

# NF Discovery and Management


- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- NF Management, on page 2
- NF Discovery, on page 24
- Selection of Alternate AMF, on page 46
- Static Configuration for Peer NF Management, on page 47
- NRF Failure Handling, on page 52
- Discovery Cache Optimization for Range-based Query Parameters, on page 53


# Feature Summary and Revision History

## Summary Data

***Table 1: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | SMF |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Enabled – Always-on |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

## Revision History

***Table 2: Revision History***

| Revision Details | Release |
|---|---|
| As part of associating NRF management and SMF locality to NRF endpoint, added the message handling profile for NRF. | 2023.03.0 |

| Revision Details | Release |
|---|---|
| As part of the IP pool allocation per slice and DNN feature, added configuration procedures for NRF registration and discovery. | 2022.04.0 |
| Added support for the following functionality:<br>• Configurable retry actions for specific error codes.<br>• Flexible options for the retry action associated with an error code.<br>• Httpv2 status code range in the failure handling templates of NFs. | 2021.02.0 |
| Introduced support for individual NF Profile member changes through NRF notification.<br>Included the following new parameters as part of the NRF discovery query:<br>• limit<br>• max-payload-size<br>• requester-snssais | 2020.03.0 |
| First introduced. | Pre-2020.02.0 |

# Feature Description

The Network Function (NF) Repository Function (NRF) supports the following functionality:

- Maintains the NF profile of available NF instances and their supported services;
- Allows other NF instances to subscribe to, and get notified about, the registration in NRF of new NF instances of a given type;
- Supports service discovery function. It receives NF Discovery Requests from NF instances, and provides the information of the available NF instances fulfilling certain criteria (for example, supporting a given service).

# NF Management

## Feature Description

This section describes the NF management procedures and their configurations that SMF supports. These procedures are NF registration, NF deregistration, NF heartbeat, and NF Update. The NF registration, update,

and heartbeat are sent from only one of the rest-ep pods, which is the elected primary node. After this nodes is elected, the instance remains as primary node for the NF management activities till the pod crashes or is removed.

NF management supports dynamic configuration change. With this feature, if the configurations were modified in the middle of the transaction or procedure, the ongoing transactions are not impacted.

The dynamic configuration change feature supports the following:

- NRF transaction or procedure picks a configuration version (v1) and uses the same version until the NRF transaction or procedure completes.
- If you change the configuration during an ongoing NRF transaction, then a new configuration version (v2) is created. However, the new configuration is applied in the new transaction.

The dynamic configuration changes apply to the following data structures:

- NrfFailureProfileSt
- NrfClntProfileSt
- NrfGrpSt
- NrfPairProfileSt
- NrfMgmtGrpSt

# Registration

SMF registers with NRF. During registration with NRF, SMF includes at least one of the addressing parameters, such as FQDN, IPv4 or IPv6 address in the NF profile. Including at least one of the addressing parameters in the NF profile registration is mandatory. If SMF supports "https" uri scheme, then SMF provides FQDN in the NFProfile or NFService.

## Configuring NRF Endpoints Profile Parameters for NF Management

The SMF provides CLI for configuring NRF endpoints for **nnrf-nfm** (NF Management).



**Note** For NF management, you can configure only the **nnrf-nfm** service.

The CLI configuration allows configuring multiple endpoints under each endpoint profile. The SMF uses the priority and capacity parameters to load balance between these endpoints. Primary, secondary, and tertiary hosts [ip:port] can be configured within each endpoint. Both IPv4 and IPv6 addresses can be specified. If both are specified, then the IPv4 address is preferred.

A URI uniquely identifies a resource. In the 5GC SBI APIs, when a resource URI is an absolute URI, its structure is specified as follows:

```
{apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}
```

**Note** In this release of the specification, both HTTP and HTTPS scheme URIs are allowed. See the *3GPP TS 33.501, subclause 13.1* for more information on security of service-based interfaces.

" apiRoot " is a concatenation of the following parts: scheme ("http" or "https")

- fixed string "://"
- authority (host and optional port) as defined in IETF RFC 3986
- an optional deployment-specific string (API prefix) that starts with a "/" character [api-root in CLI]

To configure the NRF endpoints for different services supported by NRF, use the following sample configuration.

```
config
  group nrf mgmt mgmt_name
    service type nrf nnrf-nfm
      endpoint-profile epprofile_name
        priority priority_value
        capacity capacity
        api-root api_string
        api-uri-prefix uri_prefix_string
        uri-scheme { http | https }
        endpoint-name ep_name { capacity capacity | primary ip-address
        { ipv4 ipv4_address | ipv6 ipv6_address | port port_num}
        | secondary ip-address { ipv4 ipv4_address | ipv6 ipv6_address
        | port port_num } | tertiary ip-address { ipv4 ipv4_address
        | ipv6ipv6_address | port port_num }
        | fqdn { name fqdn_name | port port_num } }
        version [ uri-version version_num full version version_num ]
        end
```

**NOTES:**

- **group nrf mgmt** *mgmt_name* : Show the NRF self-management group configurations.
- **api-root** *api_string*: Specify the deployment-specific service API prefix that is used within the { apiRoot }.
- **api-uri-prefix** *uri_prefix_string*: Specify the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **capacity** *capacity*: Specify the profile capacity.
- **endpoint-name** *ep_name* **{ capacity** *capacity* **| primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} }**: Specify the endpoint name. You can configure the primary, secondary, and tertiary hosts (IP: Port) within each endpoint for NF server failover handling. The server failover configuration accepts both the IPv4 and IPv6 addresses. However, the SMF gives preference to the IPv4 address.
  - **capacity** *capacity*: Specify the node capacity for the endpoint. *capacity* must be an integer in the range of 0-65535.

- The endpoint selection for sending the message is based on probabilistic load-balancing algorithm (IETF RFC 2782) using the priority and capacity parameters.
- **primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the primary endpoint IPv4 address, IPv6 address, or port.
- **secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the secondary endpoint IPv4 address, IPv6 address, or port.
- **tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the tertiary endpoint IPv4 address, IPv6 address, or port.
- **priority** *priority_value*: Specify the priority for the service to select the appropriate profile using the load-balancing logic. *priority* must be an integer in the range of 0-65535.
- **fqdn name** *fqdn_name*: Specify the FQDN name.
- **fqdn port** *fqdn_port*: Specify the FQDN port number. If port is not configured, SMF uses the standard port for FQDN, that is 80 for URI scheme HTTP and 443 for URI scheme HTTPS.

- **uri-scheme { http | https }**: Specify the URI scheme as **http** or **https**.
- **version [ uri-version** *version_num* **full version** *version_num* **]**: Specify the api/version. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draft-number>].

### Verifying the NF Endpoint Profile Parameters for NF Management

Use the **show running-config group nrf** command to verify the NF endpoint profile parameters for NF management.

```
show running-config group nrf
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 209.165.200.231
   primary ip-address port 8082
   secondary ip-address ipv4 209.165.200.232
   secondary ip-address port 8082
 exit
   endpoint-name EP2
      priority 10
       primary ip-address ipv4 209.165.200.231
       primary ip-address port 8082
       secondary ip-address ipv4 209.165.200.232
       secondary ip-address port 8082
       fqdn name nrf.cisco.com
       fqdn port 9010
     exit
   exit
 exit
exit
```

# Improved Slice Utilization Using Optimized NRF Registration Request Messages

*Table 3: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| Optimized NRF Registration Request Messages | 2024.02.1 | The size of the **smfInfoList** attribute in the NRF Registration Request message is reduced by grouping its sub-attributes based on the TAC Groups.<br>Grouping the message attributes allows the service providers to scale up their slice deployments to provide additional services.<br>The **smfInfoList** optimization is enabled using the **ie smfinfolist tac-based** command in NRF Message Handling Profile Configuration mode. |

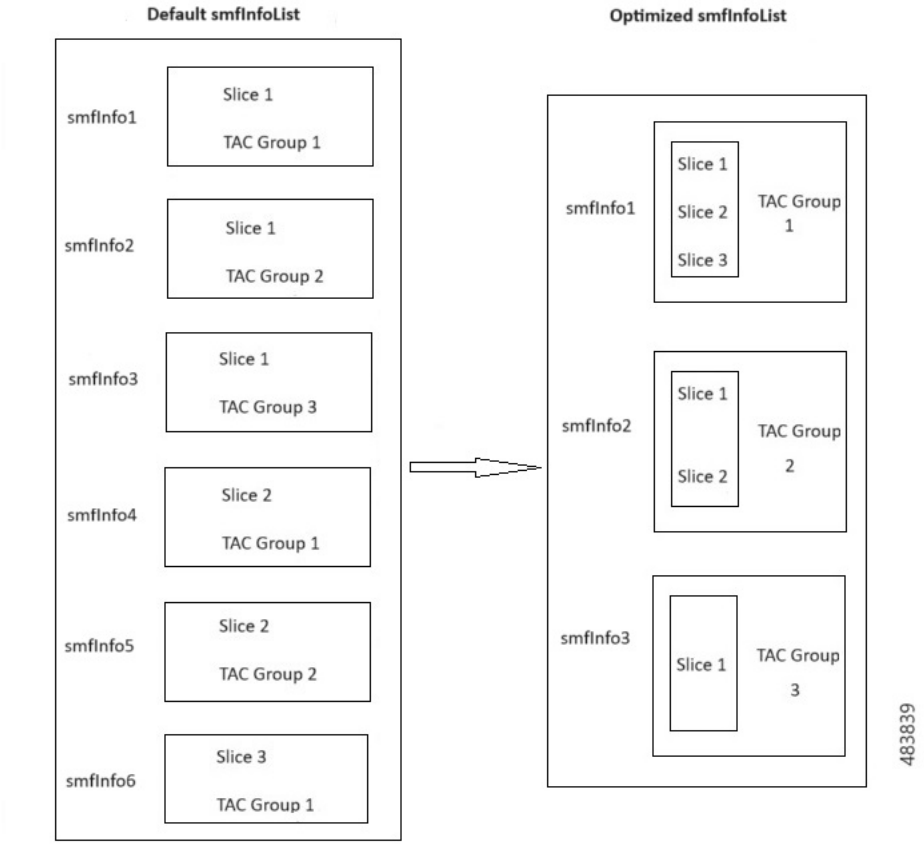
## Overview

SMF sends a registration request message to NRF with the **smfInfoList** attribute containing multiple entries of **smfInfo**. Each **smfInfo** attribute further includes sub-attributes to carry information about slice, DNN, TAC Group, and so on.

By default, one **smfInfo** attribute contains a single slice and a single **TAC Group** information. Even though, slice, DNN, and TAC Group are shared attributes, it takes multiple **smfInfo** entries to accommodate all the shared message information.

This feature groups the **smfInfo** attributes based on the TAC Groups. It categorizes the attributes in such a way that all the common elements corresponding to one TAC Group falls into one **smfInfo** entry. Grouping the common message attributes allows accomodating the existing message attributes in lesser **smfInfo** elements.

*Figure 1: smfInfoList Optimization in NRF Registration Request*




Optimizing the **smfInfoList** reduces the number of **smfInfo** entries required to accommodate the same message information. This in turn reduces the size of **smfInfoList** sent in the NRF Registration Request allowing additional slice details to be added within the limited number of **smfInfo** entries in the **smfInfoList**.

**Note** This feature is backward compatible.

## Enabling smfInfoList Grouping based on TAC Group

To enable the smfInfoList grouping based on TAC Group, use following sample configuration:

```
config
  profile message-handling message_handling_name
    nf-type nrf
      mh-profile mh_profile_name
        service name type { nnrf-at | nnrf-bs | nnrf-nfd | nnrf-nfm }
          message type { nf-deregister | nf-list-retrieval |
nf-profile-retrieval | nf-register | nf-status-notify | nf-status-subscribe
 | nf-status-unsubscribe | nf-updatenf-register }
```

```
ie smfinfolist tac-based
end
```

**NOTES**:

- **ie smfinfolist**: Specifies the **smfInfoList** information element to be grouped.

  **tac-based**: Groups the **smfInfoList** information element based on TAC group.

**Note** You are recommended to use this configuration only when you want to add new slice or TAC Group information.

# SMF Deregistration with NRF

## Feature Description

The SMF supports the deregistration of Network Function (NF) Repository Function (NRF), wherein the NF deregister service operation of the SMF removes the profile of a network function that is registered in the NRF.

The SMF starts the NF deregister service operation in the following scenarios:

- When the Service Based Interface (SBI) endpoint is not configured and all the rest endpoints stop functioning.
- When all the configured SBI endpoints VIP IP and N11 VIP IPs are offline.

**Note** SMF can't perform NRF deregistration when all the REST endpoints abruptly stop functioning. Only the REST endpoints can perform registration or deregistration of NRF.

## How it Works

The NF deregister service operation deletes the specific resource based on its NF instance ID. The NF deregistration starts when the Uniform Resource Identifier (URI) receives a request to delete a specific NF instance.

The recommended SMF shutdown process involves the following steps:

1. All N11 and SBI VIP IPs are marked as offline. After these endpoints appear offline, the NF deregistration request is sent to the NRF. The NRF notifies the peer NFs, such as AMF, about the SMF shutdown and its unavailability for traffic.
2. Wait for a grace period to allow convergence and perform a "system mode shutdown" to stop all the pods.

