

Failure Handling Support

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- Access and Mobility Management Function Failure Handling, on page 3
- SMF Failure Handling of HTTP Status Code 429, on page 5
- Charging Function Failure Handling, on page 9
- Network Repository Function Failure Handling, on page 19
- Policy Control Function Failure Handling, on page 32
- Unified Data Management Failure Handling, on page 37
- User Plane Function failure handling, on page 43

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Products or Functional Area	SMF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
Enhanced the existing failure handling configuration of N4SessionModificationReq message to support cause codes 0-255.	2021.02.3
Enhanced the existing failure handling configuration of N4SessionModificationReq message to include conditions to control the session termination based on predefined procedures.	2021.02.3
Added retransmission support for the following request messages: • Namf_Communication EBI Assignment Request • Namf_Communication N1 N2 Message Transfer Request	2021.02.2
Added permissible range values for **response-timeout** command in the PCF and UDM configuration	2021.02.0
RAT type FHT support and graceful timeout handling and its related statistics introduced.	2021.01.0
First introduced.	Pre-2020.02.0

Feature Description

The system performs error handling by segregating error codes into recoverable and non-recoverable error codes. It attempts to recover the endpoints with continuous retries when SMF receives recoverable errors from the NRF server for messages, such as NF Registration, NF Update, NF Heart Beat, and so on. This feature provides a flexible way for handling errors during the NRF interactions with SMF and other network functions, such as Charging Function (CHF), Network Repository Function (NRF), Policy Control Function (PCF), Unified Data Management (UDM), and User Plane Function (UPF).

This feature supports the following functionality:

- Configurable retry actions for specific error codes, which occur during the NRF interactions with other NFs.
- Flexibility to decide on a retry action for an error code after retrying all the endpoints in an NRF.
- CLI configuration under the **profile nf-client-failure** template to configure the error codes and the corresponding retry actions for NRF messages. You can also configure a failover option for an error code after retrying all the endpoints in an NRF.
- Provides HTTPv2 status code range support in the failure handling templates of other NFs.

Access and Mobility Management Function Failure Handling

Feature Description

The SMF supports failure handling of the Access and Mobility Management Function (AMF). Based on the request messages, SMF supports retransmission to the same endpoint.

How it Works

SMF provides retransmission support for the following request messages:

- Namf_Communication EBI Assignment Request
- Namf_Communication N1 N2 Message Transfer Request

When SMF doesn't receive a response for the preceding messages, SMF retransmits the message to the same endpoint. SMF starts the internally configured timer after sending these messages to the AMF. The timer stops after SMF receives a response from the AMF. In case the timer expires while waiting for a response, SMF uses the retry mechanism for which you have configured the number of retry attempts.

Configuring Retransmission for Request Messages

To configure retransmission for the Namf_Communication EBI Assignment and Namf_Communication EBI Assignment messages, use the following sample configuration:

```
config
  profile nf-client-failure nf-type amf
    profile failure-handling failure_handling_name
    service name type namf-comm
      message type { AmfCommEBIAssignment | AmfCommN1N2MessageTransfer
| AmfCommSMStatusChangeNotify }
       status-code httpv2 status_code
       retransmit retransmit_value
       retransmit interval retransmit_interval_value
       retry retry_value
       action retry-and-continue
    exit
  exit
```

NOTES:

- **service name type namf-comm**: Specify the AMF service name type as namf-comm.
- **message type { AmfCommEBIAssignment | AmfCommN1N2MessageTransfer | AmfCommSMStatusChangeNotify }**: Specify the message type of the namf-comm AMF service name type as **AmfCommEBIAssignment**, **AmfCommN1N2MessageTransfer**, or **AmfCommSMStatusChangeNotify**.
- **status-code httpv2** *status_code* : Specify the status code of the service. The *status_code* must be an integer in the range of 0–599.

- **retransmit** *retransmit_value*: Specify the maximum retransmission value for the same endpoint. The *retransmit_value* must be an integer in the range of 1–10.

  If SMF sends message and receive an error in the HTTP status code and if you have configured a valid retransmit count, then that number of retransmission attempts are made to the same endpoint. The maximum retransmit count is used from the first-time configuration of the HTTP status code.

  If SMF receives failure error HTTP code even after retransmission, then SMF retransmits to the same endpoint in the following conditions:

  - If valid retransmit counts are configured for the first HTTP status code that SMF received.
  - If a valid retransmit count, which must not be zero, exists for the received HTTP error code.

- **retransmit interval** *retransmit_interval_value*: Specify the retransmission interval value in milliseconds. The default value is 1000.

  If you have configured the retransmit interval, then SMF waits for the timeout between retransmissions.

- **retry** *retry_value*: Specify the number of retry attempts to the different available endpoints. The *retry_value* must be an integer in the range of 1–10.

- **action retry-and-continue**: Specify the retry as per the configured retry count and continue the session.

Configuration Example

The following is an example configuration of the retransmission for the Namf_Communication EBI Assignment message:

```
config
  profile nf-client-failure nf-type amf
    profile failure-handling FHAMF
    service name type namf-comm
      message type AmfCommEBIAssignment
      status-code httpv2 504
      retransmit 1
      retransmit interval 1000
      retry 1
      action retry-and-continue
    exit
  exit
```

SMF Failure Handling of HTTP Status Code 429

Table 3: Feature History

Feature Name	Release Information	Feature Description
SMF Failure Handling of HTTP Status Code 429	2025.03.0	This feature enables SMF to specifically handle the 429 error code, rather than treating it as a generic failure and using the standard failure handling template. When the SMF receives a 429 response, it can be configured to blocklist the congested peer NF for a specific duration. This prevents the SMF from sending additional requests to the overloaded peer, allowing the peer to recover and mitigating the impact of congestion on the SMF's operations. The duration for blocklisting can be determined by the **retry-after** value received in the 429 response or by a locally configured value Command Introduced :**show peers application-data**

Overview

This feature introduces enhanced handling for HTTP status code 429 (Too Many Requests) within the Session Management Function (SMF), specifically by implementing a peer blocklisting mechanism during network congestion.

In releases before 2025.03.0, the SMF treated HTTP 429 errors like any other HTTP error, using a standard failure handling template. This feature provides a dedicated and intelligent way to respond to 429 errors when received from the Service Communication Proxy (SCP), especially when accompanied by a `retry-after` timer. It allows the SMF to temporarily block (blocklist) congested peer Network Functions (NFs) to prevent further requests from being sent to them

How it works

When target peer is communicated via SCP model D in SMF, SMF may receive error response. This error response could be from SCP directly or from target peer via SCP. If it is from SCP, then action configured under FH of SCP is considered. If it is from target peer, nf-action from FH of SCP is considered.

The failure handling configuration is extended to include an `action-list` with an `action-type` of `blocklist-peer`. The `retry-after` duration, if provided in the HTTP header, takes precedence for how long the peer remains blocklisted. A new internal cache is used to store the status and timestamp of blocklisted peers. The SMF consults this cache before sending new requests or handling active calls, avoiding blocklisted NFs.

Information About HTTP 429 Congestion Handling with Peer Blocklisting

The SMF now supports HTTP status code 429 within its NF failure handling configuration and SCP failure handling. For 429 errors, the configured action can be `continue`, `retry`, or `terminate`.

A new `action-list` is supported specifically for status code 429. Within this `action-list`, the `action-type blocklist-peer` is supported, which can optionally include a `retry-after` attribute. The `retry-after` HTTP header received in the error response is given priority over any locally configured `retry-after` duration.

A peer is blocklisted for the duration specified by the `retry-after` value received in the response or configured locally. This blocklisting applies specifically when the 429 error response is received from the SCP (indicating the SCP itself is congested). If a 429 error is received from a target peer, the existing failure handling configuration for that peer will be applied instead.

A new cache is implemented to store NF-specific information, including the peer's status (for example, blocklisted) and the timestamp of when it was blocklisted. This cache stores information for both discovered and locally configured peers. Blocklisting is applied at the `NfInstance` level, meaning if an IP address supports multiple services, the entire instance is blocklisted.

While a peer is blocklisted, Diameter Peer Discovery (DPD) is not activated, and no pings are initiated to that peer. The blocklisted status for a discovered peer is removed if the Network Repository Function (NRF) sends a Deregister Notification or ProfileChanged Notification that changes the NF's status to suspended, or if a DPD failure is detected.



Note Since DPD is disabled for the blocklisted peers, such peers are not displayed in the output of the **show peers all** command. Once DPD is resumed, it is displayed.

The SMF checks for the blocklisted status and its expiry timestamp before selecting an NF for both new and active calls. The SMF will still handle incoming requests from a blocklisted peer and respond to them; however, receiving an incoming message from a blocklisted peer will not result in it being whitelisted. If an incoming message from a blocklisted peer triggers an outbound request *to that same peer*, the SMF will refrain from sending the outbound request.

For scenarios where an application has only one endpoint (discovered or static) and it's blocklisted, or when all available endpoints are blocklisted, a `status-code internal` configuration can be used. If the action for `status-code internal` is `retry`, the system will look for available profiles in the local cache that are not blocklisted.

The blocklist cache is synchronized periodically within the same cluster but is not replicated to other clusters (no geo-replication).

Benefits of HTTP 429 Congestion Handling with Peer Blocklisting

- **Improved Network Stability**: Prevents the SMF from continuously sending requests to already congested Network Functions (NFs), reducing the load on overloaded peers.
- **Efficient Resource Utilization**: By temporarily avoiding congested peers, the SMF can direct traffic to available and healthy NFs, optimizing overall network resource usage.
- **Automated Congestion Response**: Provides an automated mechanism to react to HTTP 429 errors and `retry-after` headers, minimizing manual intervention during congestion events.
- **Enhanced Reliability**: Contributes to a more robust and reliable network by intelligently managing traffic flow during overload situations.

Supported Scenarios

- **Network Overload Situations**: Specifically designed for scenarios where the SCP (Service Communication Proxy) sends an HTTP 429 error code along with a `NF_CONGESTION_RISK` cause and a `retry-after` timer, indicating congestion.
- **SCP-Initiated Congestion**: The blocklisting mechanism applies when the 429 error response is received directly from the SCP, implying the SCP itself is experiencing congestion

  .
- **Single or All Endpoints Blocklisted**: The `status-code internal` configuration supports use cases where an application has only one endpoint (either discovered or static) that becomes blocklisted, or when all available endpoints for an application are blocklisted.

