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 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 CI
SCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at
www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner
does not imply a partnership relationship
between Cisco and any other company.

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.

MVG Administration Guide, StarOS Release 18
© 2015 Cisco Systems, Inc. All rights reserved.
# CONTENTS

**About this Guide** .................................................................................. V  
Conventions Used ........................................................................ vi  
Supported Documents and Resources ........................................ vii  
Related Common Documentation ............................................... vii  
  Related Product Documentation ........................................ vii  
  Obtaining Documentation ................................................... vii  
Contacting Customer Support ................................................... viii  

**Mobile Video Gateway Overview** .................................................. 9  
Product Description ........................................................................ 10  
Qualified Platforms ....................................................................... 10  
Licenses .......................................................................................... 10  
Summary of Mobile Video Gateway Features and Functions .......... 10  
Network Deployments and Interfaces ............................................ 12  
  The Mobile Video Gateway in an E-UTRAN/EPC Network ........ 12  
  The Mobile Video Gateway in a GPRS/UMTS Network .......... 13  
  The Mobile Video Gateway in a CDMA2000 Network .......... 14  
  Mobile Video Gateway Logical Network Interfaces .............. 14  
Features and Functionality ................................................................. 16  
Deep Packet Inspection ................................................................. 16  
CAE Video Re-addressing ............................................................... 16  
  Platform Requirements .......................................................... 17  
  License Requirements ......................................................... 17  
  HTTP X-Header Use in CAE Video Re-addressing ............... 17  
  Mobile Video Gateway to the CAE ........................................ 18  
  Video Detection using HTTP Payload ................................. 19  
  CAE Video Re-addressing Call Flows ................................... 19  
  Bulkstatistics Support ......................................................... 22  
  Limitations ............................................................................ 23  
CAE Load Balancing ................................................................. 23  
  CAE Load Balancer Function ............................................... 24  
  CAE Health-Check Monitoring Function ........................... 24  
Video Optimization Policy Control ............................................. 24  
  Functional Overview .......................................................... 25  
Video White-listing ................................................................. 26  
Video Pacing .................................................................................. 26  
  Platform Requirements ........................................................ 27  
  Video Pacing Operation ....................................................... 27  
  Video Pacing Functions ....................................................... 29  
  Video Pacing Call Flows ....................................................... 30  
  Interactions with Related Functions ................................... 31  
Supported Video Container File Formats .................................... 31  
TCP Link Monitoring ............................................................... 32  
  TCP Link Monitoring System Flow .................................... 32  
  Functional Overview .......................................................... 33  
Congestion Management ................................................................. 34
## Contents

URL Stripping .................................................................................................................. 34  
Dynamically-enabled TCP Proxy .................................................................................. 36  
N+1 Stateful Redundancy ............................................................................................ 36  
Mobile Video Statistics ................................................................................................. 37  
Bulk Statistics for Mobile Video ................................................................................... 38  
How the Mobile Video Gateway Works ......................................................................... 39  
Mobile Video Gateway with the Content Adaptation Engine ........................................ 39  
- DPI of HTTP GET Request Identifying a Non-Video Request (MVG with the CAE) .... 39  
- DPI of HTTP RESPONSE Identifying a Non-Video Request (MVG with the CAE) ..... 40  
- DPI of HTTP GET Request Identifying a Video Request (MVG with the CAE) ........ 41  
- DPI of HTTP RESPONSE Identifying a Video Request (MVG with the CAE) ........ 42  
Mobile Video Gateway without the Content Adaptation Engine ................................... 44  
- DPI of HTTP GET Request Identifying a Non-Video Request (MVG without the CAE) 44  
- DPI of HTTP RESPONSE Identifying a Non-Video Request (MVG without the CAE) 45  
- DPI of HTTP GET Request Identifying a Video Request (MVG without the CAE) ..... 46  
- DPI of HTTP RESPONSE Identifying a Video Request (MVG without the CAE) ..... 48  

### Mobile Video Gateway Configuration ......................................................................... 51

Configuring CAE Re-addressing and Load Balancing ..................................................... 52  
- Configuring the CAE Group .................................................................................... 52  
- Configuring CAE Re-addressing ............................................................................. 53  
- Configuring Video Optimization Pre-processing ..................................................... 54  
- Configuring Content-Range HTTP Header ................................................................ 54  
- Configuring Video Detection using HTTP Payload ................................................. 54  
- Configuring Logging for CAE Health-Check Monitoring ........................................ 55  
Sample CAE Re-addressing and Load Balancing Configuration .................................... 56  
Configuring Video Optimization Policy Control .......................................................... 59  
Sample Video Optimization Policy Control Configurations ......................................... 61  
- Obtaining the Video Policy via the PCRF over a Gx Interface ............................... 61  
- Obtaining the Video Policy via the RADIUS Server over a RADIUS Interface ....... 64  
Configuring Video White-listing ................................................................................... 69  
Configuring Video Pacing ........................................................................................... 70  
Sample Video Pacing Configuration ............................................................................ 71  
Configuring TCP Link Monitoring ............................................................................. 73  
Configuring Congestion Management ....................................................................... 74  
Configuring URL Stripping .......................................................................................... 76  
Configuring Mobile Video Statistics ............................................................................ 77  
Configuring Bulk Statistics .......................................................................................... 78  

