



GTPP Endpoint

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 1](#)
- [How it Works, on page 5](#)
- [Configuring the GTPP Profile, on page 6](#)
- [Configuring the GTPP Endpoint, on page 9](#)

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Products or Functional Area	SMF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2023.03.0

Feature Description

The GTPP Endpoint is an App-infra-based service that enables the GPRS Tunneling Protocol Prime (GTPP) protocol functionality for the SMF service. The SMF with Legacy Interfaces supports the GTPP Charging (Gz) interface in the GTPP Endpoint.

For offline charging, the Gz is the reference point from a Charging Data Function (CDF) to the CGF for transporting of CDRs.

The GTPP Endpoint provides the following support to the SMF service:

- Handles the usage and event accounting at a Bearer level and the accounting information gets stored as part of the subscriber session.
- Generates CDR content from this accounting information and then transfers to the GTPP Endpoint. The GTPP Endpoint is responsible for encoding the received CDRs into ASN.1 (based on the configured dictionary).
- Sends the received CDRs to the CGF server using the GTPP protocol.

For more information on this feature, see the [UCC 5G SMF Configuration and Administration Guide > GTPP Endpoint](#) chapter.

For more information on this feature, see the [GTPP Endpoint](#) chapter.

Charging Related Transfers

Each Charging Data Function (CDF) has an O&M-configurable address list of Charging Gateway Function (CGFs) to which it can send its CDRs. The list is organized in CGF address priority order. If the primary CGF is not available, then the CDF sends the CDRs to the secondary CGF and so on.

Each CDR generating function only sends records to the CGF(s) of the same PLMN, not to CGF(s) located in other PLMN.

Each CGF in the PLMN must know the other CGFs' network addresses to recommend another CGF address in case of redundancy. This is achieved by O&M configuration facilities that enable each CGF to have a configurable list of peers' CGF addresses.

CDR Transport through GTPP

GTPP delivers Charging Data Records (CDR) from the Charging Data Function (CDF), which generates CDRs to the Charging Gateway Function CGF(s). The GTPP protocol is required if the CGF resides outside the CDFs. It utilizes some aspects of GTP, which is used for packet data tunnelling in the backbone network.

GTPP operates on the Gz interface and performs the following functions:

- Transfers CDR between the CDF and the CGF.
- Redirects CDRs to another CGF.
- Detects communication failures between the communicating peers, using echo messaging.
- Advertises to peers about its CDR transfer capability. For example, after a period of service downtime.
- Prevents duplicate CDRs that might arise during redundancy operations. If configured, the CDR duplication prevention function is carried out by marking potentially duplicated CDR packets, and delegating the final duplicate deletion task to a CGF, or the Billing Domain.

Path Protocol

GTPP uses path protocol to transport CDRs from CDF to CGF over the Gz interface to facilitate charging. The UDP path protocols is supported for GTPP:



Note Path Protocol TCP is not supported

- Ports for signalling the request messages:
 - The UDP Destination Port with the server port number 3386 is reserved for GTPP. Alternatively, another port can be used as configured by O&M.
 - The UDP Source Port is a locally allocated port number for sending network element.
- Ports for signalling the response messages:
 - The UDP Destination Port is the value of the Source Port of the corresponding request message.
 - The UDP Source Port is the value from the Destination Port of the corresponding request

GTPP Message Types

GTPP defines a set of messages between two associated nodes. The following table lists GTPP messages.

Table 3: GTPP Messages

Message Type Value (Decimal)	GTPP Message
1	Echo Request
2	Echo Response
3	Version Not Supported.
4	Node Alive Request
5	Node Alive Response
6	Redirection Request
7	Redirection Response
240	Data Record Transfer Request
241	Data Record Transfer Response

Endpoint GTP Prime

The Endpoint GTP Prime is primarily used to set the storage size limit (default 1 GB), which is applicable when **k8s use-volume-claim Pod** is set to true. Though gtp-profiles configuration alone brings up gtp-ep

pod, it is recommended to configure gtpprime endpoint as well, with basic configuration such as **storage**, or **replica**. This configuration is used by the gtp-ep pod.

Ensure to configure GTPP profiles charging agent ip address and ports in the Endpoint under the Gz interface.



Note The **gtp-ep** always ignores **nodes** config. When **k8s single-node** is set to false, it spawns 2 replicas of gtp-ep in the Active or Standby mode independent of replicas and nodes configuration.

GTPP Resiliency

The GTPP endpoint provides resiliency by supporting CDR archiving and file storage.

CDR Archiving

In an overload of CDRs scenario, the CGF server becomes slow in providing a response. This scenario leads to unacknowledged CDRs due to sendReqList buffer being full and then the CDRs drop. To avoid the CDR drop in such a case, the GTPP endpoint enqueues new CDRs into the archive list.

GTPP performs the CDR archiving in the following way:

- The archive list flush happens in a paced manner depending on the network load condition.
- After all the CGFs become inactive or due to back-to-back switchovers, an archive record is written into HDD in the paced manner to avoid losing the archived CDRs.
- Archive list is replicated on standby GTPP endpoint pod.



