



# Cisco Ultra Traffic Optimization

This chapter describes the following topics:

- [Feature Summary and Revision History, on page 1](#)
- [Overview, on page 2](#)
- [How Cisco Ultra Traffic Optimization Works, on page 3](#)
- [Configuring Cisco Ultra Traffic Optimization, on page 6](#)
- [Multi-Policy Support for Traffic Optimization, on page 11](#)
- [Monitoring and Troubleshooting, on page 24](#)

## Feature Summary and Revision History

### Summary Data

Applicable Product(s) or Functional Area	P-GW
Applicable Platform(s)	<ul style="list-style-type: none"><li>• ASR 5500</li><li>• Ultra Gateway Platform</li></ul>
Feature Default	Disabled - License Required
Related Changes in This Release	Not Applicable
Related Documentation	<ul style="list-style-type: none"><li>• <i>Command Line Interface Reference</i></li><li>• <i>P-GW Administration Guide</i></li></ul>

### Revision History



#### Important

Revision history details are not provided for features introduced before release 21.2 and N5.1.

Revision Details	Release
The Cisco Ultra Traffic Optimization library version has been upgraded from 3.0.9 to 3.0.11.	21.14.2
With this release, new keywords <b>large-flows-only</b> and <b>managed-large-flows-only</b> are implemented as part of the <b>data-record</b> command to enable the CUTO library to stream respective statistics to the external server.  New bulk statistics are added in support of this enhancement	21.14
With this release, Cisco Ultra Traffic Optimization solution is enhanced to support basic Quick UDP Internet Connections (QUIC) UDP traffic along with the existing support for TCP traffic.	21.3.17
Reboot of chassis is no longer required to enable Cisco Ultra Traffic Optimization related configuration.	21.3.x
Multi-Policy support for Cisco Ultra Traffic Optimization solution.	21.6
Cisco Ultra Traffic Optimization solution is supported in Ultra Gateway Platform (UGP).	21.6
Cisco Ultra Traffic Optimization solution is enhanced to support basic Quick UDP Internet Connections (QUIC) UDP traffic along with the existing support for TCP traffic.	21.5
Reboot of chassis is no longer required to enable Cisco Ultra Traffic Optimization related configuration.	21.5
First introduced.	21.2

## Overview

As traffic grows, Quality of Experience (QoE) drops because more consumers are sharing the radio channel with increasingly richer, more data consuming application. Connection speed (bits per second) is crucial. Traditional approaches to maintaining connection speed in the face of traffic growth (that is, Network Densification) focus on hardware-based approaches, such as adding new base stations and building the MIMO. While necessary, this approach is very time consuming and expensive.

The Cisco Ultra Traffic Optimization is a RAN optimization technology that increases subscriber connection speeds in congested cells and, as a result, increases the cell capacity significantly. The result is an optimized RAN, where Mobile Network Operators (MNOs) can deploy fewer cells, on an ongoing basis, and absorb more traffic growth while meeting network quality targets.

Large traffic flows, such as Adaptive Bit Rate (ABR) video, saturate radio resources and swamp the eNodeB scheduler. The Cisco Ultra Traffic Optimization employs machine learning algorithms to detect large traffic flows (such as video) in the network and optimize the delivery of those flows to mitigate the network congestion without changing user quality (that is, video works the same for the end user). In other words, by employing software intelligence at the network core, Cisco Ultra Traffic Optimization mitigates the overwhelming impact video has on the RAN.

The resulting benefits are seen in congested network sites. The Cisco Ultra Traffic Optimization:

- Increases average user throughput.
- Increases congested cell site capacity.
- Reduces scheduler latency.
- Maintains user quality of experience even when more users and more traffic share a cell.
- Is measured directly by eNodeB performance counters (for example, average UE throughput, scheduler latency), which are the key performance indicators that are used for network capacity planning.
- Provides permanent savings in RAN investment requirements.
- Is integrated in the Cisco StarOS P-GW.
- Requires no new hardware or cabling complexity - it can be turned on for a market in an hour.
- Supports all types of traffic, including HTTP(s) and QUIC.

## How Cisco Ultra Traffic Optimization Works

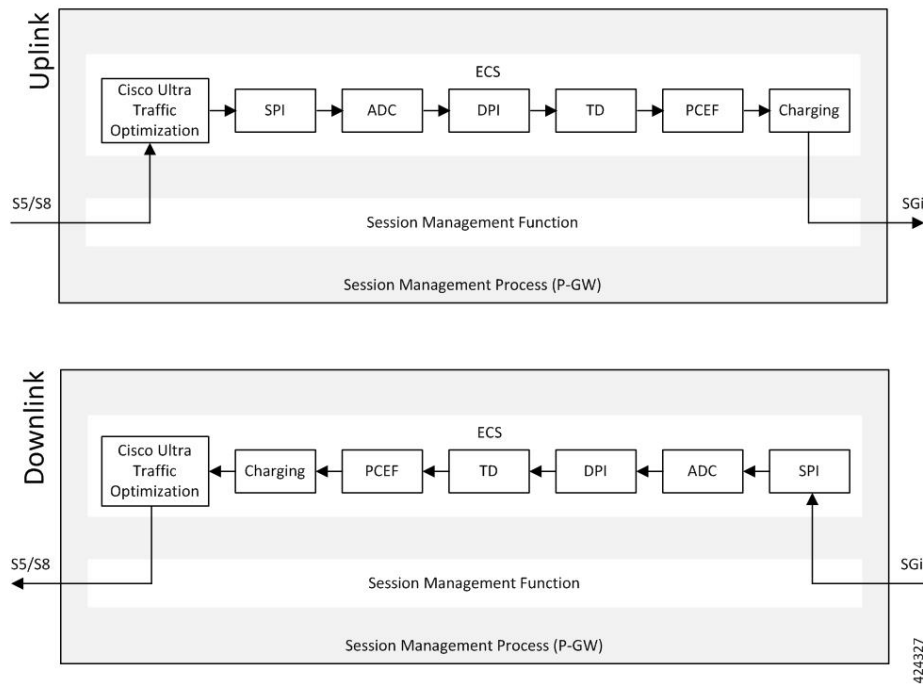
The Cisco Ultra Traffic Optimization achieves its gains by shaping video traffic during times of high network load/congestion. It monitors and profiles each individual video flow that passes through the gateway and uses its machine learning algorithms to determine whether that flow is traversing a congested channel. Cisco Ultra Traffic Optimization then flow-controls video to varying levels and time, depending on the degree of detected congestion, and efficiently aligns delivery of the video traffic to less-congested moments while still providing adequate bandwidth to videos to maintain their quality. The result is less network latency and higher user throughputs while maintaining HD video. Cisco Ultra Traffic Optimization does not drop packets or modify data payloads in any way.

The Cisco Ultra Traffic Optimization integrates with standard Cisco P-GW functions such as Application Detection and Control (ADC), allowing mobile operators to define optimization policies that are based on the traffic application type as well as APN, QCI, and other common traffic delineations. Cisco Ultra Traffic Optimization is fully radio network aware, allowing management on a per eNodeB cell basis.

## Architecture

StarOS has a highly optimized packet processing framework, the Cisco Ultra Traffic Optimization engine, where the user packets (downlink) are processed in the operating systems user space. The high-speed packet processing, including the various functions of the P-GW, is performed in the user space. The Cisco Ultra Traffic Optimization engine is integrated into the packet processing path of Cisco's P-GW with a well-defined Application Programming Interface (API) of StarOS.

The following graphic shows a high-level overview of P-GW packet flow with traffic optimization.



## Handling of Traffic Optimization Data Record

The Traffic Optimization Data Record (TODR) is generated only on the expiry of idle-timeout of the Cisco Ultra Traffic Optimization engine. No statistics related to session or flow from P-GW is included in this TODR. The data records are a separate file for the Traffic Optimization statistics, and available to external analytics platform.

