



# RADIUS Client for SMF

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 2](#)
- [How it Works, on page 7](#)
- [Configuring the RADIUS Client, on page 27](#)
- [RADIUS Test CLI support, on page 40](#)
- [RADIUS Client OA&M Support, on page 42](#)

## 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	Disabled – Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

## Revision History

*Table 2: Revision History*

Revision Details	Release
Added support for the following: <ul style="list-style-type: none"> <li>• PAP, CHAP, and MSCHAP-based RADIUS authentication</li> <li>• Multiple RADIUS NAS-IP source addresses</li> <li>• Handling RADIUS Disconnect and CoA Requests</li> <li>• RADIUS Accounting on SMF</li> <li>• New attributes in the RADIUS Access Response message</li> </ul>	2020.02.5.t1
First introduced.	Pre-2020.02.0

## Feature Description

Remote Authentication Dial-In User Service (RADIUS) is a client and server protocol. The RADIUS client is typically a Network Access Server (NAS) and the RADIUS server is usually a daemon process running on a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. RADIUS servers receive user connection requests, authenticate the user, and then return the configuration information necessary for the client to deliver service to the user.

RADIUS provides Authentication and Accounting services to the users. The SMF provides the following configuration support to —

- add RADIUS server details
- enable RADIUS accounting and authentication
- add RADIUS interface as an option for virtual APN configuration within DNN profile
- enable CC trigger reporting
- define volume and time limits

The RADIUS Client feature supports the following functions:

- **Server Selection**

RADIUS servers are configured with IP:Port as the key. The **algorithm** CLI specifies the failover or load balancing algorithm to select the RADIUS server to which the authentication or accounting request must be sent. Servers that are marked "dead" are not considered for selection until they are marked "alive". The supported algorithms are first-server and round-robin.

- **First-server**—Specifies that the request must be sent to RADIUS server with the highest priority. If the server becomes unreachable, the request is sent to the server with the next highest configured priority. This is the default algorithm.
- **Round-robin**—Specifies that the request must be sent based on load balancing in a circular queue manner. The server that is last used is stored to maintain the round-robin selection. The order of the list is purely based on the configuration sequence.

- **Monitor Server and Dead Server Detection**

Monitor Server revisits the server database and marks the server which has not received response beyond the configured "response-timeout" value after the first request is sent. The server is marked "dead" and remains in dead-state for minutes configured as "deadtime". After the "deadtime" elapses, the server's dead-variable is reset again to mark it as ready to process requests. If the server is still not reachable, it is marked "dead" as part of the next request response timeout.

- **Timeout and Retry**

After a server is selected and request is sent to the server, an entry is maintained in the request queue until response is received from the RADIUS server or until timeout occurs. Monitor Requests is called to check on the requests queue for response timeouts and retry. It walks through all the entries and checks if any request timeout value configured as "timeout" is hit. For such requests, if the number of retries is less than the configured "max-retries" value, the request is resent to the RADIUS server. Else, if the "max-retries" count is reached, the request is deleted from the request queue. After a request is deleted, even if response comes for such requests, the response is discarded and not sent to the user.

## Architecture

### RADIUS Integration in Mobile CNAT Architecture

The Mobile CNAT architecture has four distinct layers:

1. **Cloud**—Host OS + Kubernetes installation.
2. **Runtime**—Plugins to Kubernetes provided by the Cloud. This layer includes the container runtime (docker version) and Kubernetes plugins for volume (storage), networking, and load balancing.
3. **Orchestration**—Kubernetes functionality. Kubernetes provides abstractions for provided plugins (networking, volumes, load balancing) so that the CNAT components can be unaware of them.
4. **Mobile CNAT Components**—Application layer where the applications are built for mobility depending only on Kubernetes as much as possible.

5GC Network Functions (NFs) run in the Application or CNAT Component layer of this architecture. RADIUS Client is an integral part of the SMF.

