NRF Discovery

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
First introduced.	Pre-2020.02.0

Feature Description

The 3GPP-defined architecture model for 5G systems provides data connectivity based on techniques such as network function virtualization, software defined networking, and service-based interfaces. Some of the key principles are:

- Separate the User Plane (UP) functions from the Control Plane (CP) functions allowing independent scalability, evolution, and flexible deployments, such as centralized location or distributed (remote) location.
- Support "stateless" NFs where the "compute" resource is decoupled from the "storage" resource.

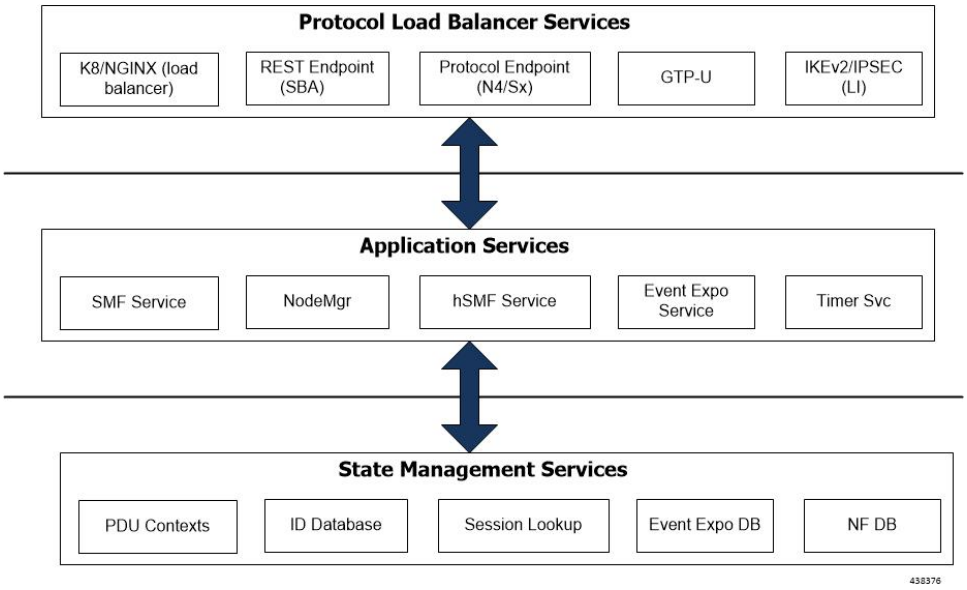
This feature discovers the set of NF instances (and their associated NF service instances), represented by their NF profile, that are currently registered in Network Repository Function (NRF) and satisfy several input query parameters.

Architecture

The SMF NF comprises of loosely coupled microservices. Microservice decomposition is based on three-layered architecture philosophies:

1. Layer 1: Protocol and Load Balancer Services (Stateless)
2. Layer 2: Application Services (Stateless)
3. Layer 3: Database Services (Stateful)

Figure 1: SMF 3-layered Micro Services Architecture



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 NF service consumer.

Call Flows

The Service Discovery Request call flow described in 3GPP TS 29.510 v15.2.0 illustrates the NF-level messages for NF discovery.

Service Discovery Request Call Flow

This section describes the Session Discovery Request call flow.

Figure 2: Service Discovery Request Call Flow

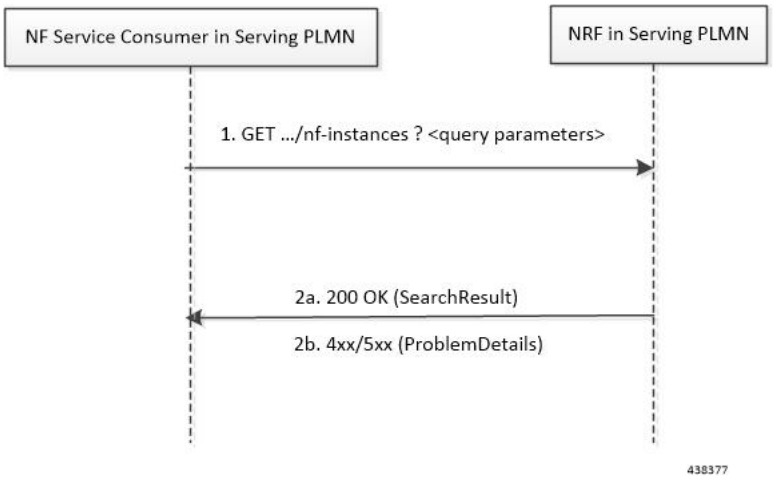


Table 3: Service Discovery Request Call Flow Description

Step	Description
1	The NF Service Consumer sends an HTTP GET request to the resource URI "nf-instances" collection resource. The input filter criteria for the discovery request is included in query parameters.
2a	On success, "200 OK" is returned. The response body contains a validity period, during which the search result can be cached by the NF Service Consumer, and an array of NF profile object that satisfy the search filter criteria (for example, all NF Instances offering a certain NF Service name).
2b	If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF returns "403 Forbidden" response. 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. 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, and it can also contain NF-specific data, for those NF instances belonging to a specific type (for example, the attribute "udrInfo" is typically present in the NF profile when the type of the NF instance takes the value "UDR"). In addition, the attribute "customInfo" can be present in the NF profile for NF instances with custom NF types. For NF instances, the "customInfo" attribute is returned by NRF, if available, as part of the NF profiles returned in the discovery response.

SMF service communicates with different NFs, such as UDM, AMF, PCF, CHF and so on, when the session is brought up. The NF discovery is based on set of filters that are associated with the session. The SMF service discovers the NFs, matching the filter criteria for the session, to send messages to NF.

NRF Library (NRF-LIB) provides APIs to discover and send a message to an NF matching a set of filter parameters. The NRF-LIB performs NF discovery for the filter and caches the discovered NFs in a local cache. The following filter parameters are supported:

- 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.

NRF-LIB sends the messages to a new target based on the Location header URL in response to initial messages sent to NF.

NRF-LIB 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 App/Rest-Ep so that the same can be specified in subsequent message (instead of discovery filter). This helps maintaining stickiness for a session to a selected NF.

Standards Compliance

The NF Discovery feature complies with the following standards:

- 5G System; Network Function Repository Services; Stage 3 (Release 15):
  - 3GPP TS 29.510 version 15.0.0 (2018-06)
  - 3GPP TS 29.510 version 15.2.0 (2018-12)

Limitations

The following are known limitations of this feature:

- The cache maintained is local to the library. Therefore, in case of deployment with multiple replicas of Rest-Ep, if two Discovery/Send messages with the same discovery filter land on different pods, then the NF discovery will be triggered by both pods.
- Only UDM, PCF, CHF, and AMF discovery and load balancing are supported. UPF discovery is not supported.
- Dynamic configuration changes of NRF endpoints are not supported.

NF Heartbeat Support

Feature Description

The NF heartbeat implementation helps the NFs to notify the NRF that the NF is operational. Each NF registered with the NRF contacts the NRF periodically by invoking the NFUpdate service operation. The time interval at which the NRF is contacted is deployment-specific and is returned by the NRF to the NF Service Consumer as a result of a successful registration.

How it Works

Call Flows

NF Heartbeat Procedure

The following figure illustrates the NF Heartbeat call flow.

