

Event Data Records in CUPS

- Revision History, on page 1
- Feature Description, on page 1
- How It Works, on page 2
- Configuring Event Data Records in CUPS, on page 5
- Monitoring and Troubleshooting, on page 6

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
Support is added to generate interim EDRs.	21.23.6
First introduced	Pre 21.24

Feature Description

Generation of Event Data Records (EDR) is supported in the CUPS architecture.

These EDRs are generated on termination of a flow. Detailed information for every flow is generated after the flow termination.

Following are the EDR fields that gets populated in the event of EDR generation due to the flow end, transaction complete, and so on or whenever the necessary conditions are met.

- P2P Duration
- Rating group
- RADIUS NAS Identifier
- 3GPP Charging-id

- SN-Parent Protocol-id

When the data traffic with TCP starts for a subscriber attached to LTE network. Need to calculate and record time difference between control packets of TCP flow in EDR. Need to record the difference between following packets:

- SYN and SYN-ACK packet
- SYN-ACK and ACK packet

TCP Fast Open

TCP Fast Open (TFO) is an extension to speed up the opening of successive TCP connections between two endpoints. It works by using a TFO cookie (a TCP option), which is a cryptographic cookie stored on the client and set upon the initial connection with the server. When the client later reconnects, it sends the initial SYN packet along with the TFO cookie data to authenticate itself. If successful, the server may start sending data to the client even before the reception of the final ACK packet of the three-way handshake. Due to this RTT between SYN-ACK and ACK is calculated based on difference between SYN-ACK packet and first uplink ACK packet.

How It Works

EDRs are generated from UP on flow termination. During call setup and call modification, all call-specific attributes required for EDR generation is sent from CP to UP as part of the Subscriber Params IE within the Sx Establishment/Modification request messages.

On flow termination, the charging counters are fetched from VPP. All configured call-level attributes in the EDR format configuration along with the charging / volume counter attributes is sent to the CDRMOD proclet. This proclet writes these records to a file/disk, which is transferred to a configured external server.



Note User-Location-Information is written in hexadecimal format.

Transaction Complete EDR

Transaction complete EDRs are generated for HTTP EDRs when a HTTP transaction is complete. On completion, the charging counter are fetched from VPP. All configured call-level attributes in the EDR format configuration along with the charging / volume counter attributes is sent to the CDRMOD proclet. This proclet writes these records to a file/disk, which is transferred to a configured external server.

The following list of EDR attributes are supported:

- attribute sn-start-time
- attribute sn-end-time
- attribute sn-start-time format MM/DD/YYYY-HH:MM:SS:sss
- attribute sn-end-time format MM/DD/YYYY-HH:MM:SS:sss
- attribute radius-calling-station-id

- attribute radius-called-station-id
- rule-variable bearer 3gpp imsi
- rule-variable bearer 3gpp imei
- rule-variable bearer 3gpp rat-type
- rule-variable bearer 3gpp user-location-information
- rule-variable ip subscriber-ip-address
- rule-variable ip dst-address
- attribute sn-ruledef-name
- attribute sn-subscriber-port
- attribute sn-server-port
- attribute sn-app-protocol
- attribute sn-volume-amt ip bytes uplink
- attribute sn-volume-amt ip bytes downlink
- attribute sn-flow-start-time format seconds
- attribute sn-flow-end-time format seconds
- attribute sn-volume-amt ip pkts uplink
- attribute sn-volume-amt ip pkts downlink
- attribute sn-direction
- rule-variable traffic-type
- rule-variable p2p protocol
- rule-variable p2p app-identifier tls-cname
- rule-variable p2p app-identifier tls-sni
- rule-variable p2p app-identifier quic-sni
- rule-variable bearer 3gpp sgsn-address
- attribute sn-rulebase
- attribute sn-charging-action
- rule-variable flow tethered-ip-ttl
- rule-variable flow ttl
- rule-variable flow ip-control-param
- rule-variable bearer qci
- rule-variable tcp flag
- rule-variable ip server-ip-address

- attribute sn-flow-id
- attribute sn-closure-reason
- attribute sn-duration
- rule-variable ip src-address
- rule-variable ip protocol
- attribute sn-charge-volume ip bytes uplink
- attribute sn-charge-volume ip bytes downlink
- tcp-state
- tcp-prev-state

The following HTTP EDR attributes are supported:

- rule-variable http url length 2000
- rule-variable http request method
- rule-variable http content type
- rule-variable http user-agent length 255
- rule-variable http reply code
- rule-variable http referer
- rule-variable http host
- rule-variable http cookie
- rule-variable http header-length
- attribute transaction-uplink-bytes
- attribute transaction-downlink-bytes

Support for Interim EDRs

ECS supports generation of Interim EDRs – EDRs that are generated for ongoing flows based on a configurable timer.

Usually, EDRs are generated for flows only when the flow terminates or when the flow reaches the configured flow idle-timeout value. These flows could have time duration that is as long as 48 hours, which makes it difficult to track subscriber activity until an EDR is generated.

Thus, with interim EDRs, ongoing flow activities are tracked by configuring an interim timeout value for a flow. On expiration of the interim timer, an EDR is generated.

For configuring an interim EDR, a new CLI keyword, **interim**, is introduced. Based on the configuration, the interim timer is applied to newly created flows. On expiration of the timer, an interim EDR is generated along with the following reason: **sn-closure-reason (23)**. The information volume available until the expiration of the timer is populated in the EDR along with its respective timestamps.