When the endpoint SBI is not configured, the system deletes the rest-ep pod immediately and avoids proper convergence. Implementing the system mode shutdown without taking the SBI and N11 VIP IPs offline also avoids convergence.

## Call Flows

This section describes the following call flows:

- NRF deregistration call flow
- NRF deregistration trigger events call flow

### *NRF Deregistration Call Flow*

This section describes the NF deregistration call flow.

***Figure 2: NRF Deregistration Call Flow***

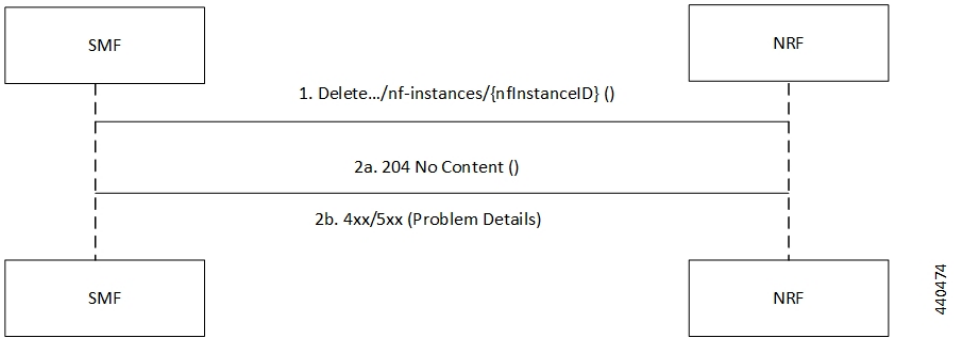



***Table 4: NRF Deregistration Call Flow Description***

| Step | Description |
|---|---|
| 1 | The SMF sends a Delete request to the resource URI that indicates the NF instance. The request body is empty. |
| 2a | If the deletion of the specified resource is successful, the "204 No Content" message appears. The response body remains empty. |
| 2b | If the NF instance, which is identified with the NF instance ID, does not exist in the list of registered NF instances in the NRF database, the NRF sends the "404 Not Found" status code with the problem details. |

### *NF Deregistration Trigger Events Call Flow*

This section describes the NF deregistration trigger events call flow.

*Figure 3: NF Deregistration Trigger Events Call Flow*

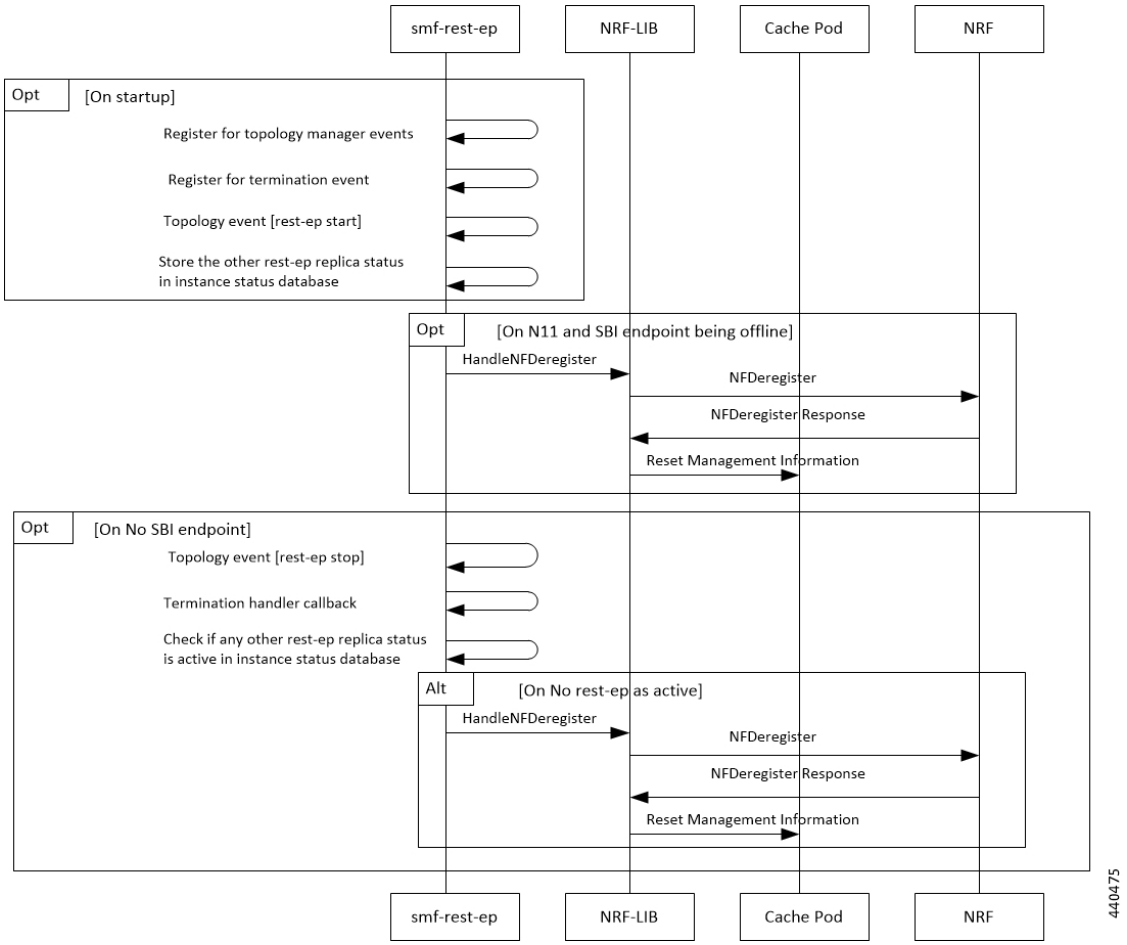



*Table 5: NF Deregistration Trigger Events Call Flow Description*

| Step | Description |
|---|---|
| On startup | |
| 1 | The SMF rest-ep registers for topology manager events to identify the state of other rest-ep instances and keeps a track of these instances in an instance state database. |
| 2 | The SMF rest-ep registers for the termination handler with the application infrastructure for receiving notification when the application infrastructure stops functioning. As part of the termination handler, the SMF rest-ep monitors the instance state database for any other working rest-ep. |
| 3 | The SMF rest-ep starts the topology event. |
| 4 | The SMF rest-ep saves the status of other rest-ep replicas in the instance state database. |
| When the N11 and SBI endpoints are offline | |
| 5 | The SMF rest-ep sends the Handle NF deregister message to the NRF-Lib. |
| 6 | When all the SBI and N11 VIP IP endpoints are offline, the SMF rest-ep sends the deregistration request to the NRF. |

| Step | Description |
|---|---|
| 7 | The NRF sends the NF deregister response to the NRF-Lib. |
| 8 | The NRF-Lib resets all the management information that is configured in the cache pod. |
| When no SBI endpoint exists | |
| 9 | The SMF rest-ep starts the topology event to stop the other rest-ep. |
| 10 | The SMF rest-ep starts the termination handler callback. |
| 11 | The SMF rest-ep checks the instance status database for any other working rest-ep. |
| When no rest-ep is functional | |
| 12 | The SMF rest-ep sends the Handle NF deregister message to the NRF-Lib. |
| 13 | The SMF rest-ep sends the deregistration request to the NRF. |
| 14 | The NRF sends the NF deregistration response to the NRF-Lib. |
| 15 | The NRF-Lib resets all the management information that is configured in the cache pod. |

### Standards Compliance

The SMF deregistration with NRF feature complies with the following standards:

- *3GPP TS 29.510 version 15.4.0 — 5G System; Network function repository services; Stage 3*

### Limitations

The SMF deregistration with NRF feature has the following limitation:

- When N11 and SBI VIP IPs are not marked offline, the NF deregistration is not sent for the system mode shutdown because no specific order for pod deletion exists. In addition, no monitoring procedure exists to check if the rest-ep pods are working.

# NF Heartbeat

## Feature Description

The NF Heartbeat feature enables the NFs to notify the NRF that the NF is operational. Each NF registered with the NRF contacts the NRF periodically by invoking the NF Update service operation. The time interval at which the NRF is contacted is deployment-specific and is returned by the NRF to the SMF as a result of a successful registration.

SMF sends the NF status and load parameter as part of NF heartbeat to NRF. SMF provides a CLI to configure the interval between periodic NF heartbeat. If the heartbeat value is configured in the NF registration response, the same value is used instead of another configured value.

### NF Heartbeat Interval

The SMF NF Heartbeat feature notifies the NRF that the SMF is operational. The default heartbeat interval is once in 10 seconds. With the **heartbeat interval** CLI command, you can configure the interval (in seconds) between the heartbeats. If NRF returns a different heartbeat time value as part of NF registration response or heartbeat response, then the same interval is used for subsequent heartbeats. As part of the heartbeat, NRF

sends the HTTP PATCH Request to the resource URI representing the NF instance. The payload body of the PATCH Request contains a "replace" operation on the "nfStatus" attribute of the NF profile of the NF instance, and configures it to the "REGISTERED" value. This release does not support parameters, such as load and capacity.

**Note** SMF uses the configured heartbeat. If the heartbeat is not configured, SMF uses the locally configured heartbeat.

# How it Works

## Call Flows

### *NF Heartbeat Call Flow*

The following figure illustrates the NF heartbeat call flow.

***Figure 4: NF Heartbeat Call Flow***

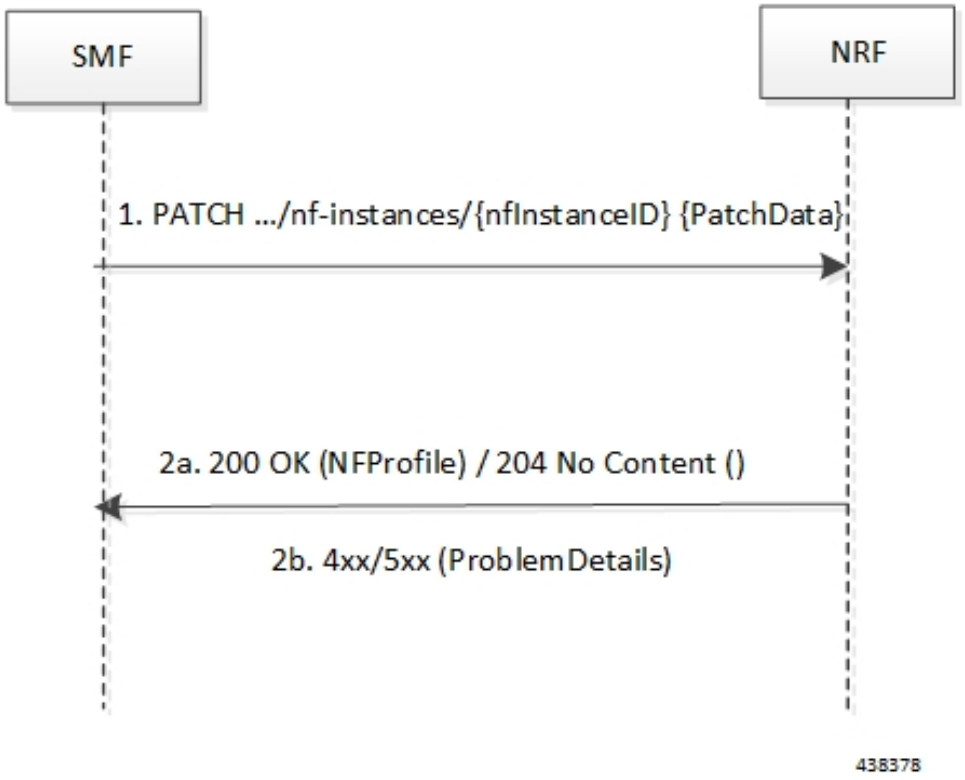



***Table 6: NF Heartbeat Call Flow Description***

| Step | Description |
|---|---|
| 1 | The NF Service Consumer sends a PATCH request to the resource URI representing the NF instance. The payload body of the PATCH request contains a replace operation on the nfStatus attribute of the NF Profile of the NF instance, and set it to the value REGISTERED. |

| Step | Description |
|---|---|
| 2 | On success, if the NF Profile changes, the NRF returns "200 OK" along with the full NF Profile data in the response body; otherwise, "204 No Content" is returned. |
| 3 | If the NF instance, identified by the "nfInstanceID", is not found in the list of registered NF instances in the NRF's database, the NRF returns "404 Not Found" status code with the ProblemDetails IE providing details of the error. Example:<br>**PATCH .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64**<br>**Content-Type: application/json-patch+json**<br>**[**<br>**{"op": "replace", "path": "/nfStatus", "value": "REGISTERED"}**<br>**]**<br>**HTTP/2 204 No Content**<br>**Content-Location:**<br>**.../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64** |

### Standards Compliance

The NF Heartbeat feature complies with the following standards:

- *3GPP TS 29.510, version 15.4.0 (2019-07) — 5G System; Network function repository services; Stage 3*

## Configuring NRF heartbeat interval

**Procedure**

**Step 1** Configure the group name of the NF management.

**Example:**

```
config
group nf-mgmt nf_mgt_name
```

**Step 2** Configure the heartbeat interval for the specified NF management group.

**Example:**

```
config
group nf-mgmt nf_mgt_name
   hearbeat interval seconds
```

- —This CLI command specifies the amount of time (in seconds) that gets subtracted from the negotiated or locally configured heartbeat timer.
- **hearbeat interval** *hearbeat_interval* —This CLI command specifies the interval between the heartbeats. The value of heartbeat interval is in seconds.