Figure 3: NF Heartbeat Call Flow

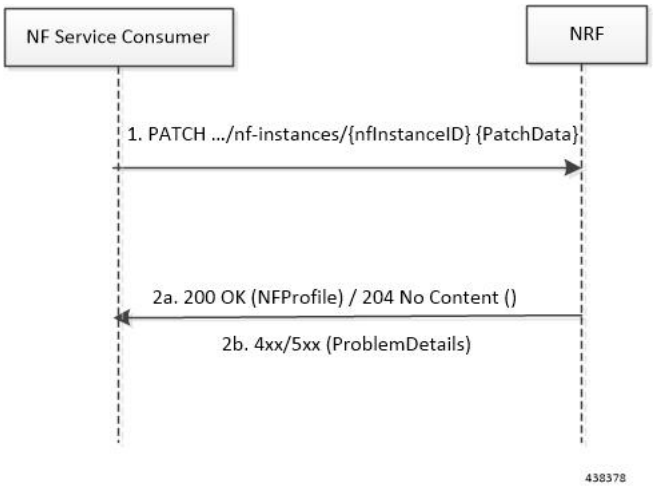


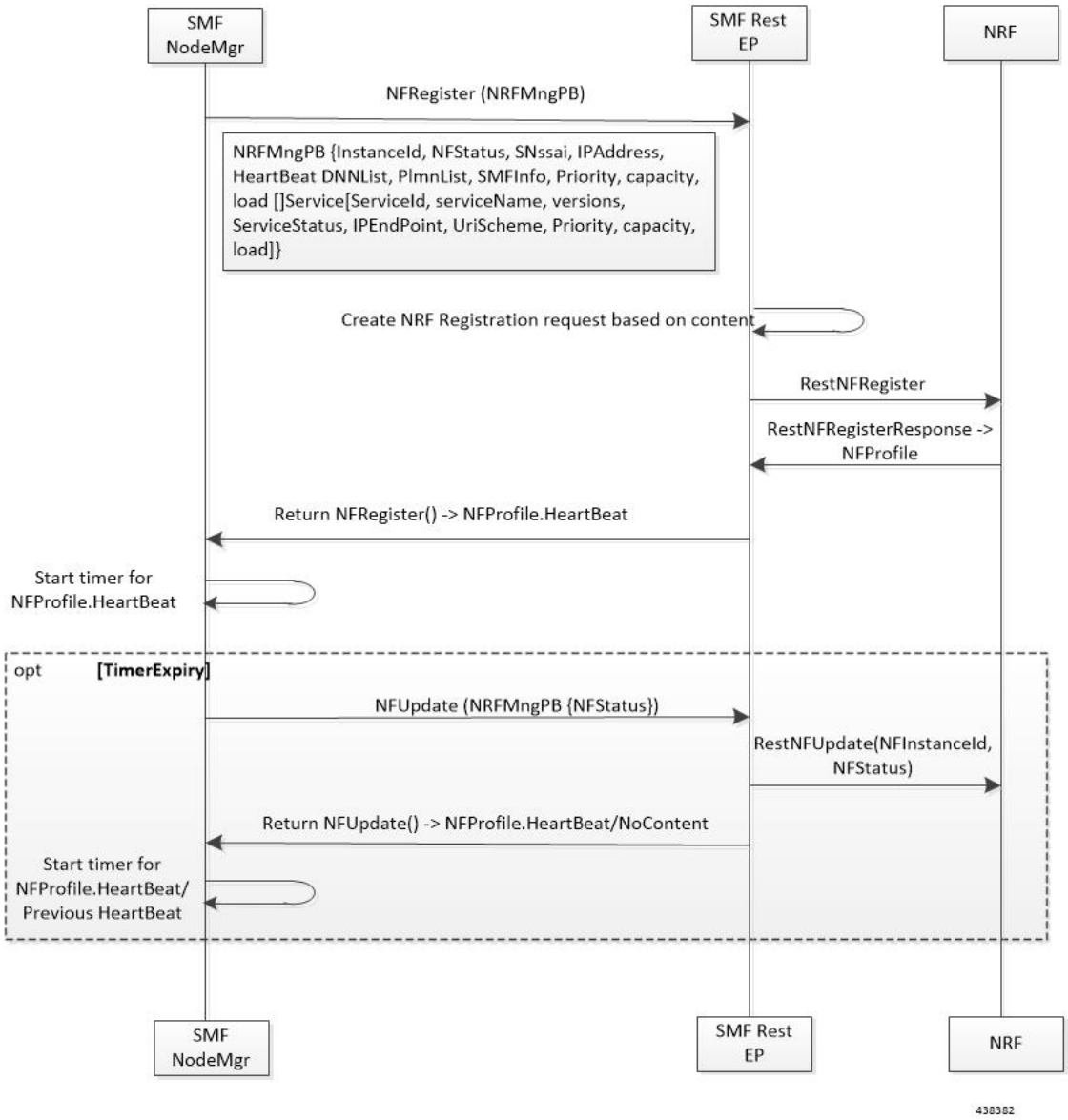
Table 4: NF Heartbeat Call Flow Description

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.
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: **PATCH .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64** **Content-Type: application/json-patch+json** **[** **{"op": "replace", "path": "/nfStatus", "value": "REGISTERED"}** **]** **HTTP/2 204 No Content** **Content-Location: .../nf-instances/4947a69a-f61b-4bc1-b9da-47c9c5d14b64**

NRF Heartbeat Internal Call Flow

The following figure shows the internal call flow related to the NRF Heartbeat feature.

Figure 4: NRF Heartbeat Internal Call Flow



The SMF NF heartbeat implementation helps in notifying the NRF that the SMF is operational. The default heartbeat frequency is once in 10 seconds. If the NRF returns a different heartbeat frequency, the same is used for the periodic heartbeat. As part of the heartbeat, HTTP PATCH Request to the resource URI representing the NF instance is sent to the NRF. The payload body of the PATCH Request contains a "replace" operation on the "nfStatus" attribute of the NF profile of the NF instance, and sets it to the value "REGISTERED". Other parameters like load and capacity are not supported in this release.

Like NF registration, NF heartbeat is also triggered from the elected master node manager. Also, the heartbeat continues even on the elected node manager restart.

Standards Compliance

The NF Heartbeat feature complies with 3GPP TS 29.510, Version 15.2.0.

Caching Support for NF Discovery

Feature Description

The SMF provides caching support for discovered caching profiles. It uses the NF discovery (nnrf-disc) function to discover profiles such as AMF, 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 Interface Per Endpoint

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 refreshes its cache data periodically with the help of the cache pod.

Currently, the SMF does not invalidate the NF discovery cache entry even on the expiration of the validity time. If a message is sent to a NF that meets a specific criterion, the SMF looks up the cache data for further processing. During a cache look-up:

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

Call Flows

Cache Lookup Call Flow

This section describes the Cache Lookup call flow.