### Monitoring the Mobile Video Gateway ........................................................................ 79

Monitoring System Status and Performance ................................................................ 80  
Clearing Statistics and Counters ................................................................................... 82
About this Guide

This preface describes the MVG Administration Guide, how it is organized and its document conventions.

Mobile Video Gateway (MVG) is a StarOS™ application that runs on Cisco® ASR 5x00 platforms.

MVG 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.
Conventions Used

The following tables describe the conventions used throughout this documentation.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Notice Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Information Note Icon]</td>
<td>Information Note</td>
<td>Provides information about important features or instructions.</td>
</tr>
<tr>
<td>![Caution Icon]</td>
<td>Caution</td>
<td>Alerts you of potential damage to a program, device, or system.</td>
</tr>
<tr>
<td>![Warning Icon]</td>
<td>Warning</td>
<td>Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.</td>
</tr>
</tbody>
</table>

Typeface Conventions

<table>
<thead>
<tr>
<th>Typeface Conventions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text represented as screen display</td>
<td>This typeface represents displays that appear on your terminal screen, for example: Login:</td>
</tr>
<tr>
<td>Text represented as commands</td>
<td>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.</td>
</tr>
<tr>
<td>Text represented as command variable</td>
<td>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.</td>
</tr>
<tr>
<td>Text represented as menu or sub-menu names</td>
<td>This typeface represents menus and sub-menus that you access within a software application, for example: Click the File menu, then click New</td>
</tr>
</tbody>
</table>
Supported Documents and Resources

Related Common Documentation

The following common documents are available:

- AAA Interface Administration and Reference
- Command Line Interface Reference
- GTPP Interface Administration and Reference
- Installation Guide (platform dependant)
- Release Change Reference
- SNMP MIB Reference
- Statistics and Counters Reference
- System Administration Guide (platform dependant)
- Thresholding Configuration Guide

Related Product Documentation

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

The following product documents are also available and work in conjunction with MVG:

- GGSN Administration Guide
- HA Administration Guide
- IPSG Administration Guide
- P-GW Administration Guide
- SAEGW 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 MVG documentation:
Products > Wireless > Mobile Internet > Platforms > Cisco ASR 5000 Series > Cisco ASR 5000
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.
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.

Depending on the feature, the Mobile Video Gateway software can run on Cisco ASR 5x00 platforms, enabling the chassis to function as an integrated Mobile Video Gateway. Depending on the feature, the Mobile Video Gateway software can be integrated with the Cisco Gateway GPRS Support Node (GGSN), the Cisco HA (Home Agent), the Cisco IP Services Gateway (IPSG), the Cisco Packet Data Network Gateway (P-GW), and the Cisco System Architecture Evolution Gateway (SAE-GW) to meet specific network deployment requirements. Specific platform compatibilities are mentioned in the feature sections that follow in this guide.

Qualified Platforms

MVG is a StarOS application that runs on Cisco ASR 5x00 platforms. For additional platform information, refer to the appropriate System Administration Guide 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.
The Mobile Video Gateway features and functions include:

- DPI (Deep Packet Inspection) to identify subscriber requests for video vs. non-video content
- CAE 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 white-listing, to exclude certain videos from video optimization.
- Video pacing for “just in time” video downloading
- TCP link monitoring
- Congestion management
- URL stripping
- Dynamically-enabled TCP proxy
- N+1 redundancy support
- 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.

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.

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.
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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCRF Interface</td>
<td>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.</td>
</tr>
<tr>
<td>RADIUS Interface</td>
<td>The Mobile Video Gateway uses a RADIUS interface to exchange signaling messages with the external RADIUS server.</td>
</tr>
<tr>
<td>Video Origin Server Interface</td>
<td>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.</td>
</tr>
<tr>
<td>CAE Interface</td>
<td>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.</td>
</tr>
</tbody>
</table>
Features and Functionality

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

- Deep Packet Inspection
- CAE Video Re-addressing
- CAE Load Balancing
- Video Pacing
- TCP Link Monitoring
- Congestion Management
- URL Stripping
- Dynamically-enabled TCP Proxy
- N+1 Stateful Redundancy
- 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.

