

CAE Health-Check Monitoring Function

The CAE health-check monitoring function is part of the CAE load balancing feature. It triggers a health-check request sent to the CAEs based on a configurable keep-alive timer. If a CAE does not respond, and after a configurable number of retries and timeouts, it marks the state of the CAE as Down. It also generates an SNMP Server-State-Down trap message, indicating that the CAE is down and unavailable. When a configurable dead-time timer expires, it sends another health-check request to the down CAE, and if the CAE sends a positive response indicating that it is back up, it marks the state of the CAE as Up and generates an SNMP Server-State-Up trap message, indicating that the CAE is back up and available.

Video Optimization Policy Control

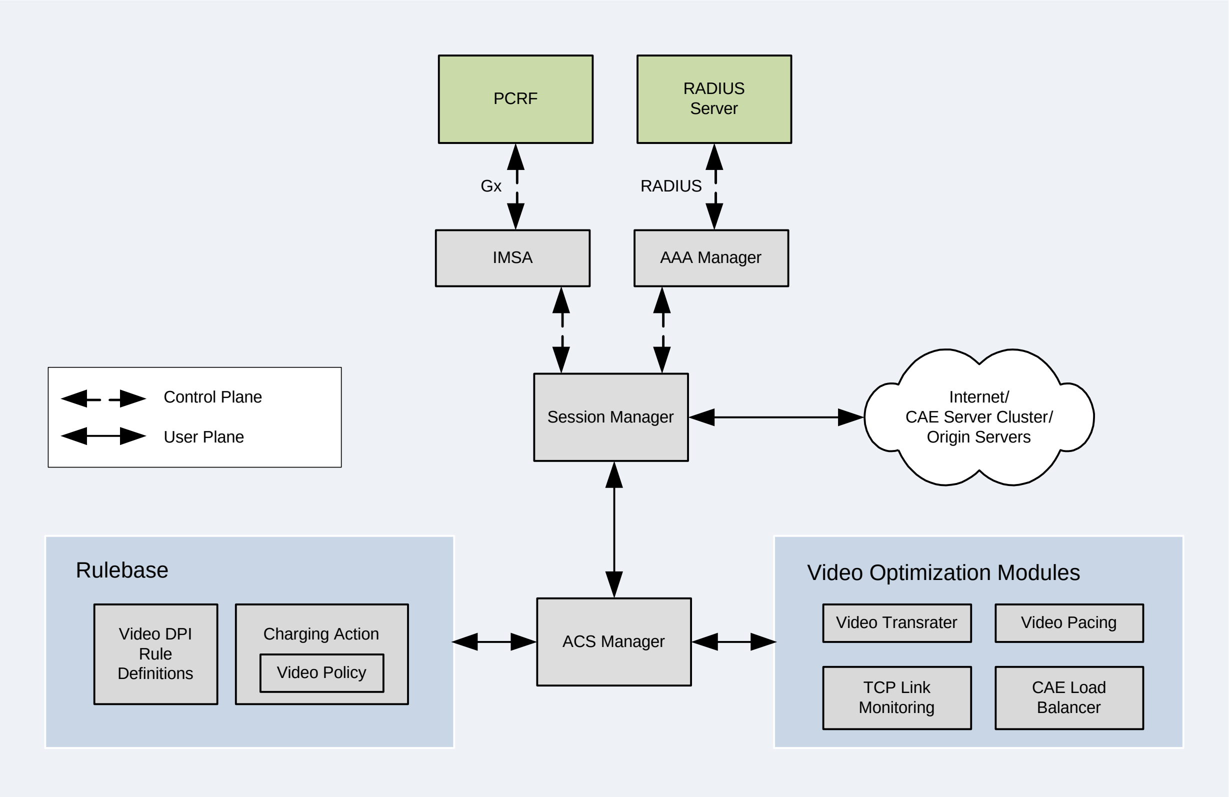
The video optimization policy control feature provides the necessary information for the Mobile Video Gateway to select the highest quality video content for a subscriber, based on information received from a PCRF or RADIUS server, or based on the subscriber's policy profile configured on the Mobile Video Gateway. The feature enables mobile operators to offer tiered video services to their subscribers with different levels of service (Gold, Silver, and Bronze levels, for example).

A video policy defines a subscriber's entitlement to the video content provided by the Mobile Video Gateway. A video policy contains various video-specific attributes, including the subscriber's video QoE (Quality of Experience).

In this software release, the video policy includes a CLI **`charging-action`** command option for specifying a suggested maximum bit rate value for video. This value, specified in bits per second (bps), is used by two of the video optimization modules on the Mobile Video Gateway, the video pacing module and the video transrater module.

The following figure shows the flow of information for the video optimization policy control feature on the Mobile Video Gateway.

Figure 8. Video Optimization Policy Control System Flow



Functional Overview

The video optimization policy control feature assigns a video policy to a subscriber via one of the following methods:

- **PCRF via the Gx interface:** Acting as a RADIUS endpoint, the Mobile Video Gateway can obtain the video policy for a subscriber using the Gx interface to the PCRF. With this method, the Charging-Rule-Name attribute received in the Charging-Rule-Install AVP in the CCA-I message contains a rule definition name that maps to the video policy. This rule definition is part of the rulebase assigned to the subscriber. The Mobile Video Gateway can assign the rulebase to the subscriber through a static configuration at the subscriber or APN level, or obtained from the RADIUS server in an Access-Accept message.

  Alternately, the Mobile Video Gateway can be configured to obtain the rulebase name itself from the PCRF via the Charging-RuleBase AVP.

- **RADIUS Server via the RADIUS interface:** In the absence of a Gx interface, the Mobile Video Gateway can obtain the video policy from the RADIUS server through the Access-Accept message. With this method, the Mobile Video Gateway applies the RuleBase-Name AVP in the Access-Accept message to the subscriber, and one of the rule definitions in the configured rulebase selected in this manner maps to the video policy. Note that one rulebase gets associated with one level of subscriber entitlement (GOLD_RULEBASE, for example).

- **Static assignment at the subscriber or APN level:** The Mobile Video Gateway can assign a video policy by assigning a rulebase at the subscriber or APN level, so that one of the rule definitions in the configured rulebase maps to the video policy. As in the RADIUS server method, one rulebase gets associated with one level of subscriber entitlement.

The video optimization policy control feature gets configured as part of an Active Charging Service, which is a component of the Enhanced Charging Services on the ASR 5000. The feature is configured by creating an Active Charging Service for the Mobile Video Gateway, specifying charging and routing rule definitions, and then creating charging actions for the tiered video service levels. Within each service level charging action, the suggested maximum video bit rate is specified.

During configuration, a rulebase is defined for each subscriber or APN and contains multiple rule definitions. When obtaining the video policy from the PCRF via the Gx interface, and when obtaining the video policy via the Charging-Rule-Install AVP, the Mobile Video Gateway enables a particular rule definition when a rule definition name matches the received Charging-Rule-Name attribute. This is achieved by using the **`dynamic-only`** option in the **`action priority`** command when configuring the rulebases. When obtaining the video policy via the RuleBase-Name AVP, note that there can be one and only one rule definition and its corresponding charging action associated with a video policy.

When a rule definition gets matched, the Mobile Video Gateway applies the corresponding charging action. For example, when the VIDEO_GOLD rule definition is matched, the Mobile Video Gateway applies the corresponding GOLD_CHARGING_ACTION. This charging action determines the video policy for the subscriber. If no rule definitions get matched, the Mobile Video Gateway uses the default value for the suggested maximum bit rate.

For configuration instructions and sample configurations, see Chapter 2. For detailed instructions for configuring the Gx interface on the Cisco P-GW, see the *Packet Data Network Gateway Administration Guide.*

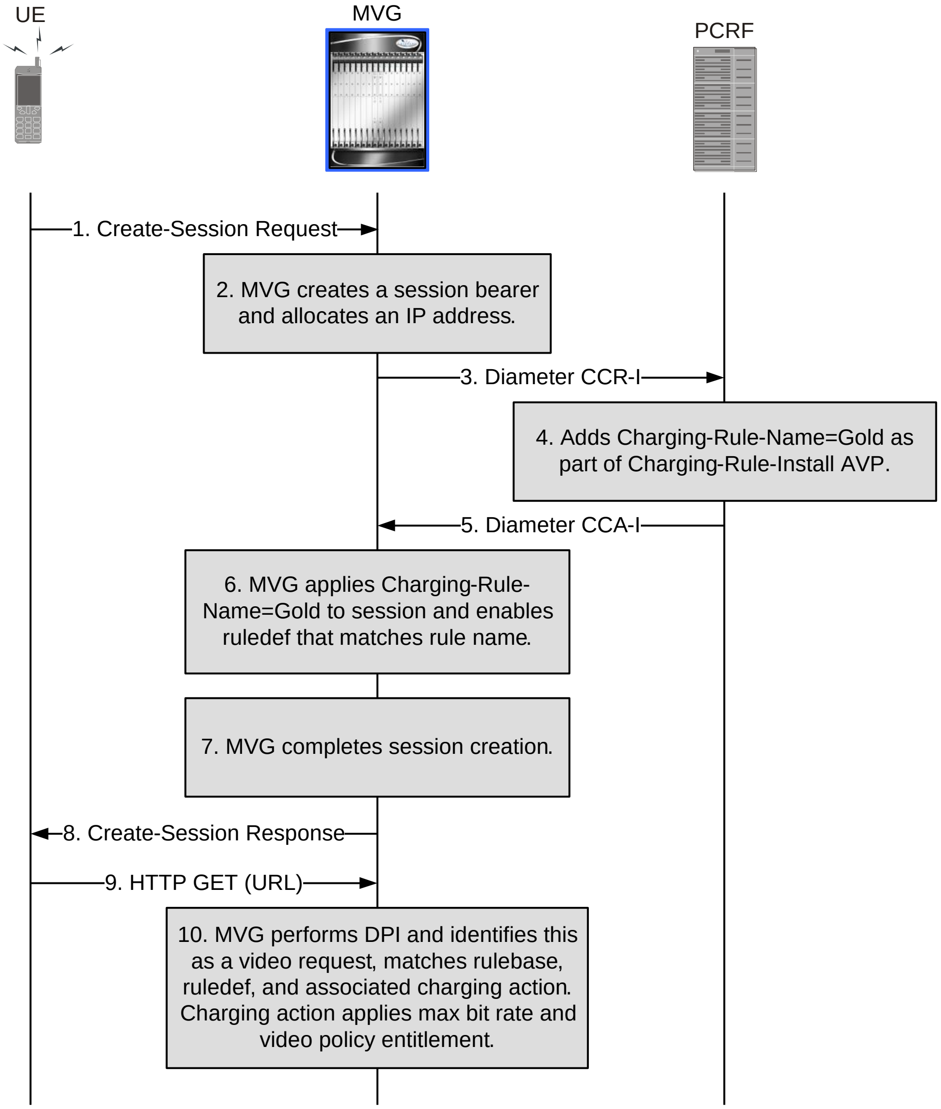
Video Optimization Policy Control Call Flows

This section includes call flows of the Mobile Video Gateway obtaining the video policy for a subscriber in two ways:

- From the PCRF over a Gx interface as it functions as a RADIUS endpoint.
- From the RADIUS server over a RADIUS interface as it functions as a RADIUS proxy.

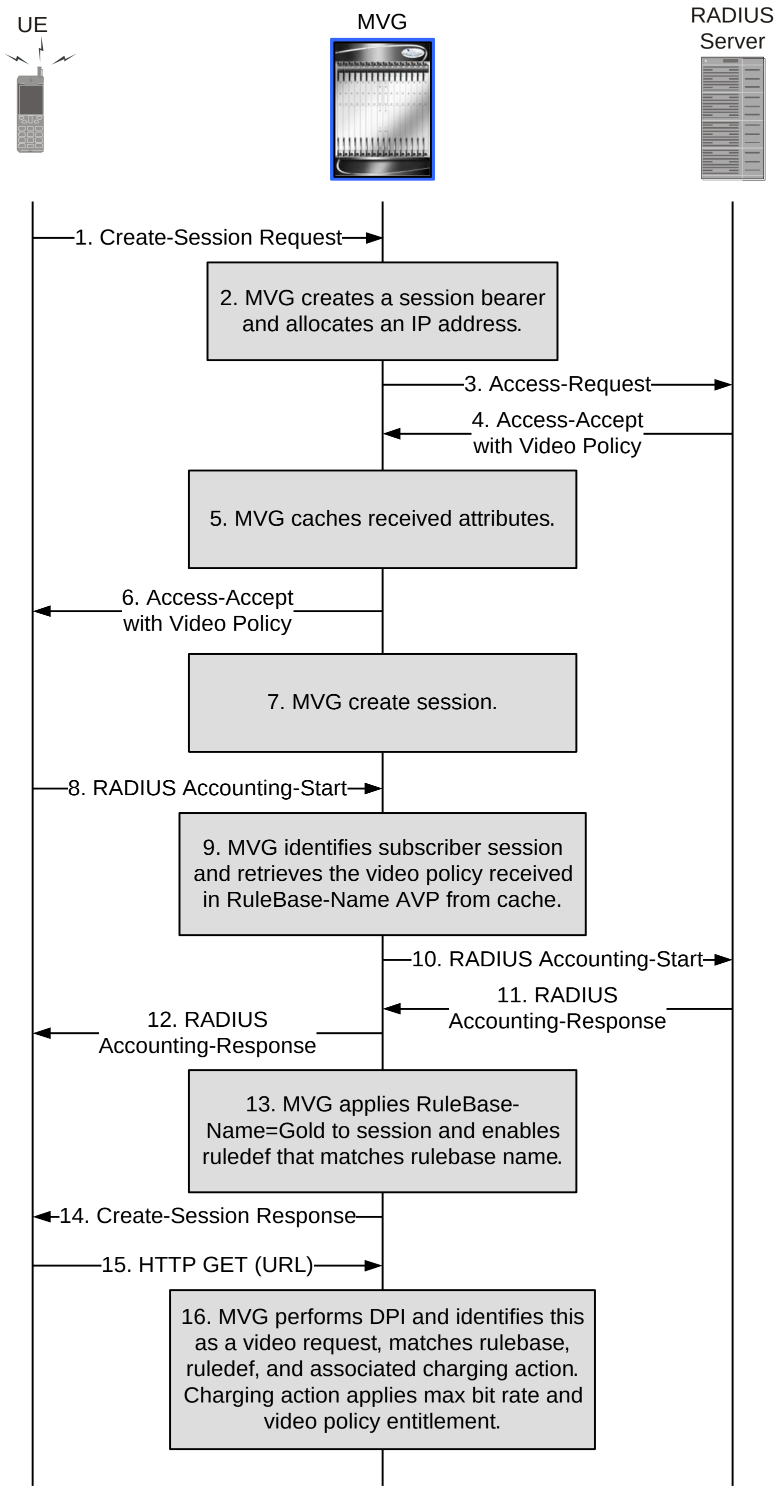
The following figure shows the Mobile Video Gateway functioning as a RADIUS endpoint obtaining the video policy via the PCRF over a Gx interface.

Figure 9. Mobile Video Gateway as a RADIUS Endpoint Obtaining the Video Policy via the PCRF


UE
MVG
PCRF
1. Create-Session Request
2. MVG creates a session bearer and allocates an IP address.
3. Diameter CCR-I
4. Adds Charging-Rule-Name=Gold as part of Charging-Rule-Install AVP.
5. Diameter CCA-I
6. MVG applies Charging-Rule-Name=Gold to session and enables ruledef that matches rule name.
7. MVG completes session creation.
8. Create-Session Response
9. HTTP GET (URL)
10. MVG performs DPI and identifies this as a video request, matches rulebase, ruledef, and associated charging action. Charging action applies max bit rate and video policy entitlement.

The following figure shows the Mobile Video Gateway functioning as a RADIUS proxy obtaining the video policy via the RADIUS server over a RADIUS interface.