### List of Attributes and File Format

All TODR attributes of traffic optimization is enabled by a single CLI command. The output is always comma separated, and in a rigid format.

#### Standard TODR

The following is the format of a Standard TODR:

```
instance_id, flow_type, srcIP, dstIP, policy_id, proto_type, dscp,
flow_first_pkt_rx_time_ms, flow_last_pkt_rx_time_ms, flow_cumulative_rx_bytes
```

Example:

```
1, 0, 173.39.13.38, 192.168.3.106, 0, 1, 0,
1489131332693, 1489131335924, 342292
```

Where:

- *instance\_id*: Instance ID.
- *flow\_type*: Standard flow (0)
- *srcIP*: Indicates the source IP address.

- *dstIP*: Indicates the destination IP address.
- *policy\_id*: Indicates the traffic optimization policy ID.
- *proto\_type*: Indicates the IP protocol being used. The IP protocols are: TCP and UDP.
- *dscp*: Indicates the DSCP code for upstream packets.
- *flow\_first\_pkt\_rx\_time\_ms*: Indicates the timestamp when the first packet was detected during traffic optimization.
- *flow\_last\_pkt\_rx\_time\_ms*: Indicates the timestamp when the last packet was detected during traffic optimization.
- *flow\_cumulative\_rx\_bytes*: Indicates the number of bytes transferred by this flow.

### Large TODR

The following is a sample output of a Large TODR.

```
2,1,2606:ae00:c663:b66f:0000:0058:be03:ae01,0172:0020:0224:0059:2200:0000:0000:0033,0,0,
1489572442058,1489572444759,5051223,1489572444755,3900,3900,0,0,0,11,0,0,11,1,1,
1489572444755,1950,4,0,0,1,1489572446761,0,2010,0,0,1,1489572448771,0,2007,0,0,1,1489572450778,0,2008,
0,0,1,1489572452786,0,2005,0,0,1,1489572454791,0,2003,0,0,1,1489572456794,0,2005,0,0,1,1489572458799,0,
2003,0,0,1,1489572460802,0,1948,0,0,0,1489572462650,0,2002,0,0,1,1489572464652,0,107,0,0,1,1515659685107,0,2007,0,0,0,3,1,1515659687114,0,2004,0,0,0,3
```

Where:

- *instance\_id*: Instance ID.
- *flow\_type*: Large flow (1)
- *srcIP*: Indicates the source IP address.
- *dstIP*: Indicates the destination IP address.
- *policy\_name*: Identifies the name of the configured traffic optimization policy.
- *policy\_id*: Indicates the traffic optimization policy ID.
- *proto\_type*: Indicates the IP protocol being used. The IP protocols are: TCP and UDP.
- *dscp*: Indicates the DSCP code for upstream packets.
- *flow\_first\_pkt\_rx\_time\_ms*: Indicates the timestamp when the first packet was detected during traffic optimization.
- *flow\_last\_pkt\_rx\_time\_ms*: Indicates the timestamp when the last packet was detected during traffic optimization.
- *flow\_cumulative\_rx\_bytes*: Indicates the number of bytes transferred by this flow.
- *large\_detection\_time\_ms*: Indicates the timestamp when the flow was detected as Large.
- *avg\_burst\_rate\_kbps*: Indicates the average rate in Kbps of all the measured bursts.
- *avg\_eff\_rate\_kbps*: Indicates the average effective rate in Kbps.
- *final\_link\_peak\_kbps*: Indicates the highest detected link peak over the life of the Large flow.
- *recovered\_capacity\_bytes*: Indicates the recovered capacity in Kbps for this Large flow.

- *recovered\_capacity\_ms*: Indicates the timestamp of recovered capacity for this Large flow.
- *phase\_count*: Indicates the Large flow phase count.
- *min\_gbr\_kbps*: Indicates the Minimum Guaranteed Bit Rate (GBR) in Kbps.
- *max\_gbr\_kbps*: Indicates the Maximum Guaranteed Bit Rate (MBR) in Kbps.
- *phase\_count\_record*: Indicates the number of phases present in this record.
- *end\_of\_phases*: 0 (not end of phases) or 1 (end of phases).
- Large flow phase attributes:
  - *phase\_type*: Indicates the type of the phase
  - *phase\_start\_time\_ms*: Indicates the timestamp for the start time of the phase.
  - *burst\_bytes*: Indicates the burst size in bytes.
  - *burst\_duration\_ms*: Indicates the burst duration in milliseconds.
  - *link\_peak\_kbps*: Indicates the peak rate for the flow during its life.
  - *flow\_control\_rate\_kbps*: Indicates the rate at which flow control was attempted (or 0 if non-flow control phase).
  - *max\_num\_queued\_packets*: Identifies the maximum number of packets queued.
  - *policy\_id*: Identifies the traffic optimization policy ID.

## Licensing

The Cisco Ultra Traffic Optimization is a licensed Cisco solution. 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*.

## Limitations and Restrictions

- The values which the P-GW chooses to send to the Cisco Ultra Traffic Optimization engine are the values associated from the bearer GBR and bearer MBR.
- In the current implementation, only downlink GBR and MBR are sent to the engine for traffic optimization.
- UDP/QUIC based Traffic Optimization is supported only on PORT 443.

## Configuring Cisco Ultra Traffic Optimization

This section provides information on enabling support for the Cisco Ultra Traffic Optimization solution.

## Loading Traffic Optimization

Use the following configuration under the Global Configuration Mode to load the Cisco Ultra Traffic Optimization as a solution:

```
configure
  require active-charging traffic-optimization
end
```

**Important**

After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

**Important**

Enabling or disabling the traffic optimization can be done through the Service-scheme framework.

**Important**

After you configure the **require active-charging traffic-optimization** CLI command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

**Important**

In 21.3, and 21.5 and later releases, the dependency on the chassis reboot is not valid anymore. The Cisco Ultra Traffic Optimization engine is loaded by default. The Cisco Ultra Traffic Optimization configuration CLIs are available when the license is enabled. As such, the **traffic-optimization** keyword has been deprecated.

## Enabling Cisco Ultra Traffic Optimization Configuration Profile

Use the following configuration under ACS Configuration Mode to enable the Cisco Ultra Traffic Optimization profile:

```
configure
  active-charging service service_name
  traffic-optimization-profile
end
```

**NOTES:**

- The above CLI command enables the Traffic Optimization Profile Configuration, a new configuration mode.

## Configuring the Operating Mode

Use the following CLI commands to configure the operating mode under Traffic Optimization Profile Configuration Mode for the Cisco Ultra Traffic Optimization engine:

```

configure
  active-charging service service_name
    traffic-optimization-profile
      mode [ active | passive ]
    end

```

**Notes:**

- **mode:** Sets the traffic optimization's mode of operation.
- **active:** Active mode where both traffic optimization and flow monitoring are done on the packet.
- **passive:** Passive mode where no flow-control is performed but monitoring is done on the packet.

## Enabling Cisco Ultra Traffic Optimization Configuration Profile Using Service-scheme Framework

The service-scheme framework is used to enable traffic optimization at APN, rule base, QCI, and Rule level. There are two main constructs for the service-scheme framework:

- **Subscriber-base** – This helps in associating subscribers with service-scheme based on the subs-class configuration.
  - **subs-class** – The conditions defined under subs-class enables in classifying the subscribers based on rule base, APN, v-APN name. The conditions can also be defined in combination, and both OR as well as AND operators are supported while evaluating them.
- **Service-scheme** – This helps in associating actions based on trigger conditions which can be triggered either at call-setup time, Bearer-creation time, or flow-creation time.
  - **trigger-condition** – For any trigger, the trigger-action application is based on conditions defined under the trigger-condition.
  - **trigger-actions** – Defines the actions to be taken on the classified flow. These actions can be traffic optimization, throttle-suppress, and so on.