Restrictions for HTTP 429 Congestion Handling with Peer Blocklisting

- **Retransmit Configuration Conflict**: Users cannot configure retransmit along with an `action-list` that includes a blocklist for a certain duration, as retransmit might attempt to re-select the same blocklisted peer.
- **`retry-after` Requirement**: The blocklist configuration will not be applied if the `retry-after` duration is not available from either the configuration or the HTTP error response.
- **Parallel Processing**: Due to parallel processing, it is possible that additional requests might briefly go to a blocklisted peer for a short duration after blocklisting.
- **Ignoring Subsequent 429s**: If a peer is already blocklisted, the SMF will ignore any new 429 responses received for that peer.
- **NRF Peer Congestion Handling**: Congestion handling for NRF peers is not supported in the July release and is planned for an upcoming release.
- **Cache Replication**: The blocklist cache is synchronized only within the same cluster and is not geo-replicated to other clusters.
- **Metric Type Limitation**: A gauge metric is not supported for the `nf_failure_handling_stats_total` statistic when labeled with `status=blocklisted`.

Configuration Overview

The configuration for HTTP 429 Congestion Handling with Peer Blocklisting involves extending the existing NF failure handling profiles. You will define a failure handling profile for a specific NF type, such as UDM, and within that profile, specify actions for HTTP status code 429. This includes associating an `action-list-profile` that defines the `blocklist-peer` action type and its `retry-after` duration.

This section shows how to configure the SMF to handle HTTP 429 status codes by blocklisting peers.

1. Define a Failure Handling Profile for an NF Client:

   ```
   profile nf-client-failure nf-type udm
   ```
2. Create a Specific Failure Handling Profile:

   ```
   profile failure-handling FHUDM
   ```
3. Specify Service Name and Type (for example, nudm-sdm):

   ```
   service name type nudm-sdm
   ```

4. Configure Response Timeout (Optional):

```
responsetimeout 2300
```

5. Configure Message Type (for example, UdmSdmGetUESMSubscriptionData):

```
message type UdmSdmGetUESMSubscriptionData
```

6. Configure Action for HTTP Status Code 429:

```
status-code httpv2 429
action retry-and-continue
action-list-profile PA1
```

7. Configure Action for Internal Status Code (for example, `blocklisted`):

```
status-code internal [blocklisted]
action { continue | terminate | retry }
```

Note Replace `{ continue | terminate | retry }` with the desired action.

8. Define the Action List Profile (for example, PA1):

```
profile action-list PA1
action-type blocklist-peer
retry-after <time in seconds>
```

Note Replace `<time in seconds>` with the desired duration for blocklisting.

Verify HTTP 429 Congestion Handling with Peer Blocklisting

You can verify the blocklisted status of peers using the `show peers application-data` command. Here is a sample output for the command.

The command output displays the blocklisted peers and theirs expiration-time in human readable format (e.g. Thu, 03 Jul 2025 05:15:06 UTC). The NFType and GrInstanceID is also displayed.

```
smf# show peers application-data
GR                      NF
INSTANCE  PEER ADDRESS    TYPE  ADDITIONAL DETAILS
-------------------------------------------------------------------------------------------------
1         10.1.34.15:8010 SCP   Blocklisted: Thu, 31 Jul 2025 05:48:31 UTC
1         10.1.34.61:9011 UDM   Blocklisted: Thu, 31 Jul 2025 05:50:31 UTC
```

Note A blocklisted peer whose `retry-after` duration has expired is removed from the blocklist cache when it is next selected for a transaction. Therefore, you might observe an expired blocklisted peer in the display until it is involved in a transaction.

Monitor RIB Metric Metric Translation

After configuring HTTP 429 Congestion Handling with Peer Blocklisting, you can monitor its activity through existing statistical metrics. The existing stat metric `nf_failure_handling_stats_total` has been enhanced. You

can use the label `status` with the value `blocklisted` to monitor statistics related to blocklisted peers. This allows you to track the total count of instances where peers have been blocklisted due to congestion handling.

Charging Function Failure Handling

Feature Description

Table 4: Feature History

Feature Name	Release Information	Description
CHF Failure Handling at Rating Group and Application level	2024.01	SMF allows you to define configurable actions to handle the failures associated with the application errors and rating group IDs. This feature introduces new command **failure-handling error-type [rg | app]** in the charging profile configuration. For more information, see the profile charging failure-handling error-type command. **Default Setting**: Disabled - Configuration Required

The SMF supports failure handling of the Charging Function (CHF) server. In the event of the failure of an online CHF server, the SMF relays the charging information to the offline CHF server.

For a seamless transfer of charging information, the SMF invokes the configurations associated with the CHF failure handling profile. If you have configured failure handling, the SMF continues the session with the selected CHF configured in another profile.

SMF supports failure handling of the Charging Function (CHF) server at the following levels.

- HTTP-level status code
- Application-level error code
- Rating group-level result code

You can configure the following actions for application errors.

- Drop data—Drop data corresponding to the Charging ID.
- Terminate—Release the PDU session.
- Continue—Disable charging.

You can configure the following actions for Rating Group errors.

- Convert-offline—Convert to offline charging.

- Delete flow—Delete the flow or rule associated with the rating group.
- Drop data—Drop the data corresponding to the rating group.
- Terminate—Release the PDU session.

Without the configuration, SMF continues with the default behaviour. See Table 5 and Table 6 for the default behavior.

For information on how to select the charging server, see the CHF Selection section in the Subscriber Charging chapter of this guide.

How it Works

This section describes how the offline failover support for charging feature works.

Handling a CHF Server Failure

The CHF server failure occurs when the selected CHF sends failure response or sends no response. For a CHF server failure, the NF library sends status code that is based on the failure template. This template is associated with the CHF network profile. The smf-service sends the profile information to smf-rest-ep while sending the IPC message.

The failure template is configured with the list of HTTP error codes and the associated failure actions and retry count, as required. This feature supports the failure actions:

- Retry and Continue—For this failure action, NF library attempts until the configured number of times before fallback. After the configured number of times complete, the NF library falls back to the lower priority CHF server IP address. If a failure or no response is received from the CHF server, the "continue" action is returned to the smf-service.
- Terminate—For this failure action, NF library does not attempt to send a message to other CHF servers. The library sends a reply to smf-service with the action as "terminate". For the "terminate" failure action, the smf-service deletes the session.
- Continue—For this failure action, the smf-service continues the session and sends the charging message to the offline CHF server. This server is configured as part of the local static CHF profile that is meant for offline purposes. In addition, the failure handling profile for offline CHF is configured.



Note For the "continue" failure action, you can configure the offline CHF server at SMF in a separate profile. SMF will use this profile after the CHF server failure. If the offline CHF server is not confgured, the session is continued without imposing any charging.

Relaying to an Offline CHF Server

After CHF server failure, when the SMF continues, it converts the ongoing charging services as follows:

- Converts the services with both online and offline charging method to the offline charging method.
- Converts the services with online charging method to the offline charging method.
- No change for the services with the offline charging method.

HTTP Cause Code Mapping with Failure Actions

The following table lists the mapping of failure actions with the associated HTTP cause code. Based on the network requirements, you can change the mapping.

Table 5: HTTP Cause Code Mapping with Failure Actions

Http-2 Cause Codes and Description		Converged CHF Failure Action			Offline CHF Failure Actions		
Code	Description	CDR-I	CDR-U	CDR-T	CDR-I	CDR-U	CDR-T
400	Bad Request	Terminate	No config	No config	Terminate	No config	No config
403	Forbidden	Terminate	No config	No config	Terminate	No config	No config
404	Not found	Terminate	No config	No config	Terminate	No config	No config
405	Method Not allowed	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	No config	No config
408	Request Timeout	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	Retry and Continue	Retry and Continue
500	Internal Server Error	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	Retry and Continue	Retry and Continue
503	Service Unavailable	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	Retry and Continue	Retry and Continue
508	Gateway Timeout	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	Retry and Continue	Retry and Continue
0	No reply from server	Retry and Continue	Retry and Continue	Retry and Continue	Terminate	Retry and Continue	Retry and Continue

Application Error Code-based Failure Handling

The following table lists the application error result codes with the associated SMF failure handling action.

Table 6: Application Error Result Code Mapping with the SMF Failure Handling Action

Application Error Result Code	SMF Failure Handling Action
CHARGING_NOT_APPLICABLE	Terminate Session
END_USER REQUEST_DENIED	Terminate Session
QUOTA_LIMIT_REACHED	Drop traffic for the session
END_USER_REQUEST_REJECTED	Terminate Session

Rating Group-level Failure Handling

The following table lists the Rating Group-level error result codes with the associated SMF failure handling action.

Table 7: Rating Group-level Error Result Code Mapping with the SMF Failure Handling Action

Rating Group-level Error Result Code	SMF Failure Handling Action
RATING_FAILED	Drop traffic corresponding to the Rating Group
QUOTA_MANAGEMENT_NOT_APPLICABLE	Convert to offline
USER_UNKNOWN	Ignored **Note** SMF supports this action only at the session level.
END_USER_SERVICE_DENIED	Drop traffic corresponding to the Rating Group
QUOTA_LIMIT_REACHED	Drop traffic corresponding to the Rating Group
END_USER_SERVICE_REJECTED	Drop traffic corresponding to the Rating Group

Call Flows

This section describes the call flows that are associated with CHF failure handling.

5G Delete Flow Action Call Flow

This section describes the 5G delete flow action call flow.