### RADIUS Client Integration in SMF

The SMF consists of loosely coupled microservices. The microservice decomposition is based on the following three-layered architecture:

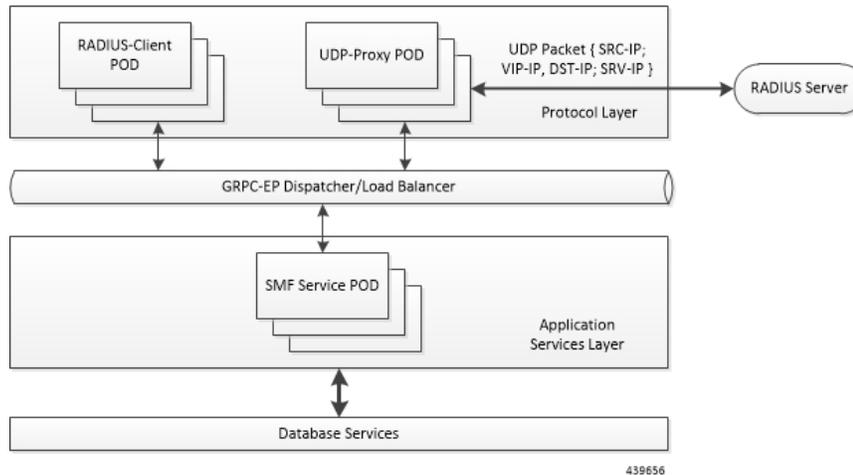
1. **Layer 1**—Protocol and Load Balancer services (stateless)
2. **Layer 2**—Application services (stateless)

### 3. Layer 3—Database services (stateful)

The Protocol layer integrates with the RADIUS Client POD.

The following figure illustrates the integration of RADIUS Client in SMF.

**Figure 1: RADIUS Client Integration**



**Radius-EP App (RADIUS-Client POD)**—The RADIUS Client functionality is added in a new POD. It handles RADIUS protocol-specific functions such as authentication and accounting.

**SMF Service App (SMF Service POD)**—The SMF Service App provides PDU session service. During session establishment, the SMF service decides if the secondary authentication is required or not, and acts accordingly.

**UDP-Proxy App (UDP-Proxy POD)**—The UDP-Proxy App is enabled with host-networking and, sends and receives packets using external Virtual-IPs. All RADIUS packets are transmitted and received from an outside cluster using this application.

## RADIUS Authentication

Authentication and key management are fundamental to the security of mobile networks because they provide mutual authentication between users and the network.

5G defines various authentication methods to authenticate a user. In the 5G architecture, the serving network authenticates the Subscription Permanent Identifier (SUPI), and key agreement between the UE and the network using primary authentication mechanism.

5G supports EAP-based secondary authentication between the UE and the network. The SMF performs the role of the EAP Authenticator and relies on external AAA server (for example, RADIUS server) to authenticate and authorize the UE's request for PDU session establishment.

The RADIUS Client function resides within the SMF to enable the generic Cloud Native 5G RADIUS functionality for authentication purposes. When the RADIUS Client feature is enabled, the SMF performs secondary authentication with the configured external RADIUS server as per 3GPP TS 23.501.

For information on enabling the RADIUS Client feature, see [Configuring the RADIUS Client, on page 27](#).

## RADIUS Accounting

Accounting collects and sends subscriber usage and access information used for billing, auditing, and reporting. For example, user identities the start and stop times, performed actions, number of packets, and number of bytes. Accounting enables an operator to analyze the services that the users access and the amount of network resources they consume. Accounting records comprise accounting Attribute Value Pairs (AVPs) and are stored on the accounting server. This accounting information can then be analyzed for network management, client billing, and/or auditing.

The SMF implements the RADIUS Accounting functionality through the use of CLI configuration. For more details on the configuration, see [Configuring the RADIUS Client, on page 27](#).

If the RADIUS accounting is enabled and server-group is configured within the DNN profile, the SMF sends server-group as AAA group in charging-params in N4 session establishment request. When the SMF sends AAA group which is not present on UPF, then it does not account the traffic for static and predefined rules in RADIUS URR and fails to report. In this scenario, the SMF considers only the dynamic rules traffic for accounting in the RADIUS URR.

## RADIUS Access Management

RADIUS supports the following timeout functionality:

- **Idle Timeout**—The SMF supports CP-Idle-Timeout which handles the Idle Timeout functionality of Control-Plane. The same functionality is leveraged for the idle timeout value received in the Radius-Access-Accept message.

The RADIUS returned values have higher precedence over the configuration unless there is a specific configuration to choose between RADIUS returned values and the pre-existing configuration in SMF.

- **Session Timeout**—The SMF supports Absolute-Session-Timeout which handles the Session Timeout functionality of a session. This functionality is leveraged for the session timeout value received in the Radius-Access-Accept message.

The RADIUS returned values have higher precedence over the configuration unless there is a specific configuration to choose between RADIUS returned values and the pre-existing configuration in SMF.

## RADIUS NAS-IP Support

SMF supports the RADIUS NAS-IP address functionality for accounting and authentication requests.

In releases prior to 2021.1.0, only one common RADIUS NAS-IP address was used for all requests. This feature is extended to support multiple RADIUS NAS-IP source addresses.