### Session Setup Trigger

The **any-match = TRUE**, a wildcard configuration, is the only supported condition for this trigger and so this is applicable to all the flows of the subscriber.

Following is a sample configuration:

```

configure
  active-charging service service_name
    service-scheme service_scheme_name
      trigger sess-setup
        priority priority_value trigger-condition trigger_condition_name1
      trigger-action trigger_action_name
        exit
      trigger-condition trigger_condition_name1
        any-match = TRUE
        exit

```



```

trigger-action sess-setup
traffic-optimization policy sess-setup
exit

```

## Bearer Creation Trigger

The trigger conditions related to QCI can be used for this trigger, and so this is applicable to all the flows of specific bearers.

The following is a sample configuration:

```

configure
  active-charging service service_name
    trigger-action trigger_action_name
      traffic-optimization
    exit
    trigger-condition trigger_condition_name1
      any-match = TRUE
    exit
    trigger-condition trigger_condition_name2
      qci = qci_value
    exit
  service-scheme service_scheme_name
    trigger bearer-creation
      priority priority_value trigger-condition trigger_condition_name2
  trigger-action trigger_action_name
    exit
  exit
  subs-class sub_class_name
    apn = apn_name
  exit
  subscriber-base subscriber_base_name
    priority priority_value subs-class sub_class_name bind service-scheme
service_scheme_name
end

```

## Flow Creation Trigger

The trigger conditions related to rule-name and QCI can be used here, and so this is related to specific flow.

The following is a sample configuration:

```

configure
  active-charging service service_name
    trigger-action trigger_action_name
      traffic-optimization
    exit
    trigger-condition trigger_condition_name1
      any-match = TRUE
    exit
    trigger-condition trigger_condition_name2
      qci = qci_value
    exit
    trigger-condition trigger_condition_name3

```

```

    rule-name = rule_name
    exit
    service-scheme service_scheme_name
    trigger flow-creation
    priority priority_value trigger-condition trigger_condition_name3
trigger-action trigger_action_name
    exit
    exit
    subs-class sub_class_name
    apn = apn_name
    exit
    subscriber-base subscriber_base_name
    priority priority_value subs-class sub_class_name bind service-scheme
service_scheme_name
    end

```

**Notes:**

- *trigger\_condition\_name3* can have only rules, only QCI, both rule and QCI, or either of rule and QCI.

The following table illustrates the different levels of Traffic Optimization and their corresponding Subscriber Class configuration and Triggers.

Traffic Optimization Levels	Subscriber Class configuration and Triggers
Applicable to all the calls or flows	<pre> subs-class sc1   any-match = TRUE   exit </pre> Sesssetup trigger condition is <b>any-match = TRUE</b>
Applicable to all calls or flows of a rulebase	<pre> subs-class sc1   rulebase = prepaid   exit </pre> Sesssetup trigger condition is <b>any-match = TRUE</b>
Applicable to all calls or flows of an APN	<pre> subs-class sc1   apn = cisco.com   exit </pre> Sesssetup trigger condition is <b>any-match = TRUE</b>
Applicable to all flows of a Bearer	<pre> trigger-condition TC1   qci = 1   exit </pre> Bearer creation trigger condition is TC1
Applicable to a particular flow	<pre> trigger-condition TC1   qci = 1   rule-name = tcp   multi-line-or all-lines   exit </pre> Flow creation trigger condition is TC1

**Important**

In case of LTE to eHRPD handover, since QCI is not valid for eHRPD, it is recommended to configure rule-name as the trigger-condition under service-scheme.

## Generating TODR

Use the following CLI commands under ACS Configuration Mode to enable Traffic Optimization Data Record (TODR) generation:

```
configure
  active-charging service service_name
    traffic-optimization-profile
      data-record
    end
```

**NOTES:**

- If previously configured, use the **no data-record** command to disable generating TODR.

## Multi-Policy Support for Traffic Optimization

Cisco Ultra Traffic Optimization engine supports Traffic Optimization for multiple policies and provides Traffic Optimization for a desired location. It supports a maximum of 32 policies that include two pre-configured policies, by default. Operators can configure several parameters under each Traffic Optimization policy.

This feature includes the following functionalities:

- By default, Traffic Optimization is enabled for TCP and UDP data for a particular Subscriber, Bearer, or Flow that use the Service-Schema.

**Important**

PORT 443 supports UDP or QUIC-based Traffic Optimization.

- Selection of a policy depends on the priority configured. A trigger-condition is used to prioritize a traffic optimization policy. The priority is configurable regardless of a specific location where the traffic optimization policy is applied. Based on the configured priorities, a traffic optimization policy can be overridden by another policy.
- A configuration to associate a traffic optimization policy with a Trigger Action, under the Service-Schema.
- A configuration to select a Traffic Optimization policy for a Location Trigger. Currently, only ECGI Change Detection is supported under the Local Policy Service Configuration mode.

**Important**

Location Change Trigger is not supported with IPSG.




---

**Important** Policy ID for a flow is not recovered after a Session Recovery (SR) or Inter-Chassis Session Recovery (ICSR).

---




---

**Important** The Multi-Policy Support feature requires the same Cisco Ultra Traffic Optimization license key be installed. Contact your Cisco account representative for detailed information on specific licensing requirements.

---

## How Multi-Policy Support Works

### Policy Selection

Cisco's Ultra Traffic Optimization engine provides two default policies – Managed and Unmanaged. When Unmanaged policy is selected, traffic optimization is not performed.

With Managed policy is selected, traffic optimization is performed using default parameters. Managed policy is applied when a policy is not specified in a Trigger Action where traffic optimization is enabled without specifying a policy.

When Managed policy is selected, traffic optimization is performed using default parameters. Managed policy is applied when a policy is not specified in a Trigger Action where traffic optimization is enabled without specifying a policy.

- Session Setup Trigger – If a trigger-action is applied only for a Session Setup in a Service-Schema, then the trigger action is only applied to new sessions only.
- Bearer Setup Trigger – If a trigger-action is applied only for a Bearer Setup, changes in the trigger action are applicable to newly created bearers and its flows.
- Flow Creation Trigger – Under a trigger condition corresponding to flow create, conditions can be added based on a rule-name, local-policy-rule or an IP protocol in addition to the trigger condition: any-match.

When the traffic optimization is disabled on existing flows based on of a trigger condition, the traffic optimization engine applies the default Unmanaged policy on them.

### Deleting a Policy

All association to a traffic optimization policy must be removed before deleting a policy.

For more information, refer to the *Traffic Optimization Policy Configuration* section.

## Configuring Multi-Policy Support

The following sections describes the required configurations to support the Multi-Policy Support.

### Configuring a Traffic Optimization Profile

Use the following CLI commands to configure a Traffic Optimization Profile.

```
configure
  require active-charging
  active-charging service service_name
```

```

traffic-optimization-profile profile_name
  data-record[ large-flows-only | managed-large-flows-only ]
  no data record
  [ no ] efd-flow-cleanup-interval cleanup_interval
  [ no ] stats-interval stats_interval
  [ no ] stats-options { flow-analyst [ flow-trace ] | flow-trace [
flow-analyst ] }
end

```

## NOTES:

- **require active-charging**: Enables the configuration requirement for an Active Charging service.




---

**Important** After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

---

- **data-record**: Enables the generation of traffic optimization data record.
- **large-flows-only**: Enables the traffic optimization data record generation for large flows.
- **managed-large-flows-only**: Enables the traffic optimization data record generation for managed large flows.