Figure 10. Mobile Video Gateway as a RADIUS Proxy Obtaining the Video Policy via the RADIUS Server


UE
MVG
RADIUS Server
1. Create-Session Request
2. MVG creates a session bearer and allocates an IP address.
3. Access-Request
4. Access-Accept with Video Policy
5. MVG caches received attributes.
6. Access-Accept with Video Policy
7. MVG create session.
8. RADIUS Accounting-Start
9. MVG identifies subscriber session and retrieves the video policy received in RuleBase-Name AVP from cache.
10. RADIUS Accounting-Start
11. RADIUS Accounting-Response
12. RADIUS Accounting-Response
13. MVG applies RuleBase-Name=Gold to session and enables ruledef that matches rulebase name.
14. Create-Session Response
15. HTTP GET (URL)
16. MVG performs DPI and identifies this as a video request, matches rulebase, ruledef, and associated charging action. Charging action applies max bit rate and video policy entitlement.

Video White-listing

Certain video clips can be excluded from video optimization. This is referred to as white-listing. The video white-listing feature can either be configured using empty charging actions that match the white-listed URLs, or using DPI rule definitions that do not match the white-listed URLs.

For configuration instructions, see Chapter 2.

Video Pacing

The video pacing feature enables mobile operators to limit the download speed of over-the-top, progressive download video (video clips provided to subscribers via HTTP downloads over TCP flows) so that their subscribers download just enough video content in time for smooth playback. By limiting the bit rate of progressive downloads to the actual encoded bit rate of each video clip, mobile operators can significantly reduce their air interface bandwidth usage.

Important: The video pacing feature has been qualified to run on the ASR 5500 for the Mobile Video Gateway integrated with the Cisco P-GW (Packet Data Network Gateway) and the Cisco HA (Home Agent).

The video pacing feature determines the optimal download speed for a video by calculating the average bit rate of the video and then, after allowing an initial burst to fill a video buffer on the subscriber UE before playback begins, by enforcing the average bit rate for the duration of the video download.

The video pacing feature is an Active Charging Service, which is a component of the Enhanced Charging Services on the ASR 5000. The video pacing feature is configured using the system CLI commands by creating an Active Charging Service for video pacing, and then specifying charging and routing rule definitions.

For configuration instructions and a sample configuration, see Chapter 2.

Video Pacing Operation

The video pacing feature operates as follows:

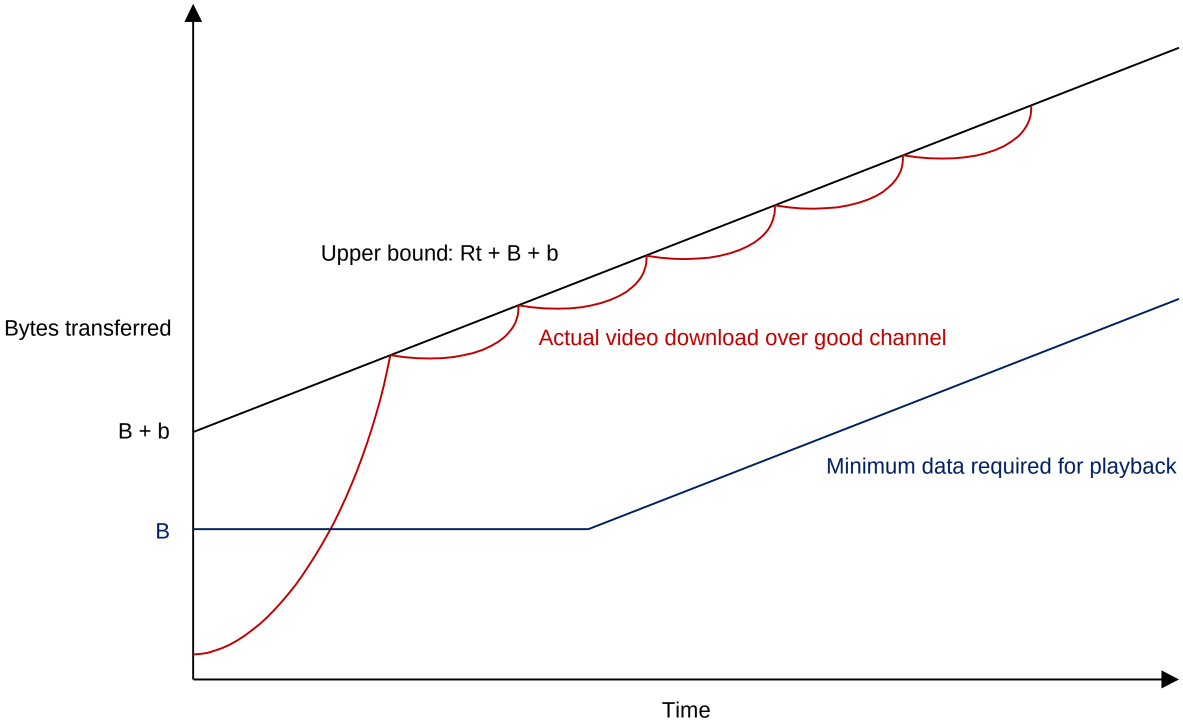
Assume a video-encoding bit rate R and a video playback start time of 0. At time t, the subscriber UE needs to receive Rt bytes of video content just in time for smooth playback. To address fluctuations over the wireless channel, assume that a video buffer is kept on the subscriber UE to accommodate these fluctuations. Assume this buffer size is the standard burst size b.

Because many software media players do not begin playback until a certain amount of video data has been buffered, the video pacing feature allows an initial burst of data, so in addition to the standard burst size b, assume an initial burst size B. This initial burst size is configured based on time duration (as t seconds of video data) and calculated for each video flow based on the determined video bit rate. The video pacing feature allows this initial burst just once, before the video begins playing.

The video pacing feature employs a token bucket algorithm to enforce the permitted video data bytes. When a video download begins, for any given time t, the token bucket algorithm disallows more than (Rt + B + b) data bytes, which is the maximum allowed data bytes. After the initial burst B is completed, the video pacing feature disallows more than (Rt + b) data bytes, and the optimal “just in time” video download rate is achieved.

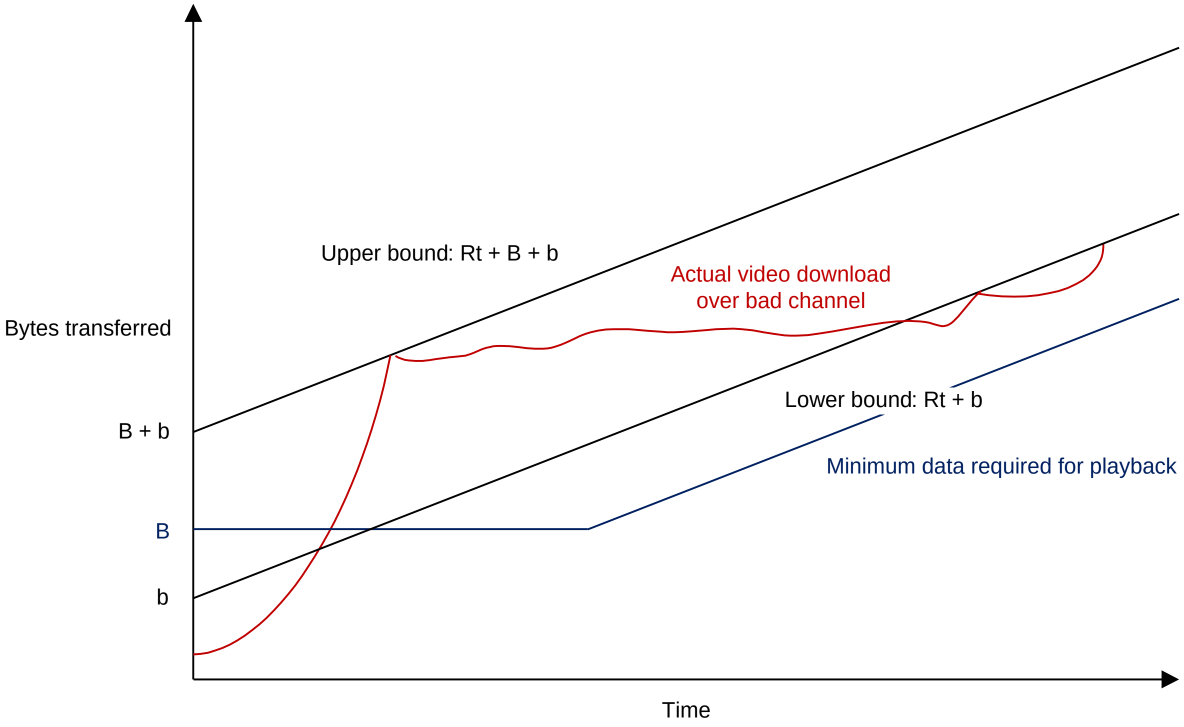
The following figures show video pacing during good and bad channel conditions.

Figure 11. Video Pacing During Good Channel Conditions



In the figure above showing good channel conditions, notice that there is a small difference between the ideal pacing rate (the black line on top) and the actual downloaded video bytes (the red line). This difference is due to network delay, and when the pacing feature begins to take action, the video content OS or Cisco CAE does not respond immediately. Even with this delay, because the video pacing feature allows the standard burst size b, the download rate never falls below the blue line representing the minimum video data required for smooth video playback. Also notice that the media player needs B (not 0) bytes of data for the video to start playing. This is why the video pacing feature allows a bigger initial burst of data (B + b), and then begins enforcing the burst size b until the completion of the download.

Figure 12. Video Pacing During Bad Channel Conditions



In the figure above showing bad channel conditions, when channel conditions worsen, the actual downloaded video bytes cannot keep up with the ideal pacing rate. Nonetheless, if the channel recovers in time, the download rate is still above the blue line representing the minimum data required for smooth playback, and video pacing continues to maintain b bytes of data above this lower limit.

Video Pacing Functions

The video pacing feature includes four main functional components, as follows:

- **Pacing Start Trigger:** The pacing start trigger is part of the Active Charging Service for video pacing. When a rule definition in the Active Charging Service identifies a packet flow as a video flow, and the corresponding charging action for video pacing is enabled, the pacing start trigger invokes video pacing enforcement for the video flow. It sets the video bit rate and initial burst size from the subscriber policy, which is configured for subscribers in the source context as part of the active charging rulebase. It then becomes dormant.

  Some mobile devices such as the Apple iPhone can serve video clips using multiple TCP sessions, such as when an iPhone user skips forward in the middle of playback and the iPhone closes the existing TCP session and starts a new one. When multiple TCP sessions are used to download the same video, the pacing start trigger gets invoked once per video flow, and the video pacing feature correlates these flows to the same video object to continue pacing enforcement from where the last TCP flow left off. When multiple TCP flows are used to download different videos, video pacing is performed independently per flow.

- **Video Pacing Enforcement:** After the initial burst of video content, the video pacing enforcement function sets the optimal video download rate for the incoming downlink packets using a token bucket algorithm. Video pacing occurs based on the settings configured via CLI command options.

- **Video Rate Determination:** The video rate determination function is a software algorithm that examines the initial HTTP RESPONSE packets and video metadata packets to determine the encoded bit rate of the video. It examines the HTTP RESPONSE headers to determine the content length of the video in total bytes as well as the total video playback duration, and then calculates the average video bit rate as: (Content length/Video playback duration). It then triggers the video pacing enforcement function to enforce the new average bit rate when the next downlink packet is received.

- **CLI Command Options:** The video pacing feature includes a set of CLI command options for the Active Charging Service `charging-action` command.

  For a description of these command options, see the *Command Line Interface Reference*.

Video Pacing Call Flows

When the Mobile Video Gateway receives an HTTP GET request from a subscriber UE, it performs DPI to determine whether it is a request for video content. If the Mobile Video Gateway cannot make this determination by inspecting the HTTP GET request, it performs DPI again when it receives the HTTP RESPONSE from the OS.

The following figures show the message flow during inspection for video content and the subsequent triggering of video pacing functions. The first figure shows the identification of a video request from an HTTP GET request, the second shows the identification of a video request from an HTTP RESPONSE.

Figure 13. DPI of HTTP GET Identifying a Video Request

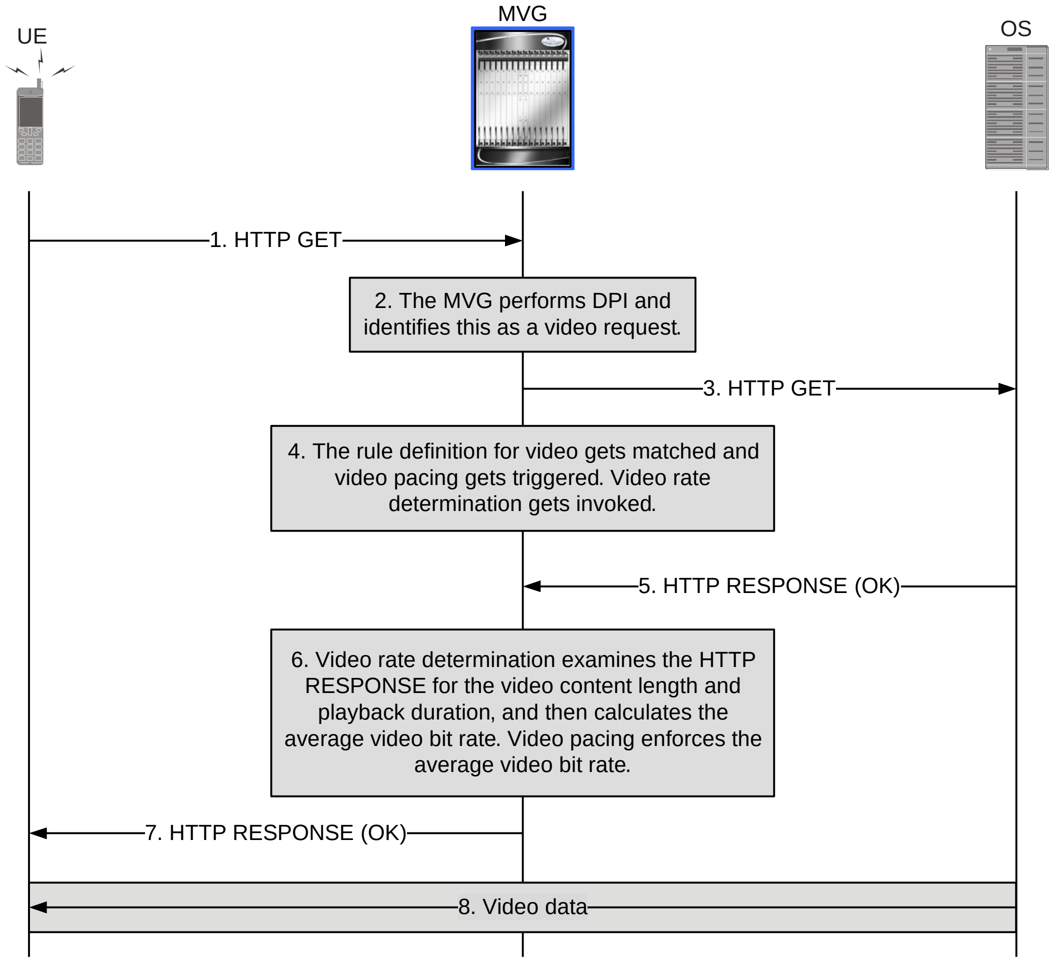
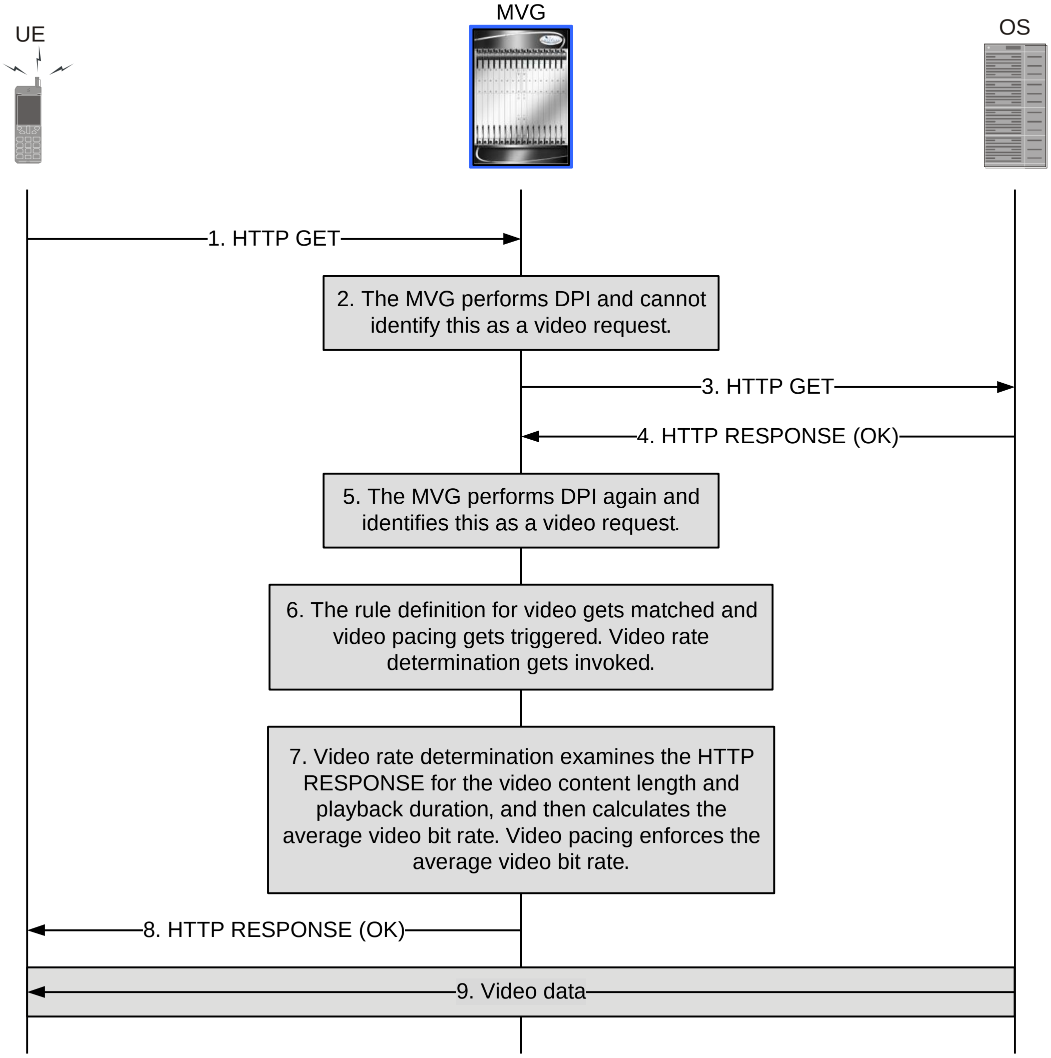