CAE Video Re-addressing

Important: In 12.3 and 14.0 releases, the CAE 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 Mobile Video Gateway can re-address HTTP video requests intended for video content Origin Servers (OS) 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.
The CAE 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.

The TCP proxy is used only for connection initiation. The CAE re-addressing functionality is invoked only on the first HTTP video request and response, after which the rest of the packets do not need the proxy functionality, and TCP proxy is dynamically disabled as soon as a response packet is received from the CAE.

For persistent connections with continuous requests from the UE, once a connection is established with the CAE for processing the first video request, TCP proxy is disabled and the connection with the CAE is kept active until the end of the flow. Both video and non-video requests are processed by a CAE that is capable of processing both types of requests.

**Platform Requirements**

In 12.3 and 14.0 releases, the CAE Video Re-addressing feature is supported on GGSN, P-GW, HA, and IPSG products as a lab demo/lab trial feature only. This feature is not fully qualified and is not supported for field deployment.

In 15.0 and later releases, the MVG CAE Video Re-addressing feature is qualified to run on the Cisco ASR 5000 chassis for PSC2 and PSC3 integrated with GGSN, P-GW, and SAE-GW (if MVG is running on P-GW) products. To use this feature in a combination of GGSN, P-GW, and HA products, MVG should run on GGSN or P-GW.

**License Requirements**

CAE Video Re-addressing is a licensed Cisco feature. It requires the Video Optimization license or the ECS Header Enrichment license to insert specific x-headers in the HTTP message before forwarding the request to the CAE. 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.

**HTTP X-Header Use in CAE 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 CAE 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.
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**: (Mandatory) The IPv4 destination address and TCP port number of the OS.
- **x-congestion**: (Optional) Cell level congestion information in the readdressed HTTP request to CAE. Values can be between 0 and 4 as described below:
  - 0: No Congestion
  - 1: Light Congestion
  - 2: Medium Congestion
  - 3: Heavy Congestion
  - 4: Extreme Congestion
- **x-device-type**: (Optional) The mobile user device type. The value is a 15-digit IMEI number of the mobile device.
- **x-rat-type**: (Optional) The RAT type.
- **x-user-profile**: (Optional) The configured subscriber profile. CAE uses this information to select the optimization parameters on a subscriber group-basis.
- **x-ToD**: (Optional) Current date, time, and time zone offset of the subscriber. This information is used to select optimization profiles based on the time zone for the subscriber in the case where CAE covers multiple time zones. This x-header is inserted only if the time zone for the subscriber is available and the timezone is represented in GMT.
• **x-forwarded-for**: (Optional) The subscriber or client’s IPv4 address. This x-header is inserted when MVG acts as a non-transparent proxy to CAE.

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

The ECS HTTP Analyzer now supports parsing for content-range HTTP headers. This header is useful in detecting HTTP video requests when using ECS-DPI ruledefs based on HTTP headers/URI.

For configuration instructions and a sample configuration, see the *Mobile Video Gateway Configuration* chapter.

**Video Detection using HTTP Payload**

Readdressing to CAE is done only for HTTP video traffic. Non-video traffic will be sent to the origin server (until flow is readdressed to CAE for the first video request). The mechanism for detection of HTTP video traffic is based on inspection of HTTP payload for video signatures. Only Response payloads will be inspected and not request payloads. MVG supports video types such as FLV, ISOM (mp4, 3gp, 3g2, qt, f4v, m4v, etc.), moov, wmv, MPEG2_TS, AVI, and so on.

This feature has the following limitations:

• This mechanism may have some performance impact.

• Only the first packet of the response video header will be inspected. If video header is not complete in the first packet of the payload due to TCP fragmentation, then video detection may not happen by this mechanism.

• Chunk encoded video responses, and compressed/encoded videos may not be detected by this mechanism.

