Cisco ASR 5000 Series Traffic Performance Optimization Administration Guide

Version 12.0

Last Updated April 30, 2011

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

Text Part Number: OL-24897-01

Cisco ASR 5000 Series Traffic Performance Optimization Administration Guide

CONTENTS

About this Guide v
Conventions Used vi
Contacting Customer Support viii
Traffic Performance Optimization Overview 9
Overview 10
TPO Deployment 10
Feature Specifications 13
Licensing 13
Supported Standards 13
TCP Optimization 13
TCP Optimization Techniques 13
HTTP Optimizations 16
HTTP Optimization Techniques 17
HTTP Compression 17
URL Rewrite 19
Advertisement Filter 22
TPO Administration 28
Disabling/Enabling TPO Optimizations 28
MUR Reporting for TPO 28
Switching TPO Policy 29
How TPO Works 30
Terms and Definitions 30
TPO Processing 30
Policy-based TPO Processing 31
Charging Action Based TPO Processing 31
Traffic Performance Optimization Configuration 33
Before You Begin 34
Configuring TPO 35
Creating and Configuring TPO Profiles 35
Creating TPO Profiles 35
Configuring TPO Profiles 35
Configuring DNS Client for URL Rewrite Feature 37
Creating and Configuring TPO Policies 37
Creating TPO Policies 37
Configuring TPO Policies 38
Applying TPO Policies to Subscribers/APNs 38
Applying TPO Policies to Subscribers 38
Applying TPO Policies to APNs 39
Configuring Default TPO Policy for Subscribers 39
Configuring TPO Profile in ECS Charging Action 39
TPO Administration and Other Configurations 40
Changing TPO Policy in Mid Session 40
Disabling/Enabling TPO Optimizations for P2P Flows 40
Verifying Your Configuration 42
Verifying and Saving Your Configuration 43

Verifying the Configuration 44
Feature Configuration 44
Service Configuration 45
Context Configuration 46
System Configuration 46
Finding Configuration Errors 46
Saving the Configuration 48
Saving the Configuration on the Chassis 49
Traffic Performance Optimization Sample Configuration 51

About this Guide

This document pertains to the features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

Conventions Used

The following tables describe the conventions used throughout this documentation.

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

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

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[**keyword** or *variable*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
|	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: **{ nonce | timestamp }** OR [**count** *number_of_packets* | **size** *number_of_bytes*]

Contacting Customer Support

Use the information in this section to contact customer support.

For New Customers: 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 is required to this site. Please contact your local sales or service representative for additional information.

For Existing Customers with support contracts through Starent Networks: Refer to the support area of https://support.starentnetworks.com/ for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

Important: For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

Chapter 1
Traffic Performance Optimization Overview

This chapter provides an overview of the Traffic Performance Optimization (TPO) in-line service.

TPO is a licensed in-line service supported on the Cisco ASR 5000 chassis running any of the following products:

- GGSN
- HA
- P-GW

The following topics are covered in this chapter:

- Overview
- Feature Specifications
- How TPO Works

Overview

Though TCP is a widely accepted protocol in use today, it is optimized only for wired networks. Due to inherent reliability of wired networks, TCP implicitly assumes that any packet loss is due to network congestion and consequently invokes congestion control measures. However, wireless links are known to experience sporadic and usually temporary losses due to several reasons, including the following, which also trigger TCP congestion control measures resulting in poor TCP performance.

Reasons for delay variability over wireless links include:

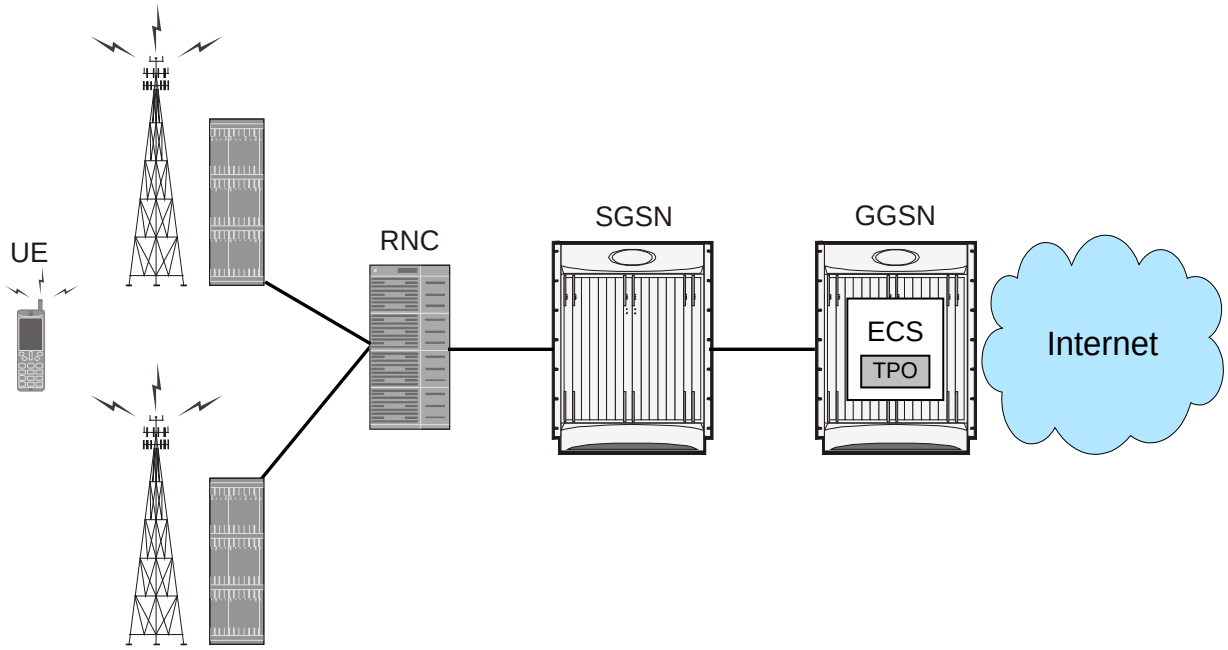
- Channel fading effect, subscriber mobility, and other transient conditions
- Link-layer retransmissions
- Handoffs between neighboring cells
- Intermediate nodes, such as SGSN and e-NodeB, implementing scheduling polices tuned to deliver better QoS for select services — resulting is variable delay in packet delivery for other services

The TPO in-line service uses a combination of TCP and HTTP optimization techniques to improve TCP performance over wireless links.

TPO Deployment

TPO uses the Enhanced Charging Service (ECS) framework for TCP Proxy functionality, and as depicted in the following figure, it is part of the ECS module in the Gateway.

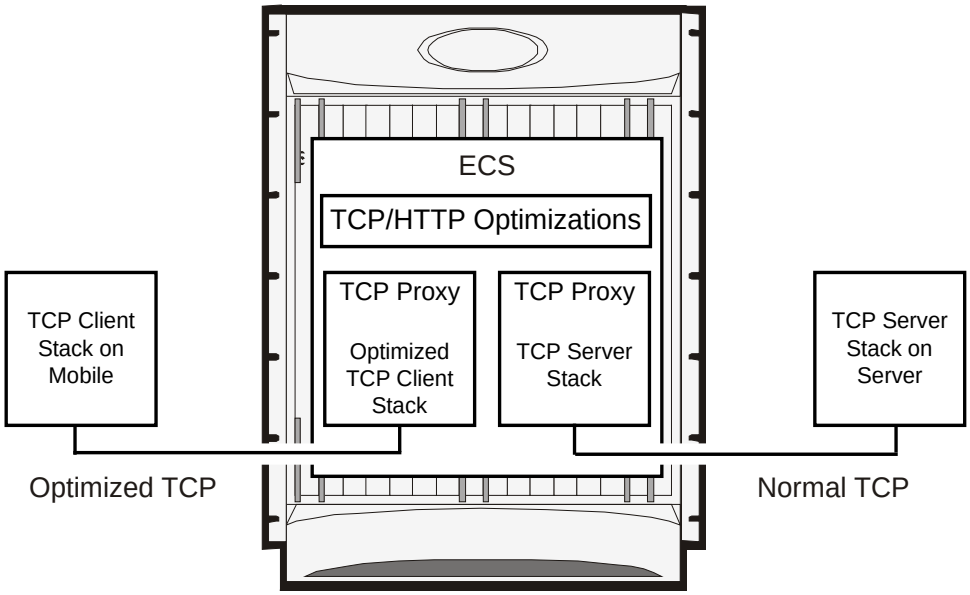
Figure 1. TPO Deployment



TPO uses the TCP Proxy to split end-to-end TCP connections between the client and server into two separate TCP connections, and apply the TCP and HTTP optimization techniques in the TCP stack towards the client. The split TCP connections isolate impacts of packet errors and delay variability for the wireless link from the wired connection, so that TCP congestion control, timeout, and retransmission mechanisms in the wired link do not suffer from the fluctuating quality of the radio channel.