MVG
UE
OS
1. HTTP GET
2. The MVG performs DPI and identifies this as a video request.
3. HTTP GET
4. The rule definition for video gets matched and video pacing gets triggered. Video rate determination gets invoked.
5. HTTP RESPONSE (OK)
6. Video rate determination examines the HTTP RESPONSE for the video content length and playback duration, and then calculates the average video bit rate. Video pacing enforces the average video bit rate.
7. HTTP RESPONSE (OK)
8. Video data

Figure 14. DPI of HTTP RESPONSE Identifying a Video Request


MVG
UE
OS
1. HTTP GET
2. The MVG performs DPI and cannot identify this as a video request.
3. HTTP GET
4. HTTP RESPONSE (OK)
5. The MVG performs DPI again and identifies this as a video request.
6. The rule definition for video gets matched and video pacing gets triggered. Video rate determination gets invoked.
7. Video rate determination examines the HTTP RESPONSE for the video content length and playback duration, and then calculates the average video bit rate. Video pacing enforces the average video bit rate.
8. HTTP RESPONSE (OK)
9. Video data

Interactions with Related Functions

The video pacing feature is designed to work with related functional components as follows:

- **Video Pacing and the CAE:** The video pacing feature is an independent software module and has no interface with the Cisco CAE. It performs its function in the same way whether a video is downloaded from the OS or from the CAE. The CAE is an optional component of the Cisco Mobile Videoscape.
- **Video Pacing and the TCP Proxy:** The video pacing feature can be configured to work with or without the TCP proxy feature with no change in its function.
- **Video Pacing and Traffic Performance Optimization:** The traffic performance optimization feature works over the interface on the client side of the TCP proxy. It handles re-transmission, TCP window size adjustment, and so on. Video pacing works over the interface on the video server side of the TCP proxy, and works independent of traffic performance optimization.

Supported Video Container File Formats

In this software release, the video pacing feature supports the following standard video container file formats:

- MP4 File Format
- FLV Files

MP4 follows the ISO Base Media File Format (MPEG-4 Part 12). We provide comprehensive support for progressive download of .FLV files, playable in Adobe® Flash® Player.

TCP Link Monitoring

TCP is the dominant transport protocol for the majority of Internet traffic, including video. For mobile networks, the available transport bandwidth can fluctuate depending on changing conditions over the wireless connections. Knowledge of the available transport bandwidth is especially important for video over mobile networks, since this bandwidth affects video delivery rates, video encoding and compression techniques, and ultimately the video playback experience of the subscribers.

The TCP link monitoring feature adds the capability to enable monitoring and logging of TCP behavior towards the subscriber UEs. Monitoring TCP behavior enables the Mobile Video Gateway to estimate transient bandwidth and identify network congestion for all TCP connections toward the clients on the subscriber UEs.

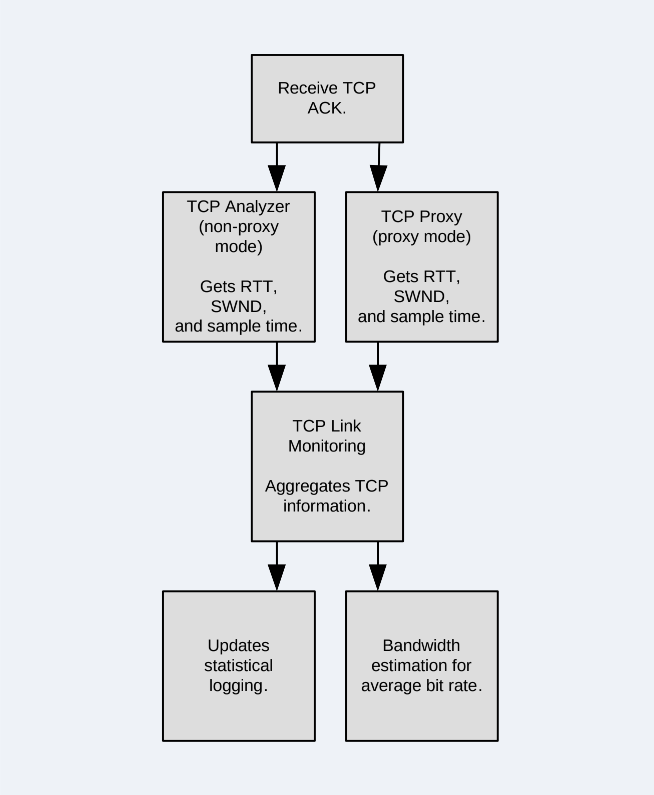
The Mobile Video Gateway services two types of TCP connections. A TCP connection can either pass through the Mobile Video Gateway intact or can be split into two connections by the TCP proxy. For the downlink data towards the subscriber UEs, the TCP link monitoring feature invokes its bandwidth estimation and statistical logging functions, which are enabled for both proxy and non-proxy modes.

TCP link monitoring statistics are gathered on a system-wide basis. This information can be periodically exported to a collection server as bulk statistics, upon which post-processing can be performed.

TCP Link Monitoring System Flow

The following figure shows the flow of information to and from the TCP link monitoring module on the Mobile Video Gateway.

Figure 15. TCP Link Monitoring System Flow


Receive TCP ACK.
TCP Analyzer (non-proxy mode)
Gets RTT, SWND, and sample time.
TCP Proxy (proxy mode)
Gets RTT, SWND, and sample time.
TCP Link Monitoring
Aggregates TCP information.
Updates statistical logging.
Bandwidth estimation for average bit rate.

The TCP link monitoring feature calculates the RTT (Round Trip Time) and estimates the link bandwidth based on the downlink data sent towards the UE and the current congestion conditions. It then collects this information at the system level to report to the bulk statistics collection server.

Note that the throughput calculation for the TCP link excludes duplicate, out-of-order, and retransmitted packets.

Functional Overview

The key functions of the TCP link monitoring feature are bandwidth estimation and system-level TCP statistical logging.

Bandwidth Estimation

Because mobile devices are served by a variety of TCP variants, either from the OS or from the Mobile Video Gateway’s TCP proxy, the TCP link monitoring feature employs an independent bandwidth estimation technique proposed by TCP Westwood+ (see “Performance Evaluation of Westwood+ TCP Congestion Control” by Mascolo, et al).

Westwood+ estimates bandwidth by calculating the ratio of the number of bytes of acknowledged TCP payload over every RTT. This rate sample is then filtered by a weighted moving average to derive a per-flow average bandwidth estimate for every RTT interval.

Statistical Logging

Statistical logging of TCP traffic supports two types of plots: histogram and time-series.

For histogram logging, the TCP link monitoring feature keeps a counter for every bit rate or RTT range. Whenever a new sample of TCP traffic is generated, a corresponding counter is updated. The collection server retrieves these values based on the configured sampling rate. There are four histogram plots: video bit rate, video RTT, non-video bit rate, and non-video RTT. For each of these plots, a total of 36 counters are used for logging.

For time-series logging, the sampling rate is the same as that of the remote update time for the collection server. Typically, this can be configured in 30-minute intervals. As with histogram logging, there are four time-series counters: video bit rate, video RTT, non-video bit rate, and non-video RTT.

Dynamically-enabled TCP Proxy

The Mobile Video Gateway can act as a dynamically-enabled TCP proxy that provides the following functions:

- Transparent video re-addressing
- Traffic performance optimization

Note that these features require the TCP proxy to function as expected.

The TCP proxy can be dynamically enabled based on Active Charging Service rule definitions. For information about the dynamically-enabled TCP proxy, including configuration instructions, see the *Enhanced Charging Services Administration Guide.*

Traffic Performance Optimization

The Mobile Video Gateway can use traffic performance optimization to improve latency and accelerate the delivery of video content, especially when network congestion and packet drops are present. The feature runs on the dynamically-enabled TCP proxy and can be enabled statically based on the subscriber profile or dynamically based on a DPI match in a charging action.

For information about traffic performance optimization, including configuration instructions, see the *Traffic Performance Optimization Administration Guide.*

N+1 Stateful Redundancy

In the telecommunications industry, over 90 percent of all equipment failures are software-related. With robust hardware failover and redundancy protection, any card-level hardware failures on the system can quickly be corrected. However, software failures can occur for numerous reasons, many times without prior indication.

This software release supports N+1 stateful redundancy for mobile video sessions. N+1 stateful redundancy provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system, preventing fully-connected subscriber sessions from being disconnected. Sessions are maintained over a software failure of a process or hardware failure.

This is an existing feature of the ASR 5000. Note that Layer 4 flows will not be maintained across switch-overs.

Threshold Crossing Alerts

Thresholding on the system is used to monitor the system for conditions that could potentially cause errors or outages. Typically, these conditions are temporary (high CPU utilization or packet collisions on a network, for example) and are quickly resolved. However, continuous or large numbers of these error conditions within a specific time interval may be indicative of larger, more severe issues. The purpose of thresholding is to help identify potentially severe conditions so that immediate action can be taken to minimize and/or avoid system downtime.

The system supports threshold crossing alerts for certain key resources such as CPU, memory, IP pool addresses, etc. With this capability, the operator can configure a threshold on these resources whereby, should the resource depletion cross the configured threshold, an SNMP trap would be sent.

The following thresholding models are supported by the system:

- **Alert:** A value is monitored and an alert condition occurs when the value reaches or exceeds the configured high threshold within the specified polling interval. The alert is generated, then generated and/or sent again at the end of the polling interval.
- **Alarm:** Both high and low threshold are defined for a value. An alarm condition occurs when the value reaches or exceeds the configured high threshold within the specified polling interval. The alert is generated, then generated and/or sent again at the end of the polling interval.

Thresholding reports conditions using one of the following mechanisms:

- **SNMP traps:** SNMP traps have been created that indicate the condition (high threshold crossing and/or clear) of each of the monitored values. Generation of specific traps can be enabled or disabled on the chassis, ensuring that only important faults get displayed. SNMP traps are supported in both Alert and Alarm modes.
- **Logs:** The system provides a facility for which active and event logs can be generated. As with other system facilities, logs are generated messages pertaining to the condition of a monitored value and are generated with a severity level of WARNING. Logs are supported in both the Alert and the Alarm models.
- **Alarm System:** High threshold alarms generated within the specified polling interval are considered outstanding until a condition no longer exists or a condition clear alarm is generated. Outstanding alarms are reported to the system's alarm subsystem and are viewable through the Alarm Management menu in the Web Element Manager.

For more information about threshold crossing alert configuration, see the *Thresholding Configuration Guide.*

Mobile Video Statistics

The mobile video statistics feature enables mobile operators to collect detailed statistics on mobile video usage to understand how subscribers behave when viewing video content, how much network resources are consumed by video, and what trends develop as video use cases evolve. The mobile video statistics feature collects important statistical data for video and presents this information in three ways: per user device type, per radio access type, and per video container type. With this information, operators can better understand evolving trends in their network and further adapt and fine tune their video optimization solution accordingly.

In this software release, the identification of a video flow is dependent on charging actions defined within the corresponding Active Charging Service. When a flow matches a rule definition for video during DPI, the mobile video statistics feature begins collecting the following statistics for the video flow:

- **Total size of the video file (the HTTP content length):** This is the size given in the HTTP RESPONSE header for the video file, represented in bytes.

- **Total duration of the video clip:** This is the video play duration identified from the video metadata, represented in seconds. If the mobile video statistics feature cannot get this information from the metadata (due to non-standard metadata formatting, etc.), this field shows 0.
- **Total bytes sent to the UE:** This is the payload data bytes (excluding TCP/IP headers) permitted to be sent towards the UE. Note that this counter includes end-to-end (TCP) retransmissions.
- **Total duration that the video object is on:** This is the time it takes for the UE to finish downloading the video, which is from the creation of the first flow to the deletion of the last flow comprising this video.
- **Total number of TCP flows used to download the video:** The total count of TCP sessions used for this video object.

The mobile video statistics feature also derives the following information from the statistics above:

- **Video delivery rate:** Total bytes sent to the UE/Total duration that the video object is on. This is the average bit rate of the video payload bytes being delivered to the UE, represented in bps.
- **Percentage of video download:** Total bytes sent to the UE/Total size of the video file. This is the percentage of the video file that the user actually downloaded. The number reflects whether users tend to watch the entire video or only a small part of it. Note that since "Total bytes sent to the UE" includes retransmissions, this number can be larger than 100%.
- **Video encoding bit rate:** Total size of the video file/Total duration of the video clip. This is the average video encoding bit rate, represented in bps.

The feature collects the information above per video object, in which each video object is defined by a unique URI. When multiple HTTP flows can be used to obtain one video object, as with Apple iOS® devices, the feature combines these flows when collecting statistics and treats them as one video object. The statistics are then aggregated per ACS manager and at the Global system level. This aggregation occurs using the following operations:

- For the first five statistics described above, when each video object terminates, the numbers are added to the aggregator at the ACS manager level. Aggregation among ACS managers happens when triggered by CLI commands or when bulk statistics are generated.
- The three derived statistics are calculated using the first five statistics after aggregation at the ACS manager level and the Global system level.