**CAE Video Re-addressing Call Flows**

**Video Request Re-addressing**

The UE requests an OTT video from the OS. In this case, the MVG recognizes that the request is for Video content and steers the request to CAE instead of the OS for optimization. CAE delivers the optimized video which is proxied back to UE by MVG.
Table 2. Video Request Re-addressing

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—3</td>
<td>UE sets up a TCP connection with Origin Server by sending SYN. The TCP three-way handshake takes place between UE and the Origin Server.</td>
</tr>
<tr>
<td>4</td>
<td>UE sends a HTTP request to the OS for “video_url” which passes through the MVG. This request could take the form of an HTTP GET.</td>
</tr>
<tr>
<td>5—8</td>
<td>MVG recognizes that the request is for video content. MVG breaks/closes the existing TCP connection with OS and establishes a new connection with CAE.</td>
</tr>
<tr>
<td>9</td>
<td>MVG now proxies the request with additional information like the congestion information etc. (x-headers) to the CAE that is configured to provide optimized video. CAE can then either send its stored copy of response or fetch it from OS (in case the response is not cached before or is cached but is stale).</td>
</tr>
<tr>
<td>10</td>
<td>CAE processes, optimizes, and streams the video in real time back to the MVG.</td>
</tr>
<tr>
<td>11</td>
<td>MVG proxies the content back to the UE.</td>
</tr>
</tbody>
</table>
The UE requests an OTT video from the OS. The MVG based on response from OS recognizes that it is configured to steer to the CAE for optimization. The UE request is then passed to CAE which delivers transcoded / transrated content, which in turn is delivered to the UE.

**Figure 8. Video Response Re-addressing**
Table 3. Video Response Re-addressing

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—3</td>
<td>UE sets up a TCP connection with Origin Server by sending SYN. The TCP three-way handshake takes place between UE and the Origin Server.</td>
</tr>
<tr>
<td>4</td>
<td>UE sends a HTTP request to the OS for “video_url” which passes through the MVG. This request could take the form of an HTTP GET.</td>
</tr>
<tr>
<td>5</td>
<td>MVG sends the request to the origin server.</td>
</tr>
<tr>
<td>6—11</td>
<td>MVG analyzes the response and checks the response for video content. If the response contains video content, the MVG breaks/closes the existing connection with OS and establishes a new connection with CAE. The request with additional information like the congestion information etc. (x-headers) is then steered to one of the CAE servers configured on the system. The CAE can either send its stored copy of response or fetch it from OS (in case the response is not cached before or is cached but is stale).</td>
</tr>
<tr>
<td>12</td>
<td>CAE processes, optimizes, and streams the video in real time back to the MVG.</td>
</tr>
<tr>
<td>13</td>
<td>MVG proxies the content back to the UE.</td>
</tr>
<tr>
<td>14</td>
<td>UE closes the TCP connection.</td>
</tr>
<tr>
<td>15</td>
<td>MVG closes the connection with CAE.</td>
</tr>
</tbody>
</table>

**Bulkstatistics Support**

Bulkstatistics reporting for the CAE Video Re-addressing feature is supported and the following bulkstatistics are available in the ECS schema:

- video-readdress-req-redirected
- video-readdress-res-redirected
- video-readdress-req-with-xheader-inserted
- video-readdress-upl-bytes-redirected
- video-readdress-upl-pkts-redirected
- video-readdress-dnl-bytes-redirected
- video-readdress-dnl-pkts-redirected
- video-readdress-req-charging-action-hit
- video-readdress-resp-charging-action-hit
- video-readdress-skipped-pipelined-reqs
- video-readdress-connect-failed-to-video-server
- video-readdress-load-balancer-failures
- video-readdress-flows-connected-to-video-server
- video-readdress-xhdr-insert-failed
For more information on these bulk statistics, refer to the *ECS Schema Statistics* chapter in the *Statistics and Counters Reference* guide.

**Limitations**

The following are limitations to using the CAE Video re-addressing feature:

- CAE video re-addressing cannot be used in parallel with the ECS IP re-addressing feature.
- CAE video re-addressing is not compatible with various ECS charging features under the **flow action** configuration ("Flow Readdress", "Flow Terminate", "Flow Discard", and "URL Redirect")

**CAE Load Balancing**

The CAE Load Balancing feature brings 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.

The following figure shows the CAE in a server cluster.

![CAE Server Cluster Diagram]

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 the Mobile Video Gateway Configuration chapter.

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

**Figure 10. 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.

This works just like for other ASR 5000 features. Ruledefs that trigger MVG features (CAE-Readdressing) can be activated/deactivated from PCRF by having them predefined.

For configuration instructions and a sample configuration, see the *Mobile Video Gateway Configuration* chapter. For detailed instructions for configuring the Gx interface on the Cisco P-GW, see the *Packet Data Network Gateway Administration Guide*.

### 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.

### 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.

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.
Platform Requirements

In 12.2 and 12.3 releases, the Video Pacing feature is qualified to run on the Cisco ASR 5000 chassis for PSC2 and PSC3 integrated with GGSN, P-GW, SAE-GW (if MVG is running on P-GW), HA, and IPSG products. To use this feature in a combination of GGSN, P-GW, and HA products, MVG should run on GGSN or P-GW.