**Note**

- If NRF returns a different heartbeat interval value as part of NF registration response or heartbeat response, the same value is used for subsequent heartbeats.
- If the skew parameter is not configured, the system uses the negotiated or locally configured heartbeat interval without modification.
- Extremely large skew values that significantly reduce the heartbeat interval should be avoided, as they may unnecessarily increase heartbeat frequency.

**Step 3** Configure the NRF group name and locality.

**Example:**

```
config
group nf-mgmt nf_mgt_name
   hearbeat interval seconds
   nrf-mgmt-group nrf_group_name
   locality locality_name
   end
```

# NRF Support for SMF Subscription and Notification

## Feature Description

The SMF uses the NRF-provided Subscription service to subscribe to NF status changes that the NF receives as a discovery response. This feature helps in updating the cached NF discovery responses.

The SMF honors only the notification changes in load, capacity, status at the NF level, and at the service level. It ignores all other parameter changes in the notification.

After the successful subscription for notification service, the SMF receives notifications of registration and deregistration of NF Instances, or notifications of NF profile changes for a given NF Instance.

The SMF supports the "NFProfile" field and "ChangeItem" field in the "NotificationData". If the notification event type is set to "NF_PROFILE_CHANGED", the SMF receives notification about the profile-level changes or a list of individual change items for the NFProfile parameters along with nfInstanceUri.

The "ChangeItem" field includes the following parameters:

- op—Indicates the type of change that happens to the resource.
- path—Contains the JSON pointer value which indicates the target location within the resource.
- from—Indicates the path of the JSON element that is moved or copied to the location indicated by the "path" attribute. It is present if the "op" attribute is of value "MOVE".
- origValue—Indicates the original value at the target location within the resource specified in the "path" attribute.
- newValue—Indicates a new value at the target location within the resource specified in the "path" attribute.



**Note** The SMF currently supports only the ADD, REPLACE, and REMOVE operations as part of the "op" parameter.

The following is an example of the notification payload sent from the NRF when an NF instance has changed its profile by updating the IP address value and the TCP port for the first endpoint of the first NF service.

Example 1:

```
{
  "event": "NF_PROFILE_CHANGED",
  "nfInstanceUri": ".../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64",
  "profileChanges": [
    {
      "op": "REPLACE",
     "path": "/nfServices/0/ipEndPoints/0/ipAddress", ===> Change ipAddress to ipv4Address

      "newValue": "209.165.201.10"
    },
    {
      "op": "REPLACE",
      "path": "/nfServices/0/ipEndPoints/0/port",
      "newValue": 8080
    }
  ]
```

Example 2:

```
{
"event": "NF_PROFILE_CHANGED",
"nfInstanceUri": ".../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64",
"nfProfile": <Newly updated complete profile>
}
```

## How it Works

This feature uses the NF Subscribe service to subscribe to changes on the status of NF instances that the NF receives as discovery responses. The SMF sends a subscription for the response validity period for each of the NF profiles that it receives in the discovery response. The SMF checks if an existing NF instance subscription time needs an extension or not depending on the current response time validity. If a subscription needs an extension, a subscription PATCH is sent with the extended validity time.

During subscription, the NRF may respond with a modified validity time. This validity time might differ from the SMF validity time request. In such a scenario, the SMF tracks the required subscription time and the actual subscription time returned by the NRF.

The SMF periodically (every two minutes) checks in database if there is any subscription with the actual subscription time ending soon (as in next five minutes) but has required validity time more than the actual validity time. In this scenario, the SMF sends a PATCH subscription to extend the subscription validity time.

The SMF fills the Status Notification URI based on the interface NRF configuration that is specified in the configuration. The notification VIP IP and VIP port are used to frame the status notification URI.

```
http://{nrfinterface.vip-ip}:{ nrfinterface.vip-port}/{notifResourceURI}
```

On status notification, the SMF updates the local cache and the external cache (cache pod) with the changed attributes.

## Call Flows

This section describes the call flows for the SMF Subscription and Notification feature.

### *Subscription (PATCH) Call Flow*

The NRF updates the subscription to notifications on NF instances to refresh the validity time, when the specified time is due to expire. The SMF can request a new validity time to the NRF. If the operation is successful, the NRF can assign and provide a new validity time to the NF.

Updating the "subscriptionID" resource, initiates the Subscription (PATCH) operation. The operation starts on issuing an HTTP PATCH request on the URI representing the individual resource.

The following figure illustrates the call flow for subscription to NF instances in the same PLMN.

***Figure 5: Subscription (PATCH) Call Flow***

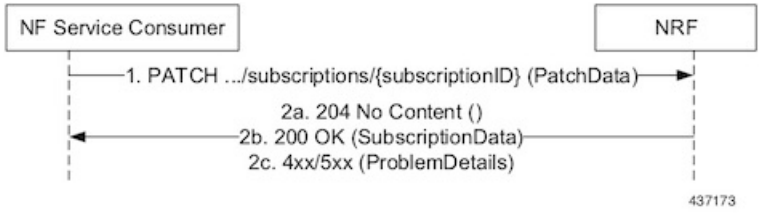



***Table 7: Subscription (PATCH) Call Flow***

| Step | Description |
|---|---|
| 1 | The SMF sends a PATCH request to the resource URI identifying the individual subscription resource. The payload body of the PATCH request contains a "replace" operation on the "validityTime" attribute of the SubscriptionData structure. The request also contains a new suggested value for the "validityTime" attribute. This replace operation does not replace any other attribute of the resource. |
| 2a | When a subscription is successful, the NRF sends a "204 No Content" response. This response indicates that the NRF accepts:<br>• Extension of the subscription lifetime<br>• Value of the "validityTime" attribute |
| 2b | If the subscription fails due to errors in the JSON Patch object in the request body, the NRF returns a "400 Bad Request" status code with the problem details. |
| 2c | If the subscription fails due to internal errors in the NRF, the NRF returns a "500 Internal Server Error" with the problem details.<br>Example:<br>`PATCH .../subscriptions/2a58bf47`<br>`Content-Type: application/json-patch+json`<br>`[`<br>`{ "op": "replace", "path": "/validityTime", "value": "2018-12-30T23:20:50Z" },`<br>`]` |

### *Subscription (POST) Call Flow*

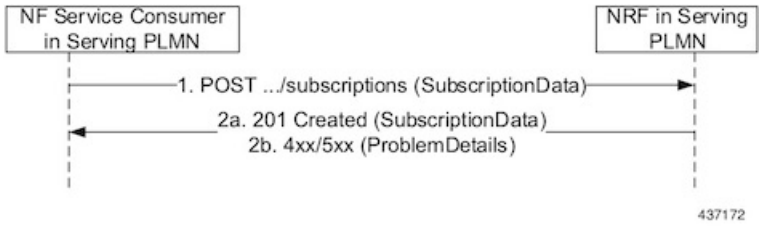
The Subscription service operation allows to:

- Create a subscription so that the SMF can request notification (depending on certain filters) in the following scenarios:
  - When there is a registration or deregistration in the NRF.
  - When there is a modification to a profile.
- Create a subscription to a specific NF instance such that the SMF can request notification in the following scenarios:
  - When there is a modification to an NF instance.
  - When there is a deregistration of an NF instance.

**Important** Currently, SMF only supports the subscription of NF instances that the NF receives as its discovery response.

The following figure illustrates the call flow for subscription to NF instances in the same PLMN.

***Figure 6: Subscription (POST) Call Flow***




Implementing the subscription to notifications on NF instances creates a new individual resource under the collection resource "subscriptions." Issuing a POST request starts the operation on the Uniform Resource Identifier (URI) representing the "subscriptions" resource.

*Table 8: Subscription (POST) Call Flow Description*

| Step | Description |
|---|---|
| 1 | The NF Service Consumer sends a POST request to the resource URI representing the "subscriptions" collection resource.<br>The request body includes data that indicates the type of notifications that the SMF has subscribed to receive. It also contains a callback URI, where the SMF prepares to receive the actual notification from the NRF. The notification contains the SMF suggested validity time, which represents the time span during which the subscription remains active.<br>The subscription request may also include more parameters indicating the list of attributes in the NF Profile to monitor (or to exclude from monitoring). This request determines if the NRF must send a notification, when there is a change in any of the profile attributes.<br>The subscription data also includes the **reqNfType** attribute, which contains the NF type of the NF Service Consumer that requests the creation of the subscription. The NRF uses it for authorizing the request. |
| 2a | When a subscription is successful, the NRF sends a "201 Created" response. This response contains newly created subscription data that includes the NRF-determined validity time beyond which, the subscription is invalid. When the subscription expires, the SMF creates a new subscription in the NRF to continue receiving status notifications. |
| 2b | If the subscription fails due to errors in the subscription data, the NRF returns a "400 Bad Request" status code with the problem details.<br>If the subscription fails due to internal errors in the NRF, the NRF returns a "500 Internal Server Error" with the problem details. |

## *NFStatus Notify Call Flow*

When a POST request is issued to each callback URI of the various subscribed NF instances, the SMF initiates the NFStatus Notify operator.

The following figure illustrates the NFStatus Notify call flow.

*Figure 7: NFStatus Notify Call Flow*

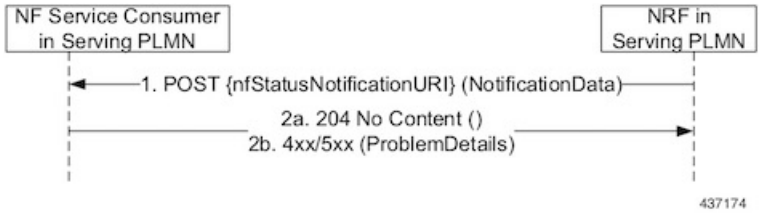


*Table 9: NFStatus Notify Call Flow Description*

| Step | Description |
|---|---|
| 1 | The NRF sends a POST request to the callback URI.<br>The request body for a profile change notification request includes the following:<br>• **event**—This attribute indicates the notification type. It can be one of the following:<br>  • NF_REGISTERED<br>  • NF_DEREGISTERED<br>  • NF_PROFILE_CHANGED<br>• **nfInstanceUri**—Uniform Resource Identifier (URI) of the NF instance associated to the notification event.<br>• **nfProfile**—Indicates the new or updated NF profile.<br>• **profileChanges**—This attribute identifies changes on the profile of the NF instance associated to the notification event. This attribute is available when the event notification type is "NF_PROFILE_CHANGED". |
| 2a | When the notification is successful, the NRF sends a "204 No content" response. |
| 2b | If the SMF disregards "nfStatusNotificationURI" as a valid notification URI, the SMF returns a "404 Not Found" status code with the problem details. For example, if the URI does not belong to any of the existing subscriptions that the SMF has created in the NRF. |

## Limitations

This feature has the following limitations:

- NF status notification supports only NF profile load, NF profile capacity, NF profile status, service load, service capacity, and service status parameter changes.
- SMF supports only the NFPofiile field in the "NotificationData." It does not support the "Change item" field.
- The SMF supports notification of the following parameter changes:
  - nfProfile
    - nfStatus
    - ipv4Address
    - ipv6Address
    - priority
    - capacity
    - load
  - nfService

- version

**Note** Change to a new version is permitted but not the deletion and modification of the existing version.

- scheme

**Note** Currently, http is only supported

- nfServiceStatus
- ipEndPoints
- apiPrefix
- capacity
- load
- priority

- The SMF currently supports only the ADD, REPLACE, and REMOVE operations as part of the "op" parameter in the "ChangeItem" field.

## Configuring NRF for Subscription and Notification

This section describes how to configure the NRF for subscription and notification.

**Note** For the subscription and notification to work, it is mandatory to configure the NRF interface within SBI interface.

When discovery is done with NRF, a subscription message for the discovered NF instances is sent. The SMF fills the Status Notification URL based on the NRF interface configuration that is specified in the configuration. The notification VIP IP and VIP port are used to frame the status notification URL. The SMF uses the URL that is included in the subscription request message for status notifications.

To configure the NRF interface, vip-ip, vip-port, and loopback port to open the server endpoints for the NF status notification, use the following sample configuration.

```
config
  instance instance-id gr_instance_id
    endpoint sbi
      replicas replica_num
        vip-ip ip_address
      interface nrf
        vip-ip ip_address
        vip-port port_number
```

```
loopbackPort port_number
end
```

**NOTES**:

- **interface nrf**: Specify the interface as NRF.
- **vip-ip** *ip_address*: Specify the virtual IP address of the virtual host. The SMF uses this as the listening IP address for the status notification.
- **vip-port** *port_number*: Specify the port number of the virtual host. The SMF uses this as the listening port for the status notification.
- **loopbackPort** *port_number*: Specify the internal port number of the loopback host. The SMF uses this port for the NF status notification.