Figure 1: 5G Delete Flow Action Call Flow

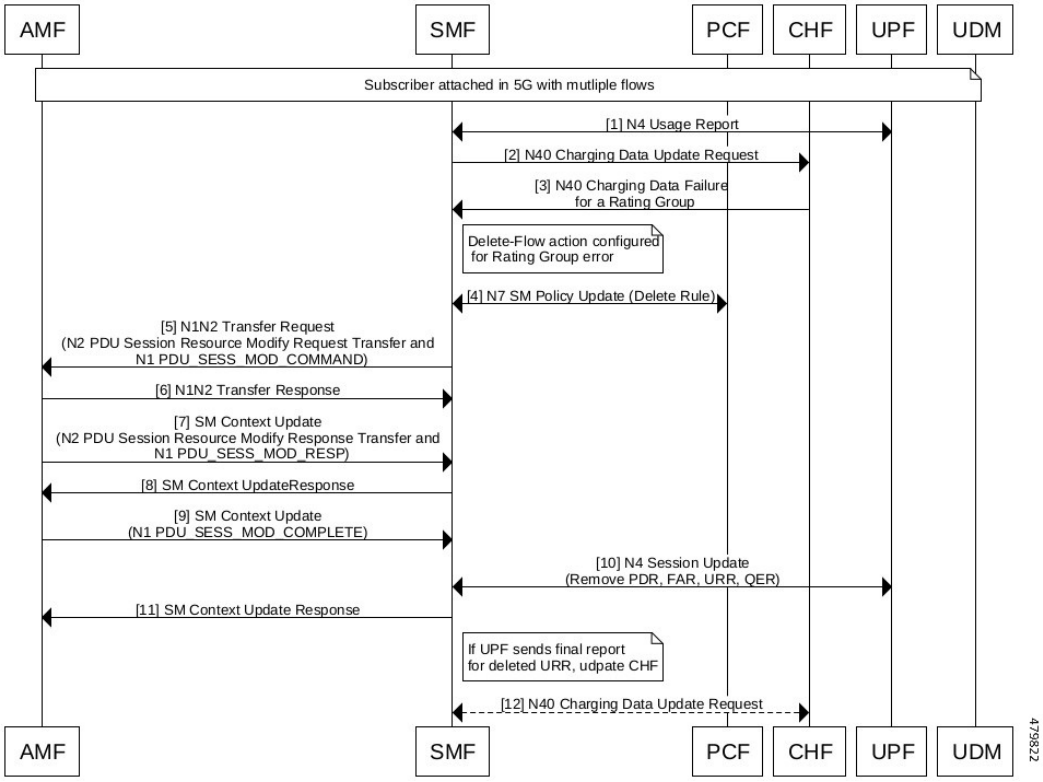


Table 8: 5G Delete Flow Action Call Flow Description

Step	Description
1	A subscriber is attached in 5G with multiple flows. UPF sends the N4 usage report to SMF.
2	SMF sends the N40 Charging Data Update Request to CHF.
3	CHF sends the N40 Charging data failure for a Rating Group to SMF. **Note** Configure the delete flow action for the Rating Group error. This action is used for deletion of the flow or rule associated with the Rating Group.
4	SMF sends the N7 SM Policy Update for deletion of a rule to PCF.
5	SMF sends the N1N2 Transfer Request to AMF. This request includes the N2 PDU Session Resource Modify Request Transfer and the N1 PDU Session Modification message.
6	AMF sends the N1N2 Transfer Response to SMF.
7	AMF sends the SM Context Update to SMF. This update message includes the N2 PDU Session Resource Modify Response Transfer and the N1 PDU Session Modify Response.

Step	Description
8	SMF sends the SM Context Update Response to AMF.
9	AMF sends the SM Context Update to SMF. This update includes the details on the completion of the N1 PDU session modification.
10	SMF sends the N4 Session Update to UPF. This update includes the details on the PDR, FAR, URR, and QER to be removed.
11	SMF sends the SM Context Update Response to AMF. **Note** If the UPF sends the final report for the deleted URR, then SMF sends this update to CHF.
12	SMF sends the N40 Charging Data Update Request to CHF.

4G Delete Flow Action Call Flow

This section describes the 4G delete flow action call flow.

Figure 2: 4G Delete Flow Action Call Flow

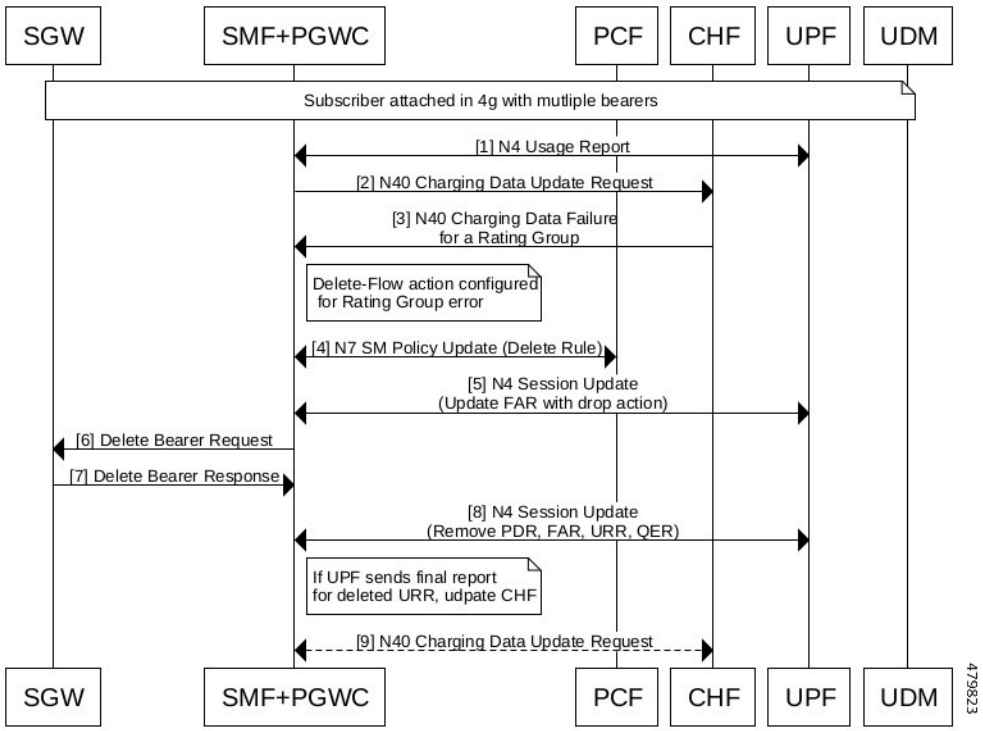


Table 9: 4G Delete Flow Action Call Flow Description

Step	Description
1	A subscriber is attached in 4G with multiple bearers. UPF sends the N4 Usage Report to SMF+PGW-C.

Step	Description
2	SMF+PGW-C sends the N40 Charging Data Update Request to CHF.
3	CHF sends the N40 Charging data failure for a Rating Group to SMF+PGW-C. **Note** Configure the delete flow action for the Rating Group error. This action is used for deletion of the flow or rule associated with the Rating Group.
4	SMF+PGW-C sends the N7 SM Policy Update for deletion of a rule to PCF.
5	SMF+PGW-C sends the N4 Session Update to UPF. This message includes details on updating FAR with the drop action.
6	SMF+PGW-C sends the Delete Bearer Request to S-GW.
7	S-GW sends the Delete Bearer Response SMF+PGW-C.
8	SMF+PGW-C sends the N4 Session Update to PCF. This update includes the details on the PDR, FAR, URR, and QER to be removed. **Note** If the UPF sends the final report for the deleted URR, then SMF sends this update to CHF.
9	SMF+PGW-C sends the N40 Charging Data Update Request to CHF.

SMF Behaviour for Failure Actions

The following table describes the SMF behaviour on receiving different failures (Continue, Ignore, and Terminate) in CDR-(I/U/T).

CHF Failure Actions	Converged CHF			Offline CHF		
	CDR-I	CDR-U	CDR-T	CDR-I	CDR-U	CDR-T
Continue	Send CDR to offline CHF. If offline CHF is not configured, continue the session without charging.	Send CDR to offline CHF. If offline CHF is not configured, continue the session without charging.	Send CDR to offline CHF if offline CHF is configured	Continue the session without charging	Continue the session without charging	Continue the session deletion
Terminate	Delete the session	Delete the session	Continue the session deletion	Delete the session	Delete the session	Continue the session deletion

CHF Failure Actions	Converged CHF			Offline CHF		
	CDR-I	CDR-U	CDR-T	CDR-I	CDR-U	CDR-T
Ignore	Delete the session	No action taken. Record(s) will be reattempted in the next CDR request.	Continue the session deletion	Delete the session	No action taken. Record(s) will be reattempted in the next CDR request.	Continue the session deletion

Standards Compliance

The offline failover support for charging feature complies with the following standards:

- *3GPP TS 32.255, version 15.3.0*
- *3GPP TS 32.290, version 15.4.0*
- *3GPP TS 32.291, version 15.3.0*

Limitations

The offline failover support for charging feature has the following limitation:

- Session Level limits are mandatory from CHF or you must configure them locally. As per the 3GPP specification, the last linked URR can’t be removed when the online URR needs to be delinked from the offline URR.

The CHF failure handling at Rating Group and application levels feature has the following limitations:

- In vSMF, only the **terminate** action for the application error is applicable.
- The **drop-data** action is supported only for the online charging.
- If multiple Rating Groups return an error and if one of the errors has the **terminate** action configured, then the PDU session is terminated.
- If multiple Rating Groups return an error and if the Rating Group mapping to the default flow has the **delete-flow** action configured, then the PDU session is terminated.
- If multiple PCC rules and Rating Groups are mapped to a single flow, then any Rating Group error that is configured for **delete-flow** action, results in the deletion of the entire flow.
- For multiple Rating Groups mapped to a single flow, multiple Rating Groups result in failure in the following scenarios:
  - If the **terminate** action is configured, then this action takes the highest priority.
  - If both the **delete-flow** and **drop-data** actions are configured, then the **delete-flow** action takes the higher priority.
  - If both the **convert-to-offline** and **drop-data** actions are configured, then **drop-data** takes the higher priority.

Configuring CHF Failure Handling

This section describes how to configure the CHF failure handling.

This feature involves the following steps:

1. Configuring Failure Handling Profile, on page 17
2. Configuring Offline Server Client and Offline Failure Handling Profile, on page 18

Configuring Failure Handling Profile

You can configure the HTTP status code with the corresponding action for the CHF Create, Update, or Release messages. Based on the configuration of the Failure Handling profile, the SMF takes an action when the CHF server failure occurs.

To configure the failure handling profile, use the following sample configuration:

```
config
  profile nf-client-failure nf-type chf
    profile failure-handling fh_profile_name
      service name type servicename_type
        message type messagetype_value
          status-code httpv2 statuscode_value
            action { continue | retry-and-continue | retry-and-ignore
| retry-and-terminate } retry retry_value
              exit
```

NOTES:

- **profile nf-client-failure nf-type chf**: Specify the name of the network function that is required after the NF client failure.
- **profile failure-handling** *fh_profile_name*: Specify the name of the profile for failure handling.
- **service name type** *servicename_type*: Specify the name of the service type. *servicename_type* can be one of the following values for CHF:
  - nchf-convergedcharging
  - nchf-spendinglimitcontrol
- **message type** *messagetype_value*: Specify the value for the type of message. *messagetype_value* can be one of the following values for CHF:
  - ChfConvergedchargingCreate
  - ChfConvergedchargingUpdate
  - ChfConvergedchargingDelete
- **status-code httpv2** *statuscode_value* : Specify the status code as per the configured failure template. *statuscode_value* must be an integer in the range of 0–599. Use either '-' or ' as separator for the range of status codes.

- **action { continue | retry-and-continue | retry-and-ignore | retry-and-terminate } retry** *retry_value*: Specify the failure action and the number of retry attempts. *retry_value* must be an integer in the range of 1–10.

Configuring Offline Server Client and Offline Failure Handling Profile

To configure the offline client profile and offline failure handling profile for the selected CHF server, use the following sample configuration:

```
config
  profile network-element chf chf_name
    nf-client-profile nf_client_profile_name
    failure-handling-profile fh_profile_name
    nf-client-profile-offline offline_server_profile_name
    failure-handling-profile-offline fh_profile_offline_name
    exit
```

NOTES:

- **profile network-element chf** *chf_name*: Specify the name of the CHF server.
- **nf-client-profile** *nf_client_profile_name*: Specify the name of the NF client profile.
- **failure-handling-profile** *fh_profile_name*: Specify the name of the failure handling profile.
- **nf-client-profile-offline** *offline_server_profile_name*: Specify the NF client profile name for the offline server.
- **failure-handling-profile-offline** *fh_profile_offline_name*: Specify the failure handling profile name for the offline server.