The NAS-IP configuration is supported at the following levels:

- Global NAS-IP
- Global accounting NAS-IP
- Per server-group common NAS-IP
- Per server-group accounting NAS-IP

The selected NAS-IP is encoded in authentication or accounting requests as per RFC2865 and is also used in the "source-ip" address of outbound UDP packets. This feature supports only the IPv4 NAS-IP address.

For more information, see the [Configuring RADIUS NAS-IP, on page 32](#) section.

## Handling RADIUS Disconnect and CoA Requests

Dynamic Authorization Client (DAC) initiates Disconnect-Request packet through UDP port to terminate the user session(s) on Network Access Server (NAS). It also discards all the associated session contexts.

The NAS responds with a Disconnect-ACK message if the session is identified, removed, and no longer valid. The NAS sends a Disconnect-NAK message if it is unable to disconnect the session.

This feature uses a combination of the following session keys to identify the sessions for termination:

- 3GPP-IMSI + 3GPP-NSAPI
- ACCT-SESSION-ID
- CALLED-STATION-ID (DNN) + FRAMED-IP-ADDR
- CALLED-STATION-ID (DNN) + FRAMED-IPV6-PREFIX



### Important

If multiple key combination is provided for the same session, it is accepted. However, if the multiple key combination leads to multiple session contexts or non-existing session context, the behavior is non-deterministic.

The SMF supports only one session context per Disconnect-Message (DM) request. The SMF supports the following attributes in the DM request to identify the NAS and the user sessions to be terminated.

Attribute	Reference Specification	Encoding Type
3GPP-IMSI	3GPP 29.061 - 16.4.7.2-1	String
3GPP-NSAPI	3GPP 29.061 - 16.4.7.2-10 3GPP 29.561 – 11.3	String
Accounting-Session-Id	RFC2866	String
FRAMED-IP	RFC2865 - 5.1	IPV4 Address
FRAMED-IPV6-PREFIX	RFC3162	PrefixLen & String
CALLED-STATION-ID (DNN)	RFC2865 - 5.30	String
NAS-IP-Address	RFC2865 – 5.4 (optional)	String
NAS-Identifier	RFC2864 – 5.32 (optional)	String

The SMF silently discards other attributes present in the DM request if the packet decoding is successful.

The SMF supports the following attributes in the DM ACK or NAK response.

Attribute	Reference Specification	Encoding Type
ERROR-CAUSE	RFC5176 – 3.5	Integer
REPLY-MESSAGE	RFC2865 – 5.18	String

The RADIUS endpoint (radius-ep) pod supports the following error codes if the Disconnect Request is rejected by radius-ep:

- 402 (Missing Attribute) - Triggered due to invalid key combination
- 403 (NAS Identification Mismatch) - Triggered if NAS-IP attribute in DM request does not match the endpoint COA-NAS VIP-IP or if NAS-Identifier attribute in the request does not match NAS identifier configuration within RADIUS Dynamic Authorization or CoA configuration
- 407 (Invalid Attribute) - Triggered due to format error, encode error, and so on
- 405 (Unsupported Service) - Triggered if the request is not a disconnect request
- 503 (Session Context Not Found) - Triggered if the session cannot be located

For more information on configuring this feature, see the [Configuring the Session Disconnect Feature, on page 38](#) section.

## How it Works

This section describes how the SMF supports RADIUS authentication and accounting functionality.

### RADIUS Interaction for Authentication

The RADIUS server supports various methods to authenticate the user. When the server is provided with the username and original password of the user, it can support Point-to-Point Protocol (PPP), Password Authentication Protocol (PAP), Challenge-Handshake Authentication Protocol (CHAP), or Microsoft CHAP (MSCHAP), UNIX login, and other authentication methods.

In releases prior to 2021.01.0: The SMF uses only MSISDN values for user authentication.

In release 2021.01.0 and later: The SMF supports user authentication using PAP, CHAP, or MSCHAP protocol. The SMF configuration aids in the protocol selection for the user authentication. If the secondary authentication is enabled in DNN profile, the SMF interacts with the RADIUS server to perform RADIUS authentication. To implement the authentication, the RADIUS client residing within the SMF sends the User-Name and User-Password attributes in Access-Request message to the RADIUS server.

The SMF uses more attributes to facilitate the RADIUS authentication function. For the complete list of attributes supported, see the [RADIUS Attribute Definition, on page 18](#) section.

The RADIUS server validates the user with the authentication information. If the validation is successful, the server sends the Access-Accept response to the SMF.