Important: In this release TPO optimizes only downlink radio usage.

Figure 2. TPO Optimization Model



Feature Specifications

This section describes features of the TPO in-line service.

Licensing

TPO is a licensed in-line service feature requiring the following license to be installed on the chassis:

Cisco PID [ASRK-00-CS0ITRPO] *Traffic Packet Optimization,1K sessions*, or Starent part number [600-00-7523] *Optimization*.

For information on obtaining and installing licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

Supported Standards

TPO is based on the following RFCs and standards:

- RFC 793 Transmission Control Protocol; 1981-09
- RFC 2616 Hypertext Transfer Protocol — HTTP/1.1; 1999-06
- RFC 3481 TCP over Second (2.5G) and Third (3G) Generation Wireless Networks; 2003-02

TCP Optimization

To improve TCP performance over wireless links TPO uses TCP optimization techniques that enable:

- Improved downlink throughput during periods of congestion
- Closer to theoretical bandwidth utilization by reducing TCP overheads
- Adoption to wireless network events and conditions optimizing data transmission rate for available bandwidth and RTT
- Minimized TCP retransmissions

TCP Optimization Techniques

This section describes the TCP optimization techniques supported by TPO.

Selection of the optimization techniques to gain TCP performance depends on the network characteristics and the prevalent wireless conditions. To trigger appropriate congestion control techniques for a given situation based on the wireless events, TPO utilizes available information such as subscriber QoS, system load, and so on.

Optimizations in TCP Three-way Handshake

During TCP Three-way Handshake, the TCP client and server negotiate to establish a connection. TCP requires three round-trip time (RTT) measurements between the client and server before data transfer is initiated. Determining the RTT adds extra time for each wireless connection.

TPO supports the following optimizations during TCP Three-way Handshake:

- TCP options such as SACK and timestamp are negotiated.
- RTT is calculated based on Timestamp.

Optimizations in TCP Slow Start Phase

During TCP Slow Start phase, to discover available bandwidth for a connection, TCP calculates the lowest possible bandwidth and increases exponentially until a packet loss is detected. In wireless environments, this phase implies periods during which the link is under-utilized and perceived by the subscribers as slow.

TPO supports the following fast start techniques:

- TPO uses subscriber QoS settings to set the initial congestion window.

  The subscriber QoS information is received from the AAA/OCS.
- TPO uses information from other TCP connections to set the initial congestion window, which is based on the RTT and used bandwidth for the other connection(s).
- TPO allows to configure the initial congestion window to any of the following values:
  - Units of 1 through 255 MSS segments.
  - A dynamically computed value at runtime, which is calculated as a percentage of bandwidth-delay product (BDP). The bandwidth is a CLI-configured value, and the delay is calculated using the SYN-ACK exchange.
  - A value recommended by RFC 5681, which varies from 2 through 4 based on MSS. This the default setting.

Optimizations in TCP Congestion Avoidance Phase

In the TCP Congestion Avoidance phase, TCP after detecting available bandwidth for a new connection linearly adjusts the congestion window to discover incremental bandwidth.

TPO allows to configure any of the following congestion control algorithms:

- TCP Westwood Plus: The TCP Westwood Plus algorithm can significantly increase throughput over wireless links. It relies on end-to-end bandwidth estimation to discriminate the cause of packet loss — congestion or wireless channel effect. The bandwidth estimate is determined by measuring and averaging the rate of

returning acknowledgements. This estimate is then used to compute the congestion window and slow start threshold after a congestion episode.

- Vegas: The Vegas algorithm instead of relying on detection of packet loss, uses a bandwidth estimation scheme to proactively gauge network congestion. The Sender watches for signs of congestion is setting in, such as RTT growing and sending rate flattening.
- Basic: This is a custom TCP congestion algorithm based on the TCP Reno algorithm. This is the default setting.

Fast Retransmits

To guarantee reliability of transfers, TCP requires the Receiver to respond to the Sender with an acknowledgment for each segment it receives. The Sender keeps a record of each segment it sends, and waits for an acknowledgment before sending the next segment. If the Sender does not receive an acknowledgment within the timeout period, under the assumption that the segment was lost in the network, the Sender fast-retransmits that segment (that is, retransmit without waiting for retransmission timeout (RTO)).

The Receiver will generate duplicate acknowledgements for every out-of-order (OOO) packet it receives. If the Sender receives three duplicate acknowledgements with the same acknowledgement number (that is, a total of four acknowledgements with the same acknowledgement number), the Sender decides that the segment with the next higher sequence number was dropped, and will not arrive out of order. The Sender will then fast-retransmit that segment.

TPO supports the following optimization with fast retransmits:

- TPO allows to configure the number of duplicate acknowledgements that will trigger fast retransmits. This can be:
  - Static value: When high amount of re-ordering is present in the network the static threshold of three duplicate acknowledgements does not work well. Under such conditions a higher static value is required as the threshold. This is a CLI-configurable parameter and can be a value 1 through 10.

    However, note that a higher static value will sometimes lead to not reaching the threshold because of less number of in-flight packets (which will roughly determine the number of duplicate ACKs received by the sender).
  - Dynamic value: The duplicate acknowledgement threshold is dynamically computed at runtime based on the number of in-flight packets (one-third of the in-flight packets, subject to a minimum of two). This will enable to adapt in networks where high amount of packet re-ordering is observed.

TCP Handoff Optimization

TPO supports intra-tech and inter-tech handoff events.

When an intra-tech handoff is detected, TPO takes the following actions:

1. Restarts RTT/RTO calculation based on first packet sent after the handoff.
2. Ignores duplicate acknowledgements as congestion event. Considers duplicate acknowledgements as packet drops due to handoff for next one RTT.
3. If RTO is triggered after the handoff event, for the next RTT, ignores congestion window adjustment.
4. If RTO happens and then handoff event is received, undoes congestion window due to RTO.
5. Restarts BWE for Vegas and Westwood congestion control algorithms.

6. Handles one handoff event per X RTT only.

When an inter-tech handoff is detected, TPO takes the following actions:

- Handoff Pre-event: As soon as bearer creation request is received at the gateway:
  1. Stops forwarding packets when handoff pre-event is received.
  2. Stops RTO and retransmission.
- Handoff Post-event: When bearer creation is complete at PGW/GGSN, restarts congestion control algorithm. This will take care of resending old unacknowledged packets.

Enabling/disabling TCP Handoff Optimization is a CLI-configurable parameter.