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

Figure 5: Caching Support Call Flow

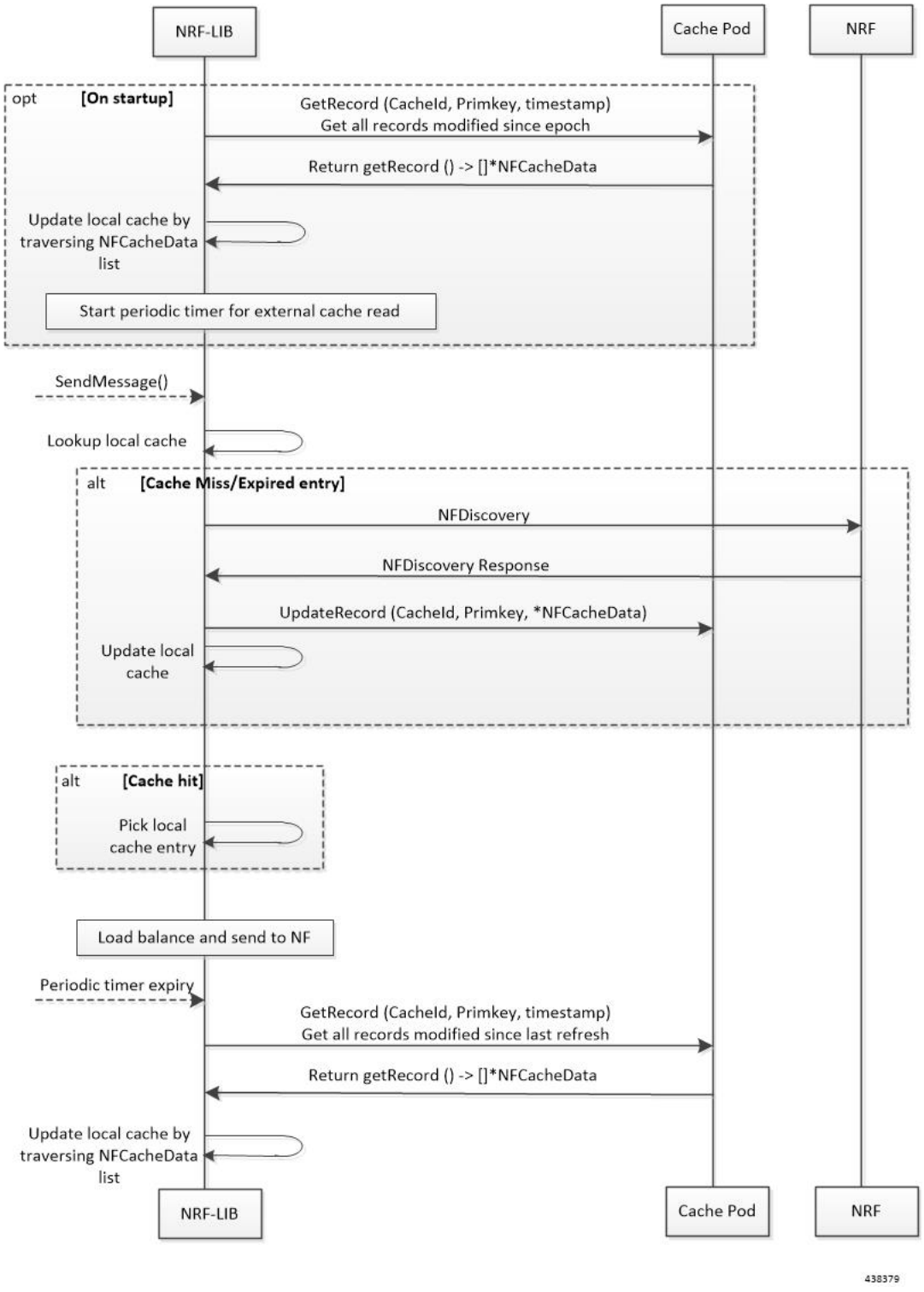


Table 5: Caching Support Call Flow Description

Step	Description
1	On Startup, NRF-LIB (in smf-rest-ep/SBI): **1.** Retrieves all the cache entries that were modified since epoch from cache-pod so that it can build the local cache. Once the local cache is built, the same cache is used in the sendmessage flow for lookup. **2.** A periodic refresh routine is initiated to refresh the local cache using the cache-pod.
2	Periodic Refresh, NRF-LIB (in smf-rest-ep/SBI): 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.
3	Send Message NRF-LIB (in smf-rest-ep/SBI): **1.** When smf-rest-ep (SBI) triggers a send message (say to UDM), NRF-LIB looks up the local cache for the cache entry with the nfType and filter key: **a.** When a cache lookup miss occurs, a discovery query is sent to NRF to fetch NF profiles from NRF. If NRF responds with NF profiles, then these NF profiles are stored in a local cache and updated in cache-pod. **b.** On a successful lookup, the cached entry is used to send a message for endpoint selection. **2.** The NF profiles are load-balanced, and a message is sent to the selected endpoint.

Standards Compliance

The Caching Support for NF Discovery feature complies with the following standards:

- 3GPP TS 29.510, V15.2.0
- 3GPP TS 29.510, V15.0.0

Limitations

The Caching Support for NF Discovery feature has the following limitations:

- This feature only supports UDM, PCF, AMF discovery, and load-balancing.
- It does not support UPF discovery.
- It does not support Dynamic Configuration changes of NRF endpoints.
- It does not support Liveliness check of the NRF endpoints.
- NF Discovery is always attempted on primary host followed by the secondary and then tertiary host.

NRF Support for SMF Subscription and Notification

Feature Description

The SMF uses the NRF-provided Subscribe service to subscribe to NF status changes that the NF receives as a discovery response. This 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.

How it Works

The NRF Support for 5G-SMF Subscription and Notification 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 provides the call flows for this 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. The NRF can assign and provide a new validity time to the NF, if the operation is successful.

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

Figure 6: Subscription to NF Instances in the Same PLMN

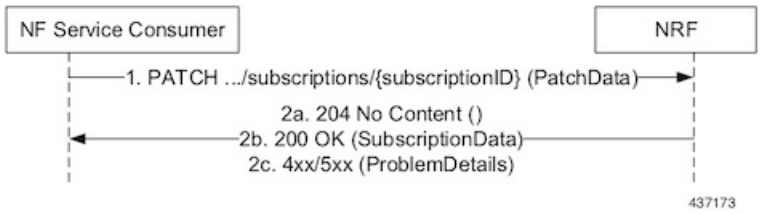


Table 6: Call Flow Description for Subscription to NF Instances in the Same PLMN

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 indicates that the NRF accepts: • Extension of the lifetime of the subscription • Value of the "validityTime" attribute
2b	The NRF returns a "400 Bad Request" status code with the problem details if the subscription fails due to errors in the JSON Patch object in the request body.

2c	The NRF returns a "500 Internal Server Error" with the problem details if the subscription fails due to internal errors in the NRF.
	Example:
	`PATCH .../subscriptions/2a58bf47`
	`Content-Type: application/json-patch+json`
	`[`
	`{ "op": "replace", "path": "/validityTime", "value": "2018-12-30T23:20:50Z"`
	`},`
	`]`

Subscription(POST) Call Flow

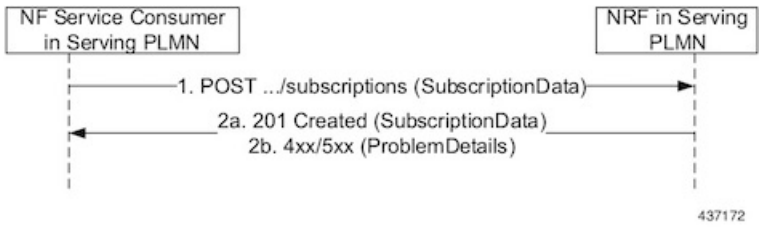
The Subscription service operation allows to:

- Create a subscription such 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 subscription of NF instances that the NF receives as its discovery response.

Figure 7: Subscription to NF Instances in the Same PLMN



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 7: Call Flow Description for Subscription to NF Instances in the Same PLMN

Step	Description
1	The NF Service Consumer sends a POST request to the resource URI representing the "subscriptions" collection resource. 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. 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 attributes of the profile.
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	The NRF returns a "400 Bad Request" status code with the problem details if the subscription fails due to errors in the subscription data. The NRF returns a "500 Internal Server Error" with the problem details if the subscription fails due to internal errors in the NRF.

NFStatus Notify Call Flow

Issuing a POST request to each callback URI of the various subscribed NF Instances, initiates the NFStatus Notify operator.

Figure 8: Notification from NRF in the Same PLMN

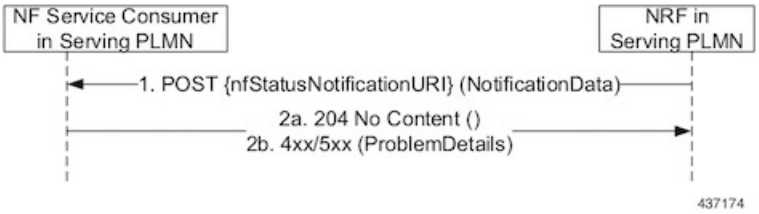


Table 8: Call Flow Description for Notification from NRF in the Same PLMN

Step	Description

1	The NRF sends a POST request to the callback URI.
	The request body for a profile change notification request includes the following:
	• **event**: This attribute indicates the notification type. It can be one of the following:
	  • NF_REGISTERED
	  • NF_DEREGISTERED
	  • NF_PROFILE_CHANGED
	• **nfInstanceUri**: Uniform Resource Identifier (URI) of the NF Instance associated to the notification event.
	• **nfProfile**: Indicates the new or updated NF profile.

2a	When the notification is successful, the NRF sends a "204 No content" response.
2b	The SMF returns a "404 Not Found" status code with the problem details if the SMF disregards the "nfStatusNotificationURI" as a valid notification URI. For example, if the URI does not belong to any of the existing subscriptions that the SMF has created in the NRF.

Limitations

In this release, the NRF Support for SMF Subscription and Notification 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.

Configuring NRF Support for SMF Subscription and Notification

This section describes how to configure the NRF Support for SMF Subscription and Notification feature.

Use the following commands to configure the NRF interface, vip-ip, vip-port, and loopback Port to open the server endpoints for the NF status notification.

```
configure
  endpoint sbi
    replicas integer
      vip-ip ip_address
        interface nrf
          vip-ip ip_address
          vip-port port_number
          loopbackPort port_number
          end
```

NOTES:

- **endpoint sbi**: Specifies the service-based interface (sbi) as the endpoint.
- **replicas**: Specifies the number of instances of the service-based interface.
- **vip-ip** *ip_address*: Specifies the virtual IP address of the virtual host.
- **interface nrf**: Specifies the interface as NRF.
- **vip-ip** *ip_address*: Specifies 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*: Specifies the port number of the virtual host. The SMF uses this as the listening port for the status notification.
- **loopbackPort** *port_number:*: Specifies the internal port number of the loopback host. The SMF uses this port for the NF status notification.