Configuring Action for Rating Group-level and Application-level Errors

Use the following sample configuration to configure an action for Rating Group-level and Application-level errors.

```
config
  profile charging profile_name
    failure-handling error-type [ rg | app ] error-value error_value action
 action_value
    exit
```

NOTES:

- **profile charging** *profile_name*: Enter the Charging Profile configuration mode.
- **failure-handling error-type [rg | app] error-value** *error_value* **action** *action_value*: Configure the failure handling error type as Rating Group or Application with an error value. For the application-level errors, enter one of the following values for *action_value*:
  - **drop-data**—Specify this value to drop data corresponding to the charging ID.
  - **terminate**—Specify this value to release the PDU session.
  - **continue**—Specify this value to disable charging.

For the Rating Group-level errors, enter one of the following values for *action_value*:

- **convert-offline**—Specify this value to convert to offline charging.
- **delete-flow**—Specify this value to delete the flow or rule associated with the Rating Group.
- **drop-data**—Specify this value to drop data corresponding to the Rating Group.
- **terminate**—Specify this value to release the PDU session.

Note

- If you haven't configured an action for the Rating Group-level and application-level errors, the SMF continues with the default behavior.
- SMF doesn’t allow mapping of multiple QoS flows to the same Rating Group for dynamic rules.
- The **drop-data** action is applicable to all the flows that are associated to the Rating Group.

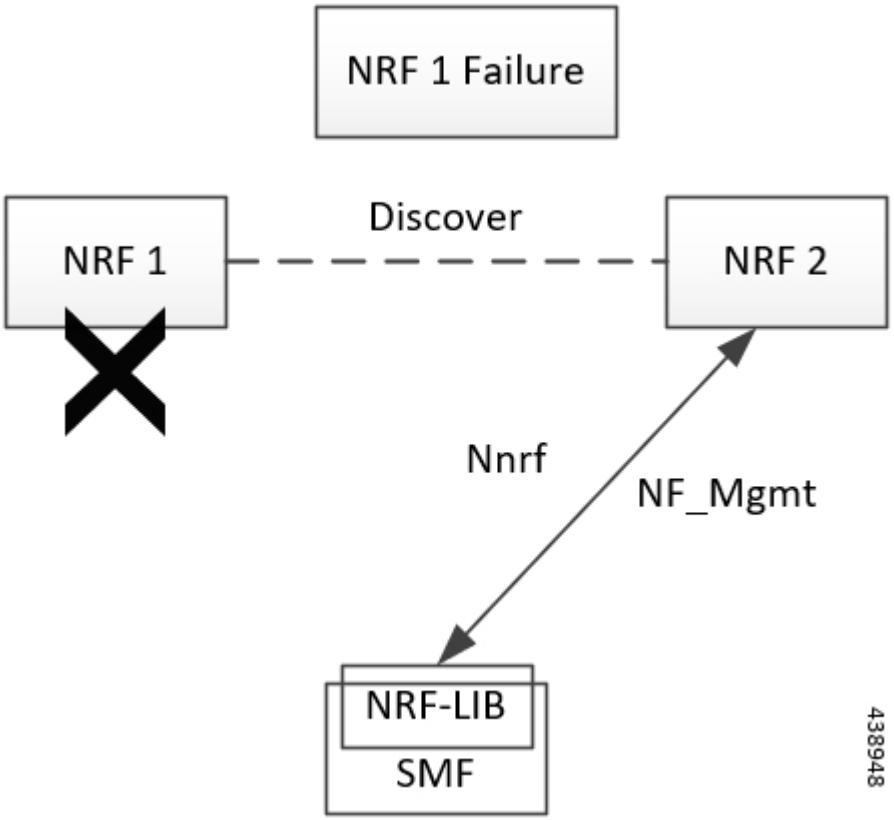
Network Repository Function Failure Handling

Feature Description

The Network Repository Function (NRF) communication failure handling logic is implemented within the SMF. The SMF uses the NF registration messages for tracking the management NRF group operational status.

How it Works

The following figure shows how the SMF handles NRF failures.



In the preceding diagram, NRF 1 is Primary and NRF 2 is secondary for SMF. On bringing up, the SMF registers (NF registration) with NRF 1 and starts NF heartbeat with NRF 1. The SMF uses the heartbeat response to track the operational status.

In case the SMF detects NRF 1 failure by missing NF heartbeat response, the SMF registers to NRF 2 (secondary NRF) and starts sending NF heartbeat. The SMF continues to send NF Register message to NRF 1 to keep track of its status.

If the SMF receives register response from NRF 1, it detects that the NRF 1 is up again. The SMF marks NRF 1 as active once it recovers and stops sending NF heartbeats to NRF 2.

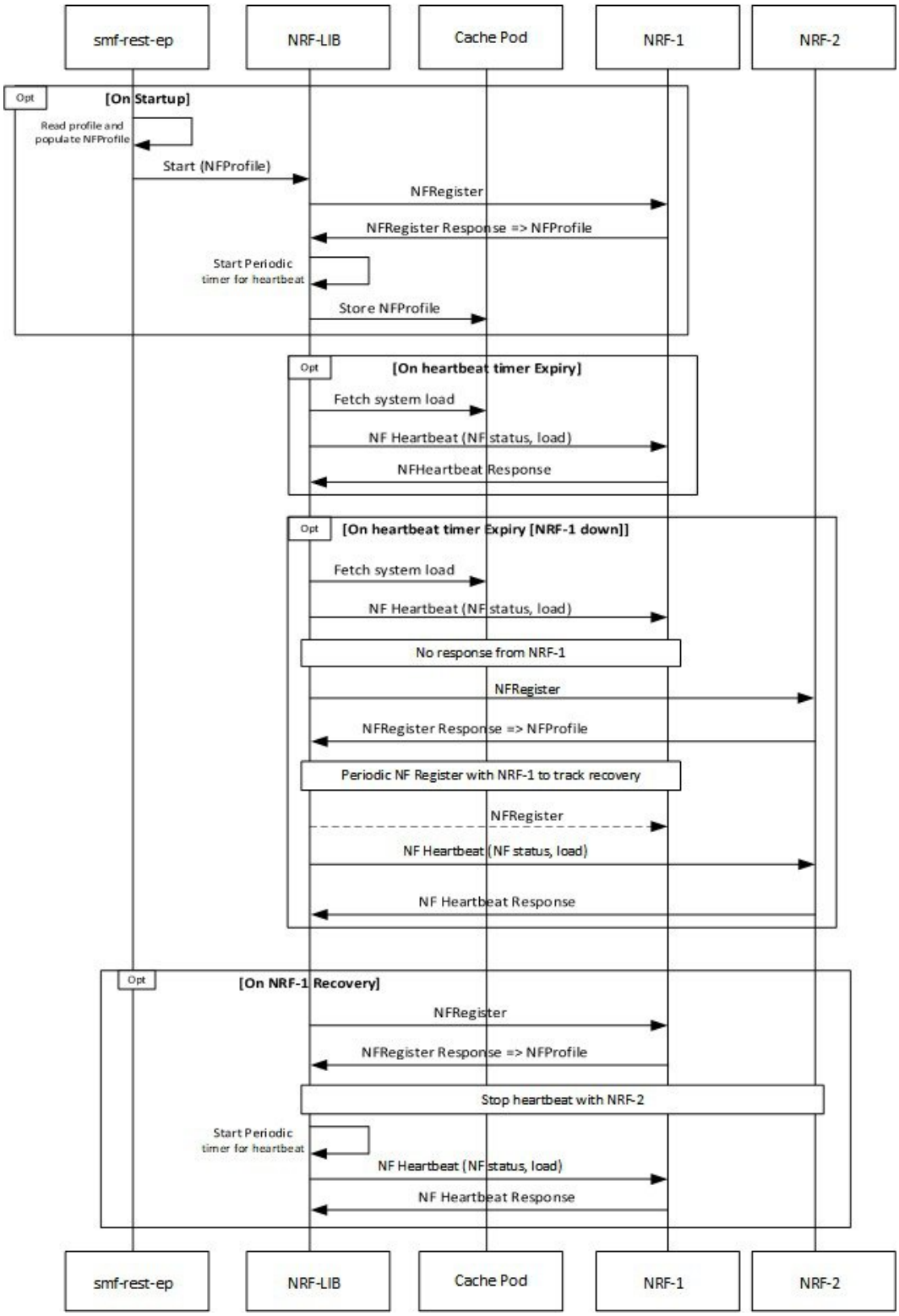
Note NF Reregistration (default behavior) on failover and fallback is configuration driven. When NRF 2 detects that the SMF has stopped sending heartbeats, it checks from NRF 1 if it has received SMF registration by using discovery with SMF instance ID.

As the management and discovery endpoint groups are separate, the Registration based operation status check is not used for NRF failure handling during NF discovery. During NF discovery, the configured NRF endpoints within the group are attempted in the priority order.

Call Flow

The following diagram shows the basic NF management call flow covering the NF registration, NF management and the NRF failure handling.

Figure 3: NF Management Call Flow



Configuring NRF Failure Handling

This section provides the NRF configurations that are required for the failure handling of other NFs.

Configuring the Failure Handling Template

To configure the failure handling template, use the following sample configuration:

```
config
  profile nf-client-failure { nf-type { amf | chf | nrf | pcf | udm }
    profile failure-handling failure_handling_name
    end
```

NOTES:

- **profile nf-client-failure { nf-type { amf | chf | nrf | pcf | udm }**: Specify the required NF client failure profile and provide the local configuration support for the following configured NFs:
  - **amf**: Enable the AMF local configuration
  - **chf**: Enable the CHF local configuration
  - **nrf**: Enable the NRF local configuration
  - **pcf**: Enable the PCF local configuration
  - **udm**: Enable the UDM local configuration

    For example, if the NF type selected is **udm**, then this command enables the UDM local configuration. The same approach applies for the other configured NFs.

- **profile failure-handling** *failure_handling_name*: Specify the failure handling profile name. For example, "udmFail".

