



# 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 2](#)
- [Configuring Cisco Ultra Traffic Optimization, on page 6](#)
- [Multi-Policy Support for Traffic Optimization, on page 12](#)
- [Monitoring and Troubleshooting, on page 18](#)

## Feature Summary and Revision History

### Summary Data

Applicable Product(s) or Functional Area	<ul style="list-style-type: none"><li>• IPSPG</li><li>• P-GW</li></ul>
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>IPSPG Administration Guide</i></li></ul>

### Revision History

Revision Details	Release
With this release, Cisco Ultra Traffic Optimization is qualified on IPSPG.	21.8

# 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

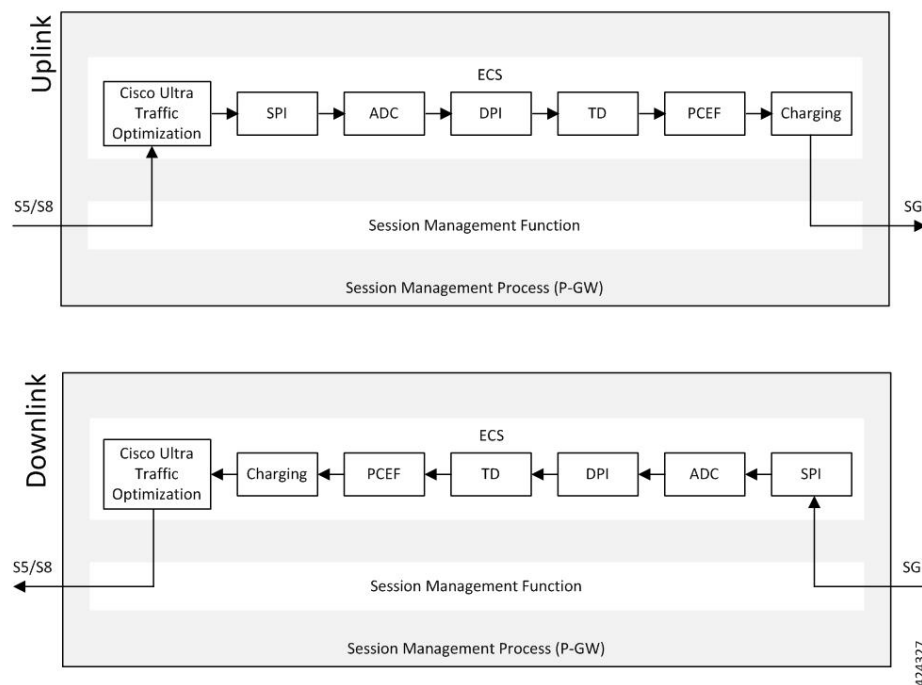
The Cisco Ultra Traffic Optimization achieves its gains by shaping video traffic during times of high network load/congestion. It monitors and profiles each individual video flow that passes through the gateway and uses its machine learning algorithms to determine whether that flow is traversing a congested 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, 1848, 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.

The IPSG supports only certain triggers for which the information is available with the IPSG service.

## 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.

## Configuring Threshold Value

Use the following CLI commands to configure the threshold value for the TCP flow to be considered for the traffic optimization:

```
configure
  active-charging service service_name
  traffic-optimization-profile
  heavy-session detection-threshold bytes
end
```

**Notes:**

- **detection-threshold *bytes*:** Specifies the Detection Threshold (in bytes), beyond which it is considered as heavy session.

*bytes* must be an integer from 1 to 4294967295.

For optimum traffic optimization benefits, it is recommended to set the threshold above 3 MB.

## 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.

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.

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

Traffic Optimization Levels	Subscriber Class configuration and Triggers
Applicable to a particular flow	<pre>trigger-condition TC1   qci = 1   rule-name = tcp   multi-line-or all-lines   exit</pre> <p>Flow creation trigger condition is TC1</p>

**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.

## Configuring Rulebase to Allow UDP Traffic Optimization

**Important**

From Release 21.8 onwards, it is recommended to enable TCP and UDP protocol for Traffic Optimization by using the CLI commands mentioned in the *Enabling TCP and UDP* section of this chapter.

Use the following configuration in ACS Rulebase Configuration Mode to turn ON/OFF the traffic optimization for UDP traffic.

**Important**

Enabling/Disabling the Cisco Ultra Traffic Optimization solution is controlled by Service-scheme Framework.

```
configure
  active-charging service service_name
  rulebase rulebase_name
  [ no ] traffic-optimization udp
  end
```

**NOTES:**

- **udp**: Specifies traffic optimization for UDP traffic.
- By default, UDP traffic optimization is disabled.
- If previously configured, use the **no traffic-optimization udp** CLI command to disable traffic optimization for UDP traffic.

## 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. With the Multi-Policy Support feature, traffic optimization in a service-scheme framework allows the operator to configure multiple policies and to configure traffic optimization based on a desirable location.

The service-scheme 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.

## 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 rulebase name <rulebase\_name>

The output of this show command has been enhanced to display if the UDP traffic optimization is Enabled or Disabled. Following are the fields that has been introduced:

- Traffic Optimization:
  - UDP: Enabled/Disabled

### 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

show active-charging traffic-optimization policy