#### PAP-, CHAP-, MSCHAP-based Authentication

The SMF decodes the Protocol Configuration Options (PCO), Extended PCO (ePCO), or Additional PCO (APCO) IE received from UE. Then, the SMF retrieves the values related to PAP (User Name and Password), CHAP (Challenge and Response), or MSCHAP (Challenge and Response) from the IE. If any of the protocols have higher precedence in configured priority under DNN, the SMF sends the received values in RADIUS Access-Request message to the RADIUS server.



**Note** The SMF does not include the authentication information received from the UE in the RADIUS Access-Request message if the priority is not configured.

By default, the SMF uses the configured host password under DNN for authentication until additional configuration is enabled to use the password received in PCO, ePCO, or APCO. The SMF allows the operator to configure the host password at DNN profile either in plain-text or encrypted form and always displays the same in encrypted format only wherever applicable.

The SMF sends MSISDN as the User Name if the UE does not provide the username explicitly in PCO IE for PAP-based authentication.

For CHAP-based authentication, the SMF converts the received CHAP Challenge and Response to MSCHAP if the **convert-to-mschap** command option is enabled, CHAP is enabled, and the received CHAP Response length is 49 bytes. By default, the SMF uses MSCHAPv1 as the authentication algorithm.

For MSCHAP-based authentication, the SMF sends User Name, Challenge, and Response received in PCO to the RADIUS server if Protocol ID is LCP and LCP container specifies the algorithm as CHAP/MSCHAPv1 (128) as per RFC 2433 or CHAP/MSCHAPv2 (129) as per RFC 2795.

The SMF forwards the authentication information from RADIUS server to UE in Create-Session-Response PCO/EPCO/APCO IE for a 4G/Wi-Fi session, and in N1 Container EPCO IE for a 5G session.

Consider the following important points while implementing the RADIUS authentication functionality.

- Perform the length validation of different AVPs applicable for this feature based on RFC 2865. Also, reject the authentication if any violation is identified.
  - The minimum length of CHAP Challenge is 5 bytes (even though it is 1 byte as per RFC 1334 and RFC 1994).
- The SMF sends the received authentication information from UE to RADIUS server based on the configured authentication algorithm at DNN level. The SMF does not manipulate any data received from UE and it only applies the configurations related to authentication before sending the information to RADIUS server.
- The SMF does not validate the use case of incrementing the Identifier value for every authentication as it does not allow multiple authentication during the PDU session lifetime.
- The SMF sends the encrypted NULL (empty) password in Access-Request when it receives empty password from UE and no host level password configured at SMF or **password-use-pco** option is enabled.
- The SMF falls back to the default authentication where Access-Request carries the configured server secret as User Password in the following scenarios:
  - If none of the algorithm preference is enabled with priority
  - If the UE provided information is not applicable for the configured algorithm preferences, if any
  - When the UE sends the empty PAP or CHAP containers without any data (the container length is 0)
- The SMF rejects the authentication in the following scenarios:
  - When there is no other algorithm configured for authentication

- Whenever there is a mismatch in CHAP identifier received in both CHAP Challenge and CHAP Response containers (the SMF currently copies the CHAP-ID from CHAP Challenge container)
  - CHAP-ID in CHAP Password must be taken from CHAP Response as per RFC 2865.
  - Response Identifier must be copied from the Identifier field of the Challenge Response as per RFC 1334.
  - Whenever the validation criteria of the current algorithm fails
- The SMF allows to configure the same priority through CLI for different algorithms because configuring 0 explicitly disables the configuration. In this scenario, any one of the algorithms is considered and the selection is purely implementation dependent. It is the responsibility of operator to ensure different algorithms have different priorities to resolve the conflicts whenever UE sends multiple authentication containers to the SMF.
- The SMF allows to configure the **password-use-pco** option without configuring PAP due to the limitation of Yang defined syntax format. The same is applicable for **convert-to-mschap** option. But the functionality will work only if the corresponding algorithm is enabled with the valid priority.
- By default, the SMF encrypts the operator given Host level password using AES-128-CFB encryption algorithm, if it's a plain-text. It ignores the encryption if the operator gives the already encrypted password which has to meet the AES-128-CFB encryption standard.
- By default, the SMF considers the authentication algorithm as MSCHAPv1(128) whenever the received CHAP Challenge and Response converted to MSCHAP if received CHAP-Response length is 49 bytes and **convert-to-mschap** option is enabled.
- The following are the list of MSCHAP specific AVPs supported at SMF and its RFC references:
  - MSCHAP-CHALLENGE (MSCHAP) □ RFC2548 Section 2.1.2
  - MSCHAP-RESPONSE □ RFC2548 Section 2.1.3
  - MSCHAP2-RESPONSE □ RFC2548 Section 2.3.2
  - MSCHAP-ERROR □ RFC2548 Section 2.1.5
  - MS-CHAP2-Success (RFC 2548, Section 2.3.3) is not supported as there is no clear information on MS-CHAP success AVP for v1 in RFC 2548.
- When the RADIUS server sends both MSCHAP-Error and Reply-Message AVPs in Access-Reject message, the preference is given to MSCHAP-ERROR while filling the CHAP container for NACK in PCO/APCO/EPCO. MSCHAP-Error is common for both MSCHAPv1 and MSCHAPv2 algorithm and it is encapsulated in the Message field of the CHAP Failure container.
- In MSCHAP, only the authentication functionality is supported.