Configuration Example

The following is an example configuration of NRF failure handling.

```
group nf-mgmt NFMGMT1
   nrf-mgmt-group nrf-nfmgmt-grp
      failure-handling-profile FHNRF
         locality       LOC1
            heartbeat interval 50
         exit

profile nf-client-failure nf-type nrf
   profile failure-handling FHNRF
      service name type nrf-nfm
         responsetimeout 2300
            message type NRFRegistration
               failover-enabled true
               status-code httpv2 400,500
            action retry
         exit
         status-code httpv2 401,504
         action retry-next
      exit
   exit
message type NFUpdate
   failover-enabled true
```

```
      status-code httpv2 400,503
   action retry
exit
status-code httpv2 411,500
   action retry-next
   exit
exit
message type Heartbeat
      re-registration-enabled true
      status-code httpv2 400,429
   action retry
exit
         status-code httpv2 411,500
            action retry-next
          exit
        exit
     exit
  exit
exit
```

When an AMF failure occurs, use the following example configuration for the range of error codes with the same retry-action and retry-count in the failure handling template.

Note You can use similar configuration during the failure of other NFs.

```
profile nf-client-failure nf-type amf
 profile failure-handling FH1
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 100,200,300,400-410
     retry  4
     action continue
    exit
   exit
  exit
 exit
 profile failure-handling FH2
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 401
     retry  4
     action continue
    exit
   exit
  exit
 exit

profile failure-handling FH3
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 250-260
     retry  4
     action continue
    exit
   exit
  exit
 exit
 profile failure-handling FH4
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 100,200,300,400-410
```

```
    action continue
   exit
  exit
 exit
exit
profile failure-handling FH5
 service name type namf-loc
  message type AmfCommEBIAssignment
   status-code httpv2 150,160,170-175
    action continue
   exit
  exit
 exit
exit
```

The following configuration is an example of the failure template mapping to DNN.

```
profile dnn intershat
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV4V6 ]
 upf apn intershat
exit
```

The following configuration is an example of the failure template mapping to SMF.

```
profile smf smf1
 node-id          12b888e1-8e7d-49fd-9eb5-e2622a57722
 locality         LOC1
 bind-address ipv4 209.165.202.129
 bind-port        8008
 fqdn             example.com.apn.epc.mnc456.mcc123
 plmn-id mcc 123
 plmn-id mnc 456
exit

profile network-element amf amf1
 nf-client-profile        AMF-L1
 failure-handling-profile FH1
 query-params [ target-nf-instance-id ]
exit
profile network-element pcf pcf1
 nf-client-profile        PCF-L1
 failure-handling-profile FH1
exit
profile network-element udm udm1
 nf-client-profile        UDM-L1
 failure-handling-profile FH1
exit
profile network-element chf chf1
 nf-client-profile        CHF-L1
 failure-handling-profile FH2
exit
end
```

Configuring Failure Handling Actions

To configure the failure retry and action for each NF service and the different message types, use the following sample configuration:

```
config
  profile nf-client-failure { nf-type { amf | chf | pcf | udm }
  profile failure-handling failure_handling_name
     service name type service_type
        message type message_type
           status-code httpv2 status_code
           retry retry_count
           action { continue | retry-and-continue | retry-and-terminate |
 terminate }
           end
```

NOTES:

- **service name type** *service_type*: Specify the configured NF service types and provide the local configuration support for the following configured NFs. The service types vary depending on the configured service.

  The AMF service supports the following service types:

  - **namf-comm**
  - **namf-evts**
  - **namf-loc**
  - **namf-mt**

  The CHF service supports the following service types:

  - **nchf-convergedcharging**
  - **nchf-spendinglimitcontrol**

  The NRF service supports the following service type:

  - **nrf-nfm**

  The PCF service supports the following service types:

  - **npcf-am-policy-control**
  - **npcf-bdtpolicycontrol**
  - **npcf-eventexposure**
  - **npcf-policyauthorization**
  - **npcf-smpolicycontrol**
  - **npcf-ue-policy-control**

  The UDM service supports the following service types:

  - **nudm-ee**
  - **nudm-pp**
  - **nudm-sdm**
  - **nudm-ueau**

- **nudm-uecm**

For example, if the *service_type* that is selected is **nudm-sdm**, then this command enables the UDM local configuration. The same approach applies for the other configured NFs.

- **message type** *message_type*: Specify the configured NF message type and provide the local configuration support for the configured NF.

  The message types vary depending on the configured profile and service type.

- **status code httpv2** *status_code* : Specify the status code for the retry and action for the NF service. Currently only "http" status code is provided. *status_code* must be an integer in the range of 0–599.

- **retry** *retry_count*: Specify the number of times the NF service must retry before proceeding with the action. *retry_count* must be an integer in the range of 1–10.

- **action**: Specify the action. The supported actions are:

  - **continue**: Specify to continue the session without any retry. The retry count configuration is invalid with this action.

  - **retry-and-continue**: Specify to retry as per the configured retry count and continue the session.

  - **retry-and-terminate**: Specify to retry as per the configured retry count and terminate the session in case all retry fails.

  - **terminate**: Specify to terminate the session without any retry. The retry count configuration is invalid with this action.

    The retry and action for a message send is picked based on the first sent status code failure. A different status code in the retry does not lead to picking a new retry count and action.

The following table provides a sample of the configured profile, service, and message type options.

Profile	Service Type	Message Type Options
amf	namf-comm	• AmfCommEBIAssignment • AmfCommN1N2MessageTransfer • AmfCommSMStatusChangeNotify • range
chf	nchf-convergedcharging	• ChfConvergedchargingCreate • ChfConvergedchargingDelete • ChfConvergedchargingUpdate • range
nrf	nrf-nfm	• Heartbeat • NFUpdate • NRFRegistration

Profile	Service Type	Message Type Options
pcf	npcf-am-policy-control	• PcfSmpolicycontrolCreate • PcfSmpolicycontrolDelete • PcfSmpolicycontrolUpdate • range
udm	nudm-sdm	• UdmRegistrationReq • UdmSdmGetUESMSubscriptionData • UdmSdmSubscribeToNotification • UdmSubscriptionReq • UdmUecmRegisterSMF • UdmUecmUnregisterSMF • UdmSdmUnsubscribeToNotification • range

Note The example does not cover all the message options that are provided for each profile and service type.

Configuring NRF Registration Failover Option

The NRF Registration Failover feature enables the user to configure the retry actions for every error code, which occurs during the NRF interactions with SMF and other NFs.

After trying all the hosts or endpoints, the next action is decided based on the failover options, which are configured for the error codes.

To configure the NRF failover functionality, use the following sample configuration:

```
config
  profile nf-client-failure nrf
    profile failure-handling failure_handling_name
      service name type nrf-nfm
        message type { Heartbeat [ re-registration-enabled { false |
true } ] | NFUpdate [ failover-enabled { false | true } ] | NRFRegistration
 [ failover-enabled { false | true } ] }
          status-code httpv2 status_code  action { retry | retry-next }
          end
```

NOTES:

- **message type { Heartbeat [re-registration-enabled { false | true }] | NFUpdate [failover-enabled { false | true }] | NRFRegistration [failover-enabled { false | true }] }** : Specify the NRF message type and enable failover functionality.

The failover options for the NRF messages are as follows:

- **NRFRegistration** or **NFUpdate**
  - **true**—After trying all the hosts or endpoints in an NRF, the system selects the next available NRF.
  - **false**—After trying all the hosts or endpoints in an NRF, the system does not select the next available NRF.

  Spawning of backup routine is only available for those NRFs in which the endpoints have been tried.
- **Heartbeat**
  - **true**—After trying all the hosts or endpoints in an NRF, if the start reregistration option is enabled, then the system starts the reregistration process for the NF clients.
  - **false**—After trying all the hosts or endpoints in an NRF, the system continues the heart beat routine with the same registered NRF.

- **status-code httpv2** *status_code* **action { retry | retry-next }**: Specify the status code and retry action for the NRF service. Currently only "http" status code is provided. *status_code* must be an integer in the range of 0–599.
  - **retry**—The system attempts one more retry to the same endpoint or host.
  - **retry-next**—The system does not retry the same endpoint or host, but it attempts the retry action to the next available endpoint or host.
- The error handling for NF Registration, NF Heartbeat, and NF Update is based on status codes. This functionality is not available for subscription and NF Deregister messages. The user can configure the max retry-count for the subscription and NF Deregister messages by using the endpoint configuration available in the **group nrf management** CLI. The system attempts the retry action based on that configuration.
- The failover-enabled option is applicable for the NF Registration and NF Update messages.
- The reregistration-enabled option is applicable for the NF Heartbeat message.
- The failover-enabled or reregistration-enabled options are not applicable for the NF Deregister message.
- The failover and reregistration options are enabled by default.

Configuring Failure Handling in Network Element Profile

To configure the failure handling in the network element profile, use the following sample configuration:

```
config
  profile network-element { { amf | chf | pcf | udm } nf_profile_name }
    failure-handling-profile profile_name
    end
```

NOTES:

- **failure-handling-profile** *profile_name*: Specify the NRF failure handling network profile for the configured NF type. *profile_name* must be an alphanumeric string representing the corresponding NRF failure handling network profile name.

Configuration Example

The following is an example configuration.

```
group nf-mgmt NFMGMT1
   nrf-mgmt-group nrf-nfmgmt-grp
      failure-handling-profile FHNRF
         locality       LOC1
            heartbeat interval 50
         exit

profile nf-client-failure nf-type nrf
   profile failure-handling FHNRF
      service name type nrf-nfm
         responsetimeout 2300
            message type NRFRegistration
               failover-enabled true
               status-code httpv2 400,500
            action retry
         exit
         status-code httpv2 401,504
         action retry-next
      exit
   exit
message type NFUpdate
   failover-enabled true
      status-code httpv2 400,503
   action retry
exit
status-code httpv2 411,500
   action retry-next
   exit
exit
message type Heartbeat
      re-registration-enabled true
      status-code httpv2 400,429
   action retry
exit
         status-code httpv2 411,500
            action retry-next
          exit
        exit
     exit
  exit
exit
```

When an AMF failure occurs, use the following example configuration for the range of error codes with the same retry-action and retry-count in the failure-handling template.

```
profile nf-client-failure nf-type amf
 profile failure-handling FH1
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 100,200,300,400-410
     retry  4
     action continue
    exit
   exit
  exit
```

```
 exit
 profile failure-handling FH2
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 401
     retry  4
     action continue
    exit
   exit
  exit
 exit

profile failure-handling FH3
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 250-260
     retry  4
     action continue
    exit
   exit
  exit
 exit
 profile failure-handling FH4
  service name type namf-comm
   message type AmfCommEBIAssignment
    status-code httpv2 100,200,300,400-410
     action continue
    exit
   exit
  exit
 exit
 profile failure-handling FH5
  service name type namf-loc
   message type AmfCommEBIAssignment
    status-code httpv2 150,160,170-175
     action continue
    exit
   exit
  exit
 exit
exit
```

Verifying the NRF Failure Handling

NF Management Failure Handling

The following is an example of management NRF endpoint configuration.

```
product smf# show running-config group nf-mgmt
group nf-mgmt MGM
 nrf-mgmt-group mgmt_group
 locality       LOC1
exit
product smf# show running-config group nrf mgmt
group nrf mgmt mgmt_group
 service type nrf nnrf-nfm
  endpoint-profile epprof
   uri-scheme http
   endpoint-name EP1
    priority 2
    primary ip-address ipv4 209.165.200.237
    primary ip-address port 8082
```

```
    secondary ip-address ipv4 209.165.200.238
    secondary ip-address port 8082
   exit
   endpoint-name EP2
    priority 10
    primary ip-address ipv4 209.165.200.237
    primary ip-address port 8082
    secondary ip-address ipv4 209.165.200.238
    secondary ip-address port 8082
   exit
  exit
 exit
exit
product smf#
```

In the sample configuration, EP1 is the higher priority endpoint name as its priority is lesser than EP2 (2 against 10). On bringing up, SMF sends NF registration to primary ip:port of EP1 [209.165.200.235:8082]. SMF uses secondary ip:port of EP1 if the primary is down. SMF performs a failover of endpoint to EP2 only if all ip:port of EP1 is down.