The keywords - **large-flows-only** and **managed-large-flows-only** when configured along with **data-record** enables the CUTO library to stream the respective statistics as part of the **stats-options** command, to the external server. The operator can configure a combination of the **stats-options** keywords **flow-trace** and **flow-analyst** and the **data-record** command to notify the CUTO library accordingly.




---

**Note** One of the above the two keywords can be configured as part of the data-record, which enables the CUTO library to stream the respective statistics.

---

The default behavior of the **data-record** command is not affected with the above implementation . If configured without any of the options, then TODRs are generated for all standard and large flows, which is the existing behavior.

- **efd-flow-cleanup-interval**: Configures the EFD flow cleanup interval. The interval value is an integer that ranges 10–5000 milliseconds.
- **stats-interval**: Configures the flow statistics collection and reporting interval in seconds. The interval value is an integer that ranges 1–60 seconds.
- **stats-options**: Configures options to collect the flow statistics. It only specifies whether the stream must be a Flow Trace or a Flow Analyst or both, to an external server.




---

**Note** From Release 21.6 onwards, the **heavy-session** command is deprecated.

---

## Configuring a Traffic Optimization Policy

Use the following CLI commands to configure a Traffic Optimization Policy.

```
configure
  require active-charging
  active-charging service service_name
    [ no ] traffic-optimization-policy policy_name
      bandwidth-mgmt { backoff-profile [ managed | unmanaged ] [
min-effective-rate effective_rate [ min-flow-control-rate flow_rate ] |
min-flow-control-rate flow_rate [ min-effective-rate effective_rate ] ] |
min-effective-rate effective_rate [ backoff-profile [ managed | unmanaged ]
[ min-flow-control-rate flow_rate ] | min-flow-control-rate control_rate [
backoff-profile [ managed | unmanaged ] ] | min-flow-control-rate [ [
backoff-profile [ managed | unmanaged ] [ min-effective-rate effective_rate
] | [ min-effective-rate effective_rate ] [ backoff-profile [ managed |
unmanaged ] ] ] }
      [ no ] bandwidth-mgmt
        curbing-control { max-phases max_phase_value [ rate curbing_control_rate
[ threshold-rate threshold_rate [ time curbing_control_duration ] ] ] | rate
curbing_control_rate [ max-phases [ threshold-rate threshold_rate [ time
curbing_control_duration ] ] ] | threshold-rate [ max-phases max_phase_value [
rate curbing_control_rate [ time curbing_control_duration ] ] ] | time [ max-phases
max_phase_value [ rate curbing_control_rate [ threshold-rate threshold_rate] ] ]
}
        [ no ] curbing-control
        heavy-session { standard-flow-timeout [ threshold threshold_value |
threshold threshold_value [ standard-flow-timeout timeout_value ] }
        [ no ] heavy-session
        link-profile { initial-rate initial_seed_value [ max-rate
max_peak_rate_value [ peak-lock ] ] | max-rate [ initial-rate initial_seed_value
[ peak-lock ] ] | peak-lock [ initial-rate initial_seed_value [ max-rate
max_peak_rate_value ] ] }
        [ no ] link-profile
        session-params { tcp-ramp-up tcp_rampup_duration [ udp-ramp-up
udp_rampup_duration ] | udp-ramp-up udp_rampup_duration [ tcp-ramp-up
tcp_rampup_duration ] }
        [ no ] session-params
      end
```

### NOTES:

- **no:** Overwrites the configured parameters with default values. The operator must remove all associated policies in a policy profile before deleting a policy profile. Otherwise, the following error message is displayed:  
Failure: traffic-optimization policy in use, cannot be deleted.
- **bandwidth-mgmt:** Configures bandwidth management parameters.
- **backoff-profile:** Determines the overall aggressiveness of the back off rates.
- **managed:** Enables both traffic monitoring and traffic optimization.
- **unmanaged:** Only enables traffic monitoring.

- **min-effective-rate**: Configures minimum effective shaping rate in Kbps. The shaping rate value is an integer ranging 100–10000.
- **min-flow-control-rate**: Configures the minimum rate that is allowed in Kbps to control the flow of heavy-session-flows during congestion. The control rate value is an integer ranging 100–10000.
- **curbing-control**: Configures curbing flow control related parameters.
- **max-phases**: Configures consecutive phases where the target shaping rate is below **threshold-rate** to trigger curbing flow control. The maximum phase value is an integer ranging 2–10.
- **rate**: Configures the curbing flow-control at a fixed rate in Kbps instead of a dynamic rate. The control rate value is an integer ranging 0–10000. To disable the fixed flow control rate, set the flow control rate value to 0.
- **threshold-rate**: Configures the minimum target shaping rate in kbps to trigger curbing. The threshold rate is an integer ranging 100–10000.
- **time**: Configures the duration of a flow control phase in milliseconds. The flow control duration value is an integer ranging 0–600000. To disable flow control, set the flow control duration value to 0.
- **heavy-session**: Configures parameters for heavy-session detection.
- **standard-flow-timeout**: Configures the idle timeout in milliseconds, for expiration of standard flows. The timeout value is an integer ranging 100–3000.
- **threshold**: Configures heavy-session detection threshold in bytes. On reaching the threshold, the flow is monitored and potentially managed. The threshold value is an integer ranging 0–100000000.
- **link-profile**: Configures link profile parameters.
- **initial-rate**: Configures the initial seed value of the acquired peak rate in Kbps for a traffic session. The initial seed value is an integer ranging 100–30000.
- **max-rate**: Configures the maximum learned peak rate that is allowed in Kbps for a traffic session. The max rate value is an integer ranging 100–30000.
- **peak-lock**: Confirms with the link peak rate available at the initial link peak rate setting.
- **session-params**: Configures session parameters.
- **tcp-ramp-up**: Configures the ramp-up-phase duration in milliseconds, for TCP traffic. The TCP ramp-up duration is an integer ranging 0–5000.
- **udp-ramp-up**: Configures the ramp-up-phase duration in milliseconds, for the UDP traffic. The UDP ramp-up duration is an integer ranging 0–5000.

**Important**

After you configure **require active-charging** command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

**Traffic Optimization Policy - Default Values**

Bandwidth-Mgmt:

```

Backoff-Profile      : Managed
Min-Effective-Rate  : 600 (kbps)
Min-Flow-Control-Rate : 250 (kbps)

Curbing-Control:

Time                : 0 (ms)
Rate                : 600 (kbps)
Max-Phases         : 2
Threshold-Rate     : 600 (kbps)

Heavy-Session:

Threshold           : 4000000 (bytes)
Standard-Flow-Timeout : 500 (ms)

Link-Profile:

Initial-Rate       : 7000 (kbps)
Max-Rate           : 10000 (kbps)
Peak-Lock          : Disabled

Session-Params:

Tcp-Ramp-Up        : 5000 (ms)
Udp-Ramp-Up        : 0 (ms)

```

## Associating a Trigger Action to a Traffic Optimization Policy

Use the following CLI commands to associate a Trigger Action to a Traffic Optimization Policy.

```

configure
  require active-charging
  active-charging service service_name
    trigger-action trigger_action_name
    traffic-optimization policy policy_name
    [ no ] traffic-optimization
  end

```

### NOTES:

- **traffic-optimization policy**: Configures a traffic optimization policy.
- **no**: Removes the configured traffic optimization policy.



### Important

After you configure **require active-charging** command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

## Enabling TCP and UDP

Use the following CLI commands to enable TCP and UDP protocol for Traffic Optimization:

```

configure
  require active-charging
  active-charging service service_name

```



```

trigger-condition trigger_condition_name
  [ no ] ip protocol = [ tcp | udp ]
end

```

**NOTES:**

- **no**: Deletes the Active Charging Service related configuration.
- **ip**: Establishes an IP configuration.
- **protocol**: Indicates the protocol being transported by the IP packet.
- **tcp**: Indicates the TCP protocol to be transported by the IP packet.
- **udp**: Indicates the UDP protocol to be transported by the IP packet.

