Performance Optimization Support

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Products or Functional Area	SMF
Applicable Platforms	SMI
Feature Default Setting	
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2022.04.0

Feature Description

This chapter describes the performance optimuzation features.

Some of the performance optimuzation features are common across cnSGWc and SMF. For complete information on the features, see the *UCC cnSGW Configuration and Administration Guide*.

Batch ID Allocation, Release, and Reconciliation Support

Feature Description

The nodemgr allocates a unique ID to the subscriber that is in the attached state. When the subscriber detaches, the unique ID is released to the nodemgr. If the allocation and deallocation procedures increase, the nodemgr performance is impacted and the sgw-service continues to wait longer to complete these procedures.

The Batch ID Allocation, Release, and Reconciliation Support feature provide a mechanism to reduce the interaction between the sgw-service and nodemgr, which in turn optimizes the nodemgr's performance.

How it Works

This section describes how this feature works.

Batch ID Allocation:

Allocation of batch ID involves the following steps:

- The sgw-service manages the ID store by allocating the free IDs to the subscriber. When the IDs are unavailable in the store, the sgw-service sends the Batch ID Allocation Request to nodemgr.
- In response, nodemgr returns a batch of 128 IDs with the ID Reserve Report Interval. The sgw-service updates the ID store with the IDs received from the nodemgr and starts a timer for the ID Reserve Report Interval.
- If all the IDs are used before the duration configured in the ID Reserve Report Interval, sgw-service sends a Batch of ID Allocation Request to the nodemgr with a notification to reserve all IDs from the previous request.
- If the ID Reserve Report Interval timer expires before the sgw-service allocates all the IDs, sgw-service sends the unused IDs back to nodemgr through the Reserve Report Batch operation.

Batch ID Release:

Releasing of the batch ID involves the following steps:

- The sgw-service manages the IDs that the ID store releases for each nodemgr.
- The sgw-service returns the ID to the ID store whenever an ID is deallocated. If the ID store is full, the sgw-service sends a Batch ID Release Request and the released IDs to the respective nodemgr.
- When sgw-service starts adding IDs to the ID store, the ID release timer starts.

- If the ID release timer expires before the batch IDs are releases or the batch is full, sgw-service sends the released IDs to nodemgr.

Batch ID Reconciliation

Batch ID reconciliation occurs when the service pod and the nodemgr pod restarts.

On service pod restart:

1. When the service pod receives the batch IDs and becomes unresponsive before allocating the IDs, the nodemgr does not get the Batch ID Reserve Request causing the ID reserve procedure to time out. In such a scenario, the nodemgr reconciles the unreserved or unallocated IDs with CDL. The IDs that are not allocated to the subscribers are released to the ID store.
2. The service pod collects the IDs that are released and if it becomes unresponsive before releasing them to the nodemgr. In this scenario, the IDs are dropped.

On nodemgr pod restart:

1. The IDs existing in the in-flight Batch ID Reserve Request and Batch ID Release Request are dropped.
2. The nodemgr notifies cachemgr about the allocated IDs in a batch. If nodemgr becomes unresponsive before notifying the IDs to cachemgr, after a restart, nodemgr starts allocating the new IDs. The nodemgr allocated the IDs based on the last allocated ID and the batch size.

Feature Configuration

To configure this feature, use the following configuration:

```
config
   sgw sgw_name
      resmgr-batch-operation [ disable | enable ]
      end
```

NOTES:

resmgr-batch-operation [disable | enable]—Configures the batch operation. By default, **resmgr-batch-operation** is disabled.

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Bulk Statistics

The following statistics are supported for the Batch ID Allocation and Release Support feature:

- sgw_resource_mgmt_stats—Captures the total number of the cnSGW-C resource management statistics.

  Sample queries:

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",
id_req_type="id_batch_alloc",instance_id="0",service_name="sgw-service",status="attempted"}
 3
:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",
```

```
id_req_type="id_batch_alloc",instance_id="0",service_name="sgw-service",status="success"}
 3
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1"
,id_req_type="id_batch_dealloc",instance_id="0",service_name="sgw-service",status="attempted"}
 2
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",id_req_type=
"id_batch_dealloc",instance_id="0",service_name="sgw-service",status="success"}
 2
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",
id_req_type="id_batch_dealloc_timeout",instance_id="0",service_name="sgw-service",status="attempted"}
 1
:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",id_req_type
="id_batch_dealloc_timeout",instance_id="0",service_name="sgw-service",status="success"}
 1
```