Note

- The archive list is maintained per GTPP profile level. The maximum size of the archive list is 5M.
- After reaching the maximum configured size, the oldest record is purged and a new record is added.
- Archive list gets replicated on the standby pod. When the active pod is inactive, the standby pod becomes the active pod and starts processing the CDRs to retain all the archive records.

CDR File Storage

GTPP performs the CDR file storage in the following way:

1. When all the CGFs are inactive, all the CDRs are written into a file in HDD on the active GTPP endpoint.
2. The CDRs are stored on the standby GTPP endpoint.
3. After the inactive CGF becomes active, the CDRs are read from the file on the active GTPP endpoint and sent to the CGF.
4. After all the CDRs are read from one file, the file gets deleted from the active GTPP endpoint. Then, the file delete replication information is sent to the standby GTPP endpoint.
5. Standby GTPP endpoint deletes the file from HDD.



Note During the CDR reading from HDD process, when some CDRs are sent to CGF and some CDRs are still in the file and the active GTPP endpoint becomes inactive, then the last read file is sent from the new active GTPP endpoint pod. The CGF receives various duplicate CDRs in such a case.

How it Works

This section describes how this feature works.

Call Flows

Information Elements Support

Following are the Information Elements (IEs) supported for the Gz support with GTPP feature.

Table 4: TLV and TV IEs

Information Element	Description
TLV Information Elements	
254	Address of Recommended Node
253	Requests Responded
252	Data Record Packet
251	Charging Gateway Address (this IE is also used in TS 29.060 [200])
250	Sequence numbers of Cancelled Packets
249	Sequence numbers of Released Packets
TV Information Elements	
127	Charging ID
126	Packet Transfer Command

Standards Compliance

The GTPP endpoint support complies with the following Charging specific 3GPP standards:

- *3GPP TS 32.251 "Telecommunication management; Charging management; Packet Switched (PS) domain charging"*
- *3GPP TS 32.295 "Telecommunication management; Charging management; Charging Data Record (CDR) transfer"*

- 3GPP TS 32.297 "Telecommunication management; Charging management; Charging Data Record (CDR) file format and transfer"
- 3GPP TS 32.298 "Telecommunication management; Charging management; Charging Data Record (CDR) parameter description"

Limitations

The following are the limitations in the SMF with Diameter Interfaces Charging feature:

- The SMF with Diameter Interface does not support Local storage mode, TCP, and Redirect Request or Responses.
- The SMF with Diameter Interfaces supports only the custom24 CDR Dictionary.
- In case of switchover, only the new CDR Send Request buffer is checked on the new standby GTPP pod.
- In case of switchover, only the new archive is checked on the new stand by GTPP pod.
- In case of double fault, old archive CDR and old Send Request buffer are lost.

Configuring the GTPP Profile

To configure the GTPP profile, use the following configuration:

```

config
  profile gtp profile_name gtp
  dictionary
  ignore ignore_value
  instance-id
  charging-agent address IPv4_address port UDP_port
  server{ cgf address IPv4_address max-cdrs max_cdrs { node-alive Enable
| Disable}
  port UDP_port priority priority deadtime time_interval
  echo-interval echo_interval timeout timeout_val max-retry
max_retry max-pdu-size max_pdu_size wait-time time_interval }
  local-storage
  mode
  local
  streaming-parallel
  cgf-server-redundancy-support
  exit
exit
exit

```

NOTES:

- **dictionary**: Specify a dictionary for ASN.1 based encoding of a CDR.
- **ignore** *ignore_value*: Specify the configuration to ignore the echo-rc-change. This CLI control option provides a flexibility to detect a CGF path failure due to a change in the echo response RC.
- **instance-id**: Specify the instance ID of a GR instance.

- **charging-agent**: Configure the charging agent.
 - **addressIPv4_address**: Specify the IP address of the interface configured within the endpoint that is used to transmit CDR records to the CGF.
 - **port**: Specify the UDP port.



Note The Charging agent IP address and port configured in GTPP profiles should also be configured in the endpoint gtpprime under the Gz interface. The Runtime configuration update of the Charging agent IP address and port is not recommended. Ensure to add new profile with new Charging agent IP address and port.

- **server**: Configures server details.
 - **cgf**: Configure the CGF server with the following parameters:
 - **address IPv4_address**: Enter the IPv4 address of CGF server, using dotted-decimal notation range.
 - **max**: Configure maximum number of unacknowledged CDRs for a CGF. Must be an integer ranging from 1 to 2000.



Note The runtime configuration change of **max** is not recommended. Follow the Method of procedure:

1. Delete the **cgf** having old **max** and then commit the change.
2. Add the **cgf** again with a new **max** value.

- **node-alive Enable | Disable** : Enable or disable sending Node Alive Request to a GTPP Server (such as CGF).
- **port**: Specify which port that the CGF is using.
- **priority**: Specify the relative priority of this server when system is selecting which CGF server to use.
- **deadtime**: Configure the deadtime in seconds. Must be an integer ranging from 1 to 65535. Default value is 120.
- **max-cdrs**: Designate the maximum number of CDRs in a GTPP message. Must be an integer ranging from 1 to 255.
- **max-pdu-size**: Designate the maximum size of the PDU, in bytes. Must be an ranging from 1024 to 1460.
- **timeout**: Specify the number of times the system attempts to communicate with a CGF that is not responding.
- **wait-time**: Specify the time to wait before sending the GTPP request.