**Important**

After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

## Service-Scheme Configuration for Multi-Policy Support

The service-schema framework enables traffic optimization at APN, rule base, QCI, and Rule level. In 21.6, with the Multi-Policy Support feature, traffic optimization in a service-schema framework allows the operator to configure multiple policies and to configure traffic optimization based on a desirable location.

The service-schema framework helps in associating actions based on trigger conditions, which can be triggered either at call-setup time, Bearer-creation time, or flow-creation time.

### Session Setup Trigger

The **any-match = TRUE**, a wildcard configuration, is the only supported condition for this trigger and so this is applicable to all the flows of the subscriber.

Use the following configuration to setup a Session Trigger:

```

configure
  active-charging service service_name
    trigger-action trigger_action_name
      traffic-optimization
    exit
  trigger-condition trigger_condition_name1
    any-match = TRUE
  exit
  service-scheme service_scheme_name
    trigger sess-setup
      priority priority_value trigger-condition trigger_condition_name1
  trigger-action trigger_action_name
    exit
  subs-class sub_class_name
    apn = apn_name
  exit
  subscriber-base subscriber_base_name

```

```

        priority priority_value subs-class sub_class_name bind service-scheme
service_scheme_name
    end

```

### Sample Configuration

Following is a sample configuration for Session Setup Trigger:

```

service-scheme SS1
    trigger sess-setup
        priority 1 trigger-condition sess-setup trigger-action sess-setup
    #exit
    trigger-condition sess-setup
        any-match = TRUE
    #exit
    trigger-action sess-setup
        traffic-optimization policy sess-setup
    #exit

```

### Bearer Creation Trigger

The trigger conditions related to QCI can be used for this trigger, and so this is applicable to all the flows of specific bearers.

Use the following configuration to configure a Bearer Creation Trigger:

```

configure
    active-charging service service_name
        service-scheme service_scheme_name
            trigger bearer-creation
                priority priority_value trigger-condition trigger_condition_name2
trigger-action trigger_action_name
                exit
                trigger-condition trigger_condition_name2
                    qci = qci_value
                exit
            trigger-action bearer-creation
                traffic-optimization policy bearer-creation
            exit

```

### Sample Configuration

The following is a sample configuration for Bearer Creation Trigger:

```

service-scheme SS1
    trigger bearer-creation
        priority 1 trigger-condition bearer-creation trigger-action bearer-creation
    #exit
    trigger-condition bearer-creation
        qci = 1 to 2
    #exit
    trigger-action bearer-creation
        traffic-optimization policy bearer-creation
    #exit

```

### Flow Creation Trigger

The trigger conditions related to rule-name and QCI can be used here, and so this is related to specific flow.

Use the following configuration to configure a flow creation trigger:

```
configure
  active-charging service service_name
    service-scheme service_scheme_name
      trigger bearer-creation
        priority priority_value trigger-condition trigger_condition_name
      trigger-action trigger_action_name
        exit
      trigger-condition trigger_condition_name
        ip-protocol = protocol_type
        rule-name = rule_name
        **Multi-line or All-lines**
      exit
```

### Sample Configuration

The following is a sample configuration for Flow Creation Trigger using the default Cisco Ultra Traffic Optimization policy:

```
service-scheme SS1
  trigger flow-create
    priority 1 trigger-condition TC5 trigger-action TA4
  #exit
  trigger-condition TC5
    ip protocol = tcp
    ip protocol = udp
    multi-line-or all-lines
  #exit
  trigger-action TA4
    traffic-optimization
  #exit
```

### Configuring: ecgi-change

The following demonstrates ecgi-change sample configuration:

#### Trigger Condition and Trigger Action in ACS Configuration

```
configure
active-charging-service ACS
  trigger-action TA1
    traffic-optimization policy flow-create-ecgi-change
  #exit
  trigger-condition TC4
    local-policy-rule = ruledef-ecgi
  #exit
end
```

#### Service Schema Configuration

```
configure
active-charging-service ACS
  service-scheme SS1
    trigger flow-create
      priority 2 trigger-condition TC4 trigger-action TA1
    #exit
  subs-class SC1
    any-match = TRUE
  #exit
```

```

subscriber-base SB1
  priority 1 subs-class SC1 bind service-scheme SS1
  #exit
end

```

## Local Policy Configuration

```

local-policy-service LP
  ruledef anymatch
    condition priority 1 imsi match *
  #exit
  ruledef ecgi-1
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AE7F0A 1AE7F0B 1AE7F28 1AE7F29
    1AE7F46 1AE7F47 1AEAC00 1AEAC01 1AEAC02 1AEAC0A 1AEAC0B 1AEAC0C 1AEAC14 1AEAC15 1AEAC16
    1AEAC28 1AEAC29 1AEAC2A 1AEAC46 1AEAC47 1AEAC48 1AEAC50 1AEAC51 1AEAC52 1AEAC6E 1AEAC6F
    1AEAC70 1AEAC78 1AEAC79 1AEAC7A
  #exit
  ruledef ecgi-10
    condition priority 1 ecgi mcc 300 mnc 235 eci match 1F36C52 1F36C6E 1F36C6F 1F36C70
    1F36C78 1F36C79 1F36C7A
  #exit
  ruledef ecgi-2
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AEBE01 1AEBE02 1AEBE0B 1AEBE0C
    1AEBE15 1AEBE16 1AEBE29 1AEBE2A 1AEBE47 1AEBE48 1AEBF00 1AEBF01 1AEBF02 1AEBF0A 1AEBF0B
    1AEBF0C 1AEBF14 1AEBF15 1AEBF16 1AEBF1E 1AEBF1F 1AEBF20 1AEBF28 1AEBF29 1AEBF2A 1AEBF46
  #exit
  ruledef ecgi-3
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AEBF47 1AEBF48 1AEBF50 1AEBF51
    1AEBF52 1AEBF6E 1AEBF6F 1AEBF70 1AEBF78 1AEBF79 1AEBF7A 1AF0E00 1AF0E01 1AF0E02 1AF0E0A
    1AF0E0B 1AF0E0C 1AF0E14 1AF0E15 1AF0E16 1AF0E28 1AF0E29 1AF0E2A 1AF0E46
  #exit
  ruledef ecgi-4
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AF0E47 1AF0E48 1AF4A0A 1AF4A0B
    1AF4A14 1AF4A15 1AF4A28 1AF4A29 1AF4A46 1AF4A47 1AF4D00 1AF4D01 1AF4D0A 1AF4D0B 1AF4D14
    1AF4D15 1AF4D28 1AF4D29 1AF4D46 1AF4D47 1AF4D50 1AF4D51 1AF4D6E 1AF4D6F
  #exit
  ruledef ecgi-5
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AF4D78 1AF4D79 1AF7200 1AF7201
    1AF7202 1AF720A 1AF720B 1AF720C 1AF7214 1AF7215 1AF7216 1AF721E 1AF721F 1AF7444 1AF7228
    1AF7229 1AF722A 1AF7246 1AF7247 1AF7248 1AF7250 1AF7251 1AF7252 1AF726E
  #exit
  ruledef ecgi-6
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1AF726F 1AF7270 1B04C00 1B04C01
    1B04C02 1B04C03 1B04C0A 1B04C0B 1B04C0C 1B04C0D 1B04C14 1B04C15 1B04C16 1B04C17 1B04C1E
    1B04C1F 1B04C20 1B04C21 1B04C28 1B04C29 1B04C2A 1B04C2B 1B04C46 1B04C47
  #exit
  ruledef ecgi-7
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1B04C48 1B04C49 1B04C50 1B04C51
    1B04C52 1B04C53 1B04C6E 1B04C6F 1B04C70 1B04C71 1B04C78 1B04C79 1B04C7A 1B04C7B 1B05300
    1B05301 1B05302 1B0530A 1B0530B 1B0530C 1B05314 1B05315 1B05316 1B05328 1B05329
  #exit
  ruledef ecgi-8
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1B0532A 1B05346 1B05347 1B05348
    1B32F00 1B32F01 1B32F02 1B32F0A 1B32F0B 1B32F0C 1B32F14 1B32F15 1B32F16 1B32F28 1B32F29
    1B32F2A 1B32F46 1B32F47 1B32F48 1B76400 1B76401 1B76402 1B7640A 1B7640B 1B7640C 1B76428
  #exit
  ruledef ecgi-9
    condition priority 1 ecgi mcc 111 mnc 444 eci match 1B76429 1B7642A 1B76446 1B76447
    1B76448 1F36C00 1F36C01 1F36C02 1F36C0A 1F36C0B 1F36C0C 1F36C14 1F36C15 1F36C16 1F36C1E
    1F36C1F 1F36C20 1F36C28 1F36C29 1F36C2A 1F36C46 1F36C47 1F36C48 1F36C50 1F36C51
  #exit
  actiondef activate_lp_action
    action priority 1 activate-lp-rule name ruledef-tai
  #exit