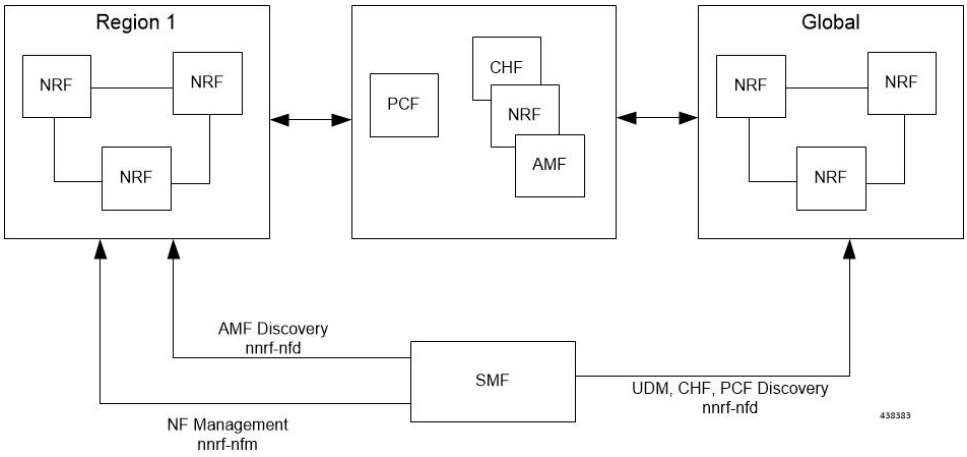
NRF Interface per Endpoint

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. With associating a single NRF interface for each endpoint, the self-management of an NF improves the productivity.

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

Figure 9: NRF Deployment



Standards Compliance

The NRF Interface Per Endpoint feature complies with the following standards:

- 3GPP TS 29.510 V15.2.0 (2018-12)
- 3GPP TS 29.510 V15.0.0 (2019-06)

Limitations

The NRF Interface Per Endpoint feature has the following limitations:

- The NF discovery and load-balancing capabilities are available only for UDM, PCF, CHF, and AMF.
- The dynamic configuration changes of NRF endpoints is not available.
- Support for the liveliness check of the NRF endpoints is not available.
- The SMF attempts the NF discovery first on the primary host. In the absence of the primary host, SMF attempts the discovery on the secondary host and switches to the tertiary if both primary and secondary are unavailable.

Configuring the NRF Interface Per Endpoint

This section describes how to configure the NRF Interface Per Endpoint feature.

Configuring the NRF Interface Per Endpoint feature involves the following steps:

1. Associating a Discovery Group with NF Type
2. Configuring Locality for NF Types
3. Associating NRF Management and SMF Locality to NRF Endpoint
4. Configuring the NRF Group
5. Configuring Locality for SMF

6. Configuring NF Profiles for a DNN
7. Configuring Network Element Profile Parameters for the NF

Associating a Discovery Group with NF Type

Use the following CLI commands for pairing a discovery group with NF types.

```
configure
   profile nf-pair nf-type nf_type
     profile nf-pair nf-type nrf-discovery-group nrf_discovery_group_name

      end
```

NOTES:

- **nf-type** *nf_type*: Specifies the NF type. The *nf_type* can be: 5G_EIR, AF, AMF, AUSF, BSF, CHF, GMLC, LMF, N3IWF, NEF, NRF, NSSF, NWDAF, PCF, SEPP, SMF, SMSF, UDM, UDR, UDSF, UPF, or range.
- **nrf-discovery-group** *nrf_discovery_group_name*: Specifies the 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 Locality for NF Types

The SMF provides locality aware NF discovery.

Use the following configuration to configure locality for NF types.

```
configure
   profile nf-pair nf-type nf_type locality { client client_name | geo-server
 geo_server_name | preferred-server preferred_server_name }
   end
```

NOTES:

- **client** *client_name*: Specifies the client locality information. Client locality is the SMF's locality and is a mandatory parameter.
- **geo-server** *geo_server_name*: Specifies the geo-server locality information. Geo-server locality is geo redundant site for the preferred locality and is generally used as the next best server locality after preferred locality, during NF discovery.
- **preferred-server** *preferred_server_name*: Specifies the preferred server locality information. Preferred server locality is the locality that should be considered as the locality of preference during the corresponding NF discovery.

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
```

Associating NRF Management and SMF Locality to NRF Endpoint

Use the following CLI commands for configuring NRF Management (nrf-group) and SMF Locality and associating them to NRF Endpoint.

```
configure
   group nf-mgmt mgmt_name
      nrf-mgmt-group nrf_group_name
      locality locality_name
      end
```

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 the NRF Endpoints Profile Parameters

The SMF provides CLI for configuring NRF endpoints for different services that are supported by NRF, such as **nnrf-nfm** (NF management) and **nnrf-nfd** (NF Discovery).

Note For a discovery group, only the **nnrf-disc** service can be configured. For management service, only **nnrf-nfm** can be configured.

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 host [ip:port] can be configured within each endpoint. Both IPv4 and IPv6 address can be specified. If both are specified, then 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}
```

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

Note In this release of the specification, both HTTP and HTTPS scheme URIs are allowed. See 3GPP TS 33.501, Subclause 13.1 for further details on security of service-based Interfaces.

- the 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]

```
configure
   group nrf { mgmt mgmt_name | discovery discovery_name }
      service type nrf { nnrf-nfm | nnrf-disc }
         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 | ipv6 ipv6_address | port port_num } }
            version [ uri-version version_num full version version_num ]
            end
```

NOTES:

- **api-root** *api_string* **:** Specifies the deployment-specific service API prefix that is used within the { apiRoot }.
- **api-uri-prefix** *uri_prefix_string:* Specifies the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **capacity** *capacity:* Specifies 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* **} }:** Specifies the endpoint name. You can configure the primary, secondary, and tertiary host (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*: Specifies 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* **}**: Specifies the primary endpoint IPv4 address, IPv6 address or port.
- **secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specifies the secondary endpoint IPv4 address, IPv6 address or port.
- **tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specifies the tertiary endpoint IPv4 address, IPv6 address, or port.
- **priority** *priority_value*: Specifies 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.

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

Verifying the NRF Endpoints Profile Parameters

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 1.1.1.[1]
    exit
   exit
   endpoint-name endpointName
    priority 1
    capacity 100
    primary ip-address ipv4 231.1.1.1
    primary ip-address port 3021
   exit
  exit
 exit
exit
```

Configuring Locality for SMF

This section describes how to configure the locality for SMF.

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

```
configure
   profile smf smf_profile_name
      locality value
      end
```

NOTES:

- **locality** *value*: Specifies 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

This section describes how to configure the NF profile that the configured Data Network Name (DNN) uses.

```
configure
   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*: Specifies 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 this configuration, use the **no network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name* command.

Configuring Network Element Profile Parameters for the NF

This section describes how to configure the network element profile parameters for the configured NF.

```
configure
   profile network-element { { amf | chf | pcf | udm } nf_profile_name }
      failure-handling-profile profile_name
      nf-client-profile profile_name
     query-params { dnn | supi | tai | target-nf-instance-id | target-plmn
 }
      end
```

NOTES:

- **failure-handling-profile** *profile_name*: Specifies the NRF failure handling network profile for the configured NF type. *profile_name* must be an alphanumeric string representing the corresponding NRF failure handling network profile name.
- **nf-client-profile** *profile_name*: Specifies the local NF client profile. *profile_name* must be an alphanumeric string representing the corresponding NF client profile name.
- **query-params { dnn | supi | tai | target-nf-instance-id | target-plmn }**: Specifies one of the following query parameters to include in the NF discovery request towards the NRF.
  - **dnn**: Specifies a DNN as the query parameter in the NF discovery request towards the NRF.
  - **supi**: Specifies a SUPI as the query parameter in the NF discovery request towards the NRF.
  - **tai**: Specifies a TAI as the query parameter in the NF discovery request towards the NRF.
  - **target-nf-instance-id**: Specifies a target NF instance Identifier as the query parameter in the NF discovery request towards the NRF.
  - **target-plmn**: Specifies a target PLMN as the query parameter in the NF discovery request towards the NRF.
  - **limit**: Specifies a limit for the maximum number of profiles that the NRF sends in the NF discovery response.
  - **max-payload-size**: Specifies the maximum payload size as the query parameter in the NF discovery request towards the NRF.

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

- This is an optional configuration. By default, these 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 feature.