```
sgw_resource_mgmt_stats{app_name="smf",cluster="Local",data_center="DC",gr_instance_id="1",
id_req_type="id_batch_release_timeout",instance_id="0",service_name="sgw-service",status="attempted"}
 1
-:sgw_resource_mgmt_stats{app_name="smf",cluster="Local",d-ata_center="DC",gr_instance_id="1",id_req_type
="id_batch_release_timeout",instance_id="0",service_name="sgw-service",status="success"}
 1
```

- nodemgr_rmgr_batch_reconcile_stats—Captures the total count of batches that are sent for reconciliation.

  Sample queries:

```
nodemgr_rmgr_batch_reconcile_stats{app_name="smf",cluster="Local",data_center="DC",instance_id="0",
service_name="nodemgr",status="success"} 1
```

- nodemgr_resource_mgmt_resp_stats—Captures the total number of IDs released due to reconciliation.

  Sample queries:

```
nodemgr_resource_mgmt_resp_stats{app_name="smf",cluster="Local",data_center="DC",error="",
gr_instance_id="0",instance_id="0",ip_ver_type="IP_TYPE_NONE",req_type="ID_REQ_REL_RECONCILE",
service_name="nodemgr",status="success"} 16
```

For more information on bulk statistics support, see *UCC Serving Gateway Control Plane Function Metrics Reference*.

CDL Flush Interval and Session Expiration Tuning Configuration

Feature Description

You can modify the default service-pod parameters to fine-tune the throughput performance and optimize the load performance.

Feature Configuration

To configure this feature, use the following configuration:

```
config
   profile sgw sgw_name
      timers [ session-expiration-in-secs session_expiration |
affinity-expiration-in-secs affinity_expiration | session-dbsync-interval-in-ms
 database_sync ]
      end
```

NOTES:

- **session-expiration-in-secs** *session_expiration* —Specify the duration for which the session is cached on service pod. *session_expiration* accepts value in the range of 1-600 milliseconds. The default value is 30 milliseconds.
- **affinity-expiration-in-secs** *affinity_expiration* —Specify the duration for which the session affinity keys are valid on the service pod and other pods. *affinity_expiration* accepts value in the range of 1-1200 seconds. The default value is 80 seconds.
- **session-dbsync-interval-in-ms** *database_sync* —Specify the duration after which the session is synchronized in the database. *database_sync* accepts value in the range of 1-10000 milliseconds. The default value is 500 milliseconds.

Domain-based User Authorization Using Ops Center

Feature Description

SMF and cnSGW-C support domain-based user authorization using the Ops Center. To control the access on a per-user basis, use the TACACS protocol in Ops Center AAA. This protocol provides centralized validation of users who attempt to gain access to a router or NAS.

Configure the NETCONF Access Control (NACM) rules in the rule list. Then, map these rules in the Ops center configuration to map the group to appropriate operational authorization. Use the configurations that are based on the following criteria and products:

- With the NACM rules and SMF domain-based group, configure the Ops center to allow only access or update SMF-based configuration.
- With the NACM rules and cSGW-C domain-based group, configure the Ops center to allow only access or update cSGW-C-based configuration.
- With the NACM rules and cSGW-C domain-based group, configure the Ops center to allow only access or update CCG-based configuration.

Note The NSO service account can access the entire configuration.

How it Works

To support this feature configuration in Ops Center, the domain-based-services configuration is added in the TACACS security configuration. The TACACS flow change works in the following way:

- If you have configured the **domain-based-services** parameter, then the configured user name that is sent to the TACACS process, splits user ID into user ID and domain. The split character, which is a domain delimiter, is configured in domain-based-services. These split characters can be "@", "/", or "\" and are used in the following format to get the domain and user ID information.
  - @ — <user id>@<domain>
  - / — <domain>/<user id>
  - \ — <domain>\<user id>
- The TACACS authenticates and authorizes as per the existing flow. However, if the domain-based-services feature is enabled and TACACS authenticates and authorizes the user, following steps are added to the TACACS flow procedure.
  - If Network Services Orchestrator (NSO) logs in as the NSO service account, then that session receives a specific NACM group that you configured in **domain-based-services nso-service-account group** *group-name*. This functionally is the same as the way NSO works.
  - If the specified domain exists in the group mapping, then the NACM group that you configured in **domain-based-services domain-service** *domain* **group** *group-name* is applied.
  - If the user does not have a domain or the domain does not exist in the domain to group mapping, then **no-domain** NACM group that you configured in **domain-based-services no-domain group** *group-name* is applied. If the **no-domain** configuration does not exist, then the user value is rejected.

To enable this feature, you must configure the **domain-based-services** CLI command with the following options:

- NSO service account
- Domain service
- Domain delimiter
- No domain

Feature Configuration

To enable domain-based user authorization using Ops Center, use the following sample configuration:

```
config
   tacacs-security domain-based-services [ domain-delimiter delimiter_option
 | domain-service domain_service_name [ group service_group_name  ] | no-domain
group service_group_name | nso-service-account [ group service_group_name | id
service_account_id ] ]
   end
```

NOTES:

- **domain-based-services [domain-delimiter** *delimiter_option* **| domain-service** *domain_service_name* **[group** *service_group_name* **] | no-domain group** *service_group_name* **| nso-service-account [group** *service_group_name* **| id** *service_account_id* **]]**: Configure the required domain-based-services value. The **domain-based-services** includes the following options:
  - **domain-delimiter**: Specify the delimiter to use to determine domain. This option is mandatory and allows the following values:
    - @—If domain-delimiter is "@", the user value is in the format: <user>@<domain>.
    - /—If domain-delimiter is "/", the user value is in the format: <domain>/<user>.
    - \—If domain-delimiter is "\", the user value is in the format: <domain>\<user>.
  - **domain-service**: Specify the list of domains and their group mapping. The key is the name of the domain and group is the group that is assigned to the domain. You must configure at least one option in this list.
  - **no-domain**: Specify the group that has no domain or if the domain is unavailable in the domain-service mapping, then this group is sent in the accept response.
  - **nso-service-account**: Specify the NSO service account that has the ID and group. If you configure this parameter, then you must configure the ID and group fields. The ID and group must have string values.

Configuration Example

The following is an example of the domain-based user authorization in the tacacs-security mode:

```
config
   tacacs-security domain-based-services nso-service-account id nsid
      tacacs-security domain-based-services nso-service-account group nso-group
   tacacs-security domain-based-services no-domain group read-operational
   tacacs-security domain-based-services domain-delimiter @
   tacacs-security domain-based-services domain-service etcd
      group etcd
exit
tacacs-security domain-based-services domain-service sgw
   group sgw_1
exit
tacacs-security domain-based-services domain-service smf
   group smf
exit
```

Configuration Verification

To verify the configuration, use the following show command:

show running-config tacacs-security

The output of this show command displays all the configurations of the domain-based services within the TACACS security.

```
[smf] smf# show running-config tacacs-security
tacacs-security service smf
tacacs-security server 1
address 209.165.200.234
key $8$+twbdL2ZCgmjVswgp7kFJp8+SMXDjQRTZgoPVa3oEwY=
exit
```

```
tacacs-security domain-based-services nso-service-account id nsid
tacacs-security domain-based-services nso-service-account group nso-group
tacacs-security domain-based-services no-domain group read-operational
tacacs-security domain-based-services domain-delimiter @
tacacs-security domain-based-services domain-service etcd
group etcd
exit
tacacs-security domain-based-services domain-service sgw
group sgw_1
exit
tacacs-security domain-based-services domain-service smf
group smf
exit
```

Edge Echo Implementation

Feature Description

In a nonmerged mode, the udp-proxy pod acts as an endpoint, and the gtpc-ep responds to the Echo Requests from the peer node.

The gtpc-ep experiences traffic when the system receives a high number of inputs CEPS leading to a discrepancy between the rate at which gtpc-ep picks up the messages from udp-proxy and the rate at which udp-proxy gets the messages.

If the gtpc-ep is loaded, the queue between the udp-proxy and gtpc-ep gets full, and some of the messages at udp-proxy might get dropped. The peer detects path failure if these are Echo Request messages because an Echo Response is not received. Further, the peer clears all the sessions sent to the sgw-service.

How it Works

This section describes how this feature works.

Nodemgr processes the Echo Request in the following steps:

- The nodemgr preserves a self-restart counter cache for each GR instance ID and the GTPC peer.
- When the udp-proxy pod receives an Echo Request from a peer and the self-restart counter value is not available in the self-restart counter cache, the udp-proxy pod forwards the Echo Request to gtpc-ep.
- The gtpc-ep sends the self-restart counter as part of the UDP proxy message metadata in the Echo Response. The udp-proxy stores the self-restart counter in the self-restart counter cache. When the udp-proxy receives an Echo Request from a peer, and a self-restart counter value is available in the self-restart counter cache, the udp-proxy sends an Echo Response with the restart counter.
- The udp-proxy forwards the Echo Request message to the gtpc-ep. The gtpc-ep processes the Echo Request and forwards it to nodemgr, if necessary.
- If the peer restart counter value is modified, the nodemgr detects a path failure.
- In the Echo Response, the gtpc-ep sends the self-restart counter in the UDP Proxy Message metadata to the udp-proxy. If the self-restart counter differs from the counter that is stored in the self-restart counter cache, the udp-proxy updates the self-restart counter in the cache and drops the Echo Response received from the gtpc-ep.

Note The Edge Echo feature is not supported when the gtpc-ep is started in the merged mode.

Heartbeat

To handle the Echo Request and Echo Response messages for the GTPV2 interface, a heartbeat queue is implemented between the gtpc-ep and the udp-proxy pod. The heartbeat queue is responsible for handling the HeartBeat Request and HeartBeat Response Messages between the protocol and udp-proxy pod for the PFCP interface.

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Bulk Statistics Support

The following statistics are supported for the Edge Echo Implementation feature:

- Heartbeat queue status:

```
sum(irate(ipc_response_total{rpc_name~=".ipc_stream_hb."}[10s])) by
(service_name,
instance_id, status, status_code, rpc_name, dest_host)
```

- Check the EdgeEcho messages:

```
sum(irate(udp_proxy_msg_total{ message_name ="edge_echo"}[30s])) by
(message_name,
message_direction, status)
```

To enable the Heartbeat queue and EdgeEcho messages statistics, configure the trace-level statistics for udp_proxy_msg_total using the following:

```
infra metrics verbose application
   metrics udp_proxy_msg_total level trace
   exit
```

Note Enabling the heartbeat and EdgeEcho messages statistics may lead to a performance degradation on the udp-proxy pod.

ETCD Peer Optimization Support

Feature Description

When large numbers of GTPC peers are connected with SMF or cnSGW-C, the performance of ETCD is impacted. Each peer is a considered as a record in the ETCD, and the timestamp is updated every 30 seconds for each peer. This causes continuous updates on ETCD and generates huge traffic that impacts the overall system performance.

The ETCD Peer Optimization feature facilitates optimization in peer management and enables reduced performance impact on ETCD.

How it Works

This section describes how this feature works.

Instead of considering each peer as an ETCD record entry, several peers are grouped as a peer group based on the hash value of the IP address of each peer. This reduces the number of entries in ETCD. By default, a maximum of 200 peer groups can be created. For any changes related to a peer in a peer group:

- For a new peer, the peer group is persisted immediately in ETCD.
- For the change in timestamp for existing peers, the peer group is updated once every 3 seconds. This update:
  - Results in a cumulative group update for many peers that have undergone timestamp change within each peer group.
  - Reduces frequent updates to ETCD.

Flag DB Database Updates

Feature Description

cnSGW-C updates the CDL whenever the subscriber state changes from idle to active, and the ULI, UeTz, UCI, or the serving network is modified.

When the transaction requests driven to CDL increases, cnSGW-C incurs a higher CPU utilization. To prevent the needless CPU utilization, cnSGW-C updates only a subset of the CDL with the changed attributes.

Flag DB Database for the DDN Procedure

When the DDN procedure completes, sgw-service updates the CDL which impacts the CPU utilization. To optimize the CPU usage, the CDL is notified about the DDN only with the partial updates.

DDN Internal timer

cnSGW-C implements the DDN Retry Timer by applying the CDL's timer functionality. Every DDN transaction starts the DDN Retry Timer that requires the complete CDL instance to be updated, which results in an increase in the CPU usage of the CDL and sgw-service.

cnSGW-C is modified to have an integrated DDN Retry Timer that is configurable from sgw-profile. With this approach, the performance is improved because the cnSGW-C does not communicate with the CDL for starting the DDN Retry Timer as it is an internal timer. The DDN Retry Timer is started for a duration of 10 seconds.

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Bulk Statistics Support

The following statistics are supported for the Flag DB Database Updates feature:

- mbr_partial_cdl_update: Captures the total number of partial CDL update procedures invoked by the Modify Bearer Request.

  Sample query:

```
sgw_cdl_update_stats{app_name="smf",cdl_update_type="
mbr_partial_cdl_update",cluster="Local",data_center="DC",gr_instance_id="1",
instance_id="0",rat_type="EUTRAN",service_name="sgw-service"} 1
```

- ddn_partial_cdl_update: Captures the total number of partial CDL update procedures that DDN has invoked

  Sample query:

```
sgw_cdl_update_stats{app_name="smf",cdl_update_type="
ddn_partial_cdl_update",cluster="Local",data_center="DC",gr_instance_id="1",
instance_id="0",rat_type="EUTRAN",service_name="sgw-service"} 1
```

For more information on bulk statistics support, see *UCC Serving Gateway Control Plane Function Metrics Reference*.

GTPC IPC Cross-rack Support

Feature Description

This chapter describes the performance optimuzation features.

Some of the performance optimuzation features are common across cnSGWc and SMF. For complete information on the features, see the *UCC cnSGW Configuration and Administration Guide*.

How it Works

This section describes how this feature works:

- In `SMF-Ops-Center`, you can configure `GTPC-EP Geo` endpoints for each rack in IMS and data racks.
- In `SMF-Ops-Center` CLI, you can configure `GTPC-EP Geo` endpoints for each rack in IMS and DATA racks.

Note The optimization of GTPC messages can be applied to all four instances or a subset of instances of GTPC endpoints within these racks. A new element is added under the GTPC endpoints to configure a list of IP addresses where SMF and cnSGW-C service pods can route the GTPv2 messages over the IPC interface.

Upgrading and Enabling Inter-rack GTPC IPC

This section describes how to upgrade and enable the inter-rack GTPC IPC.

Before you configure, upgrade, and enable the GTPC IPC optimization on cnSGW-C and SMF interfaces, you must perform the following:

- The inter-rack routing must be applied to the branched core network.
- More configuration parameters such as GTPC, cnSGW-C, and SMF are enabled, when extra routes are added for supporting a new routable network across rack servers.

Note The following are configuration-related points:

- The SGW service pods for the S5 egress MBR request are used for IPC messages toward PGWc and SMF IP in the list.
- The SMF service pods for S5 CBR, UBR, and DBR requests are used for IPC messages toward the cnSGW-C in the list.
- IPC messages reuse the N3 or the T3 configuration for the respective interfaces to retry messages, whenever the timeout in the peer node occurs.

Call Flows

This section describes the key call flows for this feature.

cnSGW-C GTPC Optimization with Inter-rack IPC Call Flow

This section describes the cnSGW-C GTPC Optimization with Inter-rack IPC call flow.

Figure 1: cnSGW-C GTPC Optimization with Inter-rack IPC Call Flow

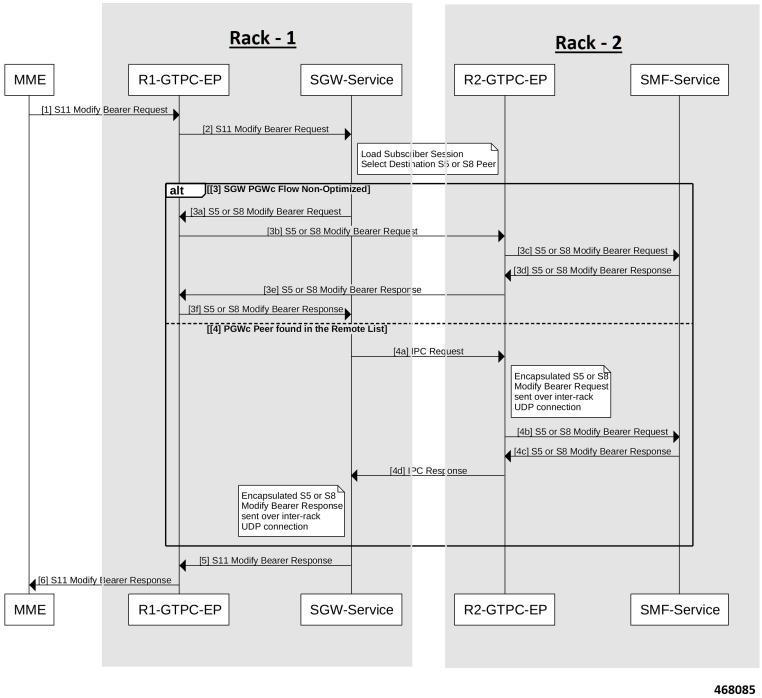


Table 3: cnSGW-C GTPC Optimization with Inter-rack IPC Call Flow Description

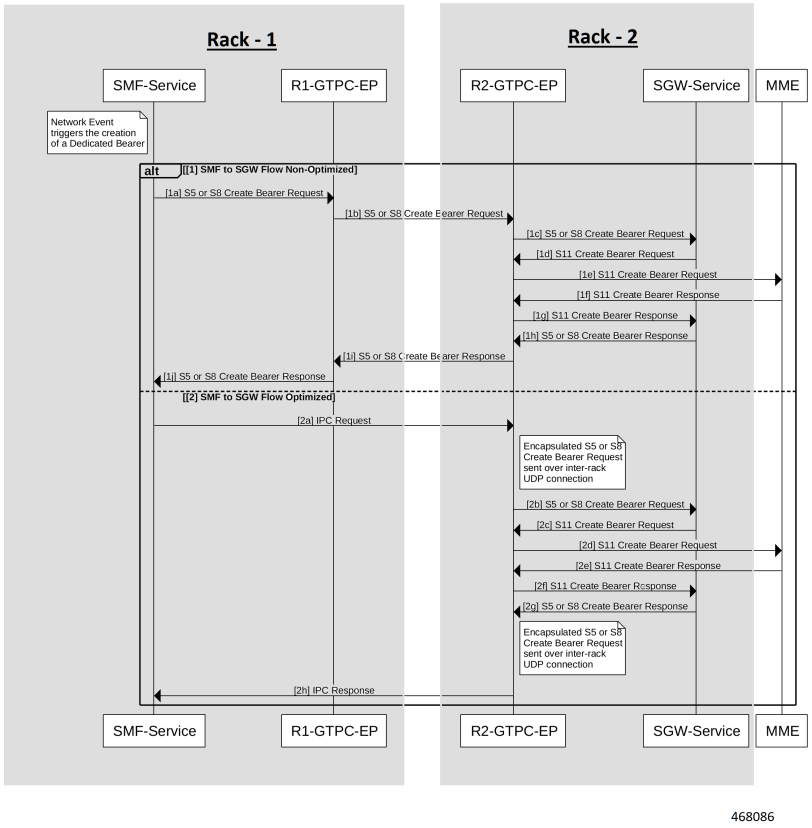
Step	Description
1	MME sends the S11 Modify Bearer Request to R1-GTPC-EP.
2	R1-GTPC-EP sends the S11 Modify Bearer Request to SGW-service. **Note** The SGW-service section performs the following: • Loads the subscriber session. • Selects the destination as the S5 or the S8 peer.
3	The following are sub-steps in the Alt SGW PGWc flow non-optimized scenario.
3a	SGW-service sends the S5 or the S8 Modify Bearer Request to R1-GTPC-EP.
3b	R1-GTPC-EP sends the S5 or the S8 Modify Bearer Request to R2-GTPC-EP.