On successful registration with EP1 primary, SMF starts heartbeat with EP1 primary. If EP1 primary goes down, SMF detects the same by missing heartbeat response. On detecting that the EP1 primary is down, SMF sends heartbeat to EP1 secondary without reregistration. Also, it periodically sends NF heartbeat to EP1 primary to detect if it has recovered.

If SMF detects that EP1 primary and secondary is down, SMF performs a failover of endpoint to EP2. After the successful failover to EP2 primary, it sends reregistration (default behavior). It is assumed that all the endpoints with an endpoint name shares the same database and so reregistration is only supported when the failover is across endpoint names. In this case, EP1 primary and secondary share the same database. Similarly, EP2 primary and secondary share another database. On failover to EP2 primary, periodic NF registration is sent to primary of the EP1 only (to detect recovery).

Whenever a higher priority endpoint name is detected to be recovered, SMF falls back to the recovered IP:Port. For example, the current active NRF endpoint is EP2 primary and SMF detects that EP1 primary has recovered, then SMF performs reregistration with EP1 primary (default behavior) and stops heartbeat on EP2 primary.

Within endpoint NF heartbeat is used to track operational status. Across endpoints, registration is used to track the operational status. Request message timeout, RPC error, and HTTP response codes 408, 429, 500, 501, 502, 503 are considered as failure to move to the next NRF.

NF Discovery Failure Handling

The following is an example of discovery NRF endpoint configuration.

```
product smf# show running-config profile nf-pair nf-type UDM
profile nf-pair nf-type UDM
 nrf-discovery-group others_group
 locality client LOC1
exit
product smf# show running-config group nrf discovery others_group
group nrf discovery others_group
 service type nrf nnrf-disc
  endpoint-profile ep1
   capacity 30
   priority 50
   uri-scheme http
   endpoint-name ED1
    priority 56
    primary ip-address ipv4 209.165.201.19
    primary ip-address port 8082
    secondary ip-address ipv4 209.165.201.20
```

```
    secondary ip-address port 8082
    exit
   endpoint-name ED2
    priority 10
    primary ip-address ipv4 209.165.201.21
    primary ip-address port 8082
    secondary ip-address ipv4 209.165.201.22
    secondary ip-address port 8082
   exit
  exit
 exit
exit
product smf#
```

In the sample configuration, ED1 is the higher priority endpoint name as its priority is lesser than ED2 (2 against 10). Whenever a NRF discovery is required, primary ip:port of ED1 [209.165.201.19:8082] is attempted. SMF uses secondary ip:port of ED1 if the primary is down. SMF performs a failover of endpoint to ED2 only if all ip:port of ED1 is down. There is no state maintained regarding NRF discovery failure with any NRF endpoint. The SMF always starts with ED1 primary and falls back to ED1 secondary in case of failure, followed by ED2 primary, and so on.

Policy Control Function Failure Handling

Feature Description

The SMF utilizes the NF Failover support to achieve the PCF failover functionality.

The NF Failover feature supports the following functionality:

- Multiple endpoints for a service as primary and secondary endpoints. The endpoints can be configured using the NRF Client Profile configuration and the NRF Failure Profile configuration.
- Failure behavior based on:
  - Message Type
  - HTTP Status Codes in the response messages

Once the PCSCF profile is configured, the SMF assumes that the PDU activation is for the IMS and IMS requires PCF, then SMF rejects the PDU regardless of the failure handling configuration.

- The SMF ignores the failure handling configuration and rejects the PDU creation if there is “pcscf-profile” under profile dnn.
- Operator can’t configure the following parameters under the same “profile dnn”:
  - pcf-interaction false
  - pcscf-profile

How it Works

This section describes how the SMF handles message-level failures and the corresponding HTTP status code-based failures.

The SMF initiates the following messages:

- PcfSmpolicycontrolCreate
- PcfSmpolicycontrolUpdate
- PcfSmpolicycontrolDelete

During the PDU session lifecycle, the SMF exchanges the messages at various stages with the PCF. Depending on the HTTP status code configured in the NRF failure profile, the SMF takes one of the following actions:

- Ignore
- Continue
- Terminate

Table 10: Relationship between PCF Failover Messages and Actions

	PcfSmpolicy controlCreate	**PcfSmpolicy controlUpdate**	**PcfSmpolicy controlDelete**
Ignore	Continue with locally configured/UDM-provided policy parameters. **Note** Do not contact PCF for subsequent messages. **PCF-Interaction Status: OFF**	Continue with currently available snapshot of policy parameters. Contact PCF for subsequent messages. **PCF-Interaction Status: ON**	Ignore the current failure and delete the session. **PCF-Interaction Status: Session deleted**
Continue	Continue with locally configured/UDM-provided policy parameters. **Note** Do not contact PCF for subsequent messages. **PCF-Interaction Status: OFF**	Continue with currently available snapshot of policy parameters. **Note** Do not contact PCF for subsequent messages. **PCF-Interaction Status: OFF**	Ignore the current failure and delete the session. **PCF-Interaction Status: Session deleted**
Terminate	Terminate the session.	Terminate the session.	Terminate the session.

PCF Interaction Status

This feature supports the following status messages for SMF-initiated and PCF-initiated messages:

- **PCF-Interaction Status: ON**

  SMF-initiated messages—The SMF continues to initiate the messages towards the PCF whenever the criteria is met.

  PCF-initiated messages—The SMF continues to accept all the messages initiated from the PCF towards the SMF.

- **PCF-Interaction Status: OFF**

SMF-initiated messages—The SMF does not initiate or send the messages towards the PCF whenever the criteria is met. The SMF treats the PCF as if it is not available and continues further actions.

PCF-initiated messages—There are two messages initiated by the PCF.

- SmPolicyUpdateNotifyReq: On receiving this message, the SMF sends a 404 error code in response and cleans up the session and does not send the Delete Request to the PCF.



Note The SMF also sends FIVEGSM_CAUSE value as **REACTIVATION REQUESTED** in the FIVEG_PDU_SESSION_RELEASE_COMMAND to UE for 5G. In case of 4G, the SMF sends cause **REACTIVATION REQUESTED** in DELETE BEARER REQUEST message to the S-GW.

- SmPolicyAssociationTerminationReq—On receiving this message, the SMF sends a success response and cleans up the session. As part of this interaction, the SMF sends a Delete Request to the PCF.



Note This is an exception when the PCF-Interaction Status is set to OFF.

Configuring the PCF Failure Handling Feature

This section desribes how to configure the PCF Failure Handling feature.

Configuring the PCF Failure Handling feature involves the following steps:

- Configuring the PCF Failure Handling Profile, on page 34
- Configuring the Association of Failure Handling Profile, on page 35
- Configuring Secondary and Tertiary IP Addresses, on page 35

Configuring the PCF Failure Handling Profile

To configure the PCF failure handling profile with action, use the following sample configuration:

```
config
  profile nf-client-failure nf-type pcf
    profile failure-handling fhprofile_name
      service name type servicename_type
        message type messagetype_value
          status-code httpv2 status_code
          action { continue | retry-and-continue | retry-and-ignore |
retry-and-terminate } retry retry_value
          exit
```

NOTES:

- **profile failure-handling** *fhprofile_name*: Specify the failure handling profile name.
- **service name type** *servicename_type*: Specify the PCF service name type. *servicename_type* can be one of the following values:

  - npcf-am-policy-control
  - npcf-bdtpolicycontrol
  - npcf-eventexposure
  - npcf-policyauthorization
  - npcf-smpolicycontrol
  - npcf-ue-policy-control

- **message type** *messagetype_value*: Specify the message type. *messagetype_value* can be one of the following values:
  - PcfAmfPolicyControlCreate
  - PcfSmpolicycontrolCreate
  - PcfSmpolicycontrolDelete
  - PcfSmpolicycontrolUpdate

- **status-code httpv2** *status_code*: Specify the HTTPv2 status code. *status_code* must be an integer in the range of 0–599, separated by either '-' or ','.
- **action { continue | retry-and-continue | retry-and-ignore | retry-and-terminate } retry** *retry_value*: Specify the action and the number of retry attempts. *retry_value* must be an integer in the range of 1–10.

Configuring the Association of Failure Handling Profile

To configure the association of FH profile in PCF, use the following sample configuration:

```
config
  profile network-element pcf pcf_profile_name
    nf-client-profile nf_profile_name
    failure-handling-profile fh_profile_name
    exit
```

NOTES:

- **nf-client-profile** *nf_profile_name*: Specify the NF client profile name.
- **failure-handling-profile** *fh_profile_name*: Specify the failure handling profile name.

Configuring Secondary and Tertiary IP Addresses

To configure the secondary and tertiary IP addresses, use the following sample configuration:

```
config
  profile nf-client nf-type pcf
    pcf-profile pcfprofile_name
      locality locality_name
        service name type npcf-smpolicycontrol
          endpoint-profile endpointprofile_name
            endpoint-name endpoint_name
              primary ip-address { ipv4 primary_ipv4_address | ipv6
```

```
primary_ipv6_address | port primary_port_number }
                    secondary ip-address { ipv4 secondary_ipv4_address | ipv6
secondary_ipv6_address | port secondary_port_number }
                    tertiary ip-address { ipv4 tertiary_ipv4_address | ipv6
tertiary_ipv6_address | port tertiary_port_number }
                    end
```

NOTES:

- **primary ip-address ipv4** *primary_ipv4_address*: Specify the IPv4 address of primary endpoint.
- **primary ip-address ipv6** *primary_ipv6_address*: Specify the IPv6 address of primary endpoint.
- **primary ip-address port** *primary_port_number*: Specify the port number of primary endpoint.
- **secondary ip-address ipv4** *secondary_ipv4_address*: Specify the IPv4 address of secondary endpoint.
- **secondary ip-address ipv6** *secondary_ipv6_address*: Specify the IPv6 address of secondary endpoint.
- **secondary ip-address port** *secondary_port*: Specify the port number of secondary endpoint.
- **tertiary ip-address ipv4** *tertiary_ipv4_address*: Specify the IPv4 address of tertiary endpoint.
- **tertiary ip-address ipv6** *tertiary_ipv6_address*: Specify the IPv6 address of tertiary endpoint.
- **tertiary ip-address port** *tertiary_port_number*: Specify the port number of tertiary endpoint.