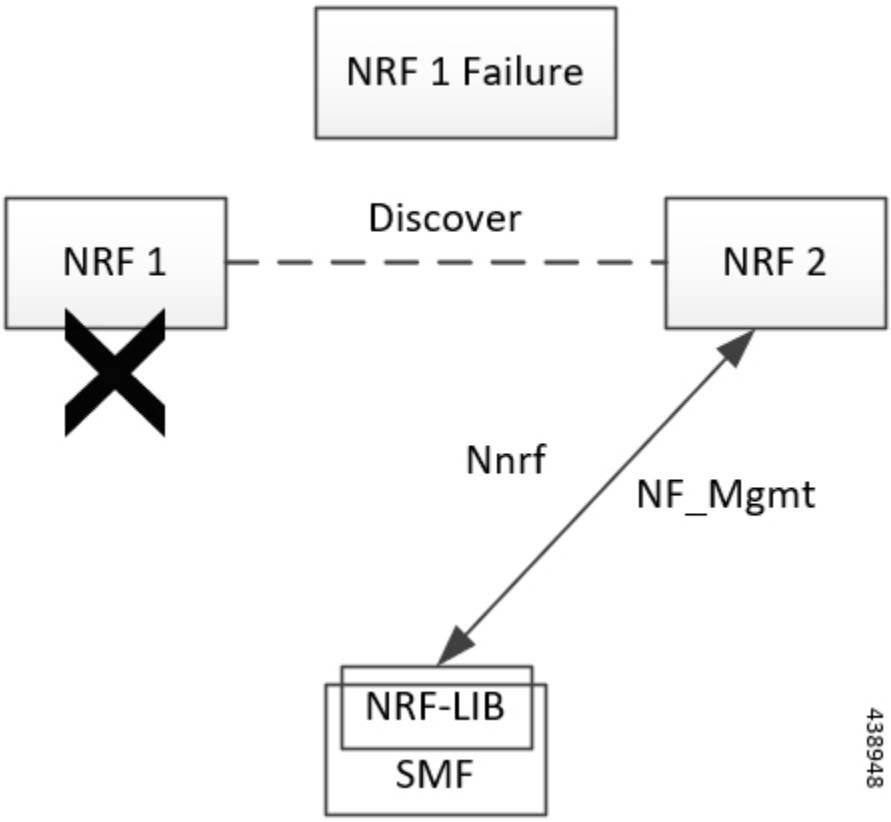
```
config
profile dnn cisco
 network-element-profiles chf chf1
 network-element-profiles amf amf1
 network-element-profiles pcf pcf1
 network-element-profiles udm udm1
 ssc-mode 2 allowed [ 3 ]
 session type IPV4 allowed [ IPV4V6 ]
 upf apn intershat
exit

profile smf smf1
 node-id          12b888e1-8e7d-49fd-9eb5-e2622a57722
 locality         LOC1
 bind-address ipv4 127.0.0.1
 bind-port        8008
 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
```

NRF Failure Handling Support

Feature Description

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

How it Works



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

In case, the SMF detects NRF 1 failure by missing NF heartbeat response, the SMF registers to NRF 2 (secondary NRF) and starts sending NF heartbeat. The SMF continues to send NF Register message1 to NRF 1 to keep track of its status.

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

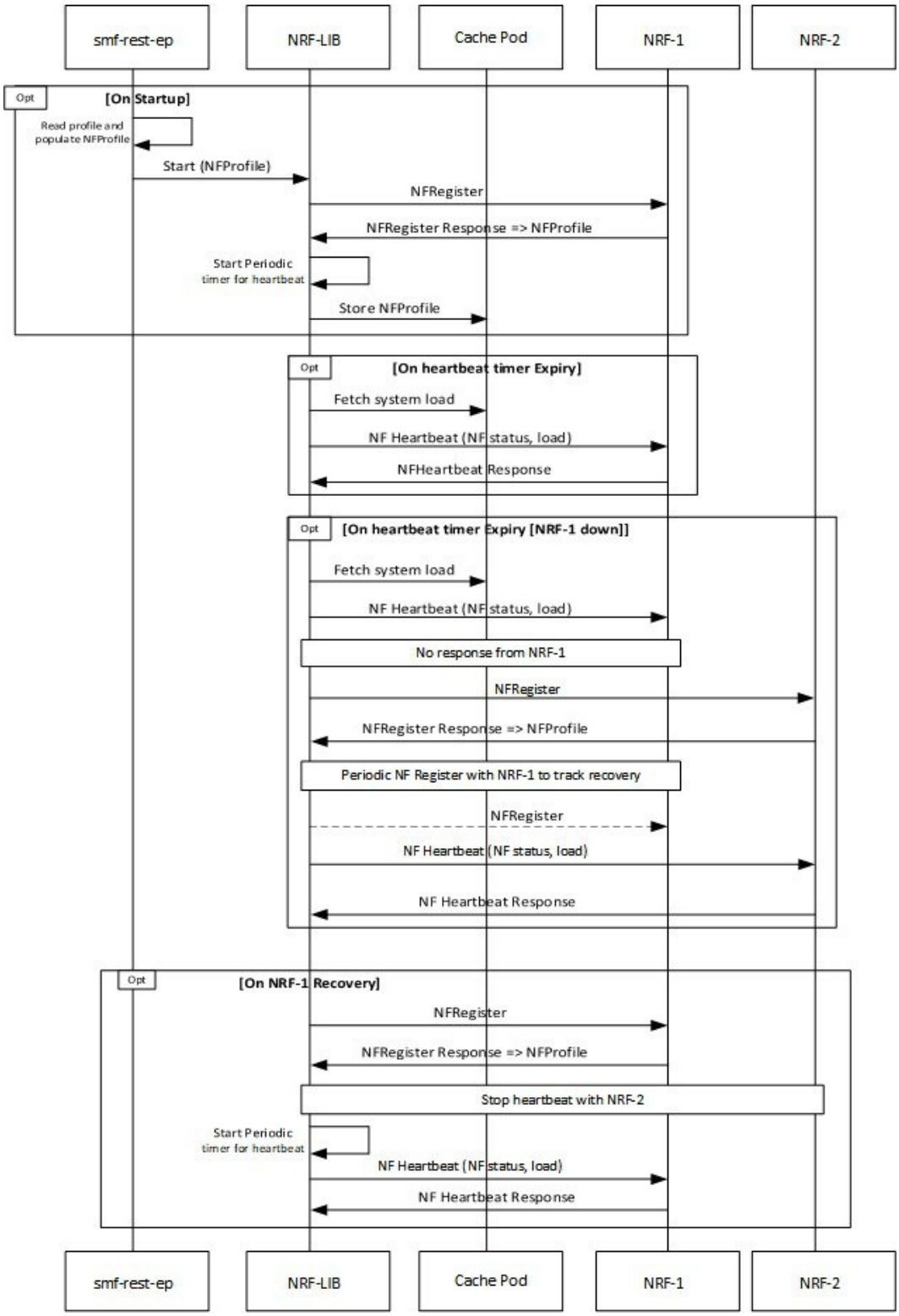
Note NF Reregistration (default behavior) on failover and fallback should be configuration driven. When NRF 2 detects that the SMF has stopped sending heartbeats, it checks from NRF 1 if it has received SMF registration by using discovery with SMF instance ID.

As the management and discovery endpoint groups are separate, the Registration based operation status check is not used for NRF failure handling during NF discovery. During NF discovery, the configured NRF endpoints within the group are attempted in the priority order. If the first choice NRF endpoint is not responding the next best NRF endpoint is chosen and so on.

Call Flow

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

Figure 10: NF Management Call Flow



Verifying the NRF Failure Handling

NF Management Failure Handling

Management NRF endpoint configuration sample is shown below.

```
product smf# show running-config group nf-mgmt
group nf-mgmt MGM
 nrf-mgmt-group mgmt_group
 locality       LOC1
exit
product smf# show running-config group nrf mgmt
group nrf mgmt mgmt_group
 service type nrf nnrf-nfm
  endpoint-profile epprof
   uri-scheme http
   endpoint-name EP1
    priority 2
    primary ip-address ipv4 10.105.227.219
    primary ip-address port 8082
    secondary ip-address ipv4 10.105.227.220
    secondary ip-address port 8082
   exit
   endpoint-name EP2
    priority 10
    primary ip-address ipv4 10.1.227.21
    primary ip-address port 8082
    secondary ip-address ipv4 10.1.227.22
    secondary ip-address port 8082
   exit
  exit
 exit
exit
product smf#
```

In the sample configuration, EP1 is the higher priority endpoint name as its priority is lesser than EP2 (2 against 10). So on bringing up, SMF sends NF registration to primary ip:port of EP1 [10.105.227.219:8082]. SMF uses secondary ip:port of EP1 if primary is down. SMF failovers to EP2 only if all ip:port of EP1 is down.

On successful registration with EP1 primary, SMF starts heartbeat with EP1 primary. If EP1 primary goes down, SMF detects the same by missing heartbeat response. On detecting EP1 primary down, SMF sends heartbeat to EP1 secondary [no reregistration]. Also, it periodically sends NF Heartbeat to EP1 primary to detect if it has recovered.