## Standard Compliance

This feature complies to *3GPP TS 29.510, Version 16.4.0*.

# NF Profile Update

## Feature Description

The SMF invokes the NF Update service operation when there are changes to the NF registration parameters due to the SMF profile configuration change.

The NF Update service updates the NF profile that was previously registered in the NRF by providing the updated profile of the requesting NF to the NRF.

The update operation can be one of the following:

- A whole NF profile update (complete replacement of the existing profile with a new profile)
- An update to only a subset of the NF profile parameters (adding, deleting, or replacing services to the NF profile)

## How it Works

This section describes the NF profile update procedure.

### Call Flows

This section describes the following call flows:

- NF Profile Complete Replacement Call Flow, on page 21
- NF Registration and NF Update Call Flow, on page 22

#### *NF Profile Complete Replacement Call Flow*

The following figure illustrates a call flow representing the complete NF profile replacement.

*Figure 8: NF Profile Complete Replacement*

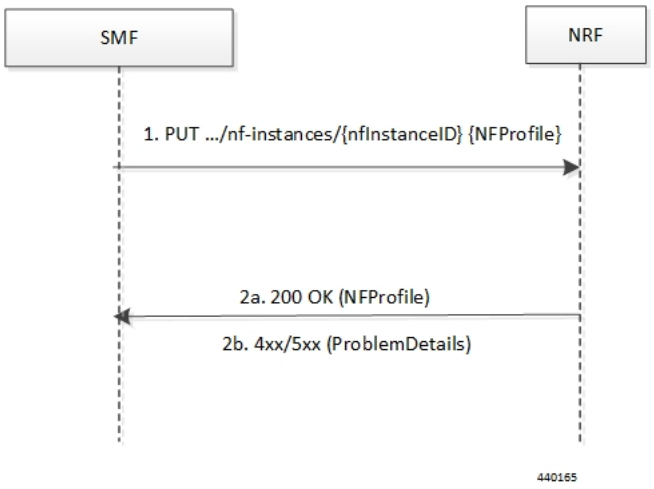


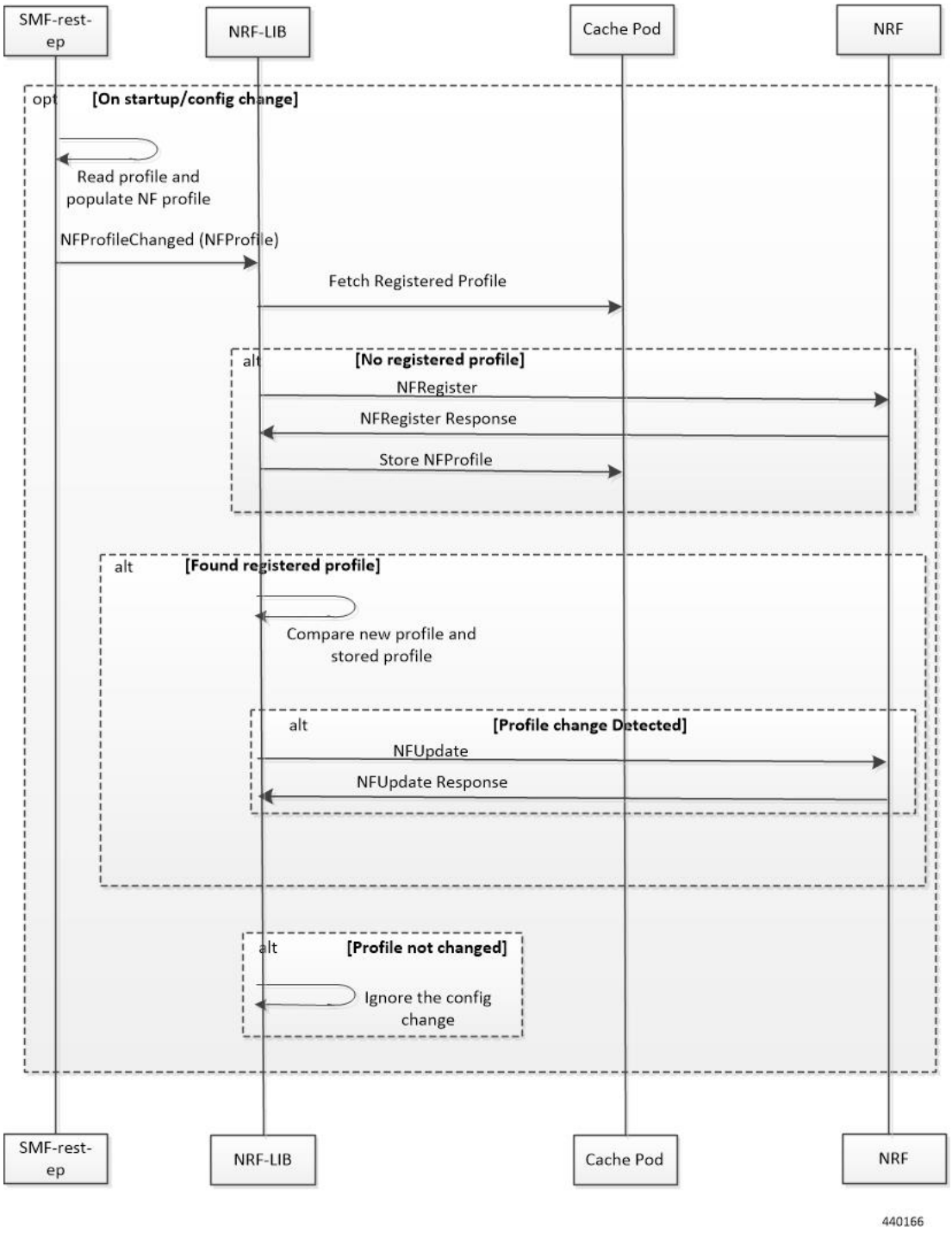

*Table 10: NF Profile Complete Replacement Call Flow Description*

| Step | Description |
|---|---|
| 1 | The SMF sends a PUT request to the resource URI representing the NF instance. The payload body of the PUT request contains an update operation on the NF Profile of the NF instance. |
| 2a | On success, if the NF Profile changes, the NRF returns "200 OK" along with the full NF Profile data in the response body. |
| 2b | If the NF instance, identified by the "nfInstanceID", is not found in the list of registered NF instances in the NRF database, the NRF returns 4xx or 5xx status code with the ProblemDetails IE providing details of the error. |

## *NF Registration and NF Update Call Flow*

The following figure illustrates the call flow representing the NF registration and NF update messaging from SMF on NF profile change trigger from REST-EP.

*Figure 9: NF Registration and NF Update Call Flow*




1. The SMF REST-EP, on start-up, reads the SMF profile configuration and accordingly populates the NF management profile. The REST-EP then triggers SMF to indicate the NF Profile change.
2. The SMF maintains the NF registration status and the registered profile in an external cache pod. The SMF detects whether the NF registration with NRF is completed. If the SMF detects that the registration

is not completed during NF profile change handling, perform Step 3. If the NF registration is complete, perform Step 4.

3. The SMF sends NF Register to NRF. It allows an NF instance to register its NF profile in the NRF. It includes the registration of the general parameters of the NF instance along with the list of services exposed by the NF instance.
4. The SMF fetches the registered NF profile and then compares it with the new profile.
5. The SMF sends NF update (PUT) request to the NRF when any of the parameters in the NF management profile changes due to SMF profile configuration change.

   Load parameter is not set as part of the PUT message. Heartbeat is set as the current active heartbeat interval.
6. The SMF ignores the trigger if there is no change detected.

**Important** The NF update is sent only from the elected SMF.

### Standards Compliance

The NF Profile Update feature complies with the following standards:

- *3GPP TS 29.510, Version 15.4.0 (2019-07) – 5G System; Network function repository services; Stage 3*

### Limitations

The NF Profile Update feature has the following limitation:

- Supports only the complete replacement of NF profile.
- Doesn't support capacity.

# NF Discovery

## Feature Description

The SMF uses the NRF-provided, NF discovery service to discover network functions (NFs), such as Access and Mobile Function (AMF), Unified Data Management (UDM), and Policy Control Function (PCF). The SMF configures the preferred locality as provided in the "profile nf-pair" configuration of Network Repository Function (NRF) in the discovery query.

For each NF, the query parameters, also known as filters, are configurable. Based on these parameters, NRF returns all the NFs matching the query criteria for the SMF to discover NF profiles.

**Note** The NF discovery and load-balancing capabilities are available only for UDM, PCF, CHF, and AMF.

NF discovery supports dynamic configuration change. With this feature, if the configurations were modified in the middle of the transaction or procedure, the ongoing transactions are not impacted.

The dynamic configuration change feature supports the following:

- NRF transaction or procedure picks a configuration version (v1) and uses the same version until the NRF transaction or procedure completes.
- If you change the configuration during an ongoing NRF transaction, then a new configuration version (v2) is created. However, the new configuration is applied in the new transaction.

The dynamic configuration changes apply to the following data structures:

- NrfFailureProfileSt
- NrfClntProfileSt
- NrfGrpSt
- NrfPairProfileSt
- NrfMgmtGrpSt

# How it Works

The service operation is executed by querying the "nf-instances" resource. The request is sent to an NRF in the same PLMN of the SMF.

## Call Flows

This section describes the call flow associated with this feature.

### Service Discovery Request Call Flow

This section describes the service discovery request call flow.

***Figure 10: Service Discovery Request Call Flow***

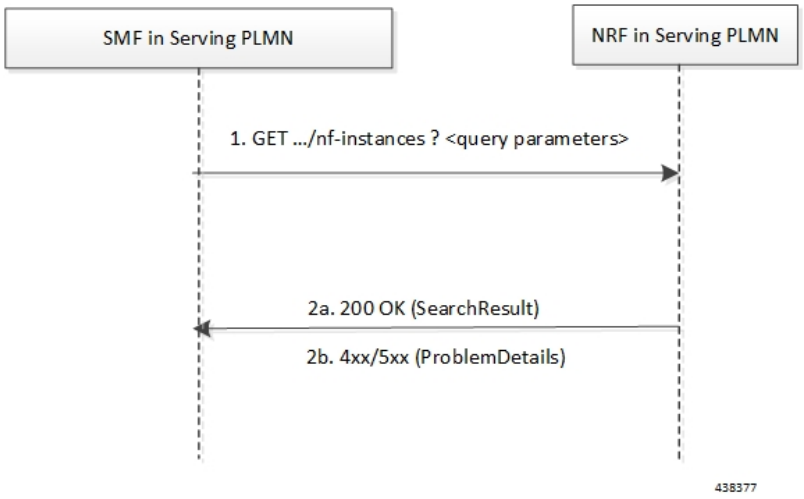


*Table 11: Service Discovery Request Call Flow Description*

| Step | Description |
|---|---|
| 1 | The SMF sends an HTTP GET request to the resource URI "nf-instances" collection resource. The input filter criteria for the discovery request exists in query parameters. |
| 2a | On success, "200 OK" is returned. The response body contains a validity period, during which the SMF caches the search result, and an array of NF profile object that satisfy the search filter criteria. For example, all NF instances displaying a certain NF Service name. |
| 2b | If the SMF is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF returns "403 Forbidden" response.<br>If the discovery request fails at the NRF due to errors in the input data in the URI query parameters, the NRF returns "400 Bad Request" status code with the "ProblemDetails" IE providing details of the error.<br>If the discovery request fails at the NRF due to NRF internal errors, the NRF returns "500 Internal Server Error" status code with the "ProblemDetails" IE providing details of the error. |

The NF profile objects that are returned in a successful result contains generic data of each NF instance, applicable to any NF type. These objects can also contain NF-specific data, for those NF instances belonging to a specific type (for example, the attribute "udrInfo" exists in the NF profile when the type of the NF instance takes the "UDR" value). In addition, the attribute "customInfo" exists in the NF profile for NF instances with custom NF types. For NF instances, the NRF returns the "customInfo" attribute, if available, as part of the NF profiles returned in the discovery response.

The SMF service communicates with different NFs, such as UDM, AMF, PCF, and CHF, when the session is active. The NF discovery is based on set of filters, also called query parameters, which are associated with the session. The SMF service discovers the NFs, matching the filter criteria for the session, to send messages to NF.

The SMF supports the following filters:

- Dnn
- Tai
- TargetNfFqdn
- TargetPlmnList
- TargetNfInstanceId
- Snssais
- Preferred locality

The discovered NFs are cached with the filter as the key. The endpoint selection for sending the message is based on probabilistic load balancing algorithm (IETF RFC 2782) using the priority and capacity parameters. The NF discovery response carries a validity time, which decides the cache validity period.

SMF sends the messages to a target based on the Location header URL in response to initial messages sent to NF.

SMF supports stickiness wherein the endpoint, service instance, and NF instance details of the selected endpoint for a message that is sent, will be provided to the application or REST-EP so that the same can be specified

in subsequent message (instead of discovery filter). This operation helps in maintaining stickiness for a session to the selected NF.