In 14.0 and later releases, in addition to the existing ASR 5000 chassis and its supported products, the Video Pacing feature is also qualified to run on the ASR 5500 chassis integrated with GGSN, P-GW, SAE-GW (if MVG is running on P-GW), and HA products. To use this feature in a combination of GGSN, P-GW, and HA products, MVG should run on GGSN or P-GW.

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.
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.
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**
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.

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.

For configuration instructions and a sample configuration, see the Mobile Video Gateway Configuration chapter.

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.
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.

### Congestion Management

Congestion can happen more frequently in a cellular network than in a wired network due to various factors such as airlink interface, high RTT for wireless connections. In this release, MVG supports TCP link state monitoring and estimation of congestion level at subscriber side airlink interface. The congestion level thus estimated can be sent to CAE during the start of a video flow. Based on this information, CAE can decide to stream video appropriately optimized for the subscriber side network. If the first flow from the subscriber itself is video, that flow cannot have congestion level value. The Congestion Management feature makes use of the existing Link Monitoring feature. So to enable congestion management, link monitoring has to be enabled.

Congestion detection is based on all types of data traffic - video and non-video, and works only for TCP traffic. CLI support is provided to configure the parameters to interpret the congestion indications per TCP flow and correlate them for a subscriber to allow experimentation. Congestion sampling time and reporting frequency can also be configured. Congestion monitoring is done in the downlink direction only.

CLI support is provided to enable or disable the Congestion Management feature at either APN or subscriber.

### URL Stripping

The URL Stripping feature allows the ASR5000 to strip a specific URL argument from the URL, based on ECS configuration. A CLI command in the ACS Charging Action Configuration mode is defined to specify the URL token to be stripped.

For the following URL, `http://www.videoserver.com?Name1=val1&Name2=val2&Name3=val3` if the CLI command is applied, this will strip parameter `name2` and its optional value `val2` from the above URL to give the following new URL: `http://www.videoserver.com?Name1=val1&Name3=val3`. If the token name does not match, then charging action will not be applied.

In case of CAE readdressing, to avoid recursive redirection of requests to the CAE, it is mandatory that the rule used to match stripurl charging action must have higher priority than the rule used to match CAE readdressing charging action. This will make sure that if the URL contains a configured token, the request will not be readdressed again and the token will be stripped by URL stripping charging action before request is sent to the origin server.

Limitations of URL stripping:

- URL stripping has a limitation in case of partial packets. In cases where the HTTP Request header is split into multiple packets, the token will not be stripped from URL while forwarding the request to the origin server. This is because ECS does not buffer the HTTP packets for rule matching and added buffering will have performance impact.

- URL stripping must have higher priority than Request-Readdressing in the rulebase. Otherwise even if the URL contains the token, Request-Readdressing will happen before token is stripped.

- URL Stripping cannot be used with Response-Readdressing feature. Once the URL is stripped and sent to the new destination, the response may contain video. Hence if Response-Readdressing is enabled, it will be readdressed back to the CAE server causing loop condition.
The following call flow is an example to show how the URL stripping feature can be used.

**Table 4. URL Stripping**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1—3</td>
<td>UE sets up TCP connection with the Origin server using TCP Handshake.</td>
</tr>
<tr>
<td>4</td>
<td>UE sends a HTTP request for “video_url” over the established connection. The request is analyzed using DPI.</td>
</tr>
<tr>
<td>5</td>
<td>The request is forwarded to the Origin server for fetching the content as MVG could not decide if it is a video request.</td>
</tr>
<tr>
<td>6</td>
<td>MVG receives a response from the Origin server and inspects the response for size, content type and other parameters to detect whether it is a video stream. MVG detects the HTTP request as a video stream.</td>
</tr>
</tbody>
</table>
### Mobile Video Gateway Overview