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

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

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

NF Discovery Failure Handling

Discovery NRF endpoint configuration sample is shown below.

```
product smf# show running-config profile nf-pair nf-type UDM
profile nf-pair nf-type UDM
 nrf-discovery-group others_group
 locality client LOC1
exit
product smf# show running-config group nrf discovery others_group
group nrf discovery others_group
 service type nrf nnrf-disc
  endpoint-profile ep1
   capacity 30
   priority 50
   uri-scheme http
   endpoint-name ED1
    priority 56
    primary ip-address ipv4 110.105.227.219
    primary ip-address port 8082
    secondary ip-address ipv4 110.105.227.220
    secondary ip-address port 8082
    exit
   endpoint-name ED2
    priority 10
    primary ip-address ipv4 110.1.227.21
    primary ip-address port 8082
    secondary ip-address ipv4 110.1.227.22
    secondary ip-address port 8082
   exit
  exit
 exit
exit
product smf#
```

In the sample configuration, ED1 has the higher priority endpoint name as its priority is lesser than ED2 (2 against 10). So, whenever there is a NRF discovery required primary ip:port of ED1 [110.105.227.219:8082] is attempted. SMF uses secondary ip:port of ED1 if primary is down. SMF failovers to ED2 only if all ip:port of ED1 is down. There is no state maintained regarding NRF discovery failure with any NRF endpoint. Every time SMF needs to send NRF discovery, SMF starts with ED1 primary and failovers to ED1 secondary in case of failure, followed by ED2 primary and so on.

Local Configuration for NF Management

Feature Description

The SMF learns about the other NF endpoints such as Unified Data Management (UDM), Access and Mobility Management Function (AMF), Policy Control Function (PCF), Charging Function (CHF) and so on, through NF discovery service exposed by Network Repository Function (NRF) or through the CLI configuration. The SMF prioritizes the NF discovery through the NRF. If the NRF is not available, then the SMF uses the local configuration of NF endpoints to discover the NFs.

Relationships

The Local Configuration for NF Discovery feature depends on the configuration of NRF endpoints, and the response from NRF. That is, the SMF uses the locally configured endpoints of the NFs only if the NRF endpoints remain unconfigured or if the NRF did not return any NFs matching the preferred server locality or geo locality.

For more information, see the NRF Interface per Endpoint, on page 15 section in this chapter.

Standards Compliance

The Local Configuration for NF Discovery feature complies with 3GPP TS 29.510, Versions 15.0.0 and 15.2.0.

Limitations

The Local Configuration for NF Discovery feature has the following limitations:

- Discovery and load balancing are available only for the UDM, PCF, CHF, and AMF but not for the UPF.
- Support for the liveliness check of the NF endpoints is currently not available.
- The SMF attempts the NF discovery first on the primary host. In the absence of the primary host, the SMF attempts the discovery on the secondary host and switches to the tertiary if both the primary and secondary are unavailable.

Configuring the NFs for NF Discovery

This section describes the Local Configuration for NF Discovery feature.

Configuring the NF for NF Discovery feature involves the following steps:

1. Configuring Locality for SMF
2. Configuring NF Profiles for a DNN
3. Configuring Network Element Profile Parameters for the NF
4. Configuring NF Client Profile
5. Defining Locality within NF Profile
6. Configuring NF Endpoint Profile Parameters

Configuring Locality for SMF

This section describes how to configure the locality for SMF. This is a mandatory configuration if the SMF performs the NF discovery using the NRF.

```
configure
   profile smf smf_profile_name
      locality value
      end
```

NOTES:

- **locality** *value*: Specifies 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

This section describes how to configure the NF profile that the configured Data Network Name (DNN) uses.

```
configure
   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* : Specifies 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 this configuration, use the **no network-element-profiles { amf | chf | pcf | udm }** *nf_profile_name*command.

Configuring Network Element Profile Parameters for the NF

This section describes how to configure the network element profile parameters for the configured NF.

```
configure
   network-element-profiles { { amf | chf | pcf | udm } nf_profile_name }
      failure-handling-profile profile_name
 nf-client-profile profile_name
     query-params { dnn | supi | tai | target-nf-instance-id | target-plmn
 }
      end
```

NOTES:

- **failure-handling-profile** *profile_name*: Specifies the NRF failure handling network profile for the configured NF type. *profile_name* must be an alphanumeric string representing the corresponding NRF failure handling network profile name.
- **nf-client-profile** *profile_name*: Specifies the local NF client profile. *profile_name* must be an alphanumeric string representing the corresponding NF client profile name.
- **query-params { dnn | supi | tai | target-nf-instance-id | target-plmn }:** Specifies to include one of the following query parameters in the NF Discovery Request towards the NRF.
  - **dnn**: Specifies a DNN as the query parameter in the NF discovery request towards the NRF.
  - **supi**: Specifies a SUPI as the query parameter in the NF discovery request towards the NRF.

- **tai**: Specifies a TAI as the query parameter in the NF discovery request towards the NRF.
- **target-nf-instance-id**: Specifies a target NF instance Identifier as the query parameter in the NF discovery request toward the NRF.
- **target-plmn**: Specifies a target PLMN as the query parameter in the NF discovery request toward the NRF.
- **limit**: Specifies a limit for the maximum number of profiles that the NRF sends in the NF discovery response.
- **max-payload-size**: Specifies the maximum payload size as the query parameter in the NF discovery request towards the NRF.
- **requester-snssais**: Specifies the list of Single Network Slice Selection Assistance Information (S-NSSAIs) as the query parameter in the NF discovery request towards the NRF.

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

Configuring NF Client Profile

This section describes how to configure the NF endpoints for AMF, CHF, PCF, and UDM.

```
configure
   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* **}**: Specifies the required NF client profiles and provides the local configuration for any of the following configured NFs:
  - **amf**: Enables the AMF local configuration
  - **chf**: Enables the CHF local configuration
  - **pcf**: Enables the AMF local configuration
  - **udm**: Enables the AMF local configuration

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

    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 this 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.

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 17 in the NRF Interface per Endpoint, on page 15 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.

```
configure
   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*: Specifies 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*: Specifies the priority for the locality configuration.
- **service name type** *service_types*: Specifies the configured NF service types. The service types vary depending the configured service.

  The AMF service supports the following service types:

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

  The CHF service supports the following service types:

  - nchf-convergedcharging
  - nchf-spendinglimitcontrol

  The PCF service supports the following service types:

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

- npcf-ue-policy-control

The UDM service supports the following service types:

- nudm-ee
- nudm-pp
- nudm-sdm
- nudm-ueau
- nudm-uecm

- **endpoint-profile** *epprofile_name:* Specifies 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

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.

NRF Library (NRF-LIB) 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, its structure is specified as follows:

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

" apiRoot " is a concatenation of the following parts:

- scheme ("http" or "https")
- the 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]

```
configure
   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
            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 }
              version [ uri-version version_num full version version_num ]
              end
```

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

NOTES:

- **api-root** *api_string* **:** Specifies the deployment-specific service API prefix that is used within the { apiRoot }.
- **api-uri-prefix** *uri_prefix_string:* Specifies the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **capacity** *capacity:* Specifies 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* **} }**: Specifies the endpoint name. You can configure the primary, secondary, and tertiary host (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*: Specifies 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* **}**: Specifies the primary endpoint IPv4 address, IPv6 address or port.
  - **secondary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specifies the secondary endpoint IPv4 address, IPv6 address or port.
  - **tertiary ip-address { ipv4** *ipv4_address* **| ipv6** *ipv6_address* **| port** *port_num* **}**: Specifies the tertiary endpoint IPv4 address, IPv6 address or port.
- **priority** *priority_value*: Specifies 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.
- **uri-scheme { http | https }**: Specifies the URI scheme as **http** or **https**.
- **version [uri-version** *version_num* **full version** *version_num* **]**: Specifies the api/Version. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draft-number>].

Verifying the Local Configuration for NF Discovery Feature

This section describes how to verify the Local Configuration for NF Discovery feature.

Use the following show command to verify the feature configuration details.

show running-config

The following is a sample output of this show command.