## Standards Compliance

The NF Discovery feature complies with the following standards:

- *3GPP TS 29.510 version 15.4.0 (2019-07) – 5G; 5G System; Network function repository services; Stage 3*

## Limitations

The NRF Discovery feature has the following limitations:

- The cache maintained is local to the library. In case of deployment with multiple replicas of REST-EP, if two Discovery or Send messages with the same discovery filter land on different pods, then both the pods trigger NF discovery.
- This feature supports only the UDM, PCF, CHF, and SMF discovery, and load balancing. It does not support UPF discovery.

# Configuring NRF for Discovery

This section provides the configurations that are required to perform the NF discovery.

## Registering NRF

To register an NRF, use the following sample configuration.

```
config
nssai name nssai_name
    sst sst ssd ssd
    dnn dnn_name_value
    end
```

**NOTES**:

- **nssai name** *nssai_name*: Configure the NSSAI name value for the slice. The *nssai_name* value must be a string.

**Note** SMF supports a maximum of 512 slices to be sent toward NRF.

### Configuration Example

The following is an example configuration of the NRF registration.

```
nssai name slice1
 sst 02
 sdt Abf123
 dnn [ dnn1 intershat intershat1 intershat2 intershat3 intershat4 intershat5 intershat6
intershat7 starosupf ]
exit
```

```
nssai name slice2
 sst 02
 sdt Abf124
 dnn [ dnn1 intershat intershat1 intershat2 intershat3 intershat4 intershat5 intershat6
intershat7 starosupf ]
exit
```

## Discovering NRF

To configure the NRF discovery, use the following sample configuration:

```
config
  profile network-element [ amf amf_profile_name| chf chf_profile_name | pcf
pcf_profile_name | udm udm_profile_name | upf upf_profile_name ]
     query-params requester-snssais
     exit
```

**NOTES:**

- **query-params requester-snssais**: Specify the list of Single Network Slice Selection Assistance Information (S-NSSAIs) as the query parameter in the NF discovery request towards the NRF.

### Configuration Example

The following is an example configuration.

```
config
   profile network-element udm udm1
      query-params requester-snssais
      exit
   profile network-element pcf pcf1
      query-params requester-snssais
      exit
   profile network-element chf chf1
      query-params requester-snssais
      exit
   profile network-element upf upf1
      query-params requester-snssais
      exit
   profile network-element amf amf1
      query-params requester-snssais
      exit
```

## Configuring NF Client Profile

To configure the NF endpoints for AMF, CHF, PCF, and UDM, use the following sample configuration:

```
config
  profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf
pcf-profile | udm udm-profile } nf_profile_name }
  end
```

**NOTES:**

- **profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf pcf-profile | udm udm-profile }** *nf_profile_name* **}**: Specify the required NF client profiles and provide the local configuration for any of the following configured NFs:
  - **amf**: Enable the AMF local configuration
  - **chf**: Enable the CHF local configuration

- **pcf**: Enable the PCF local configuration
- **udm**: Enable the UDM local configuration

  For example, if you are configuring the **amf amf-profile** keyword, this command enables the AMF local configuration. The same approach applies for the other configured NFs.

  *nf_profile_name* must be an alphanumeric string representing the corresponding NF client profile name.

- You can configure multiple NF profiles within a given service.
- To disable the configuration, use the **no profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf pcf-profile | udm udm-profile }** *nf_profile_name* **}** command.

## Configuration Example

The following is an example configuration.

```
profile nf-client nf-type pcf
pcf-profile pcf-profile
  locality LOC1
   priority 1
   service name type npcf-smpolicycontrol
    endpoint-profile epprof
     capacity   10
     priority   1
     uri-scheme http
     endpoint-name ep1
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.202.133
      primary ip-address port 8080
     exit
     endpoint-name ep2
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.201.1
      primary ip-address port 8080

     exit
    exit
   exit
  exit
exit
exit
profile nf-client nf-type pcf
pcf-profile pcf-profile
  locality LOC1
   priority 1
   service name type npcf-smpolicycontrol
    endpoint-profile epprof
     capacity   10
     priority   1
     uri-scheme http
     endpoint-name ep1
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.201.2
      primary ip-address port 8080
      fqdn name nrf.cisco.com
```

```
 fqdn port 9010
exit
```

## Associating a Discovery Group with NF Type

To pair a discovery group with NF types, use the following sample configuration.

```
config
  profile nf-pair nf-type nf_type
    nrf-discovery-group nrfdisc_group_name
    end
```

**NOTES:**

- **nf-type** *nf_type*: Specify the NF client type value as SMF.
- **nrf-discovery-group** *nrfdisc_group_name*: Specify the NRF discovery group name. Discovery group is the logical link to the NRF endpoint groups (nrf-group). For each NF type, you can associate a discovery group and the locality information.

## Configuring NF Endpoint Profile Parameters in NRF Discovery Group

The SMF provides CLI for configuring NF endpoints for **nnrf-nfd** (NF discovery).

**Note** For a discovery group, you can configure only the **nnrf-disc** service.

The CLI configuration allows configuring multiple endpoints under each endpoint profile. The SMF uses the priority and capacity parameters to load balance between these endpoints. All endpoints under an endpoint profile share the session context. That is, when selecting an endpoint profile for initial message of a session, then the SMF sends the subsequent messages (for example, update, delete, and so on) of the session to any of the endpoints in the endpoint profile.

Primary, secondary, and tertiary hosts [ip:port] can be configured within each endpoint. Both IPv4 and IPv6 addresses can be specified. If both are specified, then the IPv4 address is preferred.

SMF provides APIs to discover and send a message to an NF matching a set of filter parameters.

A URI uniquely identifies a resource. In the 5GC SBI APIs, when a resource URI is an absolute URI, the structure is specified as follows:

```
{apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}
```

apiRoot is a concatenation of the following parts:

- scheme ("http" or "https")

**Note** Both HTTP and HTTPS scheme URIs are allowed. See the *3GPP TS 33.501, subclause 13.1* for more information on security of service-based interfaces.

- fixed string "://"
- authority (host and optional port) as defined in IETF RFC 3986

- an optional deployment-specific string (API prefix) that starts with a "/" character [api-root in CLI]

To configure the NRF endpoints for different services supported by NRF, use the following sample configuration:

```
config
   group nrf discovery discovery_name
      service type nrf nnrf-disc
         endpoint-profile
            name  epprofile_name
            api-root api_string
            api-uri-prefix uri_prefix_string
            uri-scheme { http | https }
            endpoint-name ep_name { capacity capacity | primary ip-address {
ipv4 ipv4_address | ipv6 ipv6_address | port port_num } | secondary ip-address {
 ipv4 ipv4_address | ipv6 ipv6_address| port  port_num } | tertiary ip-address
{ ipv4 ipv4_address| ipv6 ipv6_address | port port_num }
            | fqdn { name  fqdn_name | port port_num } }
            version [ uri-version version_num full version version_num ]
            end
```

**NOTES:**

- **group nrf discovery** *discovery_name* : Configure the NRF discovery group.
- **api-root** *api_string*: Specify the deployment-specific service API prefix that is used within the { apiRoot }.
- **api-uri-prefix** *uri_prefix_string*: Specify the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **endpoint-name** *ep_name* **{ capacity** *capacity* **| primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} }**: Specify the endpoint name. You can configure the primary, secondary, and tertiary hosts (IP: Port) within each endpoint for NF server failover handling. The server failover configuration accepts both the IPv4 and IPv6 addresses. However, the SMF gives preference to the IPv4 address.
  - **capacity** *capacity*: Specify the node capacity for the endpoint. *capacity* must be an integer in the range of 0-65535.
  - The endpoint selection for sending the message is based on probabilistic load-balancing algorithm (IETF RFC 2782) using the priority and capacity parameters.
  - **primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the primary endpoint IPv4 address, IPv6 address, or port.
  - **secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the secondary endpoint IPv4 address, IPv6 address, or port.
  - **tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the tertiary endpoint IPv4 address, IPv6 address, or port.
  - **priority** *priority_value*: Specify the priority for the service to select the appropriate profile using the load-balancing logic. *priority* must be an integer in the range of 0-65535.
  - **fqdn name** *fqdn_name*: Specify the FQDN name.

- **fqdn port** *fqdn_port*: Specify the FQDN port number. If port is not configured, SMF uses the standard port for FQDN, that is 80 for URI scheme HTTP and 443 for URI scheme HTTPS.

- **uri-scheme { http | https }**: Specify the URI scheme as **http** or **https**.
- **version [ uri-version** *version_num* **full version** *version_num* **]**: Specify the API URI version. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draft-number>].

### Verifying the NRF Endpoints Profile Parameters for NF Discovery

This section describes how to verify the configuration of the NRF endpoints profile parameters.

```
show running-config group nrf
group nrf discovery udmdiscovery
 service type nrf nnrf-disc
  endpoint-profile epprof
   capacity       10
   priority       1
   api-uri-prefix nudm-sdm
   api-root       root
   uri-scheme     http
   version
    uri-version v1
     full-version 209.165.200.225
    exit
   exit
   endpoint-name endpointName
    priority 1
    capacity 100
    primary ip-address ipv4 209.165.200.237
    primary ip-address port 3021
    fqdn name nrf.cisco.com
    fqdn port 9010
   exit
  exit
 exit
exit
```

## Configuring Locality for NF Types

The SMF provides locality aware NF discovery.

A pair profile has the locality values configured with NF type as SMF. A locality has the following values:

- client—Specify the client locality information.
- geo-server—Specify the geo-service locality information.
- preferred-server—Specify the preferred server locality information.

For a profile selection, only the preferred-server and geo-server locality values are considered. Following are the scenarios of these locality values configuration:

- If both the preferred-server and geo-server locality values are configured, then the profiles, which exist in discovery response, matching these locality values are selected. In addition, the profiles with empty locality value are selected. Any other profile with locality other than preferred-server and geo-server locality values are not considered.

- If only the preferred-server locality value is configured, then the profiles, which exist in discovery response, matching this value is selected. In addition, the profiles with an empty locality value are selected. Any other profile with locality other than preferred-server locality value is not considered.
- If only geo-server locality value is configured then the profiles, which exist in discovery response, matching this geo-server locality value is selected. In addition, the profiles with empty locality value is selected. Any other profile with locality other than geo-server locality value is not considered.
- If both preferred-server and geo-server locality values are not configured then all the profiles, which exist in discovery response, are selected.

To configure the locality for NF types, use the following sample configuration.

```
config
  profile nf-pair nf-type nf_type
    locality { client client_name | geo-server geoserver_name | preferred-server
 prefserver_name }
     end
```

**NOTES:**

- **client** *client_name*: Specify the client locality information. Client locality is the SMF locality and is a mandatory parameter.
- **preferred-server** *prefserver_name*: Specify the preferred server locality information. The preferred server locality is the locality that should be considered as the locality of preference during the corresponding NF discovery.
- **geo-server** *geoserver_name*: Specify the geo-server locality information. The geo-server locality is the geo redundant site for the preferred locality and is generally used as the next best server locality after preferred locality, during NF discovery.



**Note** **geo-server** *geoserver_name* is not fully qualified.

### Verifying the Association of the Discovery Group and Locality Configuration

This section describes how to verify the discovery group association and locality configuration for NF.

```
show running-config profile nf-pair
profile nf-pair nf-type UDM
 nrf-discovery-group DISC1
 locality client LOC1
 locality preferred-server PREF_LOC
 locality geo-server GEO
exit
```

## Configuring Locality for SMF

To configure the locality for SMF, use the following sample configuration.

This is a mandatory configuration if the SMF performs the NF discovery using the NRF.

```
config
  profile SMF SMF_profile_name
```

```
    locality value
    end
```

**NOTES:**

- **locality** *value*: Specify the SMF locality. *value* must be an alphanumeric string representing the deployed SMF locality. By default, this CLI command is disabled.
- To disable this configuration, use the **no locality** *value* command.

## Configuring NF Profiles for a DNN

To configure the NF profile that the configured Data Network Name (DNN) uses, use the following sample configuration.

```
config
  profile dnn dnn_profile_name
    network-element-profiles { amf | chf | pcf | udm } nf_profile_name
    end
```

**NOTES**:

- **network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name*: Specify one or more NF types, such as AMF, CHF, PCF, and UDM as the network element profile. *nf_profile_name* must be an alphanumeric string representing the corresponding network element profile name.
- This is an optional configuration. By default, this CLI command is disabled.
- You can configure multiple profiles within a given service.
- To disable the configuration, use the **no network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name* command.

## Defining Locality within NF Profile

This section describes how to define the locality of the NF endpoints. For the NF endpoint selection, the SMF first considers the preferred locality that is configured with the **profile nf-pair** CLI command. The admin determines the preferred locality based on the proximity of the locality and the network function. The SMF then uses the geo-server locality configurations as the next preferred locality for the NF discovery. For information on the **profile nf-pair** command, see Configuring Locality for NF Types, on page 32 in the NRF Selection per Peer NF Type, on page 37 section.

The SMF selects the other locality endpoints if the **profile nf-pair** CLI command does not include the preferred server locality configuration, or if the **profile nf-client** CLI command does not include the endpoint configured with the preferred server or geo-server locality. For the other locality endpoint selection, the SMF uses the **priority** configuration within the **locality** CLI command.