#### Features and Functionality

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7—9</td>
<td>MVG closes the TCP connection towards the Origin server. MVG selects a CAE for processing the video stream and sets up a TCP connection. MVG forwards the request to the CAE. MVG adds the following x-headers into the HTTP request:</td>
</tr>
<tr>
<td></td>
<td>• X-forwarded-for (subscriber/client IPv4 address)</td>
</tr>
<tr>
<td></td>
<td>• X-forwarded-dest-addr-port (IPv4 address of Origin Server and TCP port of the HTTP request to Origin Server, separated by ';')</td>
</tr>
<tr>
<td></td>
<td>• X-congestion (congestion level at the MVG)</td>
</tr>
<tr>
<td>10</td>
<td>CAE decides that it cannot provide the requested video and that video must be got from the Origin server (or another CAE).</td>
</tr>
<tr>
<td>11</td>
<td>CAE responds with a HTTP redirect (302 Response) and a URL including the UUID (known to ASR5x00 and CAE).</td>
</tr>
<tr>
<td>12</td>
<td>MVG forwards the Redirect Response to the UE.</td>
</tr>
<tr>
<td>13—14</td>
<td>UE closes the current TCP connection with MVG and CAE.</td>
</tr>
<tr>
<td>15</td>
<td>UE sets up a new TCP connection with Origin Server and sends the HTTP request to the redirect location. The URL includes the UUID sent by CAE before.</td>
</tr>
<tr>
<td>16</td>
<td>MVG intercepts the request and detects the token (UUID) in the URL. It understands that this request must not be readdressed and must be sent to its original destination. MVG strips the UUID off the URL and forwards the request to the Origin server.</td>
</tr>
<tr>
<td>17</td>
<td>Origin server sends the data back to the MVG.</td>
</tr>
<tr>
<td>18</td>
<td>MVG forwards the data back to the UE.</td>
</tr>
</tbody>
</table>

### Dynamically-enabled TCP Proxy

The Mobile Video Gateway can act as a dynamically-enabled TCP proxy that provides the CAE video re-addressing functionality.

This feature requires 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*.

### 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.

### 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 and a sample configuration, see the *Mobile Video Gateway Configuration* chapter. 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 and a sample configuration, see the *Mobile Video Gateway Configuration* chapter.
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 17. DPI of HTTP GET Request Identifying a Non-Video Request

Table 5. DPI of HTTP GET Request Identifying a Non-Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP 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).</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>The OS responds with an HTTP 200 OK, including the content of the page.</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 18. DPI of HTTP RESPONSE Identifying a Non-Video Request

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>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).</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 19. DPI of HTTP GET Request Identifying a Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. TCP connection</td>
<td>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.</td>
</tr>
<tr>
<td>2. HTTP GET (URL)</td>
<td></td>
</tr>
<tr>
<td>3. TCP connection</td>
<td></td>
</tr>
<tr>
<td>4. HTTP GET (x-headers)</td>
<td>The CAE creates a TCP proxy connection with the OS and sends the original UE request to the OS.</td>
</tr>
<tr>
<td>5. TCP connection</td>
<td></td>
</tr>
<tr>
<td>6. HTTP GET (URL)</td>
<td></td>
</tr>
<tr>
<td>7. HTTP RESPONSE (OK)</td>
<td>The CAE processes the HTTP RESPONSE packets.</td>
</tr>
<tr>
<td>8. Video data</td>
<td>HTTP RESPONSE (OK)</td>
</tr>
</tbody>
</table>

Table 7. DPI of HTTP GET Request Identifying a Video Request
How the Mobile Video Gateway Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a video request (the DPI on GET/POST succeeds).</td>
</tr>
<tr>
<td>3.</td>
<td>The TCP connection gets proxied. The Mobile Video Gateway closes the TCP connection with the OS and opens a new one with the CAE.</td>
</tr>
<tr>
<td>4.</td>
<td>The Mobile Video Gateway sends the original HTTP GET request to the CAE with x-headers for transport, quality, and UE identity.</td>
</tr>
<tr>
<td>5.</td>
<td>The CAE creates a TCP proxy connection with the OS.</td>
</tr>
<tr>
<td>6.</td>
<td>The CAE sends the original HTTP GET request from the UE to the OS.</td>
</tr>
<tr>
<td>7.</td>
<td>The CAE processes the HTTP RESPONSE packets from the OS and performs video optimization.</td>
</tr>
<tr>
<td>8.</td>
<td>The Mobile Video Gateway performs additional video optimization and sends the optimized packets to the UE.</td>
</tr>
</tbody>
</table>

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 20. DPI of HTTP RESPONSE Identifying a Video Request

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)

10. Video data

Video data

Video data

Table 8. DPI of HTTP RESPONSE Identifying a Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>

MVG Administration Guide, StarOS Release 18
## Mobile Video Gateway Overview

- **How the Mobile Video Gateway Works**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>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).</td>
</tr>
<tr>
<td>5.</td>
<td>The TCP connection gets proxied. The Mobile Video Gateway closes the TCP connection with the OS and opens a new one with the CAE.</td>
</tr>
<tr>
<td>6.</td>
<td>The Mobile Video Gateway sends the original HTTP GET request to the CAE with x-headers for transport, quality, and UE identity.</td>
</tr>
<tr>
<td>7.</td>
<td>The CAE creates a TCP proxy connection with the OS.</td>
</tr>
<tr>
<td>8.</td>
<td>The CAE sends the original HTTP GET request from the UE to the OS.</td>
</tr>
<tr>
<td>9.</td>
<td>The CAE processes the HTTP RESPONSE packets from the OS and performs video optimization.</td>
</tr>
<tr>
<td>10.</td>
<td>The Mobile Video Gateway performs additional video optimization and sends the optimized packets to the UE.</td>
</tr>
</tbody>
</table>