```
config
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 127.0.0.1
 bind-port        8008
 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

profile nf-client nf-type udm
 udm-profile PROF1
  locality PREF_LOC
   priority 10
   service name type nudm-sdm
    endpoint-profile epprof
     api-uri-prefix nudm-sdm
     api-root       root
     uri-scheme     http
     version
      uri-version v1
       full-version 1.1.1.[1]
```

```
      exit
     exit
     endpoint-name endpointName
      priority 1
      capacity 100
      primary ip-address ipv4 231.1.1.1
      primary ip-address port 3021
     exit
    exit
   exit
  exit
 exit
exit
```

Fallback to Static IP Address Support

Feature Description

The SMF follows a priority order for the different NF selection options. It prioritizes the NF discovered from the network repository function (NRF) over the local configuration. The SMF uses the locally configured NFs in the following scenarios:

- When the NRF endpoints (for discovery) are not configured.
- 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 [**profile nf-pair**]. For more details, see Configuring Locality for NF Types, on page 17 in the NRF Interface per Endpoint, on page 15 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 relationship with the following features:

- NF Discovery, NF Selection, and Load Balancing
- NRF Interface Per Endpoint
- Caching Support for NF Discovery

How it Works

The SMF follows this sequence for NF selection:

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 AMF, 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 17 in the NRF Interface per Endpoint, on page 15 section.) For each NF, the query parameters are configurable. (For more details, see Configuring Network Element Profile Parameters for the NF, on page 21in the NRF Interface per Endpoint, on page 15 section) The NRF returns all the NFs matching the query criteria. When present, 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. But 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 present for the query filter in its NF discovery cache
- If matching entries are present in its NF discovery cache but the validities of these entries have expired

  The retention of an expired cache entry is configuration-based. If the expired cache entry is present 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. (For more details, see the Local Configuration for NF Management, on page 27 section.) 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. For a sample configuration, see the Configuring the Fallback to Static IP Address Support Feature, on page 38 section.

Standards Compliance

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

- 3GPP TS 29.510 V15.2.0 (2018-12)
- 3GPP TS 29.510 V15.0.0 (2019-06)

Limitations

The Fallback to Static IP Address Support feature has the following limitation:

There is no support for dynamic configuration changes of NRF endpoints.

Configuring the Fallback to Static IP Address Support Feature

This section describes how to configure the Fallback to Static IP Address Support feature.

Configuring the Failure Template

This section describes how to configure the failure template.

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

NOTES:

- **profile nf-client-failure { nf-type { amf | chf | pcf | udm }**: Specifies the required NF client failure profile and provides the local configuration support for the following configured NF:
  - **amf**: Enables the AMF local configuration
  - **chf**: Enables the CHF local configuration
  - **pcf**: Enables the PCF local configuration
  - **udm**: Enables the UDM local configuration

    For example, if the NF type that is selected is **udm**, then this command enables the UDM local configuration. The same approach applies for the other configured NFs.
- **profile failure-handling** *profile_name*: Specifies the failure handling profile name. For example, "udmFail".

Sample Configurations

The following is a sample configuration of the failure template mapping to dnn:

```
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
```

The following is a sample configuration of the failure template mapping to smf:

```
profile smf smf1
 node-id          12b888e1-8e7d-49fd-9eb5-e2622a57722
 locality         LOC1
 bind-address ipv4 127.0.0.1
 bind-port        8008
 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
```

For more information, see Configuring NF Profiles for a DNN, on page 20 in the NRF Interface per Endpoint, on page 15 section.

Configuring NF Service and Message Type

This section describes how to configure the NF service and its different message types.

```
configure
  profile nf-client-failure { nf-type { amf | chf | pcf | udm }
     profile failure-handling failure_handling_name
        service name type service_type
          message type message_type
          end
```

NOTES:

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

  The AMF service supports the following service types:

  - **namf-comm**
  - **namf-evts**

- **namf-loc**
- **namf-mt**

The CHF service supports the following service types:

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

The PCF service supports the following service types:

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

The UDM service supports the following service types:

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

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

- **message type** *message_type*: Specifies the configured NF message type and provides the local configuration support for the following configured NF.

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

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

Profile	Service Type	Message Type Options
amf	namf-comm	• AmfCommEBIAssignment • AmfCommN1N2MessageTransfer • AmfCommSMStatusChangeNotify • range

Profile	Service Type	Message Type Options
chf	nchf-convergedcharging	• ChfConvergedchargingCreate • ChfConvergedchargingDelete • ChfConvergedchargingUpdate • range
pcf	npcf-am-policy-control	• PcfSmpolicycontrolCreate • PcfSmpolicycontrolDelete • PcfSmpolicycontrolUpdate • Range
udm	nudm-sdm	• UdmRegistrationReq • UdmSdmGetUESMSubscriptionData • UdmSdmSubscribeToNotification • UdmSubscriptionReq • UdmUecmRegisterSMF • UdmUecmUnregisterSMF • UdmSdmUnsubscribeToNotification • range

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

Configuring NF Failure Retry, Action, and Message Type

This section describes how to configure the failure retry and action for each service of the NF and its different message types.

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

NOTES:

- **status code httpv2 {** *integer* **}**: Specifies the status code for the retry and action for the NF service. Currently only "http" status code is provided. *integer* specifies the status code. *integer* must be an integer in the range of 300-599.
- **retry** *integer*: Specifies the number of times the NF service must retry before proceeding with the action.
- **action**: Specifies the action. The different actions supported are:
  - **continue**: Specifies to continue the session without any retry. The retry count configuration is invalid with this action.
  - **retry-and-continue**: Specifies to retry as per the configured retry count and continue the session.
  - **retry-and-terminate**: Specifies to retry as per the configured retry count and terminate the session in case all retry fails.
  - **terminate**: Specifies to terminate the session without any retry. Retry count configuration is invalid with this action.

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

Configuring Invalidate (Purge) NF Discovery Cache

This section describes how to configure the cache entry invalidation (purge) for the NF discovery cache.