Retransmission Timeout Optimizations

TPO allows to configure the scaling factor for Round Trip Time Variation (RTTVAR). The configured scaling factor is used as a power of 2, so values of 0 through 4 correspond to 1, 2, 4, 8, and 16. In TCP RTO is calculated using the following formula:

$RTO = SRTT + K*RTTVAR$

where:

- *SRTT* = mean Round Trip Time (RTT)
- *RTTVAR* = Round Trip Time Variation

As wireless networks exhibit high RTT variation, the value of K is configurable. The value of K decides the extent to which Retransmission Timeout (RTO) timer depends on RTT variance. If RTT variance is higher, then K should be higher. The default RTTVAR scaling value, as recommended by RFC 2988, is 2.

TPO also allows to configure the RTO retransmission back-off factor. Once RTO fires for a packet, TCP will retransmit that packet and set the RTO to be a factor X, which is CLI-configurable, of the previous RTO. The default RTO backoff value, as recommended by RFC 5681 is 2.0.

HTTP Optimizations

To optimize incoming HTTP payload over TCP connections TPO uses HTTP optimization techniques that enable:

- Reducing payload by compressing text-based Web pages (packet headers are not compressed), albeit such pages do not account for much traffic and Web servers themselves could do the compression
- Minimizing number of round trips from clients, enabling faster response time
- Reducing payload by filtering advertisements

HTTP Optimization Techniques

This section describes HTTP optimization techniques supported by TPO.

HTTP Compression

The HTTP Compression feature enables to compress HTTP content transferred to the mobile client. HTTP Compression, by reducing the amount of traffic to be transported, enables better use of available bandwidth and improves transmission speeds.

Important: In this release TPO supports only the standards-based gzip compression algorithm.

Note that TPO will not attempt compression in the following cases:

- If the mobile client cannot accept compressed encodings
- If the HTTP version used is earlier than 1.1
- If the response from the Web server is already compressed

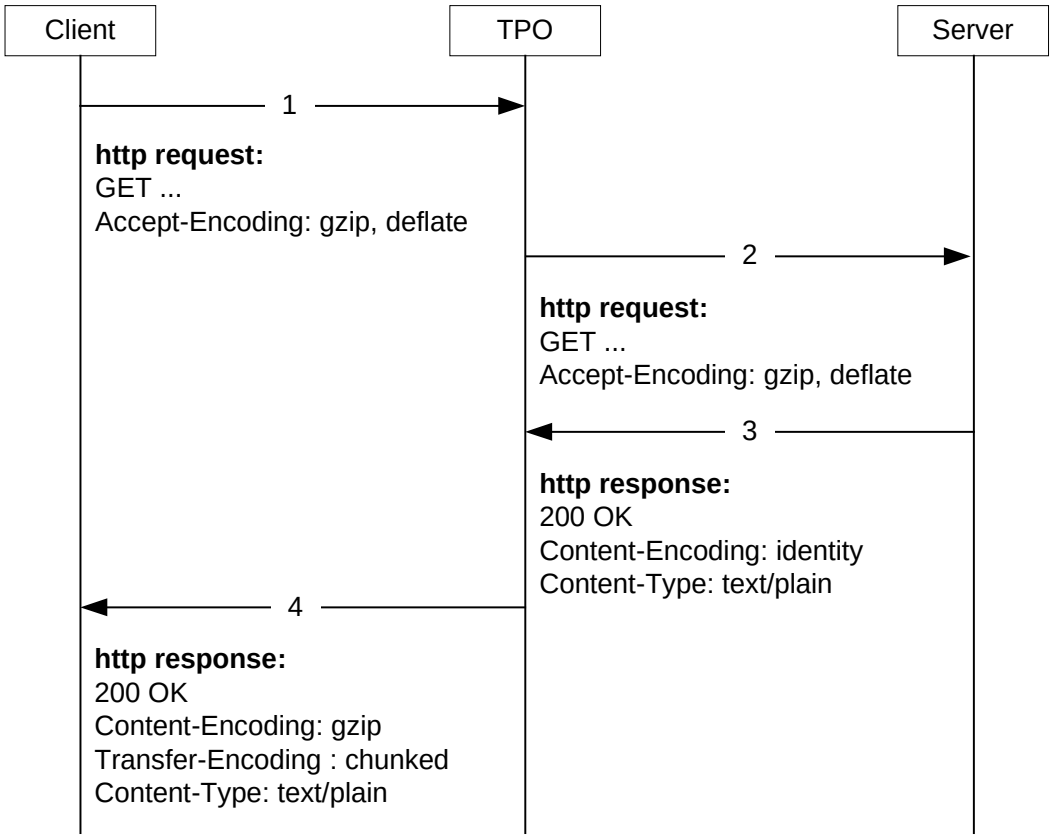
Enabling/disabling the HTTP Compression feature is a CLI-configurable parameter. By default HTTP Compression is disabled.

Compression level is a CLI-configurable parameter. Compression levels 1 through 9 are supported. The higher the compression level, the better the compression efficiency but with increased CPU and memory utilization. By default the compression level is set to 6.

TPO supports Prevention of Compression at the Web server. This enables TPO to receive uncompressed data from the Web server, on which it can apply other HTTP optimization techniques, and then compress the optimized data and send it to the mobile client. TPO achieves this by manipulating the HTTP requests. Enabling/disabling this feature is a CLI-configurable parameter. By default, the Prevention of Server Compression feature is disabled.

The following figure and steps explain how the HTTP Compression feature works:

Figure 3. HTTP Compression



1. Mobile client's browser in its HTTP request for HTML content incudes an Accept-Encoding field with comma separated list of supported compression schema names.
2. TPO forwards the HTTP request to the Web server.
3. If the Web server does not support the compression schema(s) in the Accept-Encoding field or cannot undertake compression, in the HTTP response it sends the uncompressed data.

   If the Web server supports the compression schema(s) in the Accept-Encoding field, in the HTTP response the Web server may send the compressed data, and include the Content-Encoding field with name(s) of the compression schema(s) used. TPO forwards the HTTP response to the mobile client.
4. If the Web server in its response sends uncompressed data, TPO compresses the data and then uses internal thresholds for amounts of internal resources available compared to the amount of packet size reduction achieved.

   If the comparison is favorable, TPO forwards the compressed response to the mobile client, and any subsequent response packets are also compressed. TPO updates the Content-Encoding field with name(s) of the compression schema(s) used. The mobile client's browser parses the requested data and displays it.

   If the comparison is not favorable, TPO forwards the original response to the mobile client and then forwards any subsequent response packets.

URL Rewrite

The URL Rewrite feature enables to preemptively resolve host names in embedded URLs present within HTML content and rewrite them with resolved IP addresses. This rewriting helps to eliminate DNS round trips in high latency mobile networks resulting in faster responses.

Enabling/disabling the URL Rewrite feature is a CLI-configurable parameter. By default the URL Rewrite feature is disabled.

Important: The URL Rewrite feature needs a valid DNS client to be configured in the ISP (destination) context.

When the URL Rewrite feature is enabled, TPO rewrites URLs of the following format

http://*<host_name[:port]>***/***<url_path>***/***<file_name.extension>*

into

http://*<resolved_ip_address[:port]>***/***<url_rewrite_prefix>***/***<host_name[:port]>***/***<url_path>***/***<file_name.extension>*

For example, if the URL Rewrite prefix is *urlrewrite*, TPO rewrites the URL

http://www.google.com/test.img

into

http://209.85.153.103/urlrewrite/www.google.com/test.img

When the mobile client requests for the URL

http://*<resolved_ip_address[:port]>***/***<url_rewrite_prefix>***/***<host_name[:port]>***/***<url_path>***/***<file_name.extension>*

TPO rewrites the URL back to

http://*<host_name[:port]>***/***<url_path>***/***<file_name.extension>*

The URL Rewrite prefix is a CLI-configurable parameter. By default, the prefix is set to “urlrewrite”.