OAM Support for PCF Failure Handling

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

Bulk Statistics Support

This feature supports the following statistics:

- PcfSmpolicyControlCreate
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses
- PcfSmPolicyControlUpdate
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses
- PcfSmpolicyControlDelete
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses

- PolicyUpdateNotifyReq
  - Number of accepted requests
  - Number of rejected requests
  - Number of skipped requests
- PolicyDeleteReq
  - Number of accepted requests
  - Number of rejected requests
  - Number of skipped requests
- PolicyUpdateRequest
  - Number of accepted requests
  - Number of rejected requests
  - Number of skipped requests
- Gauge counter for number of subscribers with policy type local/pcf.

Unified Data Management Failure Handling

Feature Description

The Unified Data Management (UDM) is responsible for primarily storing the subscriber data, which SMF accesses for managing the user sessions on the network.

The UDM failure handling support on SMF introduces a new failure handling template (FHT) profile. This profile is associated with the UDM profile in SMF.

The FHT template provides flexibility for SMF to fine tune its interactions with UDM over N10 for the sessions. It supports the SMF to handle the HTTP status codes in response from UDM for both new and existing sessions.

The NF failover support is available in the SMF using the NF Client profile configuration and the NF failure profile configuration. This feature supports the following functionality:

- Configure multiple endpoints for a service as primary and secondary endpoints.
- Specify the failure handling behavior based on:
  - Message Type
  - HTTP Status Codes in the response messages

How it Works

The SMF utilizes the NF Failover to achieve the UDM failover support functionality. This section provides information on how the SMF handles message-level failures and the corresponding HTTP status code-based failures.

The SMF initiates the following messages:

- UE-Connection-Management (UE-CM)
  - Nudm_UECM_Registration
  - Nudm_UECM_DeRegistration
- UE-Subscription-Management (UE-SDM)
  - Nudm_SDM_Get
  - Nudm_SDM_Subscribe
  - Nudm_SDM_Unsubscribe

During the PDU session lifecycle, the SMF exchanges the preceding messages at various stages with the UDM. Depending on the HTTP status code configured in the NF failure profile, the SMF takes one of the following actions:

- Ignore
- Continue
- Terminate

The SMF provides the following actions to attempt the same request to other available UDM servers.

- retry-and-terminate
- retry-and-ignore
- retry-and-continue

When all the retry attempts fail, the SMF takes the appropriate failure handling action. For example, if the FH action is retry-and-terminate, the SMF terminates the call after all the attempts fail.



Note The SMF allows dynamic changes to the failure handling template configuration. Any changes to the configuration apply only to the new calls.

Table 11: Relationship between N10 Messages and Failover Actions

Scenario	Service	Message	Condition	Action	Success Response	Handling of Failure Response		
						Terminate	Continue	Ignore
PDU Session Creation procedures in 5G, 4G, and Wi-Fi Inter-RAT Handover procedures	UECM	Nudm_ UECM _Registration	If the Nudm UECM Registration is not done and the access type is not 4G	Send the message	Mark the Registration is successful	Terminate call	Continue call	Continue call
		Nudm _UECM_ DeRegistration	If the Nudm UECM Registration is done	Send the message	No action	Terminate call	Terminate call	Terminate call
PDU Session Creation procedures in 5G, 4G, and Wi-Fi	SDM	Nudm_SDM _Get	If skipping the subscription fetch config is not enabled	Send the message	Mark the subscription fetch is successful	Terminate call	Continue call	Continue call
		Nudm_ SDM _Subscribe	If the subscription fetch is successful	Send the message	No action	Terminate call	Continue call if the subscription is not done	Continue call if the subscription is not done
PDU Session Release procedures in 5G, 4G, and Wi-Fi	SDM	Nudm_SDM _Unsubscribe	If the subscription fetch is successful and the registration is not done	Send the message	No action	Terminate call	Continue call	Continue call

- **Terminate**: The SMF terminates the call in any message type.
- **Continue**: The SMF ignores the current failure and skips the subsequent interaction for the other messages in the same service group.
- **Ignore**: The SMF ignores failure only for the current interaction and proceeds with the call. The SMF processes the subsequent message interaction.
- Perform UDM subscription fetch only during the session establishment in EPS and NR network.

  If the UDM subscription fetch fails and the FH action is ‘Ignore’ or the configuration to skip subscribe-to-notification is enabled, then the SMF skips the subscribe-to-notification interaction.

- When the UDM failure handling template is not configured, the default failure handling action is 'Terminate'.

Configuring UDM Failure Handling Feature

This section describes how to configure the UDM Failure Handling feature.

Configuring the UDM Failure Handling feature involves the following steps:

- Configuring UDM Failure Handling Profile, on page 40
- Configuring Association of FH profile, on page 40
- Configuring Secondary and Tertiary IP Addresses, on page 41

Configuring UDM Failure Handling Profile

Use the following sample configuration to configure the UDM failure handling profile with action.

```
config
   profile nf-client-failure nf-type udm
      profile failure-handling fh_profile_name
         service name type { nudm-ee | nudm-pp | nudm-sdm | nudm-ueau
         | nudm-uecm }
        message type { UdmRegistrationReq | UdmSdmGetUESMSubscriptionData

         | UdmSdmSubscribeToNotification | UdmSubscriptionReq
         | UdmUecmRegisterSMF | UdmUecmUnregisterSMF |
         UdmSdmUnsubscribeToNotification }
            status-code httpv2 0
               action { continue | retry-and-continue | retry-and-ignore
             | retry-and-terminate | terminate }
             end
```

Configuring Association of FH profile

To configure the association of FH profile in the UDM, use the following sample configuration:

```
config
   profile network-element udm udm_profile_name
      nf-client-profile nf_profile_name
      failure-handling-profile fh_profile_name
      failure-handling-profile-rat nr
         failure-handling-profile fh_profile_name
         exit
```

NOTES:

- **failure-handling-profile-rat nr**: Specify the failure handling profile specific to RAT type.
- **failure-handling-profile** *fh_profile_name*: Specify the failure handling network profile name. *fh_profile_name* must be a string.

Verifying the RAT-based FH Profile

This section describes how to verify RAT-based FH profile in the UDM.

Use the **show running-config profile network-element udm** *udm_profile_name* command to verify the feature configuration details.

The following is an example output.

```
nf-client-profile UP1
   failure-handling-profile FH1
   failure-handling-profile-rat nr
      failure-handling-profile FH4
      exit
exit
```

In this example, FH1 is the default failure handling profile. However, if the RAT type is configured as **nr**, then the failure handling profile FH4 is used.

Configuring Secondary and Tertiary IP Addresses

To configure secondary and tertiary IP addresses, use the following sample configuration:

```
config
   profile nf-client nf-type udm
      udm-profile udmprofile_name
         locality LOC
            service name type { nudm-ee | nudm-pp | nudm-sdm | nudm-ueau
 | nudm-uecm }
               endpoint-profile epprofile_name
                  endpoint-name endpoint_name
                     primary ip-address { ipv4 primary_ipv4_address | ipv6
primary_ipv6_address | port primary_port_number }
                    secondary ip-address { ipv4 secondary_ipv4_address | ipv6
 secondary_ipv6_address | port secondary_port_number }
                     tertiary ip-address { ipv4 tertiary_ipv4_address | ipv6
 tertiary_ipv6_address | port tertiary_port_number }
                     end
```

NOTES:

- **primary ip-address ipv4** *primary_ipv4_address*: Specify the IPv4 address of primary endpoint.
- **primary ip-address ipv6** *primary_ipv6_address*: Specify the IPv6 address of primary endpoint.
- **primary ip-address port** *primary_port_number*: Specify the port number of primary endpoint.
- **secondary ip-address ipv4** *secondary_ipv4_address*: Specify the IPv4 address of secondary endpoint.
- **secondary ip-address ipv6** *secondary_ipv6_address*: Specify the IPv6 address of secondary endpoint.
- **secondary ip-address port** *secondary_port*: Specify the port number of secondary endpoint.
- **tertiary ip-address ipv4** *tertiary_ipv4_address*: Specify the IPv4 address of tertiary endpoint.
- **tertiary ip-address ipv6** *tertiary_ipv6_address*: Specify the IPv6 address of tertiary endpoint.
- **tertiary ip-address port** *tertiary_port_number*: Specify the port number of tertiary endpoint.

Configuring Response Timeout Parameter

To configure response timeout for fail-open support over the UDM interface (N10), use the following sample configuration:

```
config
  profile network-element udm udm_profile_name
    response-timeout timeout_value
    exit
```

NOTES:

- **response-timeout** *timeout_value*: Specify the response timeout in milliseconds. *timeout_value* must be an integer in the range of 1000-30000.

  Default: 4000

Verifying the Response Timeout Configuration

The following is an example configuration.

```
[unknown] smf# show running-config profile network-element udm
profile network-element udm udm1
nf-client-profile UP1
failure-handling-profile FH4
query-params [ dnn ]
response-timeout 2000
exit
[unknown] smf#
```

Statistics

The following statistics are supported for all the UDM message status with status as Attempted/Success/Skipped/Failed for all UDM services and message combination.

```
udm_msg_processing_status{app_name="SMF",cluster="Local",data_center="DC"
,instance_id="1",msg_status="attempted",rat_type="nr",service_name="smfservice",
udm_end_point="",udm_msg="UdmSmSubscription”} 1
```

```
udm_msg_processing_status{app_name="SMF",cluster="Local",data_center="DC",
instance_id="1",msg_status="skipped",rat_type="nr",service_name="smfservice",udm_end_point="",
udm_msg="UdSmSubscription"} 1
```

OAM Support for UDM Failure Handling Feature

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

Bulk Statistics Support

The SMF maintains the following statistics in support of the UDM Failure Handling feature.

- Nudm_UECM_Registration
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses

- Nudm_UECM_DeRegistration
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses
- Nudm_SDM_Get
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses
- Nudm_SDM_Subscribe
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses
- Nudm_SDM_Unsubscribe
  - Number of ignore responses
  - Number of continue responses
  - Number of terminate responses

The "udm_msg_processing_status" statistic in smf-service tracks the number of UDM messages with status as—Attempted, Success, Skipped, and Failed.