**Important**

The SMF uses the inbuilt encryption algorithm “AES-128-CFB” for encrypting the host level password (outbound password) provided by NETCONF-YANG data model. The SMF Ops Center creates a global key, for AES-128-CFB encryption, which is used for encrypting the operator given plain-text password. It shares the key with all the pods via SSH for decrypting the encrypted data in the respective pods. The key is exported as a ENV variable “CONFD\_AES\_KEY” in SMF-SERVICE pod. If the operator wishes to configure the already encrypted password, then the AES-CFB-128 encrypted string should be prefixed with “\$\$” as follows, \$\$<encrypted-data> to indicate that the given input is already AES-128-CFB encrypted string to NETCONF-YANG model.

For CLI details associated with authentication, see the [Configuring the RADIUS Client, on page 27](#) section.

## RADIUS Authentication Attributes

### RADIUS Access Request Attributes

The following table lists the supported attributes in the RADIUS Access-Request message.

Attribute	Reference Specification	Encoding Type
USER-NAME	RFC2865 - 5.1	String
PASSWORD	RFC2865 - 5.2	Encrypted String
CALLING-STATION-ID	RFC2865 - 5.31	String
CALLED-STATION-ID	RFC2865 - 5.30	String
NAS-IP-ADDRESS	RFC2865 - 5.4	IPv4 Address
NAS-IDENTIFIER	RFC2865 - 5.32	String
SERVICE-TYPE	RFC2865 - 5.6	Octets - 4 bytes
FRAMED-PROTOCOL	RFC2865 - 5.7	Octets - 4 bytes
NAS-PORT-TYPE	RFC2865 - 5.41	Octets - 4 bytes
NAS-PORT	RFC2865 - 5.5	Octets - 4 bytes
3GPP-IMSI	3GPP 29.061 - 16.4.7.2-1	String
3GPP-CHARGING-ID	3GPP 29.061 - 16.4.7.2-2	Octets - 4 bytes
3GPP-PDP-TYPE	3GPP 29.061 - 16.4.7.2-3	Octets - 4 bytes
3GPP-CHARGING-GATEWAY-ADDR	3GPP 29.061 - 16.4.7.2-4	IPv4 Address
3GPP-GPRS-NEG-QOS-PROFILE	3GPP 29.061 - 16.4.7.2-5 3GPP 29.274 - 8.7	Special Encoded Octets
3GPP-SGSN-ADDRESS	3GPP 29.061 - 16.4.7.2-6	IPv4 Address
3GPP-GGSN-ADDRESS	3GPP 29.061 - 16.4.7.2-7	IPv4 Address
3GPP-IMSI-MCC-MNC	3GPP 29.061 - 16.4.7.2-8	String
3GPP-GGSN-MCC-MNC	3GPP 29.061 - 16.4.7.2-9	String