URL Rewrite works only on HTML content, hence it is called only in the following cases:

- Content in response to HTTP GET request where MIME type of the content is HTML/CSS/JavaScript
- Content is not encoded

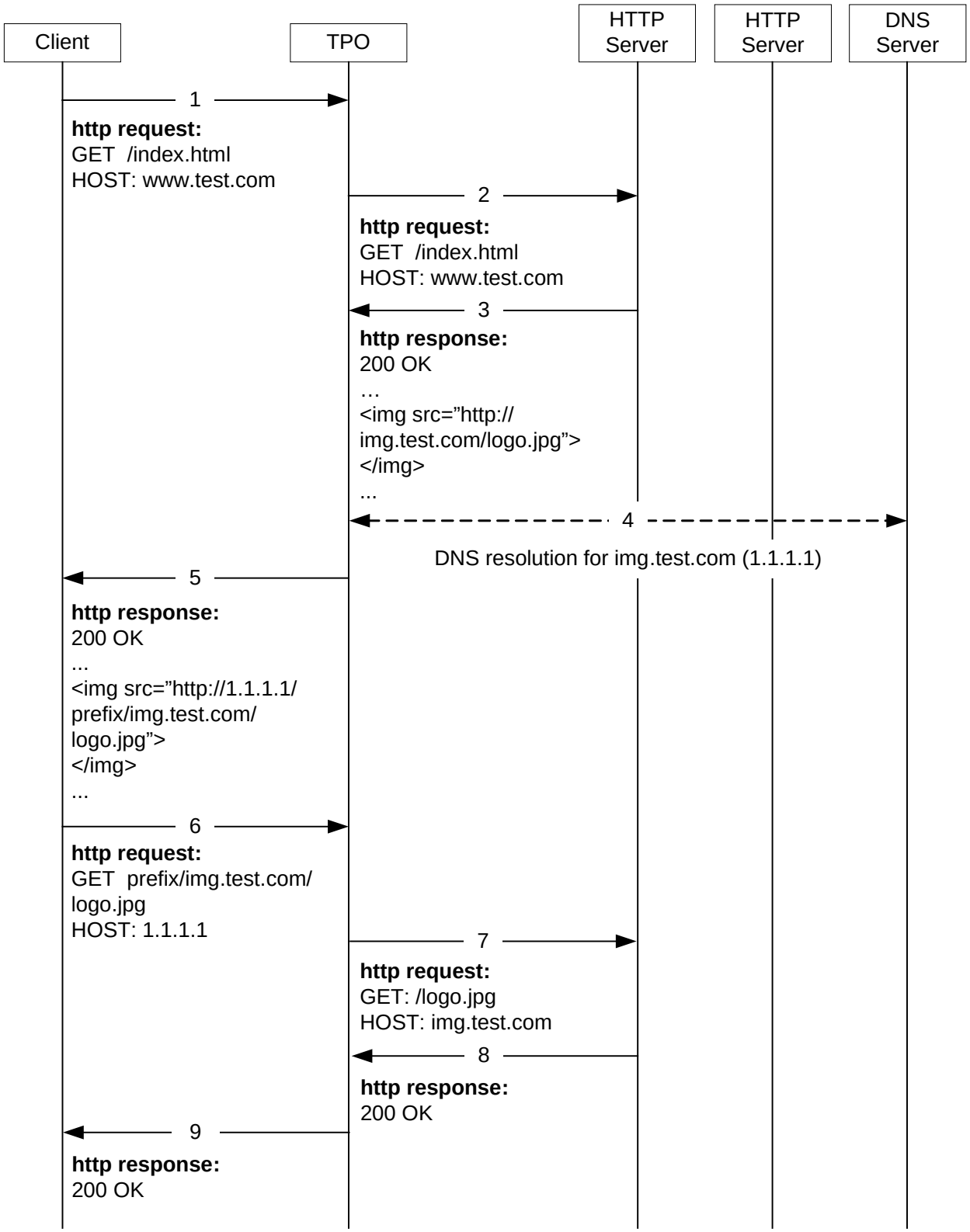
TPO rewrites only those URLs that are present in the following HTML tags:

- image
- imagepath
- img
- input
- link
- script

Important: URLs that are part of JavaScript and VBScript are not rewritten. If an HTML tag spans across packets, TPO will queue only two packets and will rewrite the URL if found.

The following figure and steps explain how the HTTP URL Rewrite feature works:

Figure 4. HTTP URL Rewrite



1. The mobile client sends HTTP request for HTML content.
2. TPO forwards the HTTP request to the Web server.
3. The Web server sends an HTTP response with the requested HTML content.
4. TPO resolves host names in embedded URLs present within the HTML content and rewrites them with corresponding IP addresses.
5. TPO forwards the HTTP response to the mobile client.
6. The mobile client's browser parses the HTML and sends HTTP requests for images and other content to the modified URLs.
7. TPO rewrites the URLs and forwards the HTTP requests to the Web server.
8. The Web server returns HTTP response with the requested content.
9. TPO forwards the HTTP response to the mobile client.

   In case both HTTP Compression and URL Rewrite features are enabled, URL Rewrite processing will happen before HTTP Compression.

Advertisement Filter

The Advertisement Filter feature enables to block advertisement content in Web pages delivered to mobile clients. This filtering reduces over-the-air bandwidth usage as advertisements are not always downloaded, and faster response times as advertisement-related server connections are eliminated.

Important: In this release, TPO considers only images and Flash objects as advertisements.

TPO is configured with URLs of the advertisement sites to be blocked, typically sites such as www.doubleclick.net/*, ad.yahoo.com, and so on. When the mobile client's browser receives HTML content, it parses the HTML and sends out requests for images and other content. If there is a request for an image or Flash object whose URL matches any of the URLs to be blocked, TPO blocks the advertisement as per the configuration.

The Advertisement Filter feature supports the following advertisement blocking methods:

- Advertisement Blocking with NO Text: In the mobile client's browser each blocked advertisement is replaced with the "cannot display image" icon (usually an X mark). Subscribers cannot view the advertisements even if they want to.
- Advertisement Blocking with Text: In the mobile client's browser each blocked advertisement is replaced with a placeholder frame. Each placeholder frame contains standard operator-configured text and the advertisement's URL. Subscribers cannot view the blocked advertisements even if they want to.
- Advertisement Blocking with On-click Function: In the mobile client's browser each blocked advertisement is replaced with a placeholder frame. Each placeholder frame contains standard operator-configured text and the advertisement's URL. To view a blocked advertisement the subscriber must click the placeholder frame.

  To enable retrieving the blocked advertisement, in the HTTP request a bypass string is added to the advertisement's URL, which TPO interprets and forwards to the Web Server allowing the advertisement to be retrieved. The bypass string is a CLI-configurable parameter.

The background color of the placeholder frames and the text displayed in them are CLI-configurable parameters. Operators can use the text to indicate that the advertisement is blocked. Note that different text strings can be configured for "Advertisement Blocking with Text" and "Advertisement Blocking with On-click Function" configurations.