To define the locality of the NF endpoints, use the following sample configuration.

```
config
  profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf
pcf-profile | udm udm-profile } nf_profile_name }
    locality locality_name [ priority priority | service name type service_types
 { endpoint-profile epprofile_name } ]
    end
```

**NOTES:**

- **locality** *locality_name*: Specify the locality of the NF endpoint. The SMF uses the locality configurations (that is, the preferred server locality and geo-server locality) to select the appropriate NF endpoints.
- **priority** *priority*: Specify the priority for the locality configuration.
- **service name type** *service_types*: Specify the configured NF service types. The service types vary depending on the configured service.
  - SMF service supports "namf-comm" as peer communication type.
  - PCF service supports "npcf-am-policy-control" as peer communication type.
  - UDM service supports "nudm-sdm" and "nudm-uecm" as peer communication type.
  - SMF service supports "nsmf-pdusession" as peer communication type.
  - EIR service supports "n5g-eir-eic" as peer communication type.
  - NSSF service supports "nnssf-nsselection" as peer communication type.
  - SMSF service supports "nsmsf-sms" as peer communication type.
  - LMF service supports "nlmf-loc" as peer communication type.
  - GMLC service supports "ngmlc-loc" as peer communication type.
  - AUSF service supports "nausf-auth" as peer communication type.
- **endpoint-profile** *epprofile_name*: Specify the endpoints at a per NF service level. The NF specific services are available within the locality configuration.
- You can configure multiple endpoints per profile name for the configured NF.

## Configuring NF Endpoint Profile Parameters in NF Client Profile

This section describes how to configure the NF endpoint profiles within the service and its associated parameters.

The CLI configuration allows configuring multiple endpoints under each endpoint profile. The SMF uses the priority and capacity parameters to load balance between these endpoints. All endpoints under an endpoint profile share the session context. That is, when selecting an endpoint profile for initial message of a session, then the SMF sends the subsequent messages (for example, update, delete, and so on) of the session to any of the endpoints in the endpoint profile.

SMF provides APIs to discover and send a message to an NF matching a set of filter parameters.

A URI uniquely identifies a resource. In the 5GC SBI APIs, when a resource URI is an absolute URI, the structure is specified as follows:

```
{apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}
```

apiRoot is a concatenation of the following parts:

- scheme ("http" or "https")



**Important** Both HTTP and HTTPS scheme URIs are allowed. See *3GPP TS 33.501, subclause 13.1* for more information on security of service-based interfaces.

- fixed string "://"
- authority (host and optional port) as defined in *IETF RFC 3986*
- an optional deployment-specific string (API prefix) that starts with a "/" character [api-root in CLI]

To configure the NF endpoint profiles within the service and its associated parameters, use the following sample configuration:

```
config
  profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf
pcf-profile | udm udm-profile }  nf_profile_name }
     locality locality_name [ priority priority | service name type service_type
 ]
        endpoint-profile epprofile_name
           api-root api_string
           api-uri-prefix uri_prefix_string
           capacity capacity
           endpoint-name ep_name { capacity capacity | primary ip-address {
ipv4 ipv4_address | ipv6 ipv6_address | port port_num }| secondary ip-address {
ipv4 ipv4_address | ipv6 ipv6_address| port  port_num } | tertiary ip-address {
 ipv4 ipv4_address| ipv6 ipv6_address | port port_num } }
           priority priority_value
           uri-scheme { http | https }
           fqdn { name  fqdn_name | port port_num } }
           version [ uri-version version_num full version version_num ]
           end
```

**NOTES:**

- **api-root** *api_string*: Specify the deployment-specific service API prefix that is used within the { apiRoot }.
- **api-uri-prefix** *uri_prefix_string*: Specify the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **capacity** *capacity*: Specify the profile capacity.
- **endpoint-name** *ep_name* **{ capacity** *capacity* **| primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} | tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **} }**: Specify the endpoint name. You can configure the primary, secondary, and tertiary hosts (IP: Port) within each endpoint for NF server failover handling. The server failover configuration accepts both the IPv4 and IPv6 addresses. However, the SMF gives preference to the IPv4 address.
  - **capacity** *capacity*: Specify the node capacity for the endpoint. *capacity* must be an integer in the range of 0–65535.

    The endpoint selection for sending the message is based on probabilistic load balancing algorithm (*IETF RFC 2782*) using the priority and capacity parameters.
  - **primary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the primary endpoint IPv4 address, IPv6 address, or port.
  - **secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the secondary endpoint IPv4 address, IPv6 address, or port.

- **tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specify the tertiary endpoint IPv4 address, IPv6 address, or port.
- **priority** *priority_value*: Specify the priority for the service to select the appropriate profile using the load balancing logic. *priority* must be an integer in the range 0–65535.
- **fqdn name** *fqdn_name*: Specify the FQDN name.
- **fqdn port** *fqdn_port*: Specify the FQDN port number. If port is not configured, SMF uses the standard port for FQDN, that is 80 for URI scheme HTTP and 443 for URI scheme HTTPS.

- **uri-scheme { http | https }**: Specify the URI scheme as **http** or **https**.
- **version [ uri-version** *version_num* **full version** *version_num* **]**: Specify the API URI version. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draft-number>].

# NRF Selection per Peer NF Type

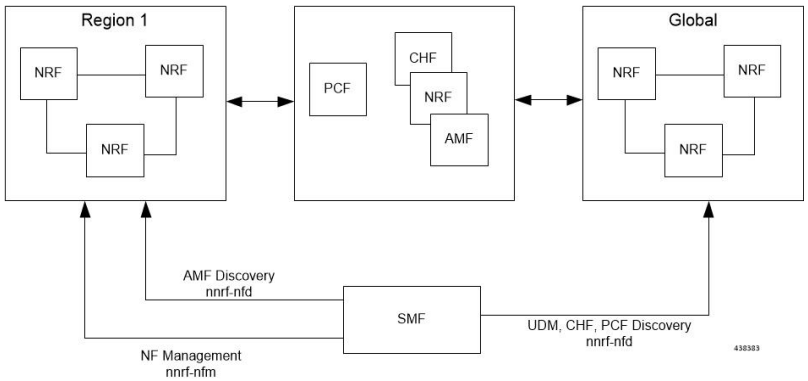
## Feature Description

The Network Repository Function (NRF) deployment can be logically segmented as global, regional, and so on, for a reliable network management. You can accomplish this segmentation by specifying different NRF endpoint groups for the discovery of different network functions.

For example, the SMF interacts with Region 1 NRF endpoints for management and SMF discovery. For UDM, CHF, and PCF discovery, the SMF communicates with the global NRF endpoints.

The following figure illustrates the NRF deployment.

***Figure 11: NRF Deployment***




### Standards Compliance

This feature complies with the following standard:

- *3GPP TS 29.510 version 15.4.0—5G System; Network function repository services; Stage 3*

## Configuring the NRF Selection per Peer NF Type

This section describes how to configure the NRF selection per peer NF type.

### Associating NRF Management and SMF Locality to NRF Endpoint

To configure the NRF management (nrf-group) and SMF locality, and associate them to NRF endpoint, use the following sample configuration.

```
config
   group nf-mgmt mgmt_name
      nrf-mgmt-group nrf_group_name
      locality locality_name
      message-handling-profile message_handling_profile_name
      end
```

**NOTES:**

- **nrf-mgmt-group** *nrf_group_name*: Specify the NRF management group.
- **locality** *locality_name*: Specify the locality information.
- **message-handling-profile** *message_handling_profile_name*: Specify the message handling profile for NRF.

#### *Verifying the Association of the NRF Management and SMF Locality to NRF Endpoint*

This section describes how to verify the configuration that associates the NRF management and SMF locality to NRF endpoint.

```
show running-config group nf-mgmt
group nf-mgmt NFMGMT1
 nrf-mgmt-group MGMT
 locality       LOC1
exit
```

### Configuring Locality for SMF

To configure the locality for SMF, use the following sample configuration.

This is a mandatory configuration if the SMF performs NF discovery using the NRF.

```
config

   profile SMF SMF_profile_name
      locality value
      end
```

**NOTES**:

- **locality** *value*: Specify the SMF locality. *value* must be an alphanumeric string representing the deployed SMF locality. By default, this CLI command is disabled.
- To disable the configuration, use the **no locality** *value* command.

### Configuring NF Profiles for a DNN

To configure the NF profile used by the configured Data Network Name (DNN), use the following sample configuration.

```
config
   profile dnn dnn_profile_name
      network-element-profiles { amf | chf | pcf | udm } nf_profile_name
```

```
    end
```

**NOTES**:

- **network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name*: Specify one or more NF types, such as AMF, CHF, PCF, and UDM as the network element profile. *nf_profile_name* must be an alphanumeric string representing the corresponding network element profile name.
- This is an optional configuration. By default, this CLI command is disabled.
- You can configure multiple profiles within a given service.
- To disable the configuration, use the **no network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name* command.

### Configuring Network Element Profile Parameters for the NF

To configure the network element profile parameters for the configured NF, use the following sample configuration.

```
config
  profile network-element { { amf | chf | pcf | udm } nf_profile_name }

    nf-client-profile profile_name
    query-params { dnn | limit | max-payload-size | requester-snssais |
 supi | tai | target-nf-instance-id | target-plmn }
    end
```

**NOTES:**

- **nf-client-profile** *profile_name*: Specify the local NF client profile. *profile_name* must be an alphanumeric string representing the corresponding NF client profile name.
- **query-params { dnn | limit | max-payload-size | requester-snssais | supi | tai | target-nf-instance-id | target-plmn }**: Specify one of the following query parameters to include in the NF discovery request towards the NRF.
  - **dnn**: Specify the DNN as the query parameter in the NF discovery request towards the NRF.
  - **limit**: Specify the limit for the maximum number of profiles that the NRF sends in the NF discovery response.
  - **max-payload-size**: Specify the maximum payload size as the query parameter in the NF discovery request towards the NRF.
  - **requester-snssais**: Specify the list of Single Network Slice Selection Assistance Information (S-NSSAIs) as the query parameter in the NF discovery request towards the NRF.
  - **supi**: Specify the SUPI as the query parameter in the NF discovery request towards the NRF.
  - **tai**: Specify the TAI as the query parameter in the NF discovery request towards the NRF.
  - **target-nf-instance-id**: Specify the target NF instance identifier as the query parameter in the NF discovery request towards the NRF.
  - **target-plmn**: Specify the target PLMN as the query parameter in the NF discovery request towards the NRF.

- This is an optional configuration. By default, the CLI commands are disabled.
- To disable this configuration, use the **no** variant of these commands. For example, **no nf-client-profile** CLI command.

*Verifying the Local Configuration for the NRF Interface Per Endpoint*

This section describes how to verify the configuration for the NRF interface per endpoint.

The following is an example of the NRF endpoint configuration.

```
show running-config profile dnn cisco
profile dnn cisco
 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

profile smf smf1
 node-id          12b888e1-8e7d-49fd-9eb5-e2622a57722
 locality         LOC1
 bind-address ipv4 209.165.200.227
 bind-port        8008
instances 1 fqdn cisco.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
```

# Caching for Discovered NF Profiles

## Feature Description

The SMF provides caching support for discovered caching profiles. It uses the NF discovery (nnrf-disc) function to discover profiles, such as SMF, UDM, PCF, and CHF. The received discovery response is associated with validity time. SMF caches the discovery response and uses the same response for future NF selections until the cache is valid. This caching support helps in reducing the number of NRF interactions during an ongoing session.

### Relationships

Caching support for NF Discovery has functional relationship with the following features:

- NRF Support for SMF Subscription and Notification
- NRF Selection per Peer NF Type

## How it Works

The SMF maintains the cache data in a cache pod. It uses the cache pod to share the NF discovery cache across multiple instances of SBI pods. The SBI pod periodically updates the cache pod on receiving an NF discovery response. All SBI pods refresh its cache data periodically with the help of the cache pod.

If a message is sent to an NF that meets a specific criterion, the SMF looks up the cache data for further processing. During a cache lookup:

- On a cache hit without an expired entry, the selected cached NF response sends a message for an endpoint selection.
- On a cache hit with an expired entry, the SMF sends NF discovery requests to the NRF to fetch a new list of NF discovery responses.
- If there is a cache miss, the SMF sends NF discovery request to the NRF to retrieve a new list of NF discovery responses.

### Call Flows

#### *Cache Lookup Call Flow*

This section describes the call flow for Cache Lookup.

SMF maintains a local cache and updates the external cache (cache-pod). The key for a cache is a combination of nfType and filter, which is a string that is prepared from multiple filter parameters in "key1=value, key2=value2" format.

On startup, SMF retrieves all the cache entries that were modified since epoch from cache-pod so that it can build the local cache. After the local cache is built, the same cache is used in the send message flow for lookup. A periodic refresh routine is initiated to refresh the local cache using the cache-pod. Local cache is periodically refreshed by getting all records from the cache-pod that were modified since last refresh. The resultant record list is traversed and the local cache is updated.