During aggregation, the mobile video statistics feature categorizes the information above based on UE device type, radio access type, and video container type, as follows:

- **UE device type:** Apple iOS devices (iPhone, iPad®, and iPod®), Android™ devices, laptops, and other devices.
- **Radio access type:** 2G, 3G, 4G-LTE, CDMA, HSPA, WLAN, and other types.
- **Video container type:** flv/f4v, mp4 (includes related types such as m4v, 3gp, 3g2, and mov), and other types.

The statistics include the total video object count for each of these categories, which is the total number of video files downloaded for a particular category.

The feature maintains two statistical arrays. The first array is arranged per UE device type, per radio access type. The second array is arranged per UE device type, per video container type.

For configuration instructions, see Chapter 2. For information about the variables in the MVS schema, see the *Statistics and Counters Reference.*

Bulk Statistics for Mobile Video

Bulk statistics on the ASR 5000 allow operators to choose to view not only statistics that are of importance to them, but to also configure the format in which they are presented. This simplifies the post-processing of statistical data, since it can be formatted to be parsed by external, back-end processors.

The system can be configured to collect bulk statistics and send them to a collection server called a receiver. Bulk statistics are statistics that are collected in a group. The individual statistics are grouped by schema. The following is a partial list of supported schemas:

- **System:** Provides system-level statistics.
- **Card:** Provides card-level statistics.
- **Port:** Provides port-level statistics.
- **MVS:** Provides statistics to support the Mobile Videoscape (MVS).

The system supports the configuration of up to four sets (primary/secondary) of receivers. Each set can be configured to collect specific sets of statistics from the various schemas. Statistics can be pulled manually from the system or sent at configured intervals. The bulk statistics are stored on the receiver(s) in files.

The format of the bulk statistic data files can be configured by the user. Users can specify the format of the file name, file headers, and/or footers to include information such as the date, system host name, system uptime, the IP address of the system generating the statistics (available for headers and footers only), and/or the time that the file was generated.

When the Web Element Manager is used as the receiver, it is capable of further processing the statistics data through XML parsing, archiving, and graphing.

The Bulk Statistics Server component of the Web Element Manager parses collected statistics and stores the information in the PostgreSQL database. If XML file generation and transfer is required, this element generates the XML output and can send it to a northbound NMS or an alternate bulk statistics server for further processing.

Additionally, if archiving of the collected statistics is desired, the Bulk Statistics Server writes the files to an alternative directory on the server. A specific directory can be configured by the administrative user or the default directory can be used. Regardless, the directory can be on a local file system or on an NFS-mounted file system on the Web Element Manager server.

For configuration instructions, see Chapter 2.

How the Mobile Video Gateway Works

This section shows how the Mobile Video Gateway works during DPI in a number of call scenarios, including scenarios involving the Mobile Video Gateway with the CAE and the Mobile Video Gateway without the CAE.

Mobile Video Gateway with the Content Adaptation Engine

This section shows call scenarios involving the Mobile Video Gateway with the Content Adaptation Engine.

DPI of HTTP GET Request Identifying a Non-Video Request (MVG with the CAE)

When the Mobile Video Gateway receives an HTTP GET request from a subscriber UE, it performs DPI to determine whether it is a request for video content. The figure below shows the Mobile Video Gateway with the CAE performing DPI on an HTTP GET request and identifying it as a non-video request. The table that follows the figure describes each step in the message flow.

Figure 16. DPI of HTTP GET Request Identifying a Non-Video Request

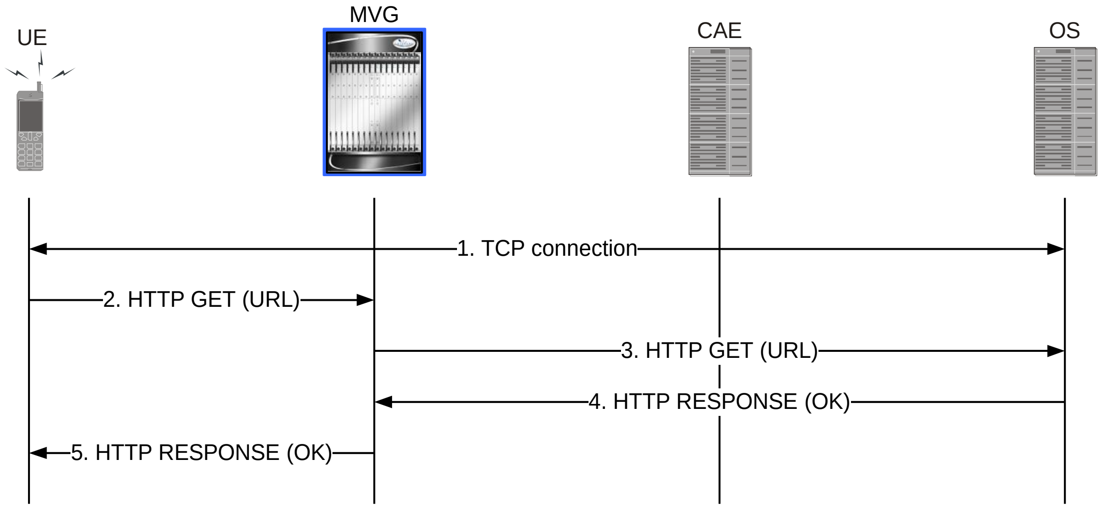


Table 2. DPI of HTTP GET Request Identifying a Non-Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a non-video request (the DPI on GET/POST fails).
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.
4.	The OS responds with an HTTP 200 OK, including the content of the page.

Step	Description
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.

DPI of HTTP RESPONSE Identifying a Non-Video Request (MVG with the CAE)

When the Mobile Video Gateway cannot determine whether an HTTP GET request is a request for video content during DPI, it performs DPI again when it receives the HTTP RESPONSE from the OS. The figure below shows the Mobile Video Gateway with the CAE performing DPI on an HTTP RESPONSE and identifying it as a response to a non-video request. The table that follows the figure describes each step in the message flow.

Figure 17. DPI of HTTP RESPONSE Identifying a Non-Video Request

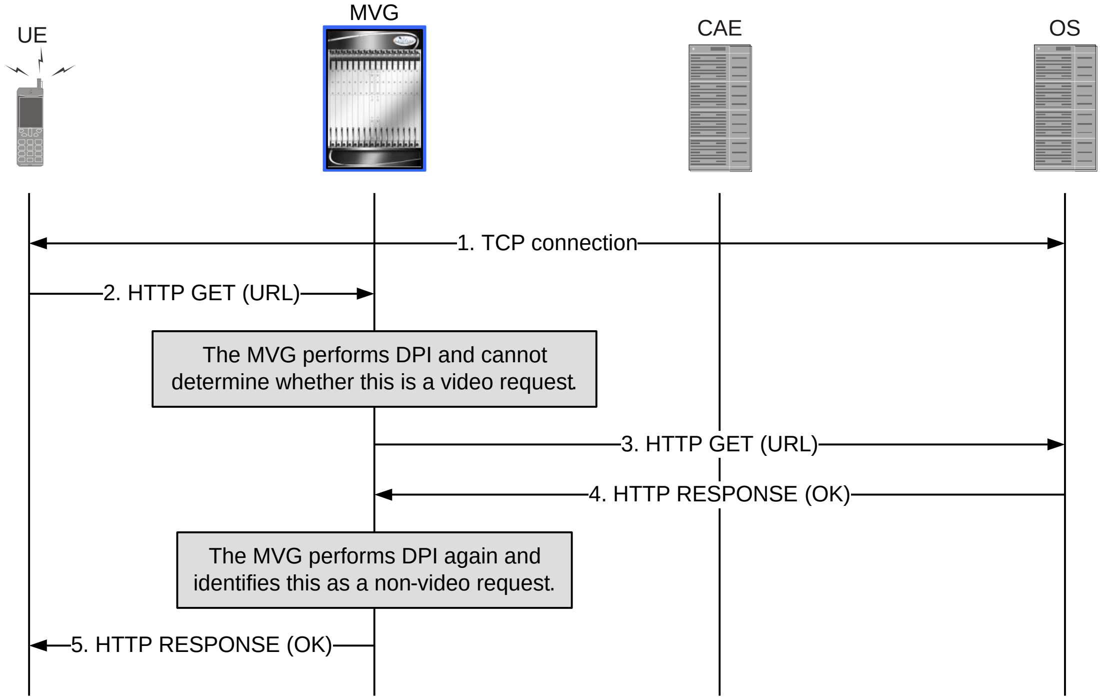

UE
MVG
CAE
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and cannot determine whether this is a video request.
3. HTTP GET (URL)
4. HTTP RESPONSE (OK)
The MVG performs DPI again and identifies this as a non-video request.
5. HTTP RESPONSE (OK)

Table 3. DPI of HTTP RESPONSE Identifying a Non-Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.

Step	Description
4.	The OS responds with an HTTP 200 OK, including the content of the page. The Mobile Video Gateway performs DPI again and identifies it as a response to a non-video request (the DPI on the RESPONSE headers fails).
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.

DPI of HTTP GET Request Identifying a Video Request (MVG with the CAE)

When the Mobile Video Gateway receives an HTTP GET request from a UE, it performs DPI to determine whether it is a request for video content. The figure below shows the Mobile Video Gateway with the CAE performing DPI on an HTTP GET request and identifying it as a video request. The table that follows the figure describes each step in the message flow.

Figure 18. DPI of HTTP GET Request Identifying a Video Request

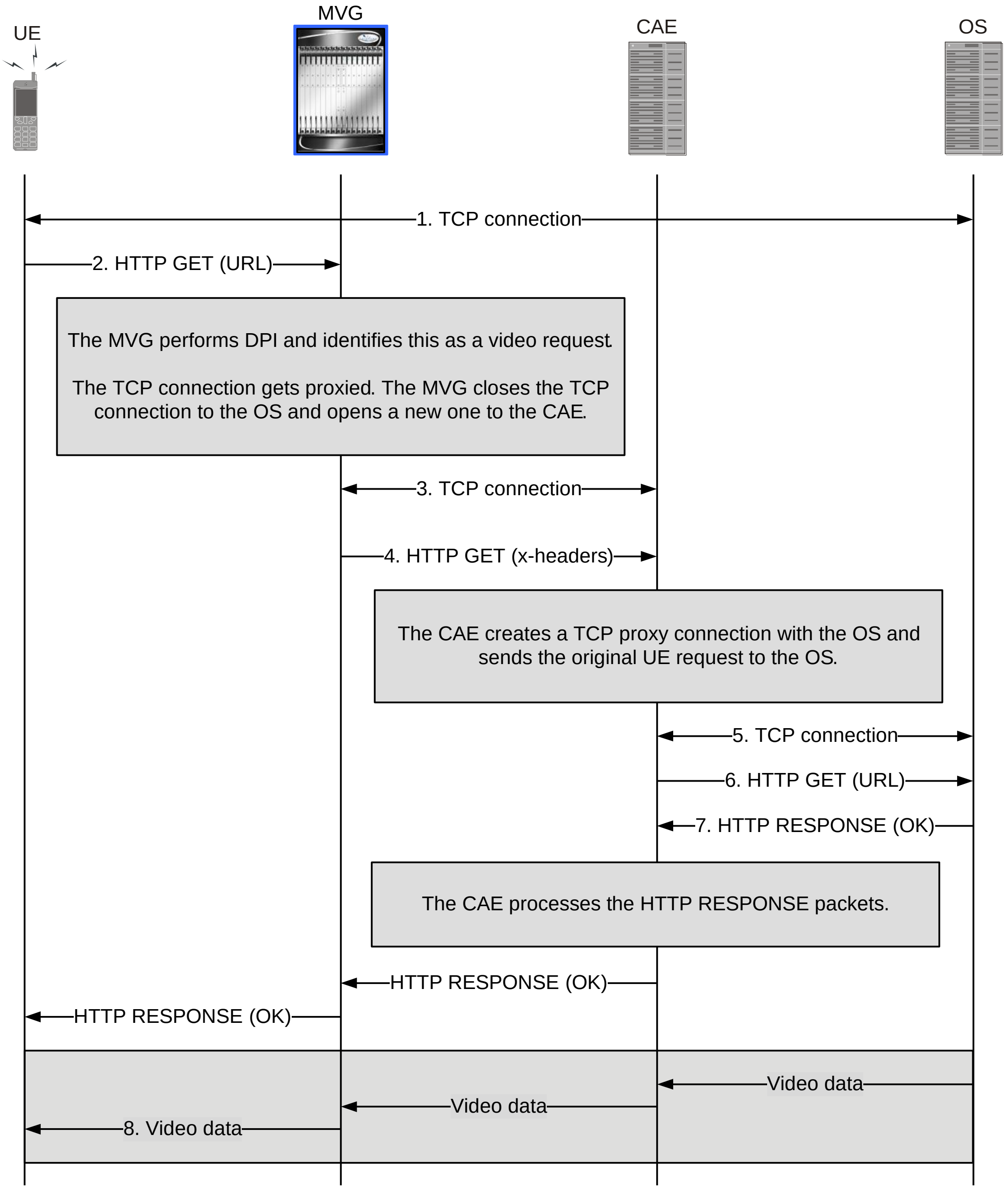

UE
MVG
CAE
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and identifies this as a video request.
The TCP connection gets proxied. The MVG closes the TCP connection to the OS and opens a new one to the CAE.
3. TCP connection
4. HTTP GET (x-headers)
The CAE creates a TCP proxy connection with the OS and sends the original UE request to the OS.
5. TCP connection
6. HTTP GET (URL)
7. HTTP RESPONSE (OK)
The CAE processes the HTTP RESPONSE packets.
HTTP RESPONSE (OK)
HTTP RESPONSE (OK)
Video data
Video data
8. Video data

Table 4. DPI of HTTP GET Request Identifying a Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a video request (the DPI on GET/POST succeeds).

Step	Description
3.	The TCP connection gets proxied. The Mobile Video Gateway closes the TCP connection with the OS and opens a new one with the CAE.
4.	The Mobile Video Gateway sends the original HTTP GET request to the CAE with x-headers for transport, quality, and UE identity.
5.	The CAE creates a TCP proxy connection with the OS.
6.	The CAE sends the original HTTP GET request from the UE to the OS.
7.	The CAE processes the HTTP RESPONSE packets from the OS and performs video optimization.
8.	The Mobile Video Gateway performs additional video optimization and sends the optimized packets to the UE.

DPI of HTTP RESPONSE Identifying a Video Request (MVG with the CAE)

When the Mobile Video Gateway cannot determine whether an HTTP GET request is a request for video content during DPI, it performs DPI again when it receives the HTTP RESPONSE from the OS. The figure below shows the Mobile Video Gateway with the CAE performing DPI on an HTTP RESPONSE and identifying it as a response to a video request. The table that follows the figure describes each step in the message flow.