Attribute	Reference Specification	Encoding Type
3GPP-NSAPI	3GPP 29.061 - 16.4.7.2-10 3GPP 29.561 – 11.3	String
3GPP-SELECTION-MODE	3GPP 29.061 - 16.4.7.2-12	String
3GPP-CHARGING-CHARACTERISTICS	3GPP 29.061 - 16.4.7.2-13	String
3GPP-SGSN-MCC-MNC	3GPP 29.061 - 16.4.7.2-18	String
3GPP-IMEISV	3GPP 29.061 - 16.4.7.2-20	String
3GPP-RAT-TYPE	3GPP 29.061 - 16.4.7.2-21	Octet - 1 byte
3GPP-USER-LOCATION	3GPP 29.061 - 16.4.7.2-22 3GPP 29.274 - 8.21-4, 8.21-5 3GPP 38.413 – 9.3.1.7, 9.3.3.10	Special Encoded Octets
3GPP-MS-TIMEZONE	3GPP 29.061 - 16.4.7.2-23 3GPP 29.274 - 8.44	Special Encoded Octets
3GPP-NEGOTIATED-DSCP	3GPP 29.061 - 16.4.7.2-26	Octet - 1 byte
CHAP-PASSWORD (CHAP)	RFC2865 – 5.3	String
CHAP-CHALLENGE (CHAP)	RFC2865 – 5.40	String
MSCHAP-CHALLENGE (MSCHAP)	RFC2548 – 2.1.2	String
MSCHAP-RESPONSE	RFC2548 – 2.1.3	Octets
MSCHAP2-RESPONSE	RFC2548 – 2.3.2	Octets
MSCHAP-ERROR	RFC2548 – 2.1.5	String
REPLY-MESSAGE	RFC2865 – 5.18	String



**Note** The WiFi call attributes are the same as the 4G call.

### RADIUS Access Response Attributes

The following table lists the supported attributes in the RADIUS Access-Response message.

Attribute	Reference Specification	Encoding Type
FRAMED-IP	RFC2865 - 5.1	IPv4 Address
FRAMED-IPv6-PREFIX	RFC3162	PrefixLen and String
IDLE-TIMEOUT	RFC2865 - 5.28	Integer
SESSION-TIMEOUT	RFC2865 - 5.27	Integer



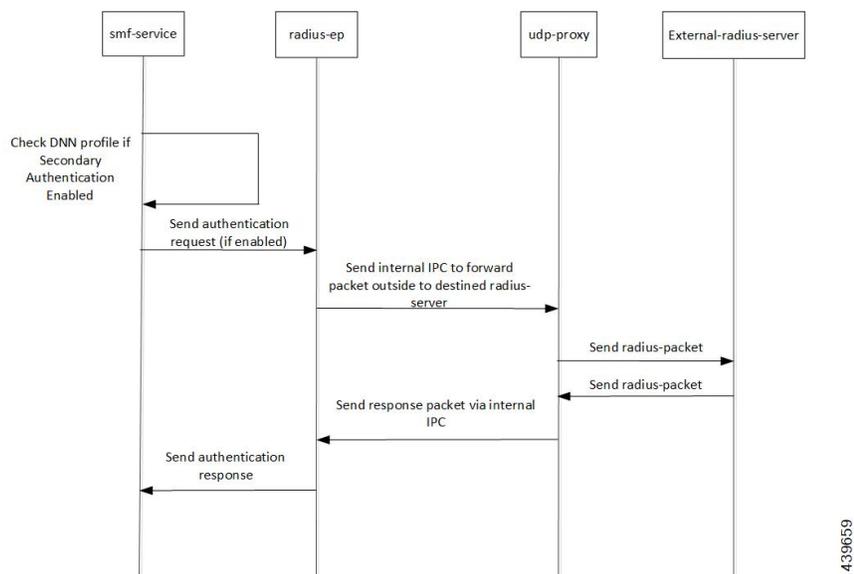
**Note** The WiFi call attributes are the same as the 4G call.

For complete description of the RADIUS authentication attributes, see the [RADIUS Attribute Definition](#), on page 18 section in this guide.

## Call Flows

### RADIUS Authentication Call Flow

The following figure illustrates the end to end call flow between the SMF server and RADIUS-EP.



**Table 3: RADIUS Authentication Call Flow**

Step	Description
1	Bringing up RADIUS-POD: Add the respective endpoint configuration, with VIP-IP similar to Protocol-EP VIP-IP. Add the RADIUS-server information to the profile-RADIUS configuration.
2	Add the secondary authentication configuration to the required DNN profiles.
3	During session-bringup, the DNN profile checks if secondary authentication is enabled after successful UDM validation. <ul style="list-style-type: none"> <li>• If authentication is not enabled, continue with PCF.</li> <li>• If authentication is enabled, send inter-process communication (IPC) message to RADIUS-POD to authenticate the subscriber.</li> </ul>
4	The RADIUS-POD prepares the Access Request packet that is destined to a configured RADIUS-server, sends the packet to UDP Proxy pod to proxy the packet out.