- **local-storage**: Specify local storage details.
- **mode**: Specify a storage mode to be used.
 - **local**: Specify the use of HDD to store CDRs
 - **streaming-parallel**: Specify the use of HDD to store CDRs, if CGF fails. When CGF comes up, stream the CDRs to the CGF. Streaming is in a parallel and newly generated CDRs are sent to CGF along with CDRs streamed from HDD.
- **cgf-server-redundancy-support**: Enable or disable the CGF server redundancy support per GTPP profile. By default this configuration is disabled.

Configuration Example

The following is an example configuration for GTPP profile.

```
profile gtp-profile pf2 gtp
dictionary custom24
mode      streaming
cgf-server-redundancy-support disable
instance-id 1
charging-agent
  address 10.10.10.205
  port    3386
exit
server
  max-pdu-size 1460
  timeout      30
  max-retry    3
  max-cdrs     5
  wait-time    30
  echo-interval 60
  deadtime     120
  cgf address 10.10.10.80 port 3386 max 100 priority 1 node-alive disable
  cgf address 10.10.10.90 port 3386 max 100 priority 2 node-alive disable
exit
exit
exit
```

Verifying GTPP Profiles

Use the **show running-config profile gtp-profile pf2 gtp** command to verify the GTPP Profile configuration.

The following is an example output of the **show running-config profile gtp-profile pf2 gtp** command.

```
profile gtp-profile pf2 gtp
dictionary custom24
mode      streaming
cgf-server-redundancy-support disable
instance-id 1
charging-agent
  address 10.10.10.205
  port    3386
exit
server
  max-pdu-size 1460
  timeout      30
  max-retry    3
```



```

max-cdrs      5
wait-time    30
echo-interval 60
deadtime     120
cgf address 10.10.10.80 port 3386 max 100 priority 1 node-alive disable
cgf address 10.10.10.90 port 3386 max 100 priority 2 node-alive disable
exit
exit
exit

```

Configuring the GTPP Endpoint



Note

- GTPP-EP pod uses this configuration.
- GTPP-EP pod always ignores nodes configuration.
- When the **k8s single-node** is set to **false**, it spawns two replicas of a GTPP-EP pod in active or standby mode, which is independent of replicas and nodes configuration.
- When the **k8s single-node** is set to **true**, the configured replicas have its impact.
- When the **k8s use-volume-claim** is set to **true**, endpoint GTP prime is used to set the storage size limit. Default value of storage size limit is one GB.
- When the system is up and running, we can't change the storage size.

To configure a GTPP endpoint, use the following commands:

config

```

instance instance-id instance_id
  endpoint gtpprime
    storage storage_capacity
    replicas replicas_count
    nodes nodes_count
  interface Gz
    vip-ip vip_ip vip-port vip-port vip-interface vip-interface vrf vrf
  end

```

NOTES:

- **endpoint gtpprime** : Specify the following parameters to configure an endpoint:
 - **storage storage_capacity**—Specify the storage size of persistent volume in GB. Must be an integer in the range of 1-100.



Note

CLI doesn't allow changing storage size while the system is running. To change the storage size, bring the system down first.

- **replicas replicas_count**—Specify the number of replicas per node. Must be an integer.
- **nodes nodes_count**—This property is ignored. You may skip configuring it.

- **interface Gz** : Configure the Gz interface details, such as vip IPv4 address, vip port, vip interface, and virtual routing and forwarding details.



Note When the system is active and under the Gz Interface, if there are any add or update **vip** configurations to use new values, ensure to restart the **udp-proxy**.

Configuration Example

The following is an example configuration for the endpoint gtpprime configuration:

```
instance instance-id 1
endpoint gtpprime
  replicas 1
  nodes 1
  storage 1
  interface gz
    vip-ip 10.10.10.205 vip-port 7202 vip-interface v303 vrf GTPP-VRF
  exit
exit
exit
```

Verifying GTPP Endpoints

Use the **show gtp-ep endpoints** command to verify the GTPP Endpoints.

The following is an example output of the show gtp-ep endpoints command.

```
result List of gtp pods with their names and IPs
gtp-ep-0 192.168.42.116
gtp-ep-1 192.168.42.126
```

Verifying GTP Prime Configuration

Use the **show running-config instance instance-id 1 endpoint gtpprime** command to verify the GTP Prime configuration.

The following is an example output of the show running-config instance instance-id 1 endpoint gtpprime command.

```
instance instance-id 1
endpoint gtpprime
  replicas 1
  nodes 1
  storage 1
  interface gz
    vip-ip 10.10.10.205 vip-port 7202 vip-interface v303 vrf GTPP-VRF
  exit
exit
exit
```