Figure 19. DPI of HTTP RESPONSE Identifying a Video Request

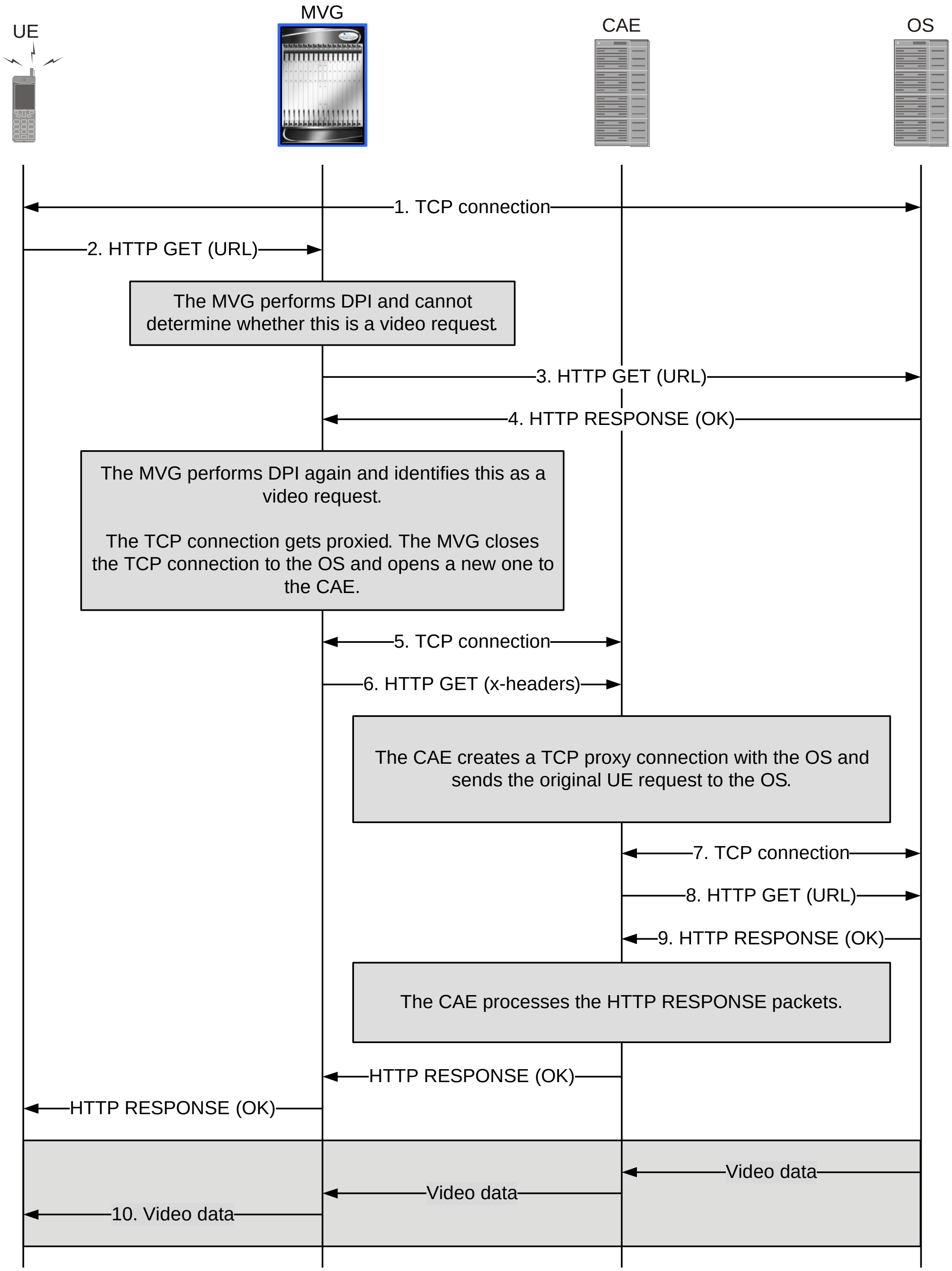

UE
MVG
CAE
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and cannot determine whether this is a video request.
3. HTTP GET (URL)
4. HTTP RESPONSE (OK)
The MVG performs DPI again and identifies this as a video request.
The TCP connection gets proxied. The MVG closes the TCP connection to the OS and opens a new one to the CAE.
5. TCP connection
6. HTTP GET (x-headers)
The CAE creates a TCP proxy connection with the OS and sends the original UE request to the OS.
7. TCP connection
8. HTTP GET (URL)
9. HTTP RESPONSE (OK)
The CAE processes the HTTP RESPONSE packets.
HTTP RESPONSE (OK)
HTTP RESPONSE (OK)
Video data
Video data
10. Video data

Table 5. DPI of HTTP RESPONSE Identifying a Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.
4.	The OS responds with an HTTP 200 OK. The Mobile Video Gateway performs DPI again and identifies it as a response to a video request (the DPI on the RESPONSE headers succeeds).
5.	The TCP connection gets proxied. The Mobile Video Gateway closes the TCP connection with the OS and opens a new one with the CAE.
6.	The Mobile Video Gateway sends the original HTTP GET request to the CAE with x-headers for transport, quality, and UE identity.
7.	The CAE creates a TCP proxy connection with the OS.
8.	The CAE sends the original HTTP GET request from the UE to the OS.
9.	The CAE processes the HTTP RESPONSE packets from the OS and performs video optimization.
10.	The Mobile Video Gateway performs additional video optimization and sends the optimized packets to the UE.

Mobile Video Gateway without the Content Adaptation Engine

This section shows call scenarios involving a Mobile Video Gateway without the Content Adaptation Engine.

DPI of HTTP GET Request Identifying a Non-Video Request (MVG without the CAE)

When the Mobile Video Gateway receives an HTTP GET request from a UE, it performs DPI to determine whether it is a request for video content. The figure below shows the Mobile Video Gateway performing DPI on an HTTP GET request and identifying it as a non-video request. The table that follows the figure describes each step in the message flow.

Figure 20. DPI of HTTP GET Request Identifying a Non-Video Request

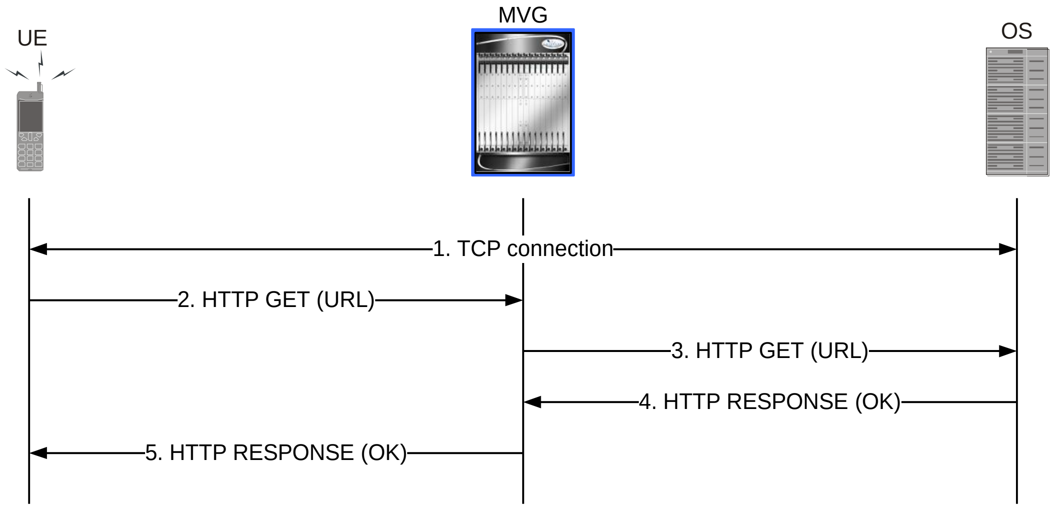


Table 6. DPI of HTTP GET Request Identifying a Non-Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a non-video request (the DPI on GET/POST fails).
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.
4.	The OS responds with an HTTP 200 OK, including the content of the page.
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.

DPI of HTTP RESPONSE Identifying a Non-Video Request (MVG without the CAE)

When the Mobile Video Gateway cannot determine whether an HTTP GET request is a request for video content during DPI, it performs DPI again when it receives the HTTP RESPONSE from the OS. The figure below shows the Mobile Video Gateway performing DPI on an HTTP RESPONSE and identifying it as a response to a non-video request. The table that follows the figure describes each step in the message flow.

Figure 21. DPI of HTTP RESPONSE Identifying a Non-Video Request

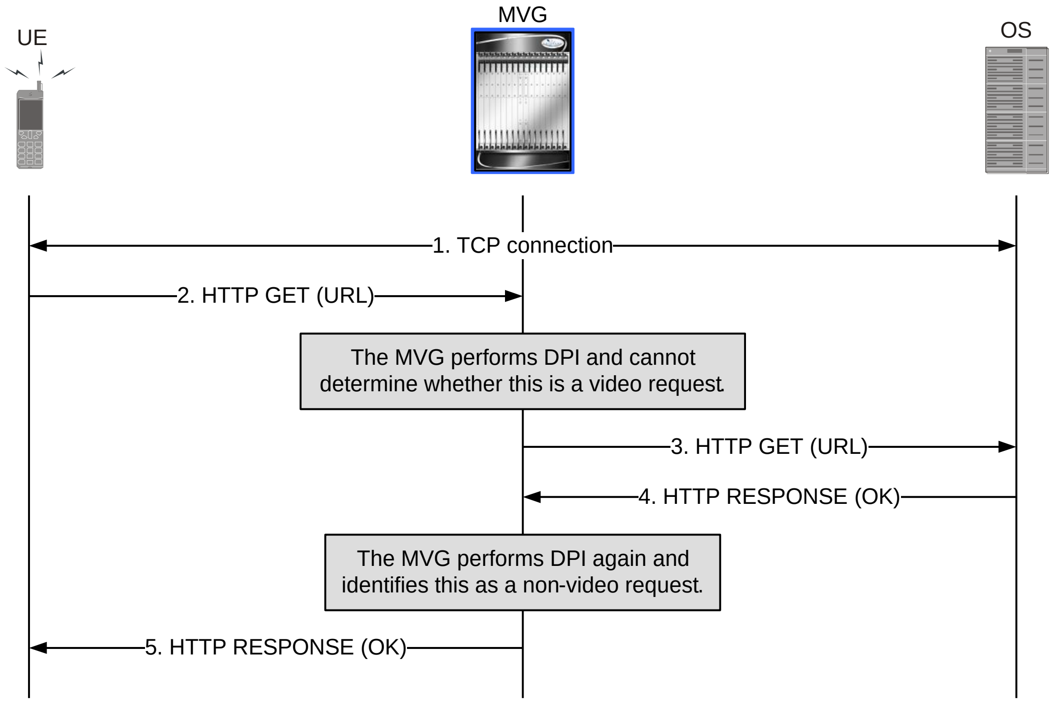

UE
MVG
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and cannot determine whether this is a video request.
3. HTTP GET (URL)
4. HTTP RESPONSE (OK)
The MVG performs DPI again and identifies this as a non-video request.
5. HTTP RESPONSE (OK)

Table 7. DPI of HTTP RESPONSE Identifying a Non-Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.
4.	The OS responds with an HTTP 200 OK, including the content of the page. The Mobile Video Gateway performs DPI again and identifies it as a response to a non-video request (the DPI on the RESPONSE headers fails).
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.

DPI of HTTP GET Request Identifying a Video Request (MVG without the CAE)

When the Mobile Video Gateway receives an HTTP GET request from a UE, it performs DPI to determine whether it is a request for video content. The figure below shows the Mobile Video Gateway performing DPI on an HTTP GET request and identifying it as a video request. The table that follows the figure describes each step in the message flow.

Figure 22. DPI of HTTP GET Request Identifying a Video Request

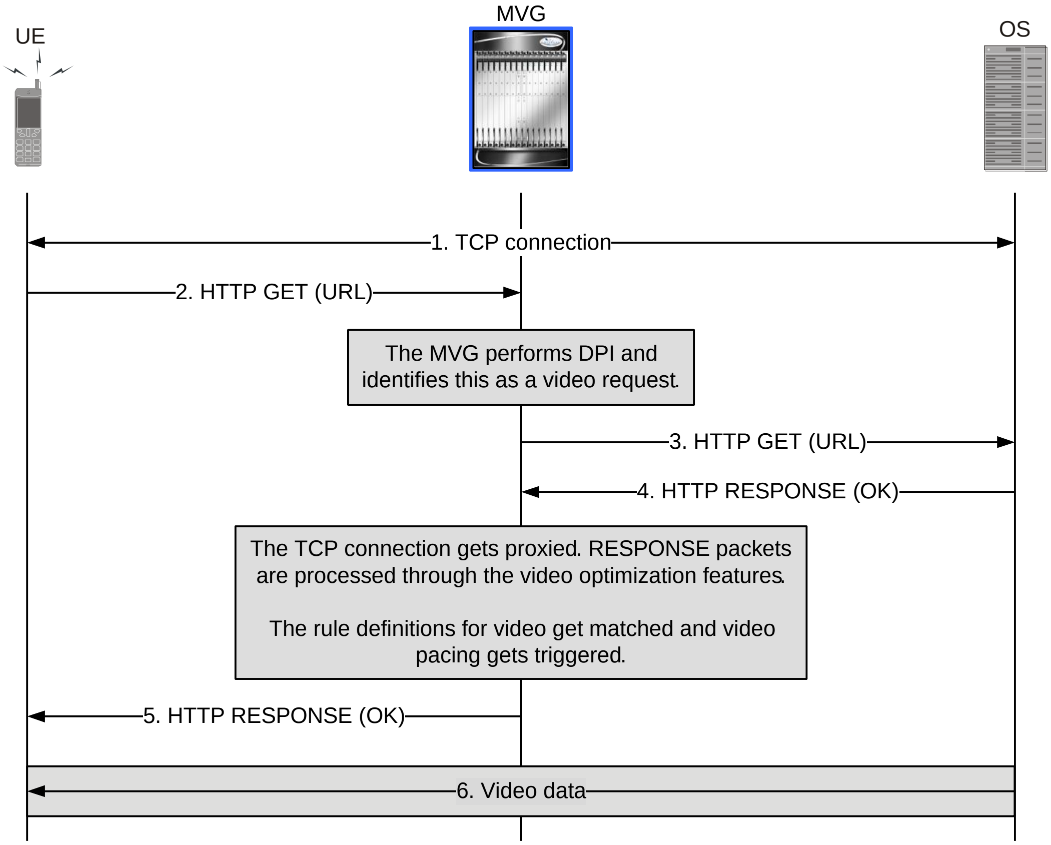

UE
MVG
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and identifies this as a video request.
3. HTTP GET (URL)
4. HTTP RESPONSE (OK)
The TCP connection gets proxied. RESPONSE packets are processed through the video optimization features.
The rule definitions for video get matched and video pacing gets triggered.
5. HTTP RESPONSE (OK)
6. Video data

Table 8. DPI of HTTP GET Request Identifying a Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a video request (the DPI on GET/POST succeeds).
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.
4.	The OS responds with an HTTP 200 OK. The Mobile Video Gateway proxies the TCP connection and the HTTP RESPONSE packets are processed through the video optimization features.
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE.
6.	The optimized TCP video flow continues to the UE.

DPI of HTTP RESPONSE Identifying a Video Request (MVG without the CAE)

When the Mobile Video Gateway cannot determine whether an HTTP GET request is a request for video content during DPI, it performs DPI again when it receives the HTTP RESPONSE from the OS. The figure below shows the Mobile Video Gateway performing DPI on an HTTP RESPONSE and identifying it as a response to a video request. The table that follows the figure describes each step in the message flow.