```

```

actiondef activate_lp_action1
  action priority 3 event-triggers ecgi-change
#exit
actiondef ecgi_change
  action priority 1 activate-lp-rule name ruledef-ecgi
#exit
eventbase default
  rule priority 1 event new-call ruledef anymatch actiondef activate_lp_action1 continue

  rule priority 11 event new-call ruledef ecgi-1 actiondef ecgi_change continue
  rule priority 12 event new-call ruledef ecgi-2 actiondef ecgi_change continue
  rule priority 13 event new-call ruledef ecgi-3 actiondef ecgi_change continue
  rule priority 14 event new-call ruledef ecgi-4 actiondef ecgi_change continue
  rule priority 15 event new-call ruledef ecgi-5 actiondef ecgi_change continue
  rule priority 16 event new-call ruledef ecgi-6 actiondef ecgi_change continue
  rule priority 17 event new-call ruledef ecgi-7 actiondef ecgi_change continue
  rule priority 18 event new-call ruledef ecgi-8 actiondef ecgi_change continue
  rule priority 19 event new-call ruledef ecgi-9 actiondef ecgi_change continue
  rule priority 20 event new-call ruledef ecgi-10 actiondef ecgi_change continue
  rule priority 21 event ecgi-change ruledef ecgi-1 actiondef ecgi_change continue
  rule priority 22 event ecgi-change ruledef ecgi-2 actiondef ecgi_change continue
  rule priority 23 event ecgi-change ruledef ecgi-3 actiondef ecgi_change continue
  rule priority 24 event ecgi-change ruledef ecgi-4 actiondef ecgi_change continue
  rule priority 25 event ecgi-change ruledef ecgi-5 actiondef ecgi_change continue
  rule priority 26 event ecgi-change ruledef ecgi-6 actiondef ecgi_change continue
  rule priority 27 event ecgi-change ruledef ecgi-7 actiondef ecgi_change continue
  rule priority 28 event ecgi-change ruledef ecgi-8 actiondef ecgi_change continue
  rule priority 29 event ecgi-change ruledef ecgi-9 actiondef ecgi_change continue
  rule priority 30 event ecgi-change ruledef ecgi-10 actiondef ecgi_change continue
#exit
#exit
end

```

### Traffic Optimization Policy Configuration

```

configure
active-charging-service ACS
traffic-optimization-policy Config:
  traffic-optimization-policy flow-create-ecgi-change
  heavy-session threshold 40000
#exit
end

```

## Local Policy Configuration



**Important** Configuring Local Policy needs a Local Policy Decision Engine License. Contact your Cisco account representative for information on specific licensing requirements.

This section describes the traffic optimization policy configuration that is based on location.

Use the following sample configuration to enable a eCGI change rule:

```

configure
  active-charging service service_name
  local-policy-service service_name
  ruledef ruledef_name
    condition priority priority_value ecgi mccmcc_value mnc mnc_value eq
    eq_value
  exit

```

```

actiondef actiondef_name1
    action priority priority_value event-triggers actiondef_name2
    exit
actiondef actiondef_name2
    action priority priority_value activate-lp-rule ruledef_name
    exit
eventbase eventbase_name
    rule priority priority_value event event_name ruledef ruledef_name
actiondef actiondef_name1 continue
    rule priority priority_value event event_name ruledef ruledef_name
actiondef actiondef_name1 continue
    exit

```

### Service-Scheme Configuration

```

configure
    active-charging service service_name
        service-scheme service_scheme_name
            trigger flow-create
                priority priority_value trigger-condition trigger_condition_name
trigger-action trigger_action_name
    exit
    trigger condition trigger_condition_name
        local-policy-rule = rule_name
    exit
    trigger action trigger_action_name
        traffic-optimization policy policy_name
    exit

```

## Configuring L7 Rule



### Important

Configuring L7 Rule needs an Application Detection Control License. Contact your Cisco account representative for detailed information on specific licensing requirements.

Use the following CLI to configure an L7 rule:

```

configure
    active-charging service service_name
        service-scheme service_scheme_name
            trigger bearer-creation
                priority priority_value trigger-condition trigger_condition_name
trigger-action trigger_action_name
    exit
    trigger-condition trigger_condition_name
        rule-name = rule_name
        rule-name = rule_name
        **Multi-line or All-lines**
    trigger-action trigger_action_name
        traffic-optimization policy policy_name
    exit

```

### Sample Configuration

The following is a sample configuration for L7 Rules:

```
service-scheme SS1
  trigger flow-create
    priority 1 trigger-condition TC6 trigger-action TA6
  #exit
  trigger-condition TC6
    rule-name = whatsapp
    rule-name = http
    multi-line-or all-lines
  #exit
  trigger-action TA6
    traffic-optimization policy flow-create-L7-Rules
  #exit
```

### Ookla Speedtest

Use the configuration information discussed in the section [Configuring L7 Rule, on page 22](#).

### Sample Configuration

The following is a sample configuration for Ookla Speedtest:

```
service-scheme SS1
  trigger flow-create
    priority 1 trigger-condition ookla trigger-action ookla
  #exit
  trigger-condition ookla
    rule-name = speedtest
  #exit
  trigger-action ookla
    no traffic-optimization
  #exit
```

### Location and App-based Configuration

#### Sample Configuration

```
service-scheme SS1
  trigger flow-create
    priority 1 trigger-condition TC3 trigger-action TA2
  #exit
  trigger-condition TC3
    local-policy-rule = ruledef-ecgi
    rule-name = youtube
    rule-name = whatsapp
    multi-line-or all-lines
  #exit
  trigger-action TA2
    traffic-optimization policy flow-create-ecgi-change
  #exi
```

### Selective Configuration by Disabling TCP and UDP

#### Sample Configuration

```
service-scheme SS1
  trigger flow-create
    priority 1 trigger-condition tcponly trigger-action tcponly
    priority 2 trigger-condition udponly trigger-action udponly
```

```

#exit
trigger-condition tcponly
    ip protocol = tcp
#exit
trigger-condition udponly
    ip protocol = udp
#exit
trigger-action tcponly
    no traffic-optimization
#exit
trigger-action udponly
    no traffic-optimization
#exit

```

## L7/ADC and Location Trigger based Configuration

### Sample Configuration

This sample configuration describes a scenario where an operator wants to always disable Traffic Optimization for Speedtest. The configuration disables traffic optimization regardless of the location. It applies a specific policy for a specific location (ECGI) (except for Speedtest) and overrides any other policy set by any trigger condition.