### 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 21. DPI of HTTP GET Request Identifying a Non-Video Request**

---

**MVG Administration Guide, StarOS Release 18**
Table 9. DPI of HTTP GET Request Identifying a Non-Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP 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).</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>The OS responds with an HTTP 200 OK, including the content of the page.</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 22. DPI of HTTP RESPONSE Identifying a Non-Video Request

Table 10. DPI of HTTP RESPONSE Identifying a Non-Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives a HTTP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether this is a video request.</td>
</tr>
<tr>
<td>3.</td>
<td>The Mobile Video Gateway performs DPI again and identifies this as a non-video request.</td>
</tr>
<tr>
<td>4.</td>
<td>The Mobile Video Gateway forwards the HTTP RESPONSE (OK) to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.</td>
</tr>
<tr>
<td>5.</td>
<td>The Mobile Video Gateway forwards the HTTP RESPONSE (OK) to the UE transparently. The connection is not proxied, and the TCP flow continues to the UE.</td>
</tr>
</tbody>
</table>
### How the Mobile Video Gateway Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>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).</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.
### Table 11. DPI of HTTP GET Request Identifying a Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and identifies it as a video request (the DPI on GET/POST succeeds).</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
</tbody>
</table>
How the Mobile Video Gateway Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>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.</td>
</tr>
<tr>
<td>5.</td>
<td>The Mobile Video Gateway forwards the HTTP 200 OK to the UE.</td>
</tr>
<tr>
<td>6.</td>
<td>The optimized TCP video flow continues to the UE.</td>
</tr>
</tbody>
</table>

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 24. DPI of HTTP RESPONSE Identifying a Video Request
Table 12. DPI of HTTP RESPONSE Identifying a Video Request

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE creates a TCP connection with the OS.</td>
</tr>
<tr>
<td>2.</td>
<td>The Mobile Video Gateway receives an HTTP GET request from the UE. The Mobile Video Gateway performs DPI and cannot determine whether it is a video request.</td>
</tr>
<tr>
<td>3.</td>
<td>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.</td>
</tr>
<tr>
<td>4.</td>
<td>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.</td>
</tr>
<tr>
<td>5.</td>
<td>The Mobile Video Gateway forwards the HTTP 200 OK to the UE.</td>
</tr>
<tr>
<td>6.</td>
<td>The optimized TCP video flow continues to the UE.</td>
</tr>
</tbody>
</table>
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
- Sample CAE Re-addressing and Load Balancing Configuration
- Configuring Video Optimization Policy Control
- Sample Video Optimization Policy Control Configurations
- Configuring Video White-listing
- Configuring Video Pacing
- Sample Video Pacing Configuration
- Configuring TCP Link Monitoring
- Configuring Congestion Management
- Configuring URL Stripping
- Configuring Mobile Video Statistics
- Configuring Bulk Statistics
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:

```plaintext
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:

```bash
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 Content-Range HTTP Header

Use the following command to define rule expressions for CAE re-addressing to verify if the HTTP Response has content-range header or not.

```bash
configure
  active-charging service acs_name
  ruledef ruledef_name
    [ no ] http content range = TRUE
  end
```

Configuring Video Detection using HTTP Payload

Readdressing to CAE is done only for HTTP video traffic. Non-video traffic will be sent to the origin server (until flow is readdressed to CAE for the first video request). Only Response payloads will be inspected and not request payloads.

Use this command to define rule expressions to enable video detection using HTTP payload content.

```bash
configure
```
active-charging service  

ruledef  

[ no ] http reply payload type = video 

end

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.

```plaintext
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 readdressing. 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-rulledef priority 1 ruledef http_youtube
    add-rulledef 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

calendar 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
```
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 Congestion Management

The congestion management feature is configured and enabled via an `active-charging` command option in either APN Configuration Mode, Subscriber Configuration Mode, or Radio Congestion Policy Configuration Mode.