When SMF-rest-ep (SBI) triggers a send message to UDM, the SMF looks up the local cache for the cache entry with the nfType and filter key. The NF profiles are load-balanced and a message is sent to the selected endpoint.

### Standards Compliance

This feature complies with the following standards:

- *3GPP TS 29.510 version 15.4.0 — 5G System; Network function repository services; Stage 3*

# NF Discovery Cache Invalidation

## Feature Description

SMF gives higher priority to the NFs that are discovered from NRF over the locally configured NFs. SMF uses the locally configured NFs only if the NRF endpoints are not configured or if no NFs are available as part of the NF discovery response. This response appears after the query filter criteria are met. Each NF discovery response has an associated validity time and SMF caches the NF discovery response, uses the cache for subsequent session activation. SMF performs NF discovery only if matching entries are unavailable for the query filter in its NF discovery response cache or if the entry in cache exists, however with the expired validity.

## How it Works

SMF provides configuration to determine the behaviour when the NRF is unreachable and has an expired cache entry. The CLI provides the following options to determine:

- If the cache entry needs to invalidate on expiry.
- If the cache entry needs to be invalidated, along with the duration to retain the cache entry after the validity expiry.

These options are applicable only if the NRF is inactive. The configuration is according to the nf-pair profile. The configuration determines if SMF should use expired cache in case NRF becomes inactive and if it uses the expired cache, along with the duration to retain the cache entry after the validity expiry.

## Configuring NF Discovery Cache Invalidation (Purge)

To configure the cache entry invalidation (purge) for the NF discovery cache, use the following sample configuration.

```
config
  profile nf-pair nf-type { amf | chf | pcf | udm }

    cache invalidation { false | true [ timeout integer ] }
    end
```

**NOTES:**

- **cache invalidation { false | true [ timeout** *integer* **] }**: Configure the interval and cache invalidation rule. The default value is false.
  - **false**: Specify that the cache entry will never be invalidated.
  - **true timeout** *integer* : Specify that the cache entry will be invalidated. **timeout***integer* specifies the time period in milliseconds (ms) for controlling the usage of the expired cache entry (when NRF is unreachable). The default value is 0 ms.

The following configuration is an example that sets the cache invalidation to false for the UDM discovery:

```
profile nf-pair nf-type UDM
  cache invalidation false
 end
```

The following configuration is an example that sets the cache invalidation to true for the UDM discovery:

```
profile nf-pair nf-type UDM
  cache invalidation true timeout 10
 end
```

# NF-Set Based Subscription for AMF Towards NRF

*Table 12: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| NF-Set based Subscription towards NRF | 2025.03.0 | SMF supports enhanced subscription criteria, enabling improved handling of NRF notifications for NF Set-based changes. Previously, SMF supported only `nfInstanceId`-based subscriptions. The enhanced subscription criteria improves SMF's ability to accurately track and respond to dynamic changes in NF Sets, such as peer additions or removals.<br>Command introduced:<br>**nf-subscription-params [nfSetCond]**— Used to configure NF Set-based subscription criteria.<br>**Default Setting**: Disabled – Configuration Required |

An **NF-Set-based subscription** is a mechanism used by SMF to subscribe to a group of Network Function (NF) instances that belong to a specific **NF Set**. An **NF Set** is a logical grouping of NF instances that share a common identifier, called the **NF Set ID**. This identifier is used to manage and monitor the group as a single entity, rather than handling each NF instance individually.

Key Characteristics of NF-Set-Based Subscription:

- **Group Subscription**: Instead of subscribing to individual NF instances, the subscription is applied to the entire NF Set. This is useful for managing NFs that are part of the same operational or administrative group.
- **Identification via NF Set ID**: Each NF Set is uniquely identified by an NF Set ID, which is used as the subscription condition. This ID enables the subscribing function to track changes within the group, such as additions or removals of NF instances.
- **Event-Based Notifications**: The subscribing NF receives notifications from the Network Function Repository Function (NRF) about changes in the NF Set, including:
  - **NF_ADDED**: When a new NF instance is added to the set.
  - **NF_REMOVED**: When an NF instance is removed from the set.
  - **NF_PROFILE_CHANGED**: When profile of a specific NF instance has been updated or modified.

- **Use Cases**: Commonly used in 5G networks to manage groups of NFs that belong to a specific NF Set.

## Usage Guidelines for NF-Set based Subscription

To use this features effectively:

- To enable this feature, 3GPP spec version of 16.9.0 must be configured for service nnrf-nfm and nnrf-nfd under the profile compliance configuration. See profile compliance for configuration details.
- Configure the subscription criteria under the profile network-element. The default criteria remain as NfInstanceId. See Configure NF-Set Based Subscription towards NRF, on page 44 for configuration details.
- Use `nfSetCond` as the subscription parameter for NF Sets, if applicable.
- SMF uses the configured criteria as the subscription parameter, regardless of the query parameter used for discovery, as long as configured criteria is present in the discovered response. If the configured parameter is unavailable, SMF uses the nfInstanceId as the subscription parameter.
- Update configurations at runtime for dynamic subscription management. Dynamic changes applies to the next subscription request not for the current or ongoing request.

## Restrictions for NF-Set based Subscription

While using this feature, consider these limitations:

- SMF supports only one NFSetId in NfProfile in discovery response and picks up the first one present in the list for subscription if nfSetCond is configured as the subscription parameter.
- NF Instances are not removed from the NF Set if the `NF_DEREGISTERED` event is received without condition event NF_REMOVED. However, SMF will not select these instances further.

## Configure NF-Set Based Subscription towards NRF

Use this task to configure NF-Set-based subscription parameters for the SMF to interact with the NRF.

### Before you begin

- Review the guidelines and limitations for configuring NF-Set subscriptions in the system.
- Ensure that you are connected to the SMF Ops Center to apply the configuration.

### Procedure

**Step 1** Enter the AMF profile configuration mode.

**`profile network-element amf`** *`profile_name`*

**Example:**

```
[smf] smf# config
[smf] smf(config)# profile network-element amf nfprf-amf1
```

**Step 2** Set the subscription parameters to nfSetCond.

**nf-subscription-params [nfSetCond]**

**Example:**

```
[smf] smf(config-network-element-amf-nfprf-amf1)# nf-subscription-params [ nfSetCond ]
```

**Step 3** Save and commit the configuration.

```
[smf] smf(config-network-element-amf-nfprf-amf1)# exit
```

This command allows you to either save or discard the configurations. Entering `yes` confirms and saves the configurations.

## Troubleshooting Information for NF-Set based Subscription

This section provide details about the show commands and bulk statistics that can be used to troubleshoot the issues.

### Show Commands

- `show nrf subscription-info`: Displays NfSetId when using `nfSetCond`.
- `show nrf discovery-info`: Displays NfSetIdList for AMFs with discovery profiles.

### Bulk Statistics Output

- The existing metric "nrf_subscription_send_messages_total" is enhanced by adding a new label, **SubscrCond**, which indicates either **"NfInstanceId"** or **"NfSetCond"** based on the subscription request.
- The existing metric **"nf_management_stats_total"** is enhanced with two new labels: **"notification_event_type"** and **"condition_event_type"**. Based on the received notification, **"notification_event_type"** reflects one of the following values: **"NF_REGISTERED"**, **"NF_DEREGISTERED"**, or **"NF_PROFILE_CHANGED"**, while **"condition_event_type"** indicates either **"NF_ADDED"** or **"NF_REMOVED"**.

# Selection of Alternate AMF

*Table 13: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| Selection of Alternate AMF | 2024.02.0 | This feature allows the SMF to choose an available AMF from the set when the AMF connected to the UE experiences an outage. SMF performs AMF selection based on the configuration of NRF query parameters or NRF query response local filters.<br><br>This feature addresses the need for uninterrupted service continuity in the event of an AMF becoming unavailable.<br><br>This feature introduces the following new CLI commands:<br>• **filter-discovery-response filter match { all \| any } attributes { target-nf-instance-id }**<br>• **filter-discovery-response filter failure-action { use-discovery-response }**<br><br>These configurations allow SMF to locally filter the received NRF discovery response and select the appropriate one which matches the configured filters.<br><br>This feature additionally supports region and set ID configuration as part of NRF query parameters.<br><br>**Default Setting**: Disabled – Configuration required to enable |

## Feature Description

The AMF selection feature addresses the critical need for uninterrupted service continuity in the event of an AMF becoming unavailable. This feature allows the SMF to choose an available AMF from the set when the AMF connected to the UE experiences an outage. SMF performs AMF selection based on the configuration of NRF query parameters or NRF query response local filters.

## Configuring Discovery Response Filter for AMF Query Parameters

To configure the discovery response filter for AMF query parameters, use the following sample configuration:

```
config
  profile network-element amf amf_profile_name
     query-params [ target-nf-instance-id | region-set ]
     filter-discovery-response filter match { all | any } attributes {
target-nf-instance-id }
     filter-discovery-response filter failure-action {
use-discovery-response }
     end
```

**NOTES:**

- **query-params [ target-nf-instance-id | region-set ]** —Specify the target NF instance ID or region set as query parameters. The **region-set** is combination of Region ID and Set ID.

  When you configure a **region-set** query parameter. SMF does NF Discovery with **amf-set-id** and **amf-region-id**. AMF learns **set-id** and **region-id** from GUAMI received in N11 create or N11 update request from AMF.

- **filter-discovery-response filter match { all | any } attributes { target-nf-instance-id }**—This filter controls the discovery response by matching 'all' or 'any' of the specified attributes.

  When you configure a local response filter parameter with **match all** option. SMF tries to filter the profiles discovered from NRF with the preferred parameters. SMF selects one of the profiles which matches the parameters. In the case of no match, AMF selection fails.

  When you configure the target-nf-instance-id as a local query parameter. SMF does NF Discovery with query params. SMF filters the discovered profile list with target-nf-instance-id = last known nf-instance-id of the AMF to select the final AMF.

- **filter-discovery-response filter failure-action { use-discovery-response }**—Determines the action to take when the filter fails.

  When you configure the local response filter parameters, SMF filters the profiles discovered from NRF with the local parameters and selects the final peer instance. If the parameters don't match, then SMF chooses one of the profiles discovered from NRF. SMF verifies if the failure-action configuration is available with **use-discovery-response** option otherwise; AMF selection fails.

# Static Configuration for Peer NF Management

## Fallback to Static IP Address Support

### Feature Description

The SMF follows a priority order for different NF selection options. The SMF prioritizes the NF discovered from the NRF over the local configuration. The SMF uses the locally configured NFs when the NF discovery response has no valid NFs.

Depending on the deployment, the preferred server and geo locality server are configured for each of the NFs. The general rule is to select NFs in the preferred server locality followed by NFs in the geo locality server in case the preferred server NFs fail.

For each NF, the SMF provides an option to configure preferred and geo server locality through the **profile nf-pair** parameter. For more details, see Configuring Locality for NF Types, on page 32 in the NRF Selection per Peer NF Type, on page 37 section.

In addition, each NF discovery response comes with associated validity time. The SMF caches this NF discovery response and uses it to fetch subsequent sessions.

The SMF performs the NF discovery in the following conditions:

- The NF discovery response cache has no matching entries.
- The NF discovery response cache has matching entries, but the validity has expired.

### Relationships

The Fallback to Static IP Address feature has functional relationships with the following features:

- Caching Support for NF Discovery
- NF Discovery, NF Selection, and Load Balancing
- NRF Selection per Peer NF Type

## How it Works

The SMF follows this sequence for NF selection if an NRF discovery group is configured:

1. It looks up the local cache (NF discovery response cache) for the NF.
2. If the NF is a valid entry (not expired), it uses that entry. Else, SMF proceeds to Step 3.
3. The SMF reaches NRF for discovery [see, NRF Discovery (Priority 1)]. Else, SMF moves to Step 4.
4. If SMF cannot use the NRF for discovery, it uses the expired NF cache [see, Expired NF Cache ( Priority 2)]. If expired NF cache is not available, SMF moves to Step 5.
5. If SMF does not find the NF in the local cache nor is it able to get it in the NRF discovery response, it uses the locally-configured NF [see, NF Local configuration (Priority 3)].

The priority order for NF selection is as follows:

1. **NRF Discovery (Priority 1)**

   The SMS uses the NRF-provided, NF discovery service to discover NFs like SMF, UDM, and PCF. The SMF sets the preferred locality as provided in the "**profile nf-pair**" configuration in the discovery query. (For more details about the "**profile nf-pair nf-type**" CLI configuration, see Configuring Locality for NF Types, on page 32 in the NRF Selection per Peer NF Type, on page 37 section.) For each NF, the query parameters are configurable. (For more details, see Configuring Network Element Profile Parameters for the NF, on page 39 in the NRF Selection per Peer NF Type, on page 37 section). The NRF returns all the NFs matching the query criteria. When available, the NRF prefers NF profiles with a locality attribute that matches the preferred-locality. The NRF could return more NFs in the response, which are not matching the preferred target NF location. This occurs when there is no NF profile that is found matching the preferred target NF location. To avoid this, the NRF could set a lower priority for any additional NFs on

the response not matching the preferred target NF location than those matching the preferred target NF location. The locality-aware NF selection logic of SMF is as follows:

**a.** If the NF has both the preferred and geo locality server configurations, all the NFs in the response that are matching these are cached. SMF ignores the balance NFs. The load-balancing logic first selects the preferred locality NFs. If the preferred locality NFs fail, SMF picks the geo locality NFs for a retry. If N retry is allowed, N-1 retries are on the preferred locality and the last retry is on the geo locality NF. If the N-1 endpoints are unavailable in the preferred locality, SMF attempts all the endpoints of the preferred locality. Else, SMF picks up the geo locality endpoints for the remaining retries. Multiple retries on the same host (port) is not attempted.