Limitations

The Event Data Record feature in CUPS has the following limitations:

- EDR will be generated only for flow end condition – idle timeout, hagr, normal flow termination & during end of session.
- Charging-Action based EDR configuration is not supported.
- Reporting EDRs are not supported.

Configuring Event Data Records in CUPS

Configuration on CP to Push EDRs to UP

Use the following configuration to push EDRs from CP to UP using PFD mechanism.



Note The CLI commands used in this configuration are part of the existing non-CUPS architecture.

```
active-charging service service_name
   rulebase rulebase_name
      flow end-condition { timeout | normal-end-signaling | session-end |
 interim  } charging-edr charging_edr_format_name
      edr transaction-complete http charging-edr charging_edr_format_name
      exit
      edr-format format_name
         attribute attribute_name
         end
```

NOTES:

- **flow end-condition**: This command allows you to configure the end condition of the session flows related to a user session and triggers EDR generation.
- **timeout**: Creates an EDR with the specified EDR format whenever a flow ends due to a timeout condition.
- **normal-end-signaling**: Creates an EDR with the specified EDR format whenever flow end is signaled normally.
- **session-end**: Creates an EDR with the specified EDR format whenever a subscriber session ends. By this option session manager creates an EDR with the specified format name for every flow that has had any activity since last EDR was created for the flow on session end.
- **charging-edr** *charging_edr_format_name*: Specifies the charging EDR format.
- **interim**: This condition specifies the interim threshold condition of the flow where an EDR is generated based on the configured timer value. The *interim_timer_value* is configured in minutes with a configurable range from 15 to 1440 minutes.
- The **interim** keyword is only applicable for new flows created and not on existing flows.

- **http**: Specifies HTTP protocol related configuration.

Configuration to Enable EDR Module on UP

Use the following configuration to enable EDR module on UP

Note The CLI commands used in this configuration are part of the existing non-CUPS architecture.

```
configure
   context context_name
      edr-module active-charging-service
      end
```

Configuring Additional TCP Fields

Prior to using the following CLI commands to configure additional TCP fields in the EDR, ensure that all the other EDR configurations are present.

Note For CUPS setup, once configuration is present on CP side, push those changes on UP using **push config-to-up all** command from CP.

```
configure
   active-charging service service_name
      edr-format edr_format_name
         [ no ] rule-variable tcp syn_synack_rtt priority 3
         [ no ] rule-variable tcp syn_synack_ack_rtt priority 4
         end
```

Monitoring and Troubleshooting

show user-plane-service statistics rulebase name *rulebase_name*

The following fields are displayed in support of this feature:

- Rulebase Name
  - EDRs
  - Charge Volume
    - Uplink Pkts
    - Uplink Bytes
    - Downlink Pkts

- Downlink Bytes

- Charging EDRs
  - Total Charging EDRs generated
  - EDRs generated for handoff
  - EDRs generated for timeout
  - EDRs generated for normal-end-signaling
  - EDRs generated for session end
  - EDRs generated for rule match
  - EDRs generated for hagr
  - EDRs generated for flow-end content-filtering
  - EDRs generated for flow-end url-blacklisting
  - EDRs generated for content-filtering
  - EDRs generated for url-blacklisting
  - EDRs generated for any-error packets
  - EDRs generated for firewall deny rule match
  - EDRs generated for transaction completion
  - EDRs generated for voip call end
  - EDRs generated for dcca failure handling
  - EDRs generated for TCP optimization on
  - EDRs generated for tethering signature change
  - EDRs generated for interim interval
  - Total Flow-Overflow EDRs
  - Total zero-byte EDRs suppressed

- EDRs generated for interim
  - Interval

- Total Rulebases

show active-charging rulebase statistics real-time

The following fields are displayed in support of this feature:

- Rulebase Name
- Charging EDRs

- Total Charging EDRs generated
  - EDRs generated for handoff
  - EDRs generated for timeout
  - EDRs generated for normal-end-signaling
  - EDRs generated for session end
  - EDRs generated for rule match
  - EDRs generated for hagr
  - EDRs generated for flow-end content-filtering
  - EDRs generated for flow-end url-blacklisting
  - EDRs generated for content-filtering
  - EDRs generated for url-blacklisting
  - EDRs generated for any-error packets
  - EDRs generated for firewall deny rule match
  - EDRs generated for transaction completion
  - EDRs generated for voip call end
  - EDRs generated for dcca failure handling
  - EDRs generated for TCP optimization on
  - EDRs generated for tethering signature change
  - EDRs generated for interim interval
  - EDRs generated for audio-end Sessions
  - EDRs generated for video-end Sessions
  - EDRs generated for voipout-end Sessions
  - Total Flow-Overflow EDRs
  - Total zero-byte EDRs suppressed

show active-charging edr-format all

The following fields are displayed in support of Additional TCP Fields in EDR feature:

- Service Name
  - EDR Format Name
    - rule-variable tcp syn-synack-rtt priority 3
    - rule-variable tcp synack-ack-rtt priority 4

Bulks Statistics

The following bulk statistic(s) are added in the ECS schema to support Event Data Records in CUPS:

- edrs-generated: Indicated the total number of EDRs generated.