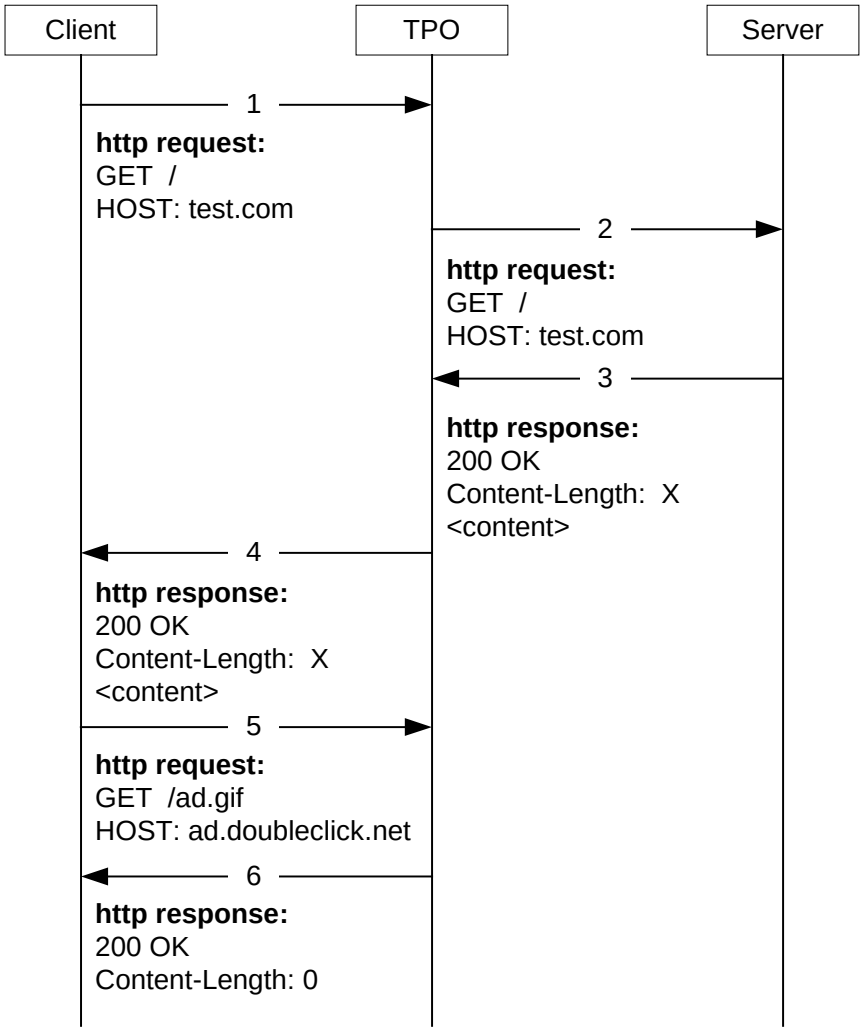
Important: The text string and URL displayed in the placeholder frames may be truncated to fit dimensions of the frames.

Important: The "Advertisement Blocking with Text" and "Advertisement Blocking with On-click Function" Advertisement Filter functionality is achieved using JavaScript code sent from TPO and executed whenever a Web page is loaded in the mobile client's browser. If the mobile client's browser does not support JavaScript or the subscriber has disabled JavaScript, instead of the placeholder frames the subscribers will see the "cannot display image" icons.

Basic Advertisement Blocking

The following figure and steps explain how the basic advertisement blocking feature works.

Figure 5. Basic Advertisement Blocking



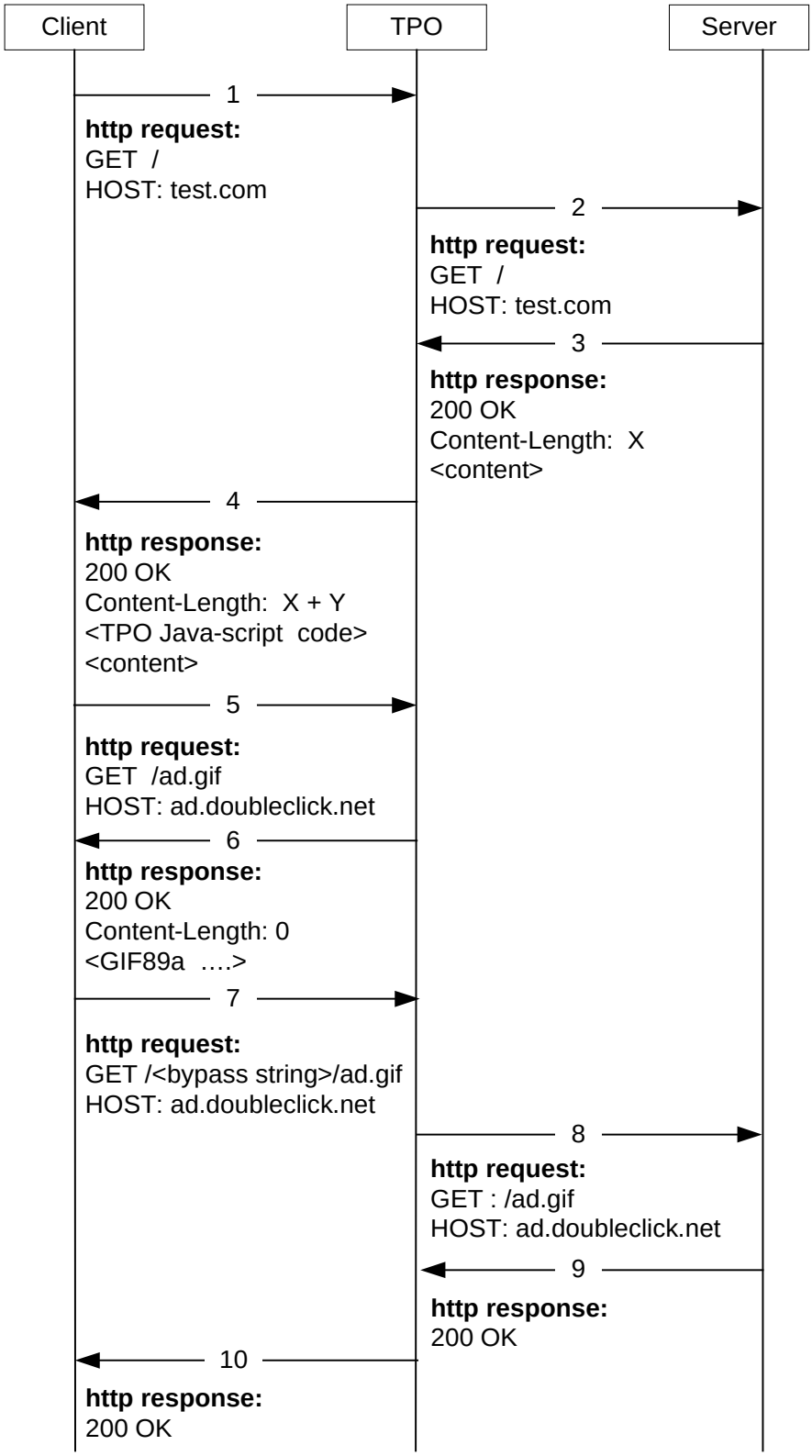
1. The mobile client sends HTTP request for HTML content.
2. TPO forwards the HTTP request to the Web server.
3. The Web server sends an HTTP response with the requested HTML content.
4. TPO forwards the HTTP response to the mobile client.
5. The mobile client's browser parses the HTML and sends HTTP requests for images, Flash objects, and other content.
6. If there is an HTTP request for an image or Flash object, TPO matches the requested URL with the list of advertisement URLs to be blocked.

   If there is a match, TPO responds with Content-Length 0 for the request thereby blocking the advertisement. In the mobile client's browser, the image or Flash object is replaced with the "cannot display image" icon.

Advertisement Blocking with On-click Function

The following figure and steps explain how the Advertisement Blocking with On-click feature works.

Figure 6. Advertisement Blocking with On-click Function



1. The mobile client's browser sends an HTTP request for HTML content.
2. TPO forwards the HTTP request to the Web server.
3. The Web server sends an HTTP response with the requested HTML content.
4. TPO forwards the HTTP response to the mobile client's browser along with a JavaScript code containing the list of advertisement site URLs to be blocked.
5. The mobile client's browser parses the HTML and sends HTTP requests for images, Flash objects, and other content.
6. If there is an HTTP request for an image or a Flash object, TPO matches the requested URL with the URLs to be blocked. If there is a match, TPO responds with local content based on the requested file's extension.
7. JavaScript in the mobile client's browser parses the HTML to check for blocked images and Flash objects, and replaces them with placeholder frames. If "Advertisement Blocking with On-click Function" is enabled, on-click functionality is added to the frames.