Figure 23. DPI of HTTP RESPONSE Identifying a Video Request

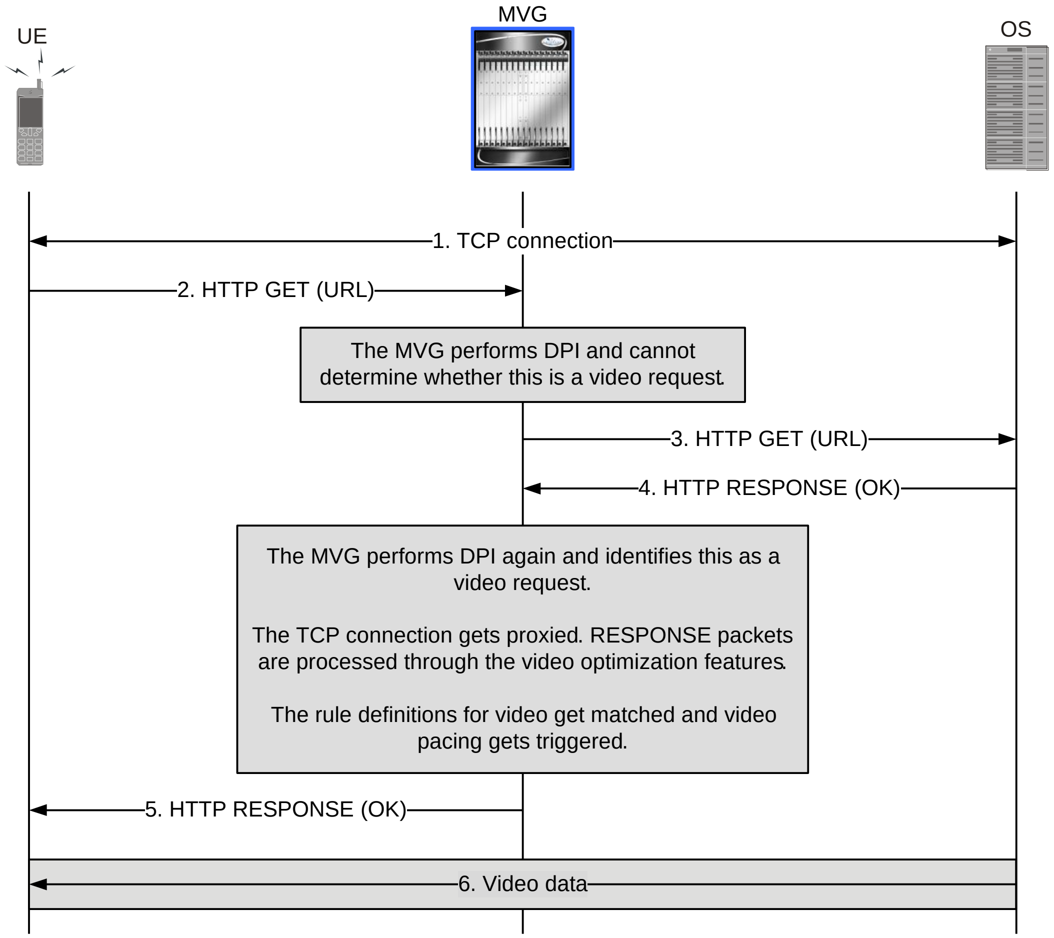

UE
MVG
OS
1. TCP connection
2. HTTP GET (URL)
The MVG performs DPI and cannot determine whether this is a video request.
3. HTTP GET (URL)
4. HTTP RESPONSE (OK)
The MVG performs DPI again and identifies this as a video request.
The TCP connection gets proxied. RESPONSE packets are processed through the video optimization features.
The rule definitions for video get matched and video pacing gets triggered.
5. HTTP RESPONSE (OK)
6. Video data

Table 9. DPI of HTTP RESPONSE Identifying a Video Request

Step	Description
1.	The UE creates a TCP connection with the OS.
2.	The Mobile Video Gateway receives an HTPP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.
3.	The Mobile Video Gateway forwards the HTTP request to the OS transparently, using the URL source IP address as the client address and the destination IP address as the OS address.

Step	Description
4.	The OS responds with an HTTP 200 OK. The Mobile Video Gateway performs DPI again and identifies this as a response to a video request (the DPI on the RESPONSE headers succeeds). The Mobile Video Gateway proxies the TCP connection and the HTTP RESPONSE packets are processed through the video optimization features.
5.	The Mobile Video Gateway forwards the HTTP 200 OK to the UE.
6.	The optimized TCP video flow continues to the UE.

Chapter 2
Mobile Video Gateway Configuration

This chapter provides configuration information for the Mobile Video Gateway.

Because each wireless network is unique, the system is designed with a variety of parameters allowing it to perform in various wireless network environments. In this chapter, only the minimum set of parameters are provided to make the system operational.

Before you perform the instructions in this chapter, confirm that the configuration for the mobile gateway upon which the Mobile Video Gateway software runs has been completed. Note that the Mobile Video Gateway features are not enabled by default, so you must follow the instructions in this chapter to configure and enable each required feature.

The following sections are included in this chapter:

Configuring CAE Re-addressing and Load Balancing

The Cisco CAE (Content Adaptation Engine) is an optional component of the Cisco Mobile Videoscape. The Mobile Video Gateway's CAE re-addressing and CAE load balancing features are configured and enabled via system CLI commands using **charging-action** command options within an Active Charging Service, which is a component of the Enhanced Charging Services. Active Charging Services employ the system's DPI capabilities and are configured in Global Configuration Mode so that the system performs DPI for CAE re-addressing and load balancing on all subscriber sessions over all system contexts.

To configure the CAE re-addressing and load balancing features, perform the following steps:

Step 1 Configure a CAE group for the CAEs in the server cluster by applying the commands in the section Configuring the CAE Group.

Step 2 Configure CAE re-addressing by applying the commands in the section Configuring CAE Re-addressing.

Step 3 Configure video optimization pre-processing by applying the commands in the section Configuring Video Optimization Pre-processing.

Step 4 Configure logging for CAE health-check monitoring by applying the commands in the section Configuring Logging for CAE Health-Check Monitoring.

Configuring the CAE Group

The CAE group is configured and enabled via system CLI commands in the context containing the interface to the CAEs, which is typically the destination context. Use the following commands to configure the CAE group:

```
configure

   context <context_name>

      cae_group <cae_group_name>

         local_address <IPv4_address>

         server <cae_name> address <IPv4_address> port <port_number>

         keepalive-server deadtime <seconds> interval <seconds> num-retry <num-retries>
port <port_number> timeout <seconds>

         end
```

The **cae-group** command specifies the name of the CAE group that will contain the CAEs servicing the video requests from the Mobile Video Gateway. <*cae_group_name*> can be a string between 1 and 79 characters. Note that there can be one and only one CAE group per context. Issuing this commands enters the Video Group Configuration Mode.

The **local-address** command specifies the local IPv4 address on the Mobile Video Gateway for the keep-alive TCP connection to the CAEs.

The **server** command specifies a CAE in the CAE group. This command must be repeated for all CAEs in the CAE group. Note that there is a system limit of 64 CAEs supported on a Mobile Video Gateway. <*cae_name*> can be a string between 1 and 15 characters. The **address** option specifies the IPv4 address of the CAE in dotted decimal notation.

This address is used for the keep-alive TCP connection to the Mobile Video Gateway. The **port** option specifies the port number on the CAE for the keep-alive TCP connection. The port number can be between 1 and 65535. This value defaults to 80 if absent from the command.

The **keepalive-server** command enables health-check monitoring for all CAEs in the CAE group. The **deadtime** option sets the periodic retry interval, in seconds, after a CAE is detected down. <*seconds*> can be between 1 and 1800 seconds, with a default value of 120 seconds. The **interval** option specifies the health-check monitoring interval, in seconds, which is how often the Mobile Video Gateway sends a keep-alive message to the CAEs. <*seconds*> can be between 0 and 120 seconds, with a default value of 10 seconds. A value of 0 turns off keep-alive detection and marks the state of all CAEs to Up. The **num-retry** option specifies the number of keep-alive retries to send after a CAE does not respond. <*num_retries*> can be between 1 and 20 retries, with a default value of 3 retries. The **port** option specifies the TCP port number for health-check monitoring. <*port_number*> can be between 1 and 65535, with a default value of 5100. The **timeout** option specifies the keep-alive timeout in seconds. <*seconds*> can be between 1 and 30 seconds, with a default value of 3 seconds.

Configuring CAE Re-addressing

In addition to the command sequence for configuring a CAE group, you need to add a charging action for CAE re-addressing, enable CAE re-addressing, and specify the HTTP x-header format to use to insert the destination IP address and TCP port number of the OS (Origin Server). Use the following commands to configure CAE re-addressing:

```
configure
   require active-charging
   active-charging service <service_name>
      charging-action cae_redirect
         video cae-readdressing xheader-format <xheader_format_name>
         video bitrate <bit_rate_in_bps>
         end
```

The **require active-charging** command enables active charging on the Mobile Video Gateway.

The **active-charging service** command specifies the name of the Active Charging Service. The <*service_name*> can be an alpha and/or numeric string between 1 and 15 characters.

The **charging-action** command specifies the name of the charging action. The <*charging_action_name*> can be an alpha and/or numeric string between 1 and 63 characters.

The **video cae-readdressing** command enables CAE re-addressing, allowing video content to be fetched from the CAEs. This command also enables CAE load balancing.

The **xheader-format** option specifies an HTTP x-header (Extension header) format for readdressing. When specified, the Mobile Video Gateway inserts a destination IP address and TCP port number in a proprietary HTTP x-header in the HTTP request to the CAE. The CAE uses this information to connect to the OS to retrieve selected video clips for adaptation. The <*xheader_format_name*> can be between 1 and 63 characters.

The **video bitrate** option specifies a suggested maximum bit rate value for video in bits per second.

Configuring Video Optimization Pre-processing

In addition to creating a CAE group and adding a charging action for CAE re-addressing, you need to add commands under the rulebase configuration to enable video optimization pre-processing for CAE re-addressing. Use the following commands to configure video optimization pre-processing:

```
rulebase base1

   tcp proxy-mode dynamic all

   video optimization-preprocessing cae-readdressing

   no tcp check-window-size

   action priority 15 group-of-ruledefs video_group charging-action cae_redirect

   route priority 10 ruledef http_routing analyzer http

   exit
```

The **`rulebase`** command creates the rulebase. The <*`rulebase_name`*> can be an alpha and/or numeric string between 1 and 63 characters.

The **`tcp proxy-mode dynamic all`** command enables the dynamic TCP proxy for subscriber-initiated TCP flows, and specifies that all TCP connections are split for all enabled Active Charging Service features.

The **`video optimization-preprocessing cae-readdressing`** command enables CAE re-addressing by enabling Enhanced Charging Services to process video requests for optimization per rulebase configuration.

Configuring Logging for CAE Health-Check Monitoring

Use the following commands to configure logging for CAE health-check monitoring:

```
logging active event-verbosity full

logging filter active facility vpn level warning
```

Note that the logging commands need to be issued from the context in which the CAE group resides (in the destination context, for example).

Sample CAE Re-addressing and Load Balancing Configuration

The following is a sample CAE re-addressing and load balancing configuration that includes a sample rule base, which acts as a subscriber’s policy in a charging service, and sample rule definitions (ruledefs), which define the packets to take action on and what action to take on them. Note that operators must create a unique configuration based on their own requirements.

```
configure
   context destination
      cae-group cae_group_1
         local-address ip_address
         server server_1 address ip_address port 80
         server server_2 address ip_address port 8080
         keepalive-server deadtime 120 interval 10 num-retry 3 port 5100 timeout 3
         exit
      exit
   context source
      apn.cisco.com
         accounting-mode radius-diameter
         active-charging rulebase base1
         ip context-name destination
         exit
      exit
   active-charging service service_1
      ruledef http_youtube
         http uri contains videoplayback
         http host contains googlevideo
         multi-line-or all-lines
         exit
      ruledef video
         http uri contains .m4v
   http uri contains .3gp
   http uri contains .mp4
   http uri contains .mov
   http uri contains .f4v
   multi-line-or all-lines
   exit
group-of-ruledefs video_group
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
charging-action mvg_1
   video cae-readdressing xheader-format xheader_format_name
   exit
rulebase base1
   action priority 1 ruledef no_redirect charging-action default
   action priority 2 group-of-ruledefs video_group charging-action mvg_1
   route priority 5 ruledef rr_http_80 analyzer http
   route priority 6 ruledef rr_http_8080 analyzer http
   exit
end
```

The association of a charging action to the CAE group in the configuration example above has the following logic:

- The system performs DPI on the HTTP GET request and determines that it is a video request based on the rule **`action priority 2 group-of-ruledefs video_group charging-action mvg_1`**.
- The system applies the charging action **`mvg_1`**. The **`video cae-readdressing`** command enables CAE re-addressing. The system examines the subscriber record and locates the interface to the CAEs in the destination context.
- The **`xheader-format`** option specifies an HTTP x-header format for readdressing. When specified, the Mobile Video Gateway inserts a destination IP address and TCP port number in a proprietary HTTP x-header in the HTTP request to the CAE. The CAE uses this information to connect to the OS to retrieve selected video clips for adaptation. The *`xheader_format_name`* can be between 1 and 63 characters.
- The system locates the CAE group **`cae_group_1`** in the destination context.
- The system selects a CAE in the group to service the video request using a round-robin selection method. The selected server information is stored in a subscriber session record. If the previously-selected server for the

same subscriber goes down, the system selects the next available CAE and updates the subscriber session record. If the system fails to find a CAE group in the configuration, or no CAEs in a group are available, the system redirects the video request to the OS.

Configuring Video Optimization Policy Control

The video optimization policy control feature is configured and enabled via system CLI commands using **charging-action** command options within an Active Charging Service. Active Charging Services employ the system's DPI capabilities and are configured in Global Configuration Mode so that the system performs DPI for video optimization policy control on all subscriber sessions over all system contexts.

Use the following commands to configure video optimization policy control:

```
configure

   require active-charging

   active-charging service <service_name>

      charging-action <charging_action_name>

         video pacing by-policing initial-burst-duration <seconds> normal-burst-duration
<seconds>

         video bitrate <bit_rate_in_bps>

         video cae-readdressing xheader-format <xheader_format_name>

         end
```

The **require active-charging** command enables active charging on the Mobile Video Gateway.

The **active-charging service** command specifies the name of the Active Charging Service. The <*service_name*> can be an alpha and/or numeric string between 1 and 15 characters.

The **charging-action** command specifies the name of the charging action. The <*charging_action_name*> can be an alpha and/or numeric string between 1 and 63 characters.

The **video pacing by-policing** command in this example enables video pacing by policing. Note that the video pacing feature is not enabled by default.

The **initial-burst-duration** option specifies the initial burst duration allowed before the feature begins to limit the bit rate to the actual encoding bit rate of the video. Note that the initial burst is configured in terms of time, so that for video files with different encoding bit rates, the amount of bytes allowed without enforcing pacing gets adjusted accordingly. The amount of bytes allowed is calculated by (video encoding rate * initial-burst-duration). The default value is 10 seconds.

The **normal-burst-duration** option specifies the normal burst duration allowed after the initial burst is completed. Like the initial burst, the normal burst is also configured in terms of time, so that for video files with different encoding bit rates, the amount of bytes allowed without enforcing pacing gets adjusted accordingly. The amount of bytes allowed is calculated by (video encoding rate * normal-burst-duration). The default value is 3 seconds.

The **video bitrate** option specifies a suggested maximum bit rate value for video. This default bit rate, in bits per second, is used on each video flow until the rate determination function calculates the optimal bit rate to use for video pacing. The default value is 0, which means that if rate determination fails on a flow identified as video, video pacing is not applied to the flow. If a value is configured for this CLI option, and if rate identification fails on a flow, instead of turning off pacing for the flow, the configured bit rate will be enforced on the flow.

The **video cae-readdressing** command enables CAE re-addressing and load balancing, allowing video content to be fetched from the CAEs.

The **xheader-format** option specifies an HTTP x-header format for readdressing. When specified, the Mobile Video Gateway inserts a destination IP address and TCP port number in a proprietary HTTP x-header in the HTTP request to the CAE. The CAE uses this information to connect to the OS to retrieve selected video clips for adaptation. The <*xheader_format_name*> can be between 1 and 63 characters.

Sample Video Optimization Policy Control Configurations

This section includes two sample video optimization policy control configurations, as follows:

- Obtaining the Video Policy via the PCRF over a Gx Interface
- Obtaining the Video Policy via the RADIUS Server over a RADIUS Interface

Note that in both sample configurations, the narrower the rule match, the higher the priority number assigned to the ruledef (rule definition) entry. Note also that operators must create a unique configuration based on their own requirements.