Step	Description
6	The UPD Proxy pod creates a socket (if not already present) and sends the packet to the RADIUS-server.
7	The RADIUS-server validates the Access Request. If accepted, it responds with the Access Accept message. Else, it responds with the Access Reject message.
8	The UDP Proxy responds to the respective RADIUS-EP instance.
9	The RADIUS-EP instance validates the response, fetches the framed-IP (if present), and updates the SMF-service.
10	The SMF-service, upon successful response from RADIUS-EP, continues with the PCF flow. Else, the SMF-service disconnects from the subscriber.

## RADIUS Interaction for Accounting

The SMF exchanges the following messages with RADIUS server through the RADIUS-client RADIUS-EP.

- **Accounting-Request:** This message carries any of the following packets to relay the accounting information to the RADIUS server.

- **Accounting Start packet:** This packet describes the type of service being delivered and the user it is being delivered to.

The SMF sends accounting-start packet during the session establishment procedure. The RADIUS Accounting server returns an acknowledgement upon receiving the accounting-start packet.

For details on configuring the RADIUS Accounting, see [Configuring the RADIUS Client, on page 27](#) section.

- **Accounting Stop packet:** This packet describes the type of service that was delivered and optionally statistics such as elapsed time, input and output octets, or input and output packets.

At the end of service delivery, the SMF sends the accounting-stop packet for all session deletion scenarios and when the RADIUS accounting is enabled during the call setup.

- **Accounting-Request Interim-Update:** During the session, the SMF sends the updated cumulative usage report to the RADIUS accounting server.

- **Accounting-Response:** For each successfully processed accounting request, the RADIUS server returns an accounting acknowledgment confirming the receipt of the information.

For CLI details associated with accounting, see the [Configuring the RADIUS Client, on page 27](#) section.

## RADIUS Accounting Attributes

The following table lists the RADIUS accounting attributes supported in the accounting-request message.

Attribute	Reference Spec	Encoding Type	Supported Accounting Type
USER-NAME	RFC 2865 - 5.1	String	Start, Stop, Interim update
CALLING-STATION-ID	RFC 2865 - 5.31	String	Start, Stop, Interim update
CALLED-STATION-ID	RFC 2865 - 5.30	String	Start, Stop, Interim update

Attribute	Reference Spec	Encoding Type	Supported Accounting Type
NAS-IP-ADDRESS	RFC 2865 - 5.4	IPV4 Address	Start, Stop, Interim update
NAS-IDENTIFIER	RFC 2865 - 5.32	String	Start, Stop, Interim update
SERVICE-TYPE	RFC 2865 - 5.6	Octets - 4 bytes	Start, Stop, Interim update
FRAMED-PROTOCOL	RFC 2865 - 5.7	Octets - 4 bytes	Start, Stop, Interim update
NAS-PORT-TYPE	RFC 2865 - 5.41	Octets - 4 bytes	Start, Stop, Interim update
NAS-PORT	RFC 2865 - 5.5	Octets - 4 bytes	Start, Stop, Interim update
3GPP-IMSI	3GPP 29.061 - 16.4.7.2-1	String	Start, Stop, Interim update
3GPP-CHARGING-ID	3GPP 29.061 - 16.4.7.2-2	Octets - 4 bytes	Start, Stop, Interim update
3GPP-PDP-TYPE	3GPP 29.061 - 16.4.7.2-3	Octets - 4 bytes	Start, Stop, Interim update
3GPP-CHARGING-GATEWAY-ADDR	3GPP 29.061 - 16.4.7.2-4	IPV4 Address	Start, Stop, Interim update
3GPP-GPRS-NEG-QOS-PROFILE	3GPP 29.061 - 16.4.7.2-5 3GPP 29.274 - 8.7	Special Encoded Octets	Start, Stop, Interim update
3GPP-SGSN-ADDRESS	3GPP 29.061 - 16.4.7.2-6	IPV4 Address	Start, Stop, Interim update This attribute is not included in the 5G accounting-start message.
3GPP-GGSN-ADDRESS	3GPP 29.061 - 16.4.7.2-7	IPV4 Address	Start, Stop, Interim update
3GPP-IMSI-MCC-MNC	3GPP 29.061 - 16.4.7.2-8	String	Start, Stop, Interim update
3GPP-GGSN-MCC-MNC	3GPP 29.061 - 16.4.7.2-9	String	Start, Stop, Interim update
3GPP-NSAPI	3GPP 29.061 - 16.4.7.2-10	String	Start, Stop, Interim update
3GPP-SELECTION-MODE	3GPP 29.061 - 16.4.7.2-12	String	Start, Stop, Interim update
3GPP-CHARGING-CHARACTERISTICS	3GPP 29.061 - 16.4.7.2-13	String	Start, Stop, Interim update
3GPP-SGSN-MCC-MNC	3GPP 29.061 - 16.4.7.2-18	String	Start, Stop, Interim update
3GPP-IMEISV	3GPP 29.061 - 16.4.7.2-20	String	Start, Stop, Interim update
3GPP-RAT-TYPE	3GPP 29.061 - 16.4.7.2-21	Octet - 1 byte	Start, Stop, Interim update