   When the subscriber clicks the placeholder frame for a blocked image or Flash object, the mobile client's browser sends an HTTP request with a bypass string appended to the requested URL.
8. TPO forwards the HTTP request to the Web server.
9. The Web server sends an HTTP response with the requested data.
10. TPO forwards the HTTP response to the mobile client's browser. The mobile client's browser displays the requested advertisement content.

TPO Administration

This section describes TPO administration activities and covers the following topics:

- DisablingEnabling TPO Optimizations
- MUR Reporting for TPO
- Switching TPO Policy

Disabling/Enabling TPO Optimizations

The TPO in-line service allows to disable/continue TPO optimizations when a peer-to-peer (P2P) flow is detected. This is a CLI-configurable feature.

Important: In this release, on disabling TPO optimizations only TCP-based optimizations are disabled. Disabling HTTP-based optimizations will be supported in a future release.

MUR Reporting for TPO

This section lists the EDR fields supported for TPO - MUR integration.

The Mobility Unified Reporting (MUR) is a Web-based application providing a unified reporting interface for a variety of data from Cisco Systems in-line service and storage applications. For more information on the MUR, see the *Mobility Unified Reporting System Online Help*.

The following are the EDR generation points:

- End of TCP connection
- End of HTTP transaction

The following EDR fields are supported for TPO - MUR integration:

- TCP flow related fields:
    - TCP data transferred
    - TCP duration
    - TPO enable/disabled
- HTTP transaction related fields:
    - HTTP URL
    - HTTP DNS local resolutions

- HTTP DNS server resolutions
- HTTP compression bytes in
- HTTP compression bytes out
- HTTP advertisement replaced (advertisement blocked)
- HTTP advertisement delivered (advertisement accessed)
- TPO enabled/disabled

Switching TPO Policy

TPO allows to switch a TPO policy in use with a different TPO policy.

Important: The switch takes effect only on current calls. For new calls, the RADIUS-returned/APN/subscriber template configured policy is used.

To be able to change the TPO policy in mid session, TPO must have been enabled for the subscriber in the APN/Subscriber template configuration during call setup.

The CLI indicates the number of sessions for which the policy got switched.

How TPO Works

This section describes how TPO works.

Terms and Definitions

The following is a list of terms specific to TPO functionality:

- TPO Policy: A TPO policy specifies the match-rule definitions to select a TPO profile. The match-rule definitions enable to use different sets of optimizations for different TCP/HTTP flows of a subscriber.

  The TPO policy to be used for a subscriber can be from one of the following:

  - AAA/OCS: The TPO policy can come from the AAA server or the OCS. During initial authentication the AAA server returns the TPO policy for the subscriber, which is applied to the corresponding session. For this purpose the system uses the RADIUS AVP SN-TPO-Policy. If the policy comes from the AAA/OCS, it will override the policy configured in the subscriber's APN/subscriber template and/or the ECS rulebase.

    Important: The TPO policy received from the AAA and OCS have the same priority. Whichever comes first, either from AAA or the OCS, is applied.

  - APN/Subscriber Template: If no TPO policy is received from the AAA/OCS, the TPO policy configured in the subscriber's APN/subscriber template is applied to the flows over the sessions using that APN/subscriber profile.
  - ECS Rulebase: The default TPO policy configured in the ECS rulebase has the least priority. If no TPO policy to use is received from the AAA/OCS, and there is no TPO policy configured in the subscriber's APN/subscriber template, only then will the default TPO policy configured in the ECS rulebase be used.

  A maximum of 2048 TPO policies can be configured in the system.
- TPO Profile: A TPO profile specifies the optimizations to be performed for a specific flow. A maximum of 2048 TPO profiles can be configured in the system. A TPO profile can be used in more than one TPO policy.
- Ruledef: A ruledef specifies the criteria to identify a specific flow, such as HTTP flow to google.com. A maximum of 2048 optimization ruledefs can be configured in the system.
- Subscriber Profile/APN: The subscriber profile/APN specifies the TPO policy to be applied to the flows over the sessions using that subscriber profile/APN.

TPO Processing

TPO can be enabled in either of the following ways:

- Policy-based TPO processing: On all flows of a subscriber based on the TPO policy specified by the AAA/OCS, APN/subscriber template, or the ECS rulebase.
- Charging Action-based TPO processing: On specific flows of a subscriber based on the flow matching a charging action with a TPO profile configured in it.

Policy-based TPO Processing

The following steps describe how policy-based TPO processing works:

1. When a subscriber initiates a data session, the TPO policy from the AAA/OCS, or the TPO policy configured in the subscriber's APN/subscriber profile, or the ECS rulebase is associated with that data session.
2. When a flow is created over the session (as when the subscriber initiates a browsing session), first the ECS routing ruledefs are applied to determine the protocol analyzer (such as HTTP).
3. The optimization ruledefs configured in the TPO policy are applied one after the other, in the specified order. When a match is found, the optimizations configured in the corresponding TPO profile are applied to the session.

   During the course of a flow, first the match-rule logic is applied at SYN time — this mostly results in a default match-rule that selects the default TPO profile. This is because many of the match-rule conditions would not apply at SYN time. The match-rule is generally more useful with deep-packet inspection (DPI). During DPI, when the complete HTTP header information is received, the match-rule is invoked and a new TPO profile (if any) is obtained and applied. This new TPO profile (selected during DPI) will be used to perform HTTP optimization. However, the original TPO profile selected during SYN time will be used for TCP optimization.

   If the first SYN does not match any TPO profile (in the absence of a default match rule), TPO is not applied to that flow. In 12.0 and later releases, with dynamic TCP Proxy this is no longer the case.

Important: The match-rule is also invoked after the HTTP request line is received. At this time, a TPO profile is used to only apply the advertisement block rule (if any). This is required to block any unwanted HTTP request packets for and advertisement site that could be potentially sent to the server. None of the other rules are applied even if present in the profile.

Charging Action Based TPO Processing

The following steps describe how charging action based TPO processing works:

1. When a flow is created (as in when the subscriber initiates a browsing session), ECS routing ruledefs are applied to determine the protocol analyzer (such as HTTP).
2. Based on the ECS rule matching the charging action to apply to the flow is selected.
3. If a TPO profile is configured in that charging action, optimizations configured in that TPO profile are applied on the flow.

Important: In this release, when the TPO profile specified by a charging action is applied on a flow, only TCP optimizations and the decision to cease/continue TPO optimizations for P2P flows will be controlled by that TPO profile. HTTP optimizations will not be affected.

When the TPO profile specified by a charging action is applied on a flow, and subsequently a different charging action that does not have a TPO profile configured in it is applied on the same flow, TPO will not be disabled.

Chapter 2
Traffic Performance Optimization Configuration

This chapter describes how to configure Traffic Performance Optimization in-line service.

The following topics are covered in this chapter:

- Before You Begin
- Configuring TPO
- Verifying Your Configuration

Before You Begin

Before you can configure the TPO in-line service:

1. Configure the core network service as described in corresponding administration guide.
2. Configure the active charging service, rule definitions, charging action(s), and rulebase(s) as required. For information refer to the *Enhanced Charging Service Administration Guide*.

Configuring TPO

This section lists the high-level steps to configure the TPO in-line service.

1. Create and configure TPO profiles as described in the *Creating and Configuring TPO Profiles* section.
2. Create and configure TPO policies as described in the *Creating and Configuring TPO Policies* section.
3. Associate TPO policies to subscribers/APNs as described in the *Applying TPO Policies to Subscribers/APNs* section.
4. Save your configuration as described in the *Verifying and Saving your Configuration* chapter.

Important: Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

Creating and Configuring TPO Profiles

This section describes how to create and configure TPO profiles.