Use the following commands to enable or disable the congestion management feature in APN Configuration Mode or Subscriber Configuration Mode:

```
configure
  require active-charging

context context_name
  subscriber default
    active-charging radio-congestion policy policy_name

  exit

exit

context context_name
apn cisco.com
  active-charging radio-congestion policy policy_name
end
```

Use the following commands to create/configure/delete Radio Congestion policy in the Radio Congestion Policy Configuration mode.

```
configure
  require active-charging

active-charging service <service_name>
  radio-congestion policy <policy_name> [ -noconfirm ]

    congestion-level low <low_value> medium <medium_value> high <high_value> extreme <extreme_value>

    correlation-method { mean | optimistic | pessimistic }

    data-loss threshold <threshold_value> weightage <weightage_value>

    sampling-interval <sampling_interval>

    reporting-interval <interval_value> min-samples-required <num_samples>

    rtt-samples <min_samples>
```
rtt-variance threshold <variance_percent> weightage <rtt_weightage>

end

For more information on the above configuration, refer to the Command Line Interface Reference.
Configuring URL Stripping

The URL stripping feature is configured via 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 URL stripping:

```plaintext
configure
  require active-charging
  active-charging service <service_name>
    charging-action <charging_action_name>
      stripurl token <token_name> [ value <token_value> ]
    no stripurl
  end
```

The `require active-charging` command enables active charging on the Mobile Video Gateway. The `active-charging service <service_name>` 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 <charging_action_name>` 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 `stripurl token <token_name>` command allows to configure the token and value to be stripped from the HTTP URL.

For more configuration information, refer to 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:

```plaintext
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 <service_name>` 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 predefined 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 provide 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 13. Mobile Video Gateway Status and Performance Monitoring Commands

<table>
<thead>
<tr>
<th>To do this:</th>
<th>Enter this command:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View Information on CAE Re-addressing and Load Balancing</strong></td>
<td></td>
</tr>
<tr>
<td>Display CAE re-addressing statistics.</td>
<td><code>show active-charging flows summary cae-readdressing</code></td>
</tr>
<tr>
<td>Display CAE configuration information, including the name of the corresponding CAE group, for all CAEs or for a specific CAE.</td>
<td>`show cae-group server { all</td>
</tr>
<tr>
<td><strong>View Information on URL-based Re-addressing</strong></td>
<td></td>
</tr>
<tr>
<td>Display statistics for URL-based addressing.</td>
<td><code>show active-charging charging-action statistics name</code></td>
</tr>
<tr>
<td></td>
<td><code>show active-charging rulebase statistics name</code></td>
</tr>
<tr>
<td><strong>View Information on Video Optimization Policy Control</strong></td>
<td></td>
</tr>
<tr>
<td>Display information on groups of rule definitions configured for tiered video policies (Gold, Silver, and Bronze levels, for example) in Active Charging Services.</td>
<td><code>show active-charging group-of-ruledefs statistics</code></td>
</tr>
<tr>
<td>Display a list of subscribers that are assigned to a particular rulebase, such as GOLD_RBASE, for example.</td>
<td><code>show active-charging sessions full rulebase name</code></td>
</tr>
<tr>
<td>Display the suggested maximum bit rate value configured for each charging action, which determines the video policy.</td>
<td><code>show active-charging charging-action name name</code></td>
</tr>
<tr>
<td><strong>View Information on Video Pacing</strong></td>
<td></td>
</tr>
<tr>
<td>Display information on TCP video flows that have been paced.</td>
<td><code>show active-charging flows</code></td>
</tr>
<tr>
<td>Display detailed statistics on TCP video flows that have been paced.</td>
<td>`show active-charging video detailed-statistics [ container { flv</td>
</tr>
<tr>
<td>Display statistics on video pacing and TCP video flows.</td>
<td><code>show active-charging subsystem all</code></td>
</tr>
<tr>
<td><strong>View Information on TCP Link Monitoring</strong></td>
<td></td>
</tr>
<tr>
<td>Display statistics on the average TCP throughput and RTT (Round Trip Time) of downlink TCP traffic towards the subscriber UE.</td>
<td><code>show active-charging flows full</code></td>
</tr>
<tr>
<td></td>
<td><code>show active-charging sessions full</code></td>
</tr>
<tr>
<td>To do this:</td>
<td>Enter this command:</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **View Information on Congestion Management** | show active-charging radio-congestion policy all  
| Display statistics for Congestion Management feature if a subscriber or APN is enabled. | show active-charging radio-congestion policy statistics  
|                                   | show active-charging sessions full all                                                  |
| **View Information on URL Stripping** | show active-charging charging-action statistics name                                   |
| Display statistics for URL Stripping feature. |                                                                                  |
| **View Information on Active Charging Services** | show active-charging subsystem                                                      |
| Display service and configuration statistics for the Active Charging Services, including statistics for video-related Active Charging Services. |                                                                                  |
| **View Bulk Statistics for the Mobile Video Gateway** | show bulkstats variables mvs                                                       |
| Display bulk statistics for the Mobile Video Gateway. |                                                                                  |
| 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.