Obtaining the Video Policy via the PCRF over a Gx Interface

The following is a sample configuration for obtaining the video policy via the PCRF over a Gx interface.

```
configure
   require active-charging
   active-charging service_1
      ruledef rr_http_80
         tcp either-port=80
         rule-application routing
         exit
      ruledef rr_http_8080
         tcp either-port=8080
         rule-application routing
         exit
      ruledef http_youtube
         http uri contains videoplayback
         http host contains googlevideo
         multi-line-or all-lines
         exit
      ruledef video
         http uri contains .m4v
         http uri contains .3gp
   http uri contains .mp4
   http uri contains .mov
   http uri contains .f4v
   multi-line-or all-lines
   exit
ruledef FACEBOOK
   http uri contains fbcdn
   exit
group-of-ruledefs VIDEO_GOLD
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
group-of-ruledefs VIDEO_SILVER
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
group-of-ruledefs VIDEO_BRONZE
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
xheader-format XHDR_GOLD
   insert X-adaptation-profile-index string-constant 4
   exit
xheader-format XHDR_SILVER
   insert X-adaptation-profile-index string-constant 3
   exit
xheader-format XHDR_BRONZE
   insert X-adaptation-profile-index string-constant 2
   exit
```

```
charging-action GOLD_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_GOLD
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
charging-action GOLD_ACTION_NO_PACING
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_GOLD
   exit
charging-action SILVER_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_SILVER
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
charging-action BRONZE_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_BRONZE
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
rulebase base1
   tcp proxy-mode static
```

```
         video optimization-preprocessing all

         action priority 10 dynamic-only group_of_ruledefs VIDEO_GOLD charging-action
GOLD_ACTION

         action priority 20 dynamic-only group_of_ruledefs VIDEO_SILVER charging-action
SILVER_ACTION

         action priority 30 dynamic-only group_of_ruledefs VIDEO_BRONZE charging-action
BRONZE_ACTION

         action priority 5 dynamic-only group_of_ruledefs VIDEO_GOLD_NO_PACING charging-
action GOLD_ACTION_NO_PACING

         route priority 2 ruledef rr_http_8080 analyzer http

         route priority 1 ruledef rr_http_80 analyzer http

         exit

      exit

   context pgw

      interface 20/2-next

         ip address <ip_address> <subnet_mask>

         exit

      subscriber default

         ip access-group acl1 in

         ip access-group acl1 out

         active-charging rulebase base2

         exit

      radius group default

      end
```

Obtaining the Video Policy via the RADIUS Server over a RADIUS Interface

The following is a sample configuration for obtaining the video policy via the RADIUS server over a RADIUS interface.

```
configure

   require active-charging

   active-charging service_1

      ruledef rr_http_80

         tcp either-port=80

         rule-application routing

         exit

      ruledef rr_http_8080

         tcp either-port=8080

         rule-application routing

         exit

      ruledef http_youtube

         http uri contains videoplayback

         http host contains googlevideo

         multi-line-or all-lines

         exit

      ruledef video

         http uri contains .m4v

         http uri contains .3gp

         http uri contains .mp4

         http uri contains .mov

         http uri contains .f4v

         multi-line-or all-lines

         exit

      ruledef FACEBOOK
   http uri contains fbcdn
   exit
group-of-ruledefs VIDEO_GOLD
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
group-of-ruledefs VIDEO_SILVER
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
group-of-ruledefs VIDEO_BRONZE
   add-ruledef priority 1 ruledef http_youtube
   add-ruledef priority 2 ruledef video
   exit
xheader-format XHDR_GOLD
   insert X-adaptation-profile-index string-constant 4
   exit
xheader-format XHDR_SILVER
   insert X-adaptation-profile-index string-constant 3
   exit
xheader-format XHDR_BRONZE
   insert X-adaptation-profile-index string-constant 2
   exit
charging-action GOLD_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_GOLD
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
charging-action GOLD_ACTION_NO_PACING
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_GOLD
   exit
charging-action SILVER_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_SILVER
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
charging-action BRONZE_ACTION
   flow idle-timeout 200
   video bitrate 1000000
   video cae-readdressing
   xheader-insert xheader-format XHDR_BRONZE
   video pacing by-policing initial-burst-duration 10 normal-burst-duration 5
   exit
rulebase GOLD_RBASE
   tcp proxy-mode static
   video optimization-preprocessing all
   action priority 10 group_of_ruledefs VIDEO_GOLD charging-action GOLD_ACTION
   action priority 5 ruledef FACEBOOK charging-action GOLD_ACTION_NO_PACING
   route priority 2 ruledef rr_http_8080 analyzer http
   route priority 1 ruledef rr_http_80 analyzer http
   exit
   rulebase SILVER_RBASE
      tcp proxy-mode static
      video optimization-preprocessing all
      action priority 10 group_of_ruledefs VIDEO_SILVER charging-action GOLD_ACTION
      route priority 2 ruledef rr_http_8080 analyzer http
      route priority 1 ruledef rr_http_80 analyzer http
      exit
   rulebase BRONZE_RBASE
      tcp proxy-mode static
      video optimization-preprocessing all
      action priority 10 group_of_ruledefs VIDEO_BRONZE charging-action BRONZE_ACTION
      route priority 2 ruledef rr_http_8080 analyzer http
      route priority 1 ruledef rr_http_80 analyzer http
      exit
   exit
context pgw
interface 20/2-next
   ip address <ip_address> <subnet_mask>
   exit
subscriber default
   ip access-group acl1 in
   ip access-group acl1 out
   active-charging rulebase BRONZE_RBASE
   exit
radius group default
end
```

Configuring Video White-listing

Certain video clips can be excluded from video optimization. This is referred to as white-listing. The video white-listing feature can either be configured using empty charging actions that match the white-listed URLs, or using DPI rule definitions that do not match the white-listed URLs.

Use the following commands to configure video white-listing:

```
rulebase whitelist

   action priority 5 ruledef facebook charging-action VIDEO_NO_PACING

   action priority 10 group-of-ruledefs all_video charging-action VIDEO_PACING

   route priority 1 ruledef rr_http_80 analyzer http

   route priority 2 ruledef rr_http_8080 analyzer http

   exit
```

The **rulebase** command creates the white-list rulebase. The <*rulebase_name*> can be an alpha and/or numeric string between 1 and 63 characters.

In the example above, the first **action priority** command specifies the action priority as 5 for the facebook ruledef, with a charging action of VIDEO_NO_PACING. When the facebook ruledef is matched during DPI, the corresponding video flows are excluded from video pacing. The second **action priority** command specifies the action priority as 10 for the all_video group-of-ruledefs, with a charging action of VIDEO_PACING. When matched during DPI, the corresponding video flows are included in video pacing.

The two **route priority** commands control the routing of packets to the appropriate protocol analyzers.

Configuring Video Pacing

The video pacing feature is configured and enabled via system CLI commands as a charging action within an Active Charging Service. Active Charging Services employ the system's DPI capabilities and are configured in Global Configuration Mode so that the system performs DPI for video pacing on all subscriber sessions over all system contexts.

Use the following commands to configure video pacing:

```
configure

   require active-charging

   active-charging service <service_name>

      charging-action <charging_action_name>

         video pacing by-policing initial-burst-duration <seconds> normal-burst-duration
<seconds>

         video bitrate <bit_rate_in_bps>

         end
```

The **require active-charging** command enables active charging on the Mobile Video Gateway.

The **active-charging service** command specifies the name of the Active Charging Service. The <*service_name*> can be an alpha and/or numeric string between 1 and 15 characters.

The **charging-action** command specifies the name of the charging action. The <*charging_action_name*> can be an alpha and/or numeric string between 1 and 63 characters.

The **video pacing by-policing** command in this example enables video pacing by policing. Note that the video pacing feature is not enabled by default.

The **initial-burst-duration** option specifies the initial burst duration allowed before the feature begins to limit the bit rate to the actual encoding bit rate of the video. Note that the initial burst is configured in terms of time, so that for video files with different encoding bit rates, the amount of bytes allowed without enforcing pacing gets adjusted accordingly. The amount of bytes allowed is calculated by (video encoding rate * initial-burst-duration). The default value is 10 seconds.

The **normal-burst-duration** option specifies the normal burst duration allowed after the initial burst is completed. Like the initial burst, the normal burst is also configured in terms of time, so that for video files with different encoding bit rates, the amount of bytes allowed without enforcing pacing gets adjusted accordingly. The amount of bytes allowed is calculated by (video encoding rate * normal-burst-duration). The default value is 3 seconds.

The **video bitrate** option specifies a suggested maximum bit rate value for video. This default bit rate, in bits per second, is used on each video flow until the rate determination function calculates the optimal bit rate to use for pacing. The default value is 0, which means that if rate determination fails on a flow identified as video, video pacing is not applied to the flow. If a value is configured for this CLI option, and if rate identification fails on a flow, instead of turning off pacing for the flow, the configured bit rate will be enforced on the flow.

Sample Video Pacing Configuration

The following is a sample video pacing configuration. Note that operators must create a unique configuration based on their own requirements.

```
configure

   require active-charging

   active-charging service_1

      charging-action video_pacing

         video pacing by-policing initial-burst-duration 15 normal-burst-duration 5

         video bitrate 1000000

         exit

      rulebase base1

         route priority 1 ruledef rr_http_80 analyzer http

         route priority 3 ruledef rr_http_8080 analyzer http

         action priority 5 group-of-ruledefs video_group charging-action video_pacing

         exit

      ruledef rr_http_80

         tcp either-port=80

         rule-application routing

         exit

      ruledef rr_http_8080

         tcp either-port=8080

         rule-application routing

         exit

      ruledef video

         http content type contains video

         http uri contains .m4v

         http uri contains .3gp

         http uri contains .mp4
      http uri contains .mov

      http uri contains .f4v

      multi-line-or all-lines

      exit

   ruledef http_youtube

      http uri contains videoplayback

      http host contains googlevideo

      multi-line-or all-lines

      exit

   group-of-ruledefs video_group

      add-ruledef priority 1 ruledef video

      add-ruledef priority 2 ruledef http_youtube

      end
```

Video pacing requires the HTTP analyzer in the Enhanced Charging Service to examine the packets before video pacing does. See the routing rule definitions in the example above for how to redirect packets to the HTTP analyzer based on TCP ports.

In this example, the operator defines a group of rule definitions called video_group. When any of the rule definitions in the group are matched, the packet flow is considered to be a video flow. As shown in the example, this can be either a URI match, which is useful for matching a certain file extension, or a string match that identifies a video from a certain website (for example, YouTube™ always has the string “videoplayback” in the URI), or a hostname match, which is useful for matching videos from a specific host, such as “googlevideo”.

Note that in ruledef video, we match with "http content type contains video". This works for most video websites, which properly identify their videos with the proper content type. However, not all websites do this correctly, thus we need to supplement the rule with matches using the common file extensions used for video files. Note a dot (.) is added before each file extension to ensure the match is applied only to the file extension, not some other character combination in the middle of a URI.

Also note that matching with file extensions works only if the original server delivers video files with extensions. If the video server identifies its video files with a random hash string (with no file extension), and does not identify it with the proper content type, we cannot identify those videos.

Configuring TCP Link Monitoring

The TCP link monitoring feature is configured and enabled via an **`active-charging`** command option in either APN Configuration Mode or Subscriber Configuration Mode. When TCP link monitoring is enabled, the system monitors the downlink TCP traffic towards the subscriber UEs for TCP proxy and non-proxy modes.

Use the following commands to configure TCP link monitoring in Subscriber Configuration Mode:

```
configure
   require active-charging
   context context_name
      subscriber default
         active-charging link-monitor tcp log
         exit
      exit
   context context_name
      apn cisco.com
         active-charging link-monitor tcp log
         end
```

The **`require active-charging`** command enables active charging on the Mobile Video Gateway.

The **`active-charging link-monitor tcp log`** command in this example enables TCP link monitoring. Note that TCP link monitoring is not enabled by default. Also note that when this command is configured without the **`log`** option, TCP link monitoring is enabled without logging.

The **`log`** option enables logging with histogram and time-series logging for both RTT and bit rate.

For additional options for this command, see the “Subscriber Configuration Mode Commands” and “APN Configuration Mode Commands” chapters of the *Command Line Interface Reference.*

Configuring Mobile Video Statistics

The mobile video statistics feature is configured and enabled via system CLI commands as a charging action within an Active Charging Service. Active Charging Services employ the system's DPI capabilities, and are configured in Global Configuration Mode so that the system performs DPI for mobile video statistics on all subscriber sessions over all system contexts. Bulk statistics can also be configured to generate mobile video statistics based on the MVS (Mobile Videoscape) schema.

Use the following commands to configure mobile video statistics:

```
configure

   require active-charging

   active-charging service <service_name>

      charging-action <charging_action_name>

         video detailed-statistics

         end
```

The **require active-charging** command enables active charging on the Mobile Video Gateway.

The **active-charging service** command specifies the name of the Active Charging Service. The <*service_name*> can be an alpha and/or numeric string between 1 and 15 characters.

The **charging-action** command specifies the name of the charging action. The <*charging_action_name*> can be an alpha and/or numeric string between 1 and 63 characters.

The **video detailed-statistics** command in this example enables mobile video statistics. When a flow matches a rule definition for video during DPI, the mobile video statistics feature begins collecting detailed statistics for the video flow. Note that the mobile video statistics feature is not enabled by default.

Configuring Bulk Statistics

Use the following commands to configure bulk statistics for the MVS (Mobile Videoscape):

```
configure
   bulkstats collection
   bulkstats mode
      sample-interval <time_interval>
      transfer-interval <xmit_time_interval>
      file <number>
         receiver <ip_address> primary mechanism ftp login <username> password <pwd>
         receiver <ip_address> secondary mechanism ftp login <username> password <pwd>
         mvs schema <file_name> format <format_string>
         end
```

The **bulkstats collection** command in this example enables bulk statistics, and the system begins collecting pre-defined bulk statistical information.

The **bulkstats mode** command enters Bulk Statistics Configuration Mode, where you define the statistics to collect.

The **sample-interval** command specifies the time interval, in minutes, to collect the defined statistics. The <*time-interval*> can be in the range of 1 to 1440 minutes. The default value is 15 minutes.

The **transfer-interval** command specifies the time interval, in minutes, to transfer the collected statistics to the receiver (the collection server). The <*xmit_time_interval*> can be in the range of 1 to 999999 minutes. The default value is 480 minutes.

The **file** command specifies a file in which to collect the bulk statistics. A bulk statistics file is used to group bulk statistics schema, delivery options, and receiver configuration. The <*number*> can be in the range of 1 to 4.

The **receiver** command in this example specifies a primary and secondary collection server, the transfer mechanism (in this example, ftp), and a login name and password.

The **mvs schema** command specifies that the MVS schema is used to gather statistics. The <*file_name*> is an arbitrary name (in the range of 1 to 31 characters) to use as a label for the collected statistics defined by the **format** option. The **format** option defines within quotation marks the list of variables in the MVS schema to collect. For example: "%date, %time%, %<*variables_to_collect*>%". The <*format_string*> can be in the range of 1 to 3599.

For descriptions of the MVS schema variables, see the "MVS Schema Statistics" chapter of the *Statistics and Counters Reference.* Note that additional options can be used to configure bulk statistics for the Mobile Videoscape. For more information on configuring bulk statistics, see the *System Administration Guide.*

Chapter 3 Monitoring the Mobile Video Gateway

This chapter provides information for monitoring the status and performance of the features and functions of the Mobile Video Gateway using the **`show`** commands found in the system CLI. These commands have many related keywords that allow them to provide useful information on all aspects of the system ranging from current software configuration through call activity and status.

The selection of keywords described in this chapter is intended to provided the most useful and in-depth information for monitoring the system. For additional information on these and other **`show`** command keywords, refer to the *Command Line Interface Reference.*

In addition to the CLI, the system supports the sending of SNMP (Simple Network Management Protocol) traps that indicate status and alarm conditions. See the *SNMP MIB Reference* for a detailed listing of these traps.