Also, for a specific policy optimization, for example: YouTube, the policy selection is prioritized as follows:

Service Scheme Configuration:

```

service-scheme SS1
trigger flow-create
    priority 1 trigger-condition speedtest-tc trigger-action speedtest-ta
    priority 2 trigger-condition location-tc trigger-action location-ta
    priority 3 trigger-condition youtube-tc trigger-action youtube-ta
#exit
trigger-condition location-tc
    local-policy-rule = ruledef-ecgi
#exit
trigger-action location-ta
    traffic-optimization policy flow-create-ecgi-change
#exit
trigger-condition speedtest-tc
    *rule-name = speedtest
#exit
trigger-action speedtest-ta
    no traffic-optimization
#exit
trigger-condition youtube-tc
    rule-name = youtube
#exit
trigger-action youtube-ta
    traffic-optimization policy youtube-policy
#exit

```

\* Provided rule-name = speedtest, is configured such that it always detects this traffic.

## Monitoring and Troubleshooting

This section provides information regarding commands available to monitor and troubleshoot the Cisco Ultra Traffic Optimization solution on the P-GW.



## Cisco Ultra Traffic Optimization Show Commands and/or Outputs

This section provides information about show commands and the fields that are introduced in support of Cisco Ultra Traffic Optimization solution.

### show active-charging traffic-optimization counters { tcp | udp } sessmgr all

The **show active-charging traffic-optimization counters sessmgr { all | instance *number* }** CLI command is introduced where:

- **counters** – Displays aggregate flow counters/statistics from Cisco Ultra Traffic Optimization engine.



---

**Important**

This CLI command is license dependent and visible only if the license is loaded.

---

Following are the new field/counters:

- Traffic Optimization Flows:
  - Active Normal Flow Count:
  - Active Large Flow Count:
  - Active Managed Large Flow Count:
  - Active Unmanaged Large Flow Count:
  - Total Normal Flow Count:
  - Total Large Flow Count:
  - Total Managed Large Flow Count:
  - Total Unmanaged Large Flow Count:
  - Total IO Bytes:
  - Total Large Flow Bytes:
  - Total Recovered Capacity Bytes:
  - Total Recovered Capacity ms:

On executing the above command, the following new fields are displayed for the Multi-Policy Support feature:



---

**Important**

This CLI command is license dependent and visible only if the license is loaded.

---

- TCP Traffic Optimization Flows:
  - Active Normal Flow Count:
  - Active Large Flow Count:
  - Active Managed Large Flow Count:

- Active Unmanaged Large Flow Count:
- Total Normal Flow Count:
- Total Large Flow Count:
- Total Managed Large Flow Count:
- Total Unmanaged Large Flow Count:
- Total IO Bytes:
- Total Large Flow Bytes:
- Total Recovered Capacity Bytes:
- Total Recovered Capacity ms:
  
- UDP Traffic Optimization Flows:
  - Active Normal Flow Count:
  - Active Large Flow Count:
  - Active Managed Large Flow Count:
  - Active Unmanaged Large Flow Count:
  - Total Normal Flow Count:
  - Total Large Flow Count:
  - Total Managed Large Flow Count:
  - Total Unmanaged Large Flow Count:
  - Total IO Bytes:
  - Total Large Flow Bytes:
  - Total Recovered Capacity Bytes:
  - Total Recovered Capacity ms:

## show active-charging traffic-optimization info

This show command has been introduced in Exec Mode, where:

- **traffic-optimization** – Displays all traffic optimization options.
- **info** – Displays Cisco Ultra Traffic Optimization engine information.

The output of this CLI command displays the version, mode, and configuration values.

Following are the new fields/counters:

- Version:
- Mode:

- Configuration:
  - Data Records (TODR)
  - Statistics Options
  - EFD Flow Cleanup Interval
  - Statistics Interval

## show active-charging traffic-optimization policy

On executing the above command, the following new fields are displayed for the Multi-Policy Support feature:

- Policy Name
- Policy-Id
- Bandwidth-Mgmt
  - Backoff-Profile
  - Min-Effective-Rate
  - Min-Flow-Control-Rate
- Curbing-Control
  - Time
  - Rate
  - Max-phases
  - Threshold-Rate
- Link-Profile
  - Initial-Rate
  - Max-Rate
  - Peak-Lock
- Session-Params
  - Tcp-Ramp-Up
  - Udp-Ramp-Up

## Bulk Statistics

The following bulk statistics are added in the ECS schema to support Large and Managed flows:

<b>Bulk Statistics</b>	<b>Description</b>
tcp-active-normal-flow-count	Indicates the number of TCP active-normal-flow count for Cisco Ultra Traffic Optimization.
tcp-active-large-flow-count	Indicates the number of TCP active-large-flow count for Cisco Ultra Traffic Optimization.
tcp-active-managed-large-flow-count	Indicates the number of TCP active-managed-large-flow count for Cisco Ultra Traffic Optimization.
tcp-active-unmanaged-large-flow-count	Indicates the number of TCP active-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-normal-flow-count	Indicates the number of TCP total-normal-flow count for Cisco Ultra Traffic Optimization.
tcp-total-large-flow-count	Indicates the number of TCP total-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-managed-large-flow-count	Indicates the number of TCP total-managed-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-unmanaged-large-flow-count	Indicates the number of TCP total-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
tcp-total-io-bytes	Indicates the number of TCP total-IO bytes for Cisco Ultra Traffic Optimization.
tcp-total-large-flow-bytes	Indicates the number of TCP total-large-flow bytes for Cisco Ultra Traffic Optimization.
tcp-total-recovered-capacity-bytes	Indicates the number of TCP total-recovered capacity bytes for Cisco Ultra Traffic Optimization.
tcp-total-recovered-capacity-ms	Indicates the number of TCP total-recovered capacity ms for Cisco Ultra Traffic Optimization.
udp-active-normal-flow-count	Indicates the number of UDP active-normal-flow count for Cisco Ultra Traffic Optimization.
udp-active-large-flow-count	Indicates the number of UDP active-large-flow count for Cisco Ultra Traffic Optimization.
udp-active-managed-large-flow-count	Indicates the number of UDP active-managed-large-flow count for Cisco Ultra Traffic Optimization.
udp-active-unmanaged-large-flow-count	Indicates the number of UDP active-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-normal-flow-count	Indicates the number of UDP total-normal-flow count for Cisco Ultra Traffic Optimization.

Bulk Statistics	Description
udp-total-large-flow-count	Indicates the number of UDP total-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-managed-large-flow-count	Indicates the number of UDP total-managed-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-unmanaged-large-flow-count	Indicates the number of UDP total-unmanaged-large-flow count for Cisco Ultra Traffic Optimization.
udp-total-io-bytes	Indicates the number of UDP total-IO bytes for Cisco Ultra Traffic Optimization.
udp-total-large-flow-bytes	Indicates the number of UDP total-large-flow bytes for Cisco Ultra Traffic Optimization.
udp-total-recovered-capacity-bytes	Indicates the number of UDP total-recovered capacity bytes for Cisco Ultra Traffic Optimization.
udp-total-recovered-capacity-ms	Indicates the number of UDP total-recovered capacity ms for Cisco Ultra Traffic Optimization.
tcp-uplink-drop	Indicates the number of TCP uplink-drop for Cisco Ultra Traffic Optimization.
tcp-uplink-hold	Indicates the number of TCP uplink-hold for Cisco Ultra Traffic Optimization.
tcp-uplink-forward	Indicates the number of TCP uplink-forward for Cisco Ultra Traffic Optimization.
tcp-uplink-forward-and-hold	Indicates the number of TCP uplink-forward and hold for Cisco Ultra Traffic Optimization.
tcp-uplink-hold-failed	Indicates the number of TCP uplink-hold-failed for Cisco Ultra Traffic Optimization.
tcp-uplink-bw-limit-flow-sent	Indicates the number of TCP uplink-bw limit-flow sent for Cisco Ultra Traffic Optimization.
tcp-dnlink-drop	Indicates the number of TCP downlink-drop for Cisco Ultra Traffic Optimization.
tcp-dnlink-hold	Indicates the number of TCP downlink-hold for Cisco Ultra Traffic Optimization.
tcp-dnlink-forward	Indicates the number of TCP downlink-forward for Cisco Ultra Traffic Optimization.
tcp-dnlink-forward-and-hold	Indicates the number of TCP downlink-forward and hold for Cisco Ultra Traffic Optimization.