3c	R2-GTPC-EP sends the S5 or the S8 Modify Bearer Request to SMF-service.
3d	SMF-service processes and sends the S5 or the S8 Modify Bearer Response to R2-GTPC-EP.
3e	R2-GTPC-EP sends the Modify Bearer Response to R1-GTPC-EP.
3f	R1-GTPC-EP sends the Modify Bearer Response to SGW-service.

Step	Description
4	Alternatively, the PGWc peer scenario is available in the remote list. The following are sub-steps in this scenario.
4a	SGW-service sends the IPC Request to R2-GTPC-EP. **Note** The R2-GTPC-EP section performs the following: • Encapsulates the S5 or the S8 Modify Bearer Request. • Sends it over the inter-rack UDP connection.
4b	R2-GTPC-EP sends the S5 or the S8 Modify Bearer Request to SMF-service.
4c	SMF-service processes and sends the S5 or the S8 Modify Bearer Response to R2-GTPC-EP.
4d	R2-GTPC-EP sends the IPC Response to SGW-service. **Note** The SGW-service section performs the following: • Encapsulates the S5 or the S8 Modify Bearer Response. • Sends it over the inter-rack UDP connection.
5	SGW-service sends the Modify Bearer Response to R1-GTPC-EP.
6	R1-GTPC-EP sends the Modify Bearer Response to MME.

SMF and PGWc GTPC Optimization with Inter-rack IPC Call Flow

This section describes the SMF and PGWc GTPC Optimization with Inter-rack IPC call flow.

Figure 2: SMF and PGWc GTPC Optimization with Inter-rack IPC Call Flow



468086

Table 4: SMF and PGWc GTPC Optimization with Inter-rack IPC Call Flow Description

Step	Description
1	The following are sub-steps in the Alt SMF to SGW flow non-optimized scenario. **Note** The SMF-service section performs the following: • Triggers the networking event. • Creates the resolute bearer.
1a	SMF-service sends the S5 or the S8 Create Bearer Request to R1-GTPC-EP.
1b	R1-GTPC-EP sends the S5 or the S8 Create Bearer Request to R2-GTPC-EP.
1c	R2-GTPC-EP sends the S5 or the S8 Create Bearer Request to SGW-service.
1d	SGW-service processes and sends the S11 Create Bearer Request to R2-GTPC-EP.
1e	R2-GTPC-EP sends the S11 Create Bearer Request to MME.
1f	MME processes and sends the S11 Create Bearer Response to R2-GTPC-EP.

Step	Description
1g	R2-GTPC-EP sends the S11 Create Bearer Response to SGW-service.
1h	SGW-service processes and sends the S5 or the S8 Create Bearer Response to R2-GTPC-EP.
1i	R2-GTPC-EP sends the S5 or the S8 Create Bearer Response to R1-GTPC-EP.
1j	R1-GTPC-EP sends the S5 or the S8 Create Bearer Response to SMF-service.
2	The following are sub-steps in the SMF to SGW flow-optimized scenario.
2a	SMF-service sends the IPC request to R2-GTPC-EP. **Note** The R2-GTPC-EP section performs the following: • Encapsulates the S5 or the S8 Create Bearer Request. • Sends it over the inter-rack UDP connection.
2b	R2-GTPC-EP sends the S5 or the S8 Create Bearer Request to SGW-service.
2c	SGW-service processes and sends the S11 Create Bearer Request to R2-GTPC-EP.
2d	R2-GTPC-EP sends the S11 Create Bearer Request to MME.
2e	MME processes and sends the S11 Create Bearer Response to R2-GTPC-EP.
2f	R2-GTPC-EP sends the S11 Create Bearer Response to SGW-service.
2g	SGW-service processes and sends the S5 or the S8 Create Bearer Response to R2-GTPC-EP. **Note** The R2-GTPC-EP section performs the following: • Encapsulates the S5 or the S8 Create Bearer Request. • Sends it over the inter-rack UDP connection.
2h	R2-GTPC-EP processes and sends the IPC Response to SMF-service.

Feature Configuration

To configure this feature, use the following configuration:

```
config
   sgw sgw_name
      resmgr-batch-operation [ disable | enable ]
      end
```

NOTES:

resmgr-batch-operation [disable | enable]—Configures the batch operation. By default, **resmgr-batch-operation** is disabled.

Configuration Example

The following is an example configuration.

```
config
instance instance-id 1
 endpoint GTP
   interface gtp-inter-rack
     vip-ip 209.165.202.130 vip-port 9084 vip-interface bd2.gtpe.2101
     gtpc-ipc
     gtp-peer-entry 209.165.202.131 port 9084 remote-gtp-peer-list [ 209.165.202.140
209.165.202.141 ]
     end
```

OAM Support

This section describes operations, administration, and maintenance support for this feature.

KPI Support

The following statistics are supported for the GTPC IPC Cross-rack Support feature.

1. KPI Name: **udp_rpc_request_total**

   The following table lists **udp_rpc_request_total** KPI details.

Description	Functionality	Label Names	Possible Values
This KPI displays the total number of UDP client endpoint request messages.	The functionality output is inter-rack RPC requests total.	MessageName	gptv2 message types like CBR, UBR, MBR
		retry	Is attempt a retry
		status	Status of the request message, which is a success or a failure
		status_code	0/1/2

2. KPI Name: **udp_rpc_response_total**

   The following table lists **udp_rpc_response_total** KPI details.

Description	Functionality	Label Names	Possible Values
This KPI displays the total number of UDP client endpoint response messages.	The functionality output is inter-rack RPC response total.	MessageName	gptv2 message types like CBR, UBR, MBR
		status	Status of the response message, which is a success or a failure
		status_code	0/1/2

Note The current implementation supports KPIs only on the client-side, as they reside on service pods, where KPIs can be enabled without impacting performance.