Attribute	Reference Spec	Encoding Type	Supported Accounting Type
3GPP-USER-LOCATION	3GPP 29.061 - 16.4.7.2-22 3GPP 29.274 - 8.21-4 3GPP 29.274 - 8.21-5	Special Encoded Octets	Start, Stop, Interim update
3GPP-MS-TIMEZONE	3GPP 29.061 - 16.4.7.2-23 3GPP 29.274 - 8.44	Special Encoded Octets	Start, Stop, Interim update
3GPP-NEGOTIATED-DSCP	3GPP 29.061 – 16.4.7.2-26	Octet – 1 byte	Start, Stop, Interim update This attribute is sent only if the associated configuration is present.
Acct-Status-Type	RFC 2866	Start/Stop/Interim	Start, Stop, Interim update
Accounting-Session-Id	RFC 2866	String	Start, Stop, Interim update
Acct-Delay-time	RFC 2866	Octet	Start, Stop, Interim update
Acct-Input-Octets	RFC 2866	Integer	Stop, Interim update
Acct-Output-Octets	RFC 2866	Integer	Stop, Interim update
Acct-Input-Gigawords	RFC 2869	Integer	Stop, Interim update
Acct-Output-Gigawords	RFC 2869	Integer	Stop, Interim update
Acct-Input-packets	RFC 2866	Integer	Stop, Interim update
Acct-Output-Packets	RFC 2866	Integer	Stop, Interim update
Acct-Session-Time	RFC 2866	Integer	Stop, Interim update
Acct-Terminate-Cause	RFC 2866	String	Stop
Framed-MTU	RFC 2866	String	Start, Stop, Interim update
3GPP-Session-Stop-Indicator	3GPP 29.061	Bit String	Stop
Framed-Ip-Addr	RFC 2866	IPV4 Address	Start, Stop, Interim update
Acct-Authentic	RFC 2866	String	Start, Stop, Interim update
EventTimeStamp	RFC 2869	String	Start, Stop, Interim update



**Note** The WiFi call attributes are the same as the 4G call.

For complete description of the RADIUS accounting attributes, see the [RADIUS Attribute Definition](#), on page 18 section in this guide.

## Call Flows

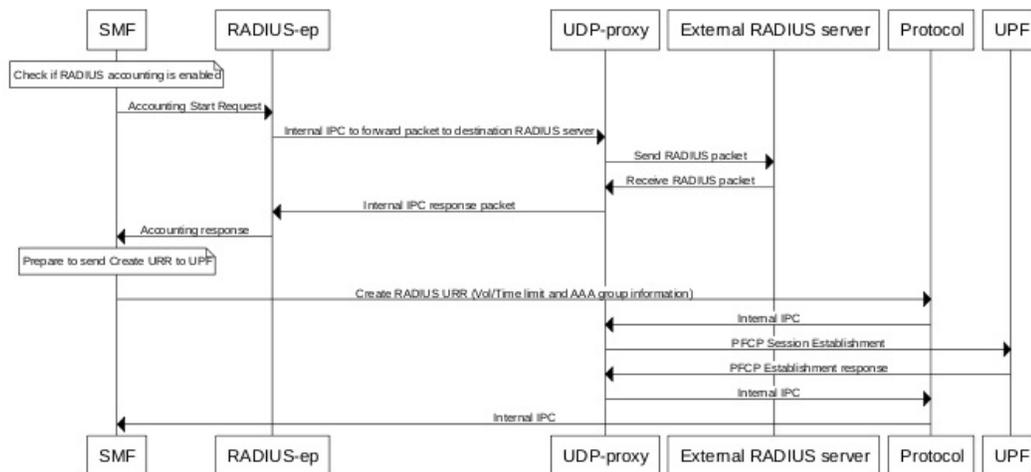
This section describes the following call flows:

- RADIUS Accounting Start Call Flow
- RADIUS Accounting Stop Call Flow
- RADIUS Accounting Interim-update Asynchronous Call Flow
- RADIUS Accounting Interim-update Synchronous Call Flow

### RADIUS Accounting Start Call Flow

This section describes the call flow associated with the initiation of RADIUS accounting procedure.