Monitoring System Status and Performance

This section contains commands used to monitor the status and performance of the features and functions of the Mobile Video Gateway. Output descriptions for most of the commands are located in the *Statistics and Counters Reference.*

Table 10. Mobile Video Gateway Status and Performance Monitoring Commands

To do this:	Enter this command:
View Information on CAE Re-addressing and Load Balancing	
Display CAE re-addressing statistics.	`show active-charging flows summary cae-readdressing`
Display CAE configuration information, including the name of the corresponding CAE group, for all CAEs or for a specific CAE.	`show cae-group server { all | name cae-name name }`
View Information on Video Optimization Policy Control	
Display information on groups of rule definitions configured for tiered video policies (Gold, Silver, and Bronze levels, for example) in Active Charging Services.	`show active-charging group-of-ruledefs statistics`
Display a list of subscribers that are assigned to a particular rulebase, such as GOLD_RBASE, for example.	`show active-charging sessions full rulebase name`
Display the suggested maximum bit rate value configured for each charging action, which determines the video policy.	`show active-charging charging-action name name`
View Information on Video Pacing	
Display information on TCP video flows that have been paced.	`show active-charging flows`
Display detailed statistics on TCP video flows that have been paced.	`show active-charging video detailed-statistics [ container { flv | mp4 | others } | rat { cdma | gprs | hspa | lte | others | umts | wlan } | ue { android | ios | laptop | others } ]`
Display statistics on video pacing and TCP video flows.	`show active-charging subsystem all`
View Information on TCP Link Monitoring	

Display statistics on the average TCP throughput and RTT (Round Trip Time) of downlink TCP traffic towards the subscriber UE.	`show active-charging flows full`
	`show active-charging sessions full`

View Information on Active Charging Services	
Display service and configuration statistics for the Active Charging Services, including statistics for video-related Active Charging Services.	`show active-charging subsystem`

To do this:	Enter this command:
View Bulk Statistics for the Mobile Video Gateway	
Display bulk statistics for the Mobile Video Gateway.	`show bulkstats variables mvs`
Display bulk statistics for the system.	`show bulkstats data`
View Session Subsystem and Task Information	
Display Session Subsystem and Task Statistics ***Important:*** Refer to the "System Software Task and Subsystem Descriptions" appendix of the *System Administration Guide* for additional information on the Session subsystem and its various manager tasks.	
View Session Manager statistics.	`show session subsystem facility sessmgr all`
View ACS Manager statistics.	`show session subsystem facility acsmgr all`
View Session Disconnect Reasons	
View session disconnect reasons with verbose output.	`show session disconnect-reasons`

Clearing Statistics and Counters

It may be necessary to periodically clear statistics and counters in order to gather new information. The system provides the ability to clear statistics and counters based on their grouping (PPP, MIPHA, MIPFA, etc.).

Statistics and counters can be cleared using the CLI **`clear`** command. Refer to the *Command Line Interface Reference* for detailed information on using this command.

Appendix A
Sample Mobile Video Gateway Configuration File

This appendix contains a sample Mobile Video Gateway configuration file.

In the following configuration example, commented lines are labeled with the number symbol (#) and variables are identified using italics within brackets (<*variable*>).

Sample Integrated MVG/P-GW Configuration

This section contains a sample configuration for the Mobile Video Gateway integrated with a P-GW. This sample configuration contains commands for the Mobile Video Gateway obtaining the video policy for subscribers from a RADIUS server and from static assignment, and without a CAE server cluster deployed in the network.

```
# Configure system settings

configure

   logging disable eventid 1 to 91289

   logging disable eventid 91291 to 200000

   license key <key_value>

   aaa large-configuration

   system hostname asr5kmvg

   card 1

      mode active

      exit

   card 2

      mode active

      exit

   card 3

      mode active

      exit

   require session recovery

   require active-charging

   exit

# Configure the local context

configure

   context local

      interface SPIO1

         ip address <ip_address> <subnet_mask>
   interface SPIO2
      ip address <ip_address> <subnet_mask>
      exit
      server ftpd
      exit
   ssh key <key_value>
   ssh key <key_value>
   ssh key <key_value>
   server sshd
      max servers 100
      subsystem sftp
      exit
   server telnetd
      exit
   subscriber default
      exit
   administrator admin encrypted password <password> ftp
   aaa group default
     exit
   gtpp group default
      exit
   ip route 0.0.0.0 0.0.0.0 <gateway_ip_addr> SPIO1
   ip route <ip_address> <subnet_mask> <gateway_ip_addr> SPIO2
   ip route <ip_address> <subnet_mask> <gateway_ip_addr> SPIO2
   exit
port ethernet 24/1
   no shutdown
   bind interface SPIO1 local
   exit
   port ethernet 24/2
      no shutdown
      bind interface SPIO2 local
      exit
   end
# Configure the SGi context
configure
   context SGi
      ip vrf test
         exit
      interface sgi
         description sgi-port-2-out
         exit
      interface sgi-port-1
         description sgi-port-1-in
         ip address <ip_address> <subnet_mask>
         exit
      interface sgi-port-2
         description sgi-port-2-out
         ip address <ip_address> <subnet_mask>
         exit
      interface sgi-port-3
         description sgi-port-to-sup
         ip address <ip_address> <subnet_mask>
         exit
      interface sgi-port-4
         description towards-www-server
         ip address <ip_address> <subnet_mask>
         exit
      subscriber default
      accounting-mode none
      ip access-group pgw-acl in
      ip access-group pgw-acl out
      active-charging rulebase no_pacing
      no tpo policy
      exit
   apn cisco.com
      accounting-mode none
      ip access-group pgw-acl in
      ip access-group pgw-acl out
      ip context-name internet
      ip address pool name testpool
      active-charging rulebase tpo
      tpo policy tpo_video
      exit
   apn common.com
      accounting-mode none
      ip access-group pgw-acl in
      ip access-group pgw-acl out
      ip context-name internet
      ip address pool name testpool
      active-charging rulebase common
      active-charging link-monitor tcp log
      exit
      radius change-authorize-nas-ip <ip_address> encrypted key <key> event-
timestamp-window 0 no-reverse-path-forward-check no-nas-identification-check
      aaa group default
         radius attribute nas-ip-address address <ip_address>
         radius server <ip_address> encrypted key <key> port <port_number>
         exit
      aaa group aaa_group1
         radius attribute nas-ip-address address <ip_address>
         exit
      gtpp group default
         exit
      gtpu-service ggsn_gtpu
         bind ipv4-address <ipv4_address>
         exit
      egtp-service pgwgtp
         interface-type interface-pgw-ingress
         associate gtpu-service ggsn_gtpu
         gtpc bind ipv4-address <ipv4_address>
         exit
      exit
   end
# Configure the PGW service
configure
   context pgw
      pgw-service pgw_service_1
         plmn id mcc <mcc> mnc <mnc>
         associate egtp-service pgwgtp
         exit
      ip route 0.0.0.0 0.0.0.0 <gateway_ip_address> sgi-port-2
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-1
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-1
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-2
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-2
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-2
      ip igmp profile default
      exit
   end
# Configure the Internet context
configure
   context internet
      ip access-list ecs-acl
         exit
      ip access-list pgw-acl
         redirect css service service1 any
         permit any
         exit
      ip pool testpool range <ip_address> <ip_address> public 0 policy allow-static-allocation
      interface sgi-port-4
      description towards-www-server
         ip address <ip_address> <subnet_mask>
         exit
      subscriber default
      exit
      aaa group default
      exit
      gtpp group default
      exit
      ip route <ip_address> <subnet_mask> <gateway_ip_address> sgi-port-4
      igmp profile default
      exit
   end
```

```
# Configure bulkstats collection

configure

   bulkstats collection

   bulkstats mode

      sample-interval 1

      transfer-interval 5

      file 1

         mvs schema ttl-rate format EMS,TTL-RATE,%date%,%time%,%tcplm-ttl-avrg-
rate%,%tcplm-ttl-rate-220-259kbps%,%tcplm-ttl-rate-260-299kbps%,%tcplm-ttl-rate-
300-339kbps%,%tcplm-ttl-rate-340-379kbps%,%tcplm-ttl-rate-380-419kbps%,%tcplm-
ttl-rate-420-459kbps%,%tcplm-ttl-rate-460-499kbps%,%tcplm-ttl-rate-500-
539kbps%,%tcplm-ttl-rate-540-579kbps%,%tcplm-ttl-rate-580-619kbps%,%tcplm-ttl-
rate-620-719kbps%,%tcplm-ttl-rate-720-819kbps%,%tcplm-ttl-rate-820-
919kbps%,%tcplm-ttl-rate-920-1019kbps%,%tcplm-ttl-rate-1020-1119kbps%,%tcplm-ttl-
rate-1120-1219kbps%,%tcplm-ttl-rate-1220-1319kbps%,%tcplm-ttl-rate-1320-
1419kbps%,%tcplm-ttl-rate-1420-1519kbps%,%tcplm-ttl-rate-1520-1619kbps%,%tcplm-
ttl-rate-gteq-1620kbps%,

         mvs schema ttl-rtt format EMS,TTL-RTT,%date%,%time%,%tcplm-ttl-avrg-rtt-
ms%,%tcplm-ttl-rtt-lt-50ms%,%tcplm-ttl-rtt-50-69ms%,%tcplm-ttl-rtt-70-
89ms%,%tcplm-ttl-rtt-90-109ms%,%tcplm-ttl-rtt-110-129ms%,%tcplm-ttl-rtt-130-
149ms%,%tcplm-ttl-rtt-150-169ms%,%tcplm-ttl-rtt-170-189ms%,%tcplm-ttl-rtt-190-
209ms%,%tcplm-ttl-rtt-210-229ms%,%tcplm-ttl-rtt-230-249ms%,%tcplm-ttl-rtt-250-
289ms%,%tcplm-ttl-rtt-290-329ms%,%tcplm-ttl-rtt-330-369ms%,%tcplm-ttl-rtt-370-
409ms%,%tcplm-ttl-rtt-410-449ms%,%tcplm-ttl-rtt-450-489ms%,%tcplm-ttl-rtt-490-
529ms%,%tcplm-ttl-rtt-530-569ms%,%tcplm-ttl-rtt-570-609ms%,%tcplm-ttl-rtt-gteq-
1650ms%,

         mvs schema video-rate format EMS,VIDEO-RATE,%date%,%time%,%tcplm-video-
avrg-rate%,%tcplm-video-rate-220-259kbps%,%tcplm-video-rate-260-299kbps%,%tcplm-
video-rate-300-339kbps%,%tcplm-video-rate-340-379kbps%,%tcplm-video-rate-380-
419kbps%,%tcplm-video-rate-420-459kbps%,%tcplm-video-rate-460-499kbps%,%tcplm-
video-rate-500-539kbps%,%tcplm-video-rate-540-579kbps%,%tcplm-video-rate-580-
619kbps%,%tcplm-video-rate-620-719kbps%,%tcplm-video-rate-720-819kbps%,%tcplm-
video-rate-820-919kbps%,%tcplm-video-rate-920-1019kbps%,%tcplm-video-rate-1020-
1119kbps%,%tcplm-video-rate-1120-1219kbps%,%tcplm-video-rate-1220-
1319kbps%,%tcplm-video-rate-1320-1419kbps%,%tcplm-video-rate-1420-
1519kbps%,%tcplm-video-rate-1520-1619kbps%,%tcplm-video-rate-gteq-1620kbps%,

         mvs schema video-rtt format EMS,VIDEO-RTT,%date%,%time%,%tcplm-video-
avrg-rtt-ms%,%tcplm-video-rtt-lt-50ms%,%tcplm-video-rtt-50-69ms%,%tcplm-video-
rtt-70-89ms%,%tcplm-video-rtt-90-109ms%,%tcplm-video-rtt-110-129ms%,%tcplm-video-
rtt-130-149ms%,%tcplm-video-rtt-150-169ms%,%tcplm-video-rtt-170-189ms%,%tcplm-
video-rtt-190-209ms%,%tcplm-video-rtt-210-229ms%,%tcplm-video-rtt-230-
249ms%,%tcplm-video-rtt-250-289ms%,%tcplm-video-rtt-290-329ms%,%tcplm-video-rtt-
330-369ms%,%tcplm-video-rtt-370-409ms%,%tcplm-video-rtt-410-449ms%,%tcplm-video-
```

```
rtt-450-489ms%,%tcplm-video-rtt-490-529ms%,%tcplm-video-rtt-530-569ms%,%tcplm-video-rtt-570-609ms%,%tcplm-video-rtt-gteq-1650ms%
         exit
      file2
         exit
      local-directory /flash/mur
         exit
      exit
   end
# Configure the active charging service
configure
   active-charging service service1
      ruledef facebook
         http uri contains fbcdn
         exit
      ruledef http_all
         http any-match = TRUE
         exit
      ruledef http_youtube
         http uri contains videoplayback
         http host contains googlevideo
         multi-line-or all-lines
         exit
      ruledef route_icmp
         icmp any-match = TRUE
         rule-application routing
         exit
      ruledef rr_http_80
         tcp either-port = 80
         rule-application routing

         multi-line-or all-lines

         exit

      ruledef rr_http_8080

         tcp either-port = 8080

         rule-application routing

         multi-line-or all-lines

         exit

      ruledef video

         http uri contains .m4v

         http uri contains .3gp

         http uri contains .mp4

         http uri contains .mov

         http uri contains .f4v

         http request method = get

         http uri contains txt

         multi-line-or all-lines

         exit

      group-of-ruledefs all_video

         add-ruledef priority 2 ruledef video

         add-ruledef priority 3 ruledef http_youtube

         exit

      charging-action VIDEO_NO_PACING

         video bitrate 1000000

         exit

      charging-action VIDEO_PACING

         video bitrate 1000000

         video pacing by-policing initial-burst-duration 15 normal-burst-duration
15
```

```
         exit
      charging-action default
         exit
      charging-action video_tpo
         tpo profile special
         exit
      charging-action common
         video bitrate 1000000
         video pacing by-policing initial-burst-duration 15 normal-burst-duration
15
         exit
      rulebase base1
         tcp proxy-mode static
         action priority 1 ruledef no-redirect charging-action pacing
         action priority 2 ruledef proxy charging-action default
         exit
      rulebase common
         tcp proxy-mode static
```

ase

```
         action priority 10 group-of-ruledefs all_video charging-action common
         action priority 50 ruledef route_http_80 analyzer http
         action priority 60 ruledef route_http_8080 analyzer http
         exit
      rulebase default
         exit
      rulebase pacing
         action priority 10 ruledef all_video charging-action VIDEO_PACING
         route priority 1 ruledef rr_http_80 analyzer http
         route priority 2 ruledef rr_http_8080 analyzer http
         exit
      rulebase tcp_proxy
         tcp proxy-mode static
         exit
      rulebase tpo
         tcp proxy-mode dynamic all
         action priority 10 group-of-ruledefs all_video charging-action video_tpo
         route priority 1 ruledef rr_http_80 analyzer http
         route priority 2 ruledef rr_http_8080 analyzer http
         exit
      rulebase whitelist
         action priority 5 ruledef facebook charging-action VIDEO_NO_PACING
         action priority 10 group-of-ruledefs all_video charging-action
VIDEO_PACING
         route priority 1 ruledef rr_http_80 analyzer http
         route priority 2 ruledef rr_http_8080 analyzer http
         exit
      tpo policy no_tpo
         exit
      tpo policy test1
         match-rule no-ruledef-match tpo profile special
         exit
      tpo policy tpo_video
         match-rule no-ruledef-match tpo profile special
         exit
      exit
   end
# Configure the SGi interfaces
configure
```

```
        port ethernet 17/1
           no shutdown
           bind interface sgi-port-1 SGi
           exit
        port ethernet 17/2
           no shutdown
           bind interface sgi-port-2 SGi
           exit
        port ethernet 17/3
           no shutdown
           bind interface sgi-port-3 SGi
           exit
        port ethernet 17/4
           no shutdown
           bind interface sgi-port-4 internet
           exit
        end
```

Notes:

- The configuration above configures the Mobile Video Gateway without a CAE server cluster deployed in the network. For instructions on configuring the Mobile Video Gateway to work with a CAE server cluster, see “Configuring CAE Re-addressing and Load Balancing” and “Sample CAE Re-addressing and Load Balancing Configuration” in Chapter 2.
- The configuration above configures the Mobile Video Gateway to obtain the video policy from a RADIUS server and from static assignment. For instructions on configuring the Mobile Video Gateway to obtain the video policy from the PCRF via the Gx interface, see “Configuring Video Policy Optimization Control” and “Sample Video Optimization Policy Control Configurations” in Chapter 2.
- The configuration above enables TPO in TCP proxy static mode. To configure TPO in TCP proxy dynamic mode, remove “match-rule” from the TPO policy “tpo_video”.
- For video pacing, under the APN, change the active charging rulebase from “tpo” to “pacing”.
- For TCP link monitoring, under the APN add “active-charging link-monitor tcp log”.