```
configure
   profile nf-pair nf-type { amf | chf | pcf | udm }
      cache invalidation { false | true [ timeout integer ] }
        end
```

NOTES:

- **cache invalidation { false | true [timeout** *integer* **] }** : Configures the interval and cache invalidation rule. The default value is false.
  - **false** : Specifies that the cache entry will never be invalidated.
  - **true timeout** *integer* : Specifies 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 is a sample configuration that sets the cache invalidation to false for the UDM discovery:

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

The following is a sample configuration that sets the cache invalidation to true for the UDM discovery:

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

NF Profile Update

Feature Description

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

NF Update service updates the profile of NF that was previously registered in the NRF by providing the updated profile of the requesting NF to the NRF. The update operation could be a whole NF profile update (complete replacement of the existing profile with a new profile), or an update to only a subset of the NF profile parameters (including adding, deleting, or replacing services to the NF profile).

Standards Compliance

The NF Profile Update feature complies with the 3GPP TS 29.510, V15.2.0 (2018-12).

Limitations

The SMF currently supports only the complete replacement of NF profile.

How it Works

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

Figure 11: NF Profile Complete Replacement

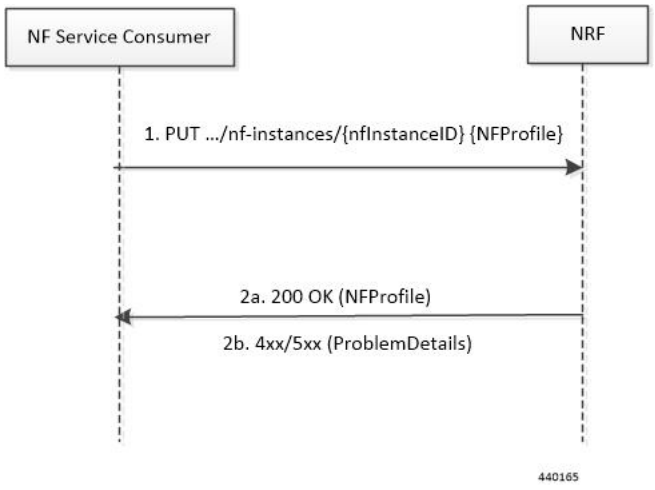


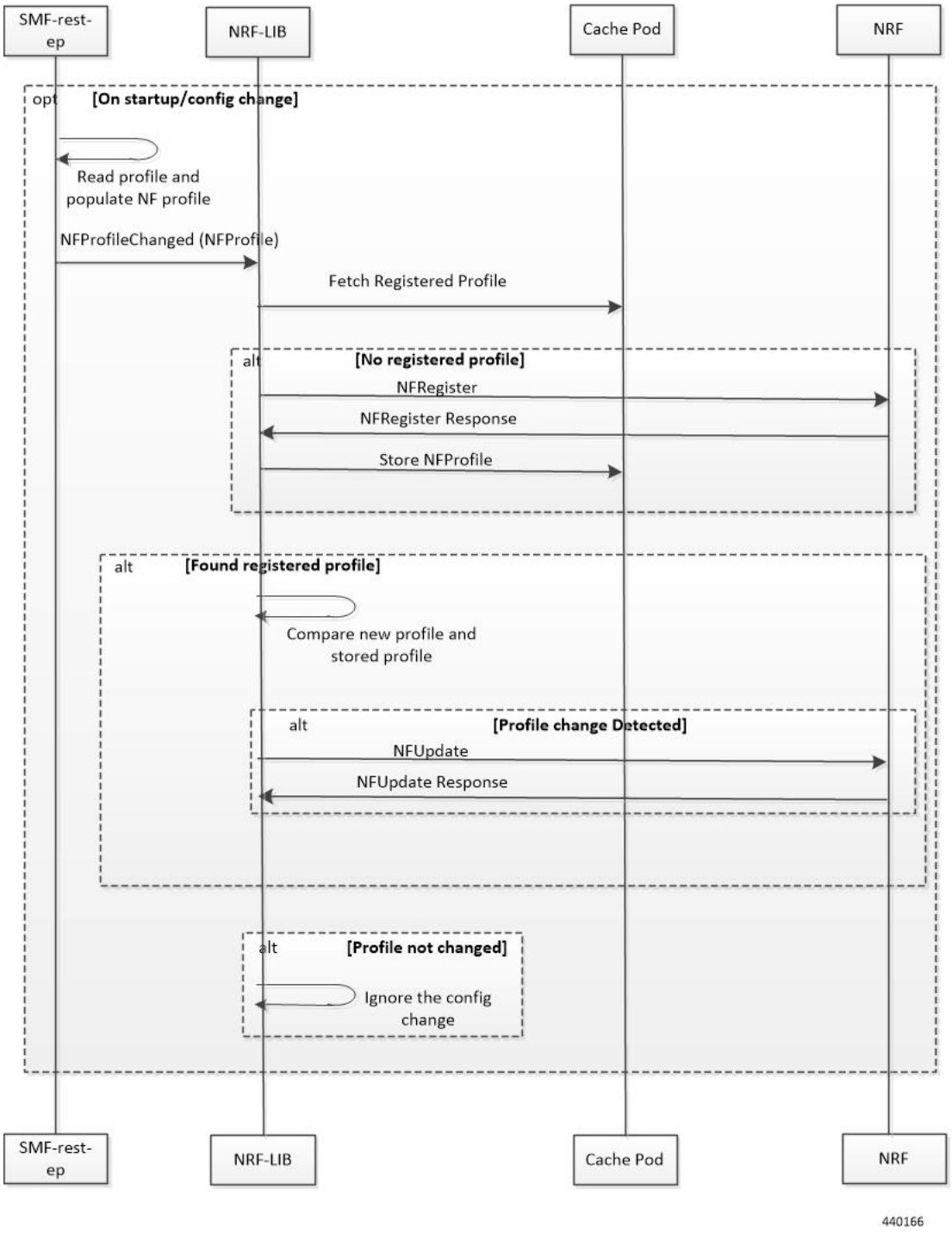
Table 9: NF Update Call Flow Description

Step	Description
1	The NF Service Consumer 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

Step	Description
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.

The following figure illustrates the call flow representing the NF registration and NF update messaging from NRF client library on NF profile change trigger from SMF-rest-ep.

Figure 12: 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 NRF-LIB to indicate the NF Profile change.
2. NRF library (NRF-LIB) maintains the NF registration status and the registered profile in an external cache pod. The NRF client detects whether the NF registration with NRF is completed. If the NRF client 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 NRF-LIB 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. NRF-LIB fetches the registered NF profile and then compares it with the new profile.
5. The NRF-LIB NF 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.
6. The NRF-LIB ignores the trigger if there is no change detected.

Important The NF update is sent only from the elected master.

Load parameter is not set as part of NF update PUT message. Heartbeat is set as the current active heartbeat interval.

Configuration Support for List of Tracking Areas and Tracking Area Ranges

Feature Description

The SMF provides an optional configuration to configure the supported list of Tracking Areas and Tracking Area Ranges for a Public Land Mobile Network (PLMN). When a new configuration is present, the SMF sends the configured Tracking Area Identity (TAI), that is, TAIList and TAIRangeList, to the Network Function (NF) Repository Function (NRF) during the SMF Service Registration.

Important Any change in the configuration results in SMF Service update towards the NRF with the new configured TAIList and TAIRangeList values.

The PLMN value sent in the NRF discovery message remains the same as the PLMN configured on the SMF.

For more details on the NF Registration and NF Registration Update, see the NF Profile Update, on page 43 section.

Configuring TAI Group

This section describes how to configure the TAI Group.

Configuring TAC List

Use the following configuration to configure the TAC list within TAI profile.

```
configure
   profile tai-group tai_group_name
      mcc mcc_value mnc mnc_value
      tac list [ tac_list_values ]
      end
```

NOTES:

- **tac list [** *tac_list_values* **]**: Configures the list of TAC values. For example, [1111 2222 3333]

Configuring TAC Range List

Use the following configuration to configure the TAC range list within TAI profile.

```
configure
   profile tai-group tai_group_name
      mcc mcc_value mnc mnc_value
      tac range start start_value end end_value
      end
```

NOTES:

- **tac range start** *start_value* **end** *end_value*: Configures a specific TAC range or multiple TAC range lists. For example, **tac range start DDDD end EEEE**

  You can configure a maximum of 16 values in a range.

- Use the **no tac range start** *start_value* **end** *end_value* command to remove a specific TAC range or TAC Ranges.

Verifying the TAI Group Configuration

Use the following show command to verify the TAI group configuration.

```
show running-config profile tai-group tai_group_name
```

The following is an example of the **show command** configuration.

```
show running-config profile tai-group t1
profile tai-group t1
mcc 111 mnc 222
  tac list [ 1111 2222 3333 ]
  tac range start 4444 end 5555
  exit
exit
mcc 333 mnc 44
  tac list [ AAAA BBBB CCCC ]
  tac range start DDDD end EEEE
  exit
exit
exit
```

Dynamic Configuration Change Support

Feature Description

Global configuration table was built for NRF configurations and rebuilt each time when there was a change in configuration. NRF transaction/procedure (such as discovery, management, and so on) picked the configuration for the respective transaction/procedure from the global configuration tables. Therefore, the ongoing transactions were impacted if the configurations were modified in the middle of the transaction/procedure.

With this feature:

- NRF transaction/procedure picks a configuration version (v1) and uses the same version till the NRF transaction/procedure complete.
- If a user changes the configuration during an ongoing NRF transaction, then a new configuration version (v2) is created. However, the new configuration is not applied to the ongoing transaction.

The dynamic configuration changes are for the following data structures:

- NrfFailureProfileSt
- NrfClntProfileSt
- NrfGrpSt
- NrfPairProfileSt
- NrfMgmtGrpSt

NRF Show Command Enhancements

show nrf registration-info

Field	Description
NF Status	Displays NRF Registration Information.
Registration Time	Displays Time of Registration with NRF.
Active MgmtEP Name	Active NRF Management End Point name.
Heartbeat Duration	Displays Heart Beat Duration.
Uri	Displays Uri Information.
Host Type	Displays NRF Host Type Information.

show nrf subscription-info

Field	Description
NF Instance Id	Displays NF instance Identity.
SubscriptionID	Displays the Subscription Identity information.
Actual Validity Time	Displays Actual Validity Time received from NRF server.
Requested Validity Time	Displays NF Requested Validity subscription Time.

show nrf discovery info

Field	Description
NF Type	Displays NF Type Information.

show nrf discovery-info AMF discovery-filter

Step	Description
Discovery Filter	Displays Discovery Filter Information.
Expiry Time	Displays Expiry Time for discovery Filter.

shownrfdiscovery-infoAMFdiscovery-filterdnn=intershatnf-discovery-profile

show nrf discovery-info AMF discovery-filter dnn=intershat nf-discovery-profile

Field	Description
NF InstanceId	Displays the NF Instance Identity.
NF Type	Displays the NF Type Information.
Discovery Filter	Displays the Discovery Filter Information.
NF Status	Displays the NF Status Information.
Priority	Displays the Priority Information.
Capacity	Displays the NF Profile Capacity Information.
Load	Displays the Load Information.
Locality	Displays the Locality Information.
ipv4 address	Displays IPv4 Address received from the discovery response for this NF profile.

Field	Description
ipv6 address	Displays the IPv6 Address received from the discovery response for this NF profile.

shownrfdiscovery-infoAMFdiscovery-filterdnn=intershatnf-discovery-profile f9882966-a253-32d1-8b82-c785b34a7cc9 nf-service

show nrf discovery-info AMF discovery-filter dnn=intershat nf-discovery-profile f9882966-a253-32d1-8b82-c785b34a7cc9 nf-service

Field	Description
ServiceInstanceId	Displays the NF Service Instance ID.
ServiceName	Displays the NF Service Name.
UriScheme	Displays the Uri Scheme Information.