For example:

```
udm_msg_processing_status
{app_name="SMF",cluster="Local",data_center="DC",instance_id="1",msg_status="attempted",
rat_type="nr",
service_name="smf-service",udm_end_point="",udm_msg="UdmSmSubscription"} 1
udm_msg_processing_status
{app_name="SMF",cluster="Local",data_center="DC",instance_id="1",msg_status="skipped",
rat_type="nr",
service_name="smf-service",udm_end_point="",udm_msg="UdSmSubscription"} 1
```

User Plane Function failure handling

The User Plane Function (UPF) failure handling allows SMF to manage the error cause code received in the UPF's responses during both new and existing sessions. The SMF provides a failure handling template (FHT) profile for Packet Forwarding Control Protocol (PFCP). This profile is associated with the UPF profile in SMF. The FHT template provides flexibility for SMF to fine tune its interactions with UPFs during sessions.

Supported configurable action in FHT

Based on the error cause codes received in the response from UPF, this feature provides configurable actions:

- ignore
- terminate
- continue
- retry-terminate
- retry-ignore
- retry-continue

How UPF failure handling works

Workflow

The stages in the UPF failure handling process are:

1. During session establishment, if the UPF is congested, it rejects PFCP establishment messages from SMF.
2. UPF sends the cause code in the response message to SMF.
3. SMF checks the user-defined failure handling template. If retry is configured in the template, the SMF reattempts to send PFCP establishment messages to another UPF to reduce call loss.
4. SMF selects a UPF based on configured priority value and capacity (load information from UPF).
5. SMF ignores, terminates, or continues the session based on user-defined actions in the FHT.

Enable UPF failure handling

Perform these steps to enable the UPF failure handling feature:

Procedure

Step 1 Configure the UPF failure handling profile.

Step 2 Configure the failure profile association.

Configure UPF failure handling profile

Configure the UPF failure handling profile in global configuration mode.

Procedure

Step 1 Specify a profile name to create the failure handling profile.

profile failure-handling *fh_profile_name*

Example:

```
[smf] smf# config
[smf] smf(config)# profile failure-handling fh1
```

Step 2 Specify the PFCP message type for which you want to configure failure handling.

interface pfcp message { *N4SessionEstablishmentReq* | *N4SessionModificationReq* | *N4SessionReportReq* | *N4AssociationUpdateReq***}**

Example:

```
[smf] smf# config
[smf] smf(config-failure-handling-fh1)# interface pfcp message N4SessionModificationReq
```

- N4SessionEstablishmentReq—Specify the failure handling message for new sessions and N4 Session Report Request.
- N4SessionModificationReq—Specify the failure handling message for existing sessions.
- N4SessionReportReq—Specify the failure handling message for session report.

  When the N4SessionReportReq keyword is configured, the SMF triggers the Session Deletion Request followed by the rejection of Session Report. The UPF responds to the delete request and clears the session.
- N4AssociationUpdateReq—Specify the failure handling message for Sx-Assoc-Update requests.

Note
UPF reselection is not applicable for message type N4SessionModificationReq because the session is already active on a UPF.

Step 3 Specify the cause codes that the SMF receives in the failure response message from the UPF.

cause-code *cause_ID*

Example:

```
[smf] smf(config-failure-handling-fh1)# cause-code pfcp-entity-in-congestion
```

Table 12: Cause codes

Cause code	Description
• *pfcp-entity-in-congestion* • *peer-overload-reduction*	• *pfcp-entity-in-congestion*—specify this cause code when the UPF is congested. • *peer-overload-reduction*—specify this cause code to clear the session on SMF, while dropping the N4 message due to peer overload reduction received on OCI.
• *system-failure* • *service-not-supported* *session-ctx-not-found*	• *system-failure*—specify this cause code for any internal or system failure during message processing. • *service-not-supported*—specify this cause code when the service is not supported. • *session-ctx-not-found*—specify this cause code when the session context referenced in a message does not exist.

Cause code	Description
• *no-resource-available* • *no-response-received*	• *no-resource-available*—specify this cause code when there is no resource available. • *no-response-received*—specify this option to determine the scenarios where SMF does not receive any response from UPF.
reject	Specify this cause code to handle the cause codes in the failure response message from UPF. The cause codes are not configured using the CLI commands available for this feature.
mandatory-ie -incorrect	Specify this cause code when the received message contains invalid data.
• *74* • *2–255*	• *74*—when SMF receives cause code 74 from UPF and if the cause-code 74 action terminate is configured in the N4 Failure handling profile, the SMF deletes the session. • *2–255*—while specifying multiple cause codes, separate the cause code value using either '-' or ',' or both. For example, cause-code 72-74,76,78-100.

The cause codes 0–255 are supported. The precedence of the cause codes is in this order:

a. Predefined string

b. Number

c. Range

d. Reject

Note
Reject is the default cause code.

Table 13: Configuration for cause code

Cause code	Configuration
0–63	Corresponding template configuration
64–255	Reject

FHT does not support these cause codes that are configured with their default behavior:

- request-reject-unspecified
- cond-ie-missing
- invalid-length
- invalid-fw-policy
- invalid-fteid-alloc-opt
- no-established-pfcp-assoc

- rule-creation-mod-failure

Step 4 Configure the condition and action for failure handling.

action {*ignore* | *retry-terminate* **{ max-retry** *retry_value* **} [condition {** *handover-execution* | *handover-preparation* | *modify* | *idft* | *handover-cancel* **}]**

Example:

```
[smf] smf(config-failure-handling-fh1)# cause-code pfcp-entity-in-congestion action terminate
  condition handover-execution
```

condition [*assoc-setup* | *route-update***] action [***terminate* | *continue* | *ignore* | *retry-ignore* | *retry-continue***] [max-retry [** *1..5***]]**

Example:

```
[smf] smf(config-failure-handling-fh1)# cause-code mandatory-ie-incorrect condition route-update
action retry-continue max-retry 2
```

Configuring the **condition** command is optional except for N4AssociationUpdateReq message type.

Table 14: Configuration matrix

Message type	Applicable action	Applicable cause code	Default behavior
N4Session EstablishmentReq	retry-terminate max-retry *retry_value*	• pfcp-entity-in-congestion • peer-overload-reduction • system-failure • service-not-supported • no-resource-available • no-response-received • reject • 74	terminate
N4Session ModificationReq	terminate	• peer-overload-reduction • mandatory-ie -incorrect • session-ctx-not-found • no-response -received • reject • no-resource -available • 74	continue
N4SessionReportReq	• ignore • terminate	2–255	terminate

Message type	Applicable action	Applicable cause code	Default behavior
N4AssociationUpdateReq	• ignore • terminate • continue • retry-continue max-retry *retry_value* • retry-ignore max-retry *retry_value*	• no-response-received • reject • mandatory-ie -incorrect • system-failure • 2-255	terminate or ignore

Table 15: Configurable actions

Action	Description
retry-terminate max-retry *retry_value*	Specify the number of retry attempts to an alternate UPF. If the retry attempt fails, the session is terminated. The retry value ranges from 1 to 5. The default value is 2. **Note** If all the UPFs are in a congested state, the call fails even if the action is set to continue.
terminate	Specify this action to terminate the session. For N4AssociationUpdateReq, specify this action to release the UPF Association.
ignore	Specify this action to ignore the session. For N4AssociationUpdateReq, specify this action to not to continue with further Association Update request for the same route-update.
continue	Specify this action to not to release the UPF Association for the condition assoc-setup. Continue with further Association Update request for the same route-update in case of condition route-update.
retry-continue max-retry *retry_value*	Specify this action to retry to send the same request before sending the subsequent Association update for the same route-update. The retry value ranges from 1 to 5. The default value is 1.
retry-ignore *retry_value*	Specify this action to retry to send the same request before terminating the subsequent Association update for the same route-update. The retry value ranges from 1 to 5. The default value is 1.

- The SMF allows configuration of one or more conditions only for failure handling of N4SessionModificationReq message type and N4AssociationUpdateReq message type.
- For the N4AssociationUpdateReq message type, the supported conditions are:
  - assoc-setup—Specify this condition during Intial Sx Association Setup procedure.

- route-update—Specify this condition during subsequent chunk addition or deletion as part of route update procedure.

Note
The max-retry can be configured only for the action retry-continue and retry-ignore. Action terminate can be configured only for condition assoc-setup. The actions ignore, retry-ignore and retry-continue can be configured only for the condition route-update. Action continue can be configured for both conditions assoc-setup and route-update.

As part of day-1 behavior, the default count of maximum retransmission at the PFCP interface level due to no response received is three.

Step 5 [Optional] Use the **show running-config profile failure-handling fh1** command to verify the UPF failure handling configuration.

```
[smf] smf# show running-config profile failure-handling FH1
Tue May  27 07:08:52.867 UTC+00:00
profile failure-handling FH1
 interface pfcp
  message N4SessionEstablishmentReq
   cause-code pfcp-entity-in-congestion action retry-terminate max-retry 2
   cause-code system-failure action terminate
   cause-code service-not-supported action terminate
   cause-code no-resource-available action retry-terminate max-retry 3
   cause-code no-response-received action retry-terminate max-retry 1
   cause-code reject action terminate
  exit
  message N4SessionModificationReq
   cause-code mandatory-ie-incorrect action terminate
   cause-code session-ctx-not-found action terminate
   cause-code reject action terminate
  exit
  message N4SessionReportReq
   cause-code 65-77 action terminate
  exit
  message N4AssociationUpdateReq
   cause-code no-response-received condition assoc-setup action terminate
   cause-code no-response-received condition route-update action retry-ignore
   cause-code mandatory-ie-incorrect condition assoc-setup action continue
   cause-code mandatory-ie-incorrect condition route-update action retry-continue
  exit
 exit
```

Configure the failure profile association

This configuration associates the UPF failure profile with the UPF group profile.

Procedure

Step 1 Define the UPF group name in global configuration mode.

profile upf-group *upf_group_name*

Example:

```
[smf] smf# config
[smf] smf(config)# profile upf-group upfg
```

Step 2 Specify the UPF failure profile name and save the configuration.

failure-profile *failure_profile_name*

Example:

```
[smf] smf(config-upf-group-upfg)# failure-profile failure_profile_name
```

Statistics for UPF failure handling

The SMF supports these statistics for UPF failure handling:

- smf_sess_pdn_rel_peer_request_reject—smf_disconnect_stats has been enhanced to include this disconnect reason, applicable for both 4G and Wi-Fi calls.
- smf_sess_pdu_rel_peer_request_reject—smf_disconnect_stats has been enhanced to include this disconnect reason, applicable for 5G calls.
- nodemgr_up_route_stats—nodemgr stats has been enhanced to include an additional field called tx_type. This field indicates whether the current transmission is for a new transmission (NewTx) or for a retransmission (RetTx).