**b.** If the NF has only the preferred locality configuration, all the NFs in the response that match the preferred locality are cached. The load-balancing logic selects the endpoints from these NFs.

**c.** If the NF does not have the preferred locality or geo locality configuration, then SMS caches all the discovery response NFs. The load-balancing logic selects from these NFs.



**Note**

- The load-balancing logic is based on priority, capacity, and load. The logic is similar to server selection as defined in IETF RFC 2782. However, the weight is considered as "capacity * (100 - load)".
- If SMF selects the NRF-discovered NFs (in any of the three cases), even when all attempts to reach preferred and geo locality fail, the SMF does not fall back to the local configuration NFs for a retry.

**2. Expired NF Cache (Priority 2)**

The SMF performs an NF discovery only in the following scenarios:

- If the matching entries are not available for the query filter in its NF discovery cache
- If matching entries are available in its NF discovery cache. However, these entries have expired validity.

  The retention of an expired cache entry is configuration-based. If the expired cache entry is available and the NRF is not reachable or returns an error, then SMF uses the expired cache entry for NF selection. You can configure the SMF to control the cache entry usage with the following options:

  - Invalidate the cache entry on expiration of validity.
  - Use the invalidated cache entry for a configurable time period (timeout) and fallback to the static configuration after the timeout expires.



**Note** The SMF controls the cache entry usage - only when the NRF is down - through these options. The configurations are based on the **profile nf-pair**. Additionally, the SMF provides flexibility in configuring different cache usage rule for different NFs. For instance, the SMF always uses the expired cache to discover PCF when the NRF is down. But, for discovering the UDM, the SMF uses the expired cache for a timeout period of 10 milliseconds (ms) when the NRF is down.

**3. NF Local Configuration (Priority 3)**

The locally configured NFs are the last option for NF endpoint selection. The local configuration too considers the preferred and geo server locality for NF selection. The priority order is as follows:

**a.** If the preferred server is configured for the NF [ in **profile nf-pair** ], SMF selects the NF endpoints under the preferred locality, first. The load-balancing logic is applicable for endpoint profiles and endpoints within the locality as per the configured priority and capacity values.

**b.** If the geo locality is configured for the NF [ in **profile nf-pair** ], SMF selects the NF endpoints under the geo locality as the fallback option. That is, if the preferred server locality NF endpoints fail or preferred server locality endpoints are not configured. The load-balancing logic is applicable for endpoint profiles and endpoints within the locality as per the configured priority and capacity values.

**c.** If the preferred server and geo locality server are not applicable, SMF picks up the locality based on the priority that is configured for each locality in the local NF configuration. The load-balancing logic is applicable for endpoint profiles and endpoints within the locality as per the configured priority and capacity values.



**Note** The priority under locality is applicable only if the preferred and geo locality servers are not applicable.

The failure template is configurable for each of the NFs. Also, the message type in the template can set the retry count and action for the possible HTTP return codes.

### Standards Compliance

The Fallback to Static IP Address feature complies with the following standards:

- *3GPP TS 29.510 version 15.4.0 (2019-07) – 5G System; Network function repository services; Stage 3*

## Configuring Fallback to Static IP Address

This section describes how to configure the support for Fallback to Static IP Address.

### Configuring NF Client Profile

To configure the NF endpoints for AMF, CHF, PCF, and UDM, use the following sample configuration:

```
config
  profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf
pcf-profile | udm udm-profile } nf_profile_name }
  end
```

**NOTES:**

- **profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf pcf-profile | udm udm-profile }** *nf_profile_name* **}**: Specify the required NF client profiles and provide the local configuration for any of the following configured NFs:
  - **amf**: Enable the AMF local configuration
  - **chf**: Enable the CHF local configuration
  - **pcf**: Enable the PCF local configuration
  - **udm**: Enable the UDM local configuration

For example, if you are configuring the **amf amf-profile** keyword, this command enables the AMF local configuration. The same approach applies for the other configured NFs.

*nf_profile_name* must be an alphanumeric string representing the corresponding NF client profile name.

- You can configure multiple NF profiles within a given service.
- To disable the configuration, use the **no profile nf-client { nf-type { amf amf-profile | chf chf-profile | pcf pcf-profile | udm udm-profile }** *nf_profile_name* **}** command.

### *Configuration Example*

The following is an example configuration.

```
profile nf-client nf-type pcf
pcf-profile pcf-profile
  locality LOC1
   priority 1
   service name type npcf-smpolicycontrol
    endpoint-profile epprof
     capacity   10
     priority   1
     uri-scheme http
     endpoint-name ep1
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.202.133
      primary ip-address port 8080
     exit
     endpoint-name ep2
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.201.1
      primary ip-address port 8080

     exit
    exit
   exit
  exit
exit
exit
profile nf-client nf-type pcf
pcf-profile pcf-profile
  locality LOC1
   priority 1
   service name type npcf-smpolicycontrol
    endpoint-profile epprof
     capacity   10
     priority   1
     uri-scheme http
     endpoint-name ep1
      priority 1
      capacity 10
      primary ip-address ipv4 209.165.201.2
      primary ip-address port 8080
      fqdn name nrf.cisco.com
      fqdn port 9010
     exit
```

### Configuring Network Element Profile Parameters for the NF

To configure the network element profile parameters for the configured NF, use the following sample configuration.

```
config
  network-element-profiles { { amf | chf | pcf | udm } nf_profile_name }

    nf-client-profile profile_name
    query-params { dnn  | limit | max-payload-size | requester-snssais
| supi | tai | target-nf-instance-id | target-plmn }
    end
```

**NOTES:**

- **nf-client-profile** *profile_name*: Specify the local NF client profile. *profile_name* must be an alphanumeric string representing the corresponding NF client profile name.
- **query-params { dnn  | limit | max-payload-size | requester-snssais | supi | tai | target-nf-instance-id | target-plmn }**: Specify one of the following query parameters to include in the NF discovery request towards the NRF.
  - **dnn**: Specify a DNN as the query parameter in the NF discovery request towards the NRF.
  - **limit**: Specify a limit for the maximum number of profiles that the NRF sends in the NF discovery response.
  - **max-payload-size**: Specify the maximum payload size as the query parameter in the NF discovery request towards the NRF.
  - **requester-snssais**: Specify the list of Single Network Slice Selection Assistance Information (S-NSSAIs) as the query parameter in the NF discovery request towards the NRF.
  - **supi**: Specify a SUPI as the query parameter in the NF discovery request towards the NRF.
  - **tai**: Specify a TAI as the query parameter in the NF discovery request towards the NRF.
  - **target-nf-instance-id**: Specify a target NF instance Identifier as the query parameter in the NF discovery request towards the NRF.
  - **target-plmn**: Specify a target PLMN as the query parameter in the NF discovery request towards the NRF.
- This is an optional configuration. By default, the CLI commands are disabled.
- To disable the configuration, use the **no** variants of these commands. For example, **no nf-client-profile** CLI command.

# NRF Failure Handling

## Feature Description

SMF uses the NF registration messages for tracking the liveliness of management NRF group. If SMF detects a failure in one of the NRFs in the management group, it uses the NRF failure handling mechanism.

Failure handling template is available for each of the NFs and its message types to set the retry count and action for the possible HTTP return codes.

For more information on failure handling, see the Failure Handling Support chapter.

# Discovery Cache Optimization for Range-based Query Parameters

*Table 14: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| Discovery cache optimization for range-based query Parameters | 2026.02.0 | With this release, SMF supports discovery cache optimization for Range-based Query Parameters. An intelligent caching mechanism is introduced that allows the system to store and reuse NF profiles based on ranges of identifiers (such as SUPI ranges) rather than individual identity entries. This feature improves the efficiency of Network Function (NF) discovery within the Cisco NF environment. |

### Information About Discovery Cache Optimization for Range-based Query Parameters

**What:** This feature introduces an intelligent caching mechanism that allows the system to store and reuse NF profiles based on ranges of identifiers (such as SUPI ranges) rather than individual identity entries.

**How:** When an NFLib triggers a discovery request, the system checks if an existing discovered profile already covers the requested SUPI range. If a valid profile is found in the local cache, the system reuses that profile instead of initiating a new discovery request to the Network Repository Function (NRF).

**Why:** Previously, the system created a unique cache entry for every single SUPI, leading to redundant entries, increased memory consumption, and unnecessary CPU cycles spent on repeated discovery requests. This optimization reduces memory overhead and improves overall network performance by minimizing redundant communication with the NRF.

In the standard discovery process, NFs register with the NRF with specific parameters, such as SUPI ranges. Previously, because discovery occurred on a per-SUPI basis, the cache could become saturated with duplicate entries for profiles that actually served a broader range of users. This feature enables the local cache to recognize and store these ranges, allowing subsequent discovery requests for different SUPIs within the same range to be satisfied by the existing cached profile. This ensures that the system remains performant even as the number of discovery transactions increases.

### Benefits of Discovery Cache Optimization for Range-based Query Parameters

- **Optimized Memory Usage:** Prevents the creation of duplicate cache entries by grouping identities into ranges.

- **Reduced CPU Load:** Decreases the number of redundant discovery requests sent to the NRF, as existing profiles are reused for new requests within the same range.
- **Improved Performance:** Enhances the speed of NF discovery by resolving queries locally from the cache whenever possible.
- **Enhanced Visibility:** Provides better insights into discovered profiles through updated CLI commands. Also provides detailed metrics and CLI commands to monitor discovery efficiency.

### Configuration Overview

This feature is enabled by default when the NRF returns profiles with defined SUPI ranges. You can manage and monitor the cache using the following CLI commands:

1. Use the `show nrf discovery-info` command to view the discovered profiles, now enhanced to display SUPI ranges.
2. Use the dedicated command handler to fetch and display the range-based cache entries.
3. Configure debug-level metrics for advanced monitoring. See Monitor Discovery Cache Optimization for Range-based Query Parameters. See Monitor Discovery Cache Optimization for Range-based Query Parameters.

### Supported Scenarios

This feature is supported in scenarios where:

- The NF discovery query parameters include a SUPI.
- The discovered profiles from the NRF contain SupiRanges with explicit "start" and "end" attributes.
- The deployment is running on supported versions (15.4.0 and 16.9.0).

### Prerequisites for Discovery Cache Optimization for Range-based Query Parameters

- The NRF server must be configured to return profiles that include SupiRanges with defined "start" and "end" attributes.
- The discovery query-parameter must include the SUPI attribute.

### Restrictions for Discovery Cache Optimization for Range-based Query Parameters

- **Pattern Attributes:**

  This feature does not support **pattern** attributes within SupiRanges. If the NRF returns a profile with a pattern attribute, it will be stored in the standard discovery cache without range-based optimization.

- **Version Dependency:**

  Optimization is only applied when the discovery query-parameter contains a SUPI and the discovered profiles meet the required attribute criteria.

- **No Range in Profile:**

  If the NRF returns a profile without a SUPI range, it is only stored in the standard discovery cache.

### Configure Discovery Cache Optimization for Range-based Query Parameters

There are no manual configuration steps required to enable the optimization logic itself, as it is handled automatically by the NfLib. To monitor or verify the cache state, you may use the following commands:

1. **Display Discovery Information:**

   The `show nrf discovery-info` command displays the discovered profiles and their associated SUPI ranges.

   Ensure that the output displays the `SupiRanges` for the discovered profiles. If the profiles are correctly indexed, subsequent discovery requests for SUPIs within those ranges should not trigger new NRF discovery traffic.

2. **Collect Cache Dump:**

   Refer to the NRF-Lib Local Cache documentation for the specific command to dump the range-based cache entries for debugging or verification purposes.

### Monitor Discovery Cache Optimization for Range-based Query Parameters

You can monitor the effectiveness of the cache using the following metrics:

- **`nf_discover_events_total`**:

  This existing metric has been updated with a new `intervaltree-cache` value for the `response_type` label. This indicates that a discovery response was successfully served from the IntervalTree cache.

- **`nf_rangebased_discovery_total`**:

  This new metric tracks Add, Update, and Remove operations on the IntervalTree cache. It includes labels for `range_type` (e.g., supi), `nf_type` (e.g., UDM), `operation` (add/update/remove), `message_type` (e.g., discovery, cache-refresh), and `status`.

Here is a sample metric output:

```
nf_rangebased_discovery_total{range_type="supi",nf_type="UDM",
   operation="add",message_type="discovery",status="success"} 150
```

You can collect a cache dump using the command handler provided for the NRF-Lib local cache to verify that range-based entries are being managed correctly.