Creating TPO Profiles

To create a TPO profile use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      tpo profile <tpo_profile_name> [ -noconfirm ]
      end
```

Notes:

- A maximum of 2048 TPO profiles can be configured in the system.

Configuring TPO Profiles

To configure a TPO profile use the following configuration:

```
configure
```

```
active-charging service <ecs_service_name>

    tpo profile <tpo_profile_name>

#configurations for HTTP optimizations:

        http ad-filter display { no-text | text-only <text_only_string> | text-
with-click <text_with_click_string> }

        http params ad-filter display bgcolor <hex_color>

        http compression

        http params compression level <compression_level>

        http optimize-compressed-page

        http prevent-server-compression

        http params url-rewrite prefix <url_rewrite_prefix>

        http url-rewrite

#configurations for TCP optimizations:

        tcp bandwidth { <bandwidth_kbps> | dynamic }

        tcp buffer-size { downlink | uplink } <buffer_size_kb>

        tcp congestion-control { basic | vegas | westwood-plus }

        tcp fast-retransmit-dupacks { <duplicate_acks> | dynamic }

        tcp handoff-optimization

        tcp initial-window { <initial_window> | dynamic bdp-percent
<bdp_percent> | rfc5681 }

        tcp mss <mss>

        tcp rto { retrans-backoff { 1.0 | 1.5 | 2.0 } | rttvar-scaling
<scaling_factor> }

        end
```

Notes:

- If you enable the URL Rewrite feature using the **http url-rewrite** command, make sure that a valid DNS client is configured in the destination context as described in the Configuring DNS Client for URL Rewrite Feature section.
- If for the Advertisement Blocking feature you configure the **http ad-filter display { no-text | text-only** *<text_only_string>* **| text-with-click** *<text_with_click_string>* **}** command, make sure that in the TPO policy, the **match-ad** and **ad-filter** commands are configured.

- The **http prevent-server-compression** command must only be configured if you want TPO to receive only uncompressed data from the Web server.

Configuring DNS Client for URL Rewrite Feature

To configure DNS client in ISP (destination) context for the HTTP URL Rewrite feature, use the following configuration:

```
configure
   context <context_name>
      interface <interface_name>
         ip address <ip_address/mask>
         exit
      ip domain-lookup
      ip name-servers <ip_address>
      dns-client <dns_client_name>
         bind address <ip_address>
         round-robin-answers
         end
```

Creating and Configuring TPO Policies

This section describes how to create and configure a TPO policy.

Creating TPO Policies

To create a TPO policy use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      tpo policy <tpo_policy_name> [ -noconfirm ]
      end
```

Notes:

- A maximum of 2048 TPO policies can be configured in the system.

Configuring TPO Policies

To configure a TPO policy use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      tpo policy <tpo_policy_name>
         match-rule priority <rule_priority> tpo-ruledef <tpo_ruledef_name> tpo
{ none | profile <tpo_profile_name> } [ description <description> ]
         match-rule no-ruledef-match tpo { none | profile <tpo_profile_name> }
         match-ad priority <rule_priority> tpo-ruledef <ecs_ruledef_name>
         ad-filter ad-click-identity <bypass_string>
         end
```

Note:

- The **match-ad** and **ad-filter** commands must be configured if the Advertisement Filter feature is to be enabled.
- If the **match-ad** command is configured, the ACS ruledef specified must be configured with relevant match conditions. For information on how to create and configure ACS ruledefs, refer to the *Enhanced Charging Service Administration Guide*.

Applying TPO Policies to Subscribers/APNs

This section describes how to associate TPO policies with subscribers/APNs:

Applying TPO Policies to Subscribers

To associate a TPO policy with subscriber(s) use the following configuration:

```
configure
   context <context_name>
      subscriber { default | name <subscriber_name> }
```

```
            tpo policy <tpo_policy_name>
            end
```

Applying TPO Policies to APNs

To associate a TPO policy with an APN use the following configuration:

```
configure
   context <context_name>
      apn name <apn_name>
         tpo policy <tpo_policy_name>
         end
```

Configuring Default TPO Policy for Subscribers

This section describes how to configure the default TPO policy for subscribers using a particular rulebase.

To configure the default TPO policy for subscribers in the rulebase use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      rulebase <rulebase_name>
         tpo default-policy <tpo_policy_name>
         end
```

Notes:

- This TPO policy is used for a subscriber only if no TPO policy to use is received from the AAA/OCS, and there is no TPO policy configured in the subscriber's APN/subscriber template.

Configuring TPO Profile in ECS Charging Action

This section describe how to configure a TPO profile within an ECS charging action to enable ECS charging-action based TPO processing.

To configure a TPO profile within an ECS charging action use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      charging-action <charging_action_name>
         tpo profile <tpo_profile_name>
         end
```

TPO Administration and Other Configurations

This section describes TPO administrative and other procedures.

This section includes the following topics:

- Changing TPO Policy in Mid Session
- Disabling/Enabling TPO for P2P Flows

Changing TPO Policy in Mid Session

To change the TPO policy in mid session use the following command available in the Exec mode:

update active-charging switch-to-tpo-policy *<tpo_policy_name>* **{ all | callid** *<call_id>* **| imsi** *<imsi>* **| ip-address** *<ip_address>* **| msid** *<msid>* **| tpo-policy** *<tpo_policy_name>* **| username** *<user_name>* **} [-noconfirm] [| { grep** *<grep_options>* **| more }]**

Notes:

- The **switch-to-tpo-policy** *tpo_policy_name* option specifies the new TPO policy to apply, and the **tpo-policy** *tpo_policy_name* option specifies to apply the new policy to sessions using this TPO policy.
- The CLI indicates the number of sessions for which the policy got switched.
- To be able to change the TPO policy in mid session, TPO must have been enabled for the subscriber in the APN/Subscriber template configuration during call setup.
- The update command takes effect only on current calls. For new calls, the RADIUS returned/APN/subscriber template configured policy is used.

Disabling/Enabling TPO Optimizations for P2P Flows

To disable/continue TPO optimizations when a peer-to-peer (P2P) flow is detected use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      tpo profile <tpo_profile_name>
         p2p-detected { cease-tpo | continue-tpo }
         end
```

Notes:

- In this release, only TPO TCP optimizations are disabled. Disabling HTTP optimizations will be supported in a future release.
- When TPO profiles are switched, if P2P flow is detected, TPO will be ceased/allowed based on the new TPO profile.

Verifying Your Configuration

This section describes how to configure your TPO configuration.

To verify your configurations, use the following steps:

1. To view TPO profile information, in the Exec Mode enter one of the following commands:
   - To view information for all TPO profiles use the following command:

     show active-charging tpo profile statistics all
   - To view detailed information for a specific TPO profile use the following command:

     show active-charging tpo profile statistics name *<tpo_profile_name>*
2. To view TPO policy information, in the Exec Mode enter one of the following commands:
   - To view information for all TPO policies use the following command:

     show active-charging tpo policy statistics all
   - To view detailed information for a specific TPO policy use the following command:

     show active-charging tpo policy statistics name *<tpo_policy_name>*
3. In the Exec Mode, enter the following command:

   show subscribers tpo { not-required | required }+

   The output of this command displays access information, call ID, MSID, user name, IP address, and idle time information for the specified subscribers — those with TPO enabled and those without.
4. In the Exec Mode, enter the following command:

   show subscribers full

   The output of this command indicates the TPO policy configured for the subscriber(s).

Chapter 3
Verifying and Saving Your Configuration

This chapter describes how to save your system configuration.

Verifying the Configuration

You can use a number of commands to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of, or specific lines in, the configuration file.

Feature Configuration

In many configurations, you have to set and verify specific features. Examples include APN and IP address pool configuration. Using these examples, enter the following commands to verify proper feature configuration:

show apn all

The output displays the complete configuration for the APN. In this example, an APN called apn1 is configured.

```
access point name (APN): apn1

authentication context: test

pdp type: ipv4

Selection Mode: subscribed

ip source violation: Checked drop limit: 10

accounting mode: gtpp No early PDUs: Disabled

max-primary-pdp-contexts: 1000000 total-pdp-contexts: 1000000

primary contexts: not available total contexts: not available

local ip: 0.0.0.0

primary dns: 0.0.0.0 secondary dns: 0.0.0.0

ppp keep alive period : 0 ppp mtu : 1500

absolute timeout : 0 idle timeout : 0

long duration timeout: 0 long duration action: Detection

ip header compression: vj

data compression: stac mppc deflate compression mode: normal

min compression size: 128

ip output access-group: ip input access-group:

ppp authentication:

allow noauthentication: Enabled imsi
```

```
 authentication:Disabled
```

Enter the following command to display the IP address pool configuration:

show ip pool

The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:

+-----Type: (P) - Public (R) - Private

 | (S) - Static (E) - Resource

 |

 |+----State: (G) - Good (D) - Pending Delete (R)-Resizing

 ||

 ||++--Priority: 0..10 (Highest (0) .. Lowest (10))

 ||||

 ||||+-Busyout: (B) - Busyout configured

 |||| |||||| vvvvv Pool Name Start Address Mask/End Address Used Avail

 ----- --------- --------------- --------------- -------- --------

PG00 ipsec 12.12.12.0 255.255.255.0 0 254 PG00

pool1 10.10.0.0 255.255.0.0 0 65534 SG00

vpnpool 192.168.1.250 192.168.1.254 0 5 Total Pool Count: 5
```

Important: To configure features on the system, use the *show* commands specifically for these features. Refer to the *Cisco Systems ASR 5000 Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *<service_type> <service_name>*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called pgw is configured.

```
Service name : pgw1

Service-Id : 1
```

```
Context : test1

Status : STARTED

Restart Counter : 8

EGTP Service : egtp1

LMA Service : Not defined

Session-Delete-Delay Timer : Enabled

Session-Delete-Delay timeout : 10000(msecs)

PLMN ID List : MCC: 100, MNC: 99

Newcall Policy : None
```

Context Configuration

Verify that your context was created and configured properly by entering the following command:

show context name *<name>*

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID     State

------------     ---------     -----

test1            2          Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the following command:

show configuration

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the following command:

show configuration errors

This command displays errors it finds within the configuration. For example, if you have created a service named "service1", but entered it as "srv1" in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu:

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##############################################################################

Displaying Global
AAA-configuration errors
##############################################################################

Total 0 error(s) in this section !
```

Saving the Configuration

Save system configuration information to a file locally or to a remote node on the network. You can use this configuration file on any other systems that require the same configuration.

Files saved locally can be stored in the SMC's CompactFlash or on an installed PCMCIA memory card on the SMC. Files that are saved to a remote network node can be transmitted through FTP, or TFTP.

Saving the Configuration on the Chassis

These instructions assume that you are at the root prompt for the Exec mode:

[local]*host_name*#

To save your current configuration, enter the following command:

save configuration *url* **[-redundant] [-noconfirm] [showsecrets] [verbose]**

Keyword/Variable	Description
url	Specifies the path and name to which the configuration file is to be stored. *url* may refer to a local or a remote file. *url* must be entered using one of the following formats: • **{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **file:/{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **tftp://**{ *ipaddress*|*host_name* [*:port#*] } [*/directory*] */file_name* • **ftp://**[*username* [*:pwd*] *@*] { *ipaddress*|*host_name* } [*:port#*] [*/directory*] */file_name* • **sftp://**[*username* [:*pwd*] *@*] { *ipaddress*|*host_name* } [*:port#*] [*/directory*] */file_name* **/flash** corresponds to the CompactFlash on the SMC. **/pcmcia1** corresponds to PCMCIA slot 1. **/pcmcia2** corresponds to PCMCIA slot 2. *ipaddress* is the IP address of the network server. *host_name* is the network server's *hostname*. *port#* is the network server's logical port number. Defaults are: • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data Note: *host_name* can only be used if the **networkconfig** parameter is configured for DHCP and the DHCP server returns a valid nameserv er.dx *username* is the username required to gain access to the server if necessary. *password* is the password for the specified username if required. */directory* specifies the directory where the file is located if one exists. */file_name* specifies the name of the configuration file to be saved. Note: Configuration files should be named with a .cfg extension.
-redundant	Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the Standby SMC, if available. Note: This keyword will only work for like local devices that are located on both the active and standby SMCs. For example, if you save the file to the /pcmcia1 device on the active SMC, that same type of device (a PC-Card in Slot 1 of the standby SMC) must be available. Otherwise, a failure message is displayed. Note: If saving the file to an external network (non-local) device, the system disregards this keyword.
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).

Keyword/Variable	Description
showsecrets	Optional: This keyword saves the CLI configuration file with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies to display every parameter that is being saved to the new configuration file.

Important: The **-redundant** keyword is only applicable when saving a configuration file to local devices. This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SMC, you must synchronize the local file system on both SMCs.

To save a configuration file called system.cfg to a directory that was previously created called cfgfiles on the SMC's CompactFlash, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called simple_ip.cfg to a directory called host_name_configs, using an FTP server with an IP address of 192.168.34.156, on which you have an account with a username of administrator and a password of secure, use the following command:

```
save configuration
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called init_config.cfg to the root directory of a TFTP server with a hostname of config_server, enter the following command:

save configuration *tftp://config_server/init_config.cfg*

Chapter 4
Traffic Performance Optimization Sample Configuration

The following is a sample TPO configuration.

```
active-charging service service_1
   ruledef ch_dns
      dns any-match = TRUE
      exit
   ruledef ch_http
      http any-match = TRUE
      exit
   ruledef rt_dns
      tcp either-port = 53
      udp either-port = 53
      rule-application routing
      multi-line-or all-lines
      exit
   ruledef rt_http
      tcp either-port = 80
      rule-application routing
      exit
   ruledef tpo_url2
      www url contains worldrediff
      www url contains info
      www url contains match
      www url contains zedo
```

```
   multi-line-or all-lines
   exit
ruledef url_fil
   www url contains ad
   www url contains dynamic
   multi-line-or all-lines
   exit
charging-action ca_nothing
   content-id 5
   exit
rulebase rulebase_1
   action priority 10 ruledef ch_dns charging-action ca_nothing
   action priority 15 ruledef ch_http charging-action ca_nothing
   route priority 25 ruledef rt_dns analyzer dns
   route priority 30 ruledef rt_http analyzer http
   exit
rulebase default
   exit
tpo profile ad-fil
   http ad-filter display text-only AdsFilTextOnly
   http params ad-filter display bgcolor ff00cc
   exit
tpo profile ad-fil2
   http ad-filter display text-with-click AdsFilteredByTPO
   http params ad-filter display bgcolor 11bbcc
   exit
tpo profile tpo
   tcp congestion-control westwood-plus
   tcp handoff-optimization
```

```
        tcp bandwidth 960
        tcp initial-window dynamic bdp-percent 90
        http compression
        http url-rewrite
        http params compression level 9
        http params url-rewrite prefix url_rewrite_prefix
        http ad-filter display text-with-click AdFilterDefault
        http params ad-filter display bgcolor ff00ff
        exit
    tpo policy tpolicy
        ad-filter ad-click-identity ad_click_identity
        match-ad priority 2 tpo-ruledef tpo_url2
        match-ad priority 10 tpo-ruledef url_fil
        match-rule priority 10 tpo-ruledef url_fil tpo profile ad-fil
        match-rule priority 20 tpo-ruledef tpo_url2 tpo profile ad-fil2
        match-rule no-ruledef-match tpo profile tpo
        end
#configuring DNS in ISP (destination) context
context isp
    interface intDns
        ip address 1.1.1.1 255.255.255.0
        exit
    ip domain-lookup
    ip name-servers 2.2.2.2
    dns-client dnshttp
        bind address 3.3.3.3
        round-robin-answers
        end
```