<b>Bulk Statistics</b>	<b>Description</b>
tcp-dnlink-hold-failed	Indicates the number of TCP downlink-hold-failed for Cisco Ultra Traffic Optimization.
tcp-dnlink-bw-limit-flow-sent	Indicates the number of TCP downlink-bw limit-flow sent for Cisco Ultra Traffic Optimization.
tcp-dnlink-async-drop	Indicates the number of TCP downlink-async-drop for Cisco Ultra Traffic Optimization.
tcp-dnlink-async-hold	Indicates the number of TCP downlink-async-hold for Cisco Ultra Traffic Optimization.
tcp-dnlink-async-forward	Indicates the number of TCP downlink-async-forward for Cisco Ultra Traffic Optimization.
tcp-dnlink-async-forward-and-hold	Indicates the number of TCP downlink-async-forward and hold for Cisco Ultra Traffic Optimization.
tcp-dnlink-async-hold-failed	Indicates the number of TCP downlink-async-hold-failed for Cisco Ultra Traffic Optimization.
tcp-process-packet-drop	Indicates the number of TCP process-packet-drop for Cisco Ultra Traffic Optimization.
tcp-process-packet-hold	Indicates the number of TCP process-packet-hold for Cisco Ultra Traffic Optimization.
tcp-process-packet-forward	Indicates the number of TCP process-packet-forward for Cisco Ultra Traffic Optimization.
tcp-process-packet-forward-failed	Indicates the number of TCP process-packet-forward-failed for Cisco Ultra Traffic Optimization.
tcp-process-packet-forward-and-hold	Indicates the number of TCP process-packet-forward and hold for Cisco Ultra Traffic Optimization.
tcp-process-packet-forward-and-hold-failed	Indicates the number of TCP process-packet-forward and hold-failed for Cisco Ultra Traffic Optimization.
tcp-pkt-copy	Indicates the number of TCP packet-copy for Cisco Ultra Traffic Optimization.
tcp-pkt-Copy-failed	Indicates the number of TCP packet-copy-failed for Cisco Ultra Traffic Optimization.
tcp-process-pkt-copy	Indicates the number of TCP process-packet-copy for Cisco Ultra Traffic Optimization.
tcp-process-pkt-copy-failed	Indicates the number of TCP process-packet-copy-failed for Cisco Ultra Traffic Optimization.

Bulk Statistics	Description
tcp-process-pkt-no-packet-found-action-forward	Indicates the number of TCP process packet, no packet found, and action forward for Cisco Ultra Traffic Optimization.
tcp-process-pkt-no-packet-found-forward-and-hold	Indicates the number of TCP process packet, no packet found, action forward and hold for Cisco Ultra Traffic Optimization.
tcp-process-pkt-no-packet-found-action-drop	Indicates the number of TCP process packet, no packet found, action drop for Cisco Ultra Traffic Optimization.
tcp-todrs-generated	Indicates the number of TCP TODRs generated for Cisco Ultra Traffic Optimization.
udp-uplink-drop	Indicates the number of UDP uplink-drop for Cisco Ultra Traffic Optimization.
udp-uplink-hold	Indicates the number of UDP uplink-hold for Cisco Ultra Traffic Optimization.
udp-uplink-forward	Indicates the number of UDP uplink-forward for Cisco Ultra Traffic Optimization.
udp-uplink-forward-and-hold	Indicates the number of UDP uplink-forward and hold for Cisco Ultra Traffic Optimization.
udp-uplink-hold-failed	Indicates the number of UDP uplink-hold failed for Cisco Ultra Traffic Optimization.
udp-uplink-bw-limit-flow-sent	Indicates the number of UDP uplink-bw limit-flow sent for Cisco Ultra Traffic Optimization.
udp-dnlink-drop	Indicates the number of UDP downlink-drop for Cisco Ultra Traffic Optimization.
udp-dnlink-hold	Indicates the number of UDP downlink-hold for Cisco Ultra Traffic Optimization.
udp-dnlink-forward	Indicates the number of UDP downlink-forward for Cisco Ultra Traffic Optimization.
udp-dnlink-forward-and-hold	Indicates the number of UDP downlink-forward and hold for Cisco Ultra Traffic Optimization.
udp-dnlink-hold-failed	Indicates the number of UDP downlink-hold failed for Cisco Ultra Traffic Optimization.
udp-dnlink-bw-limit-flow-sent	Indicates the number of UDP downlink-bw limit-flow sent for Cisco Ultra Traffic Optimization.
udp-dnlink-async-drop	Indicates the number of UDP downlink-async-drop for Cisco Ultra Traffic Optimization.

<b>Bulk Statistics</b>	<b>Description</b>
udp-dnlink-async-hold	Indicates the number of UDP downlink-async-hold for Cisco Ultra Traffic Optimization.
udp-dnlink-async-forward	Indicates the number of UDP downlink-async-forward for Cisco Ultra Traffic Optimization.
udp-dnlink-async-forward-and-hold	Indicates the number of UDP downlink-async-forward and hold for Cisco Ultra Traffic Optimization.
udp-dnlink-async-hold-failed	Indicates the number of UDP downlink-async-hold failed for Cisco Ultra Traffic Optimization.
udp-process-packet-drop	Indicates the number of UDP process-packet-drop for Cisco Ultra Traffic Optimization.
udp-process-packet-hold	Indicates the number of UDP process-packet-hold for Cisco Ultra Traffic Optimization.
udp-process-packet-forward	Indicates the number of UDP process-packet-forward for Cisco Ultra Traffic Optimization.
udp-process-packet-forward-failed	Indicates the number of UDP process-packet-forward failed for Cisco Ultra Traffic Optimization.
udp-process-packet-forward-and-hold	Indicates the number of UDP process-packet-forward and hold for Cisco Ultra Traffic Optimization.
udp-process-packet-forward-and-hold-failed	Indicates the number of UDP process-packet-forward and hold failed for Cisco Ultra Traffic Optimization.
udp-pkt-copy	Indicates the number of UDP packet-copy for Cisco Ultra Traffic Optimization.
udp-pkt-Copy-failed	Indicates the number of UDP packet-copy-failed for Cisco Ultra Traffic Optimization.
udp-process-pkt-copy	Indicates the number of UDP process-packet-copy for Cisco Ultra Traffic Optimization.
udp-process-pkt-copy-failed	Indicates the number of UDP process-packet-copy failed for Cisco Ultra Traffic Optimization.
udp-process-pkt-no-packet-found-action-forward	Indicates the number of UDP process packet, no packet found, action forward for Cisco Ultra Traffic Optimization.
udp-process-pkt-no-packet-found-forward-and-hold	Indicates the number of UDP process packet, no packet found, action forward and hold for Cisco Ultra Traffic Optimization.
udp-process-pkt-no-packet-found-action-drop	Indicates the number of UDP process packet, no packet found, action drop for Cisco Ultra Traffic Optimization.



<b>Bulk Statistics</b>	<b>Description</b>
udp-todrs-generated	Indicates the number of UDP TODRs generated for Cisco Ultra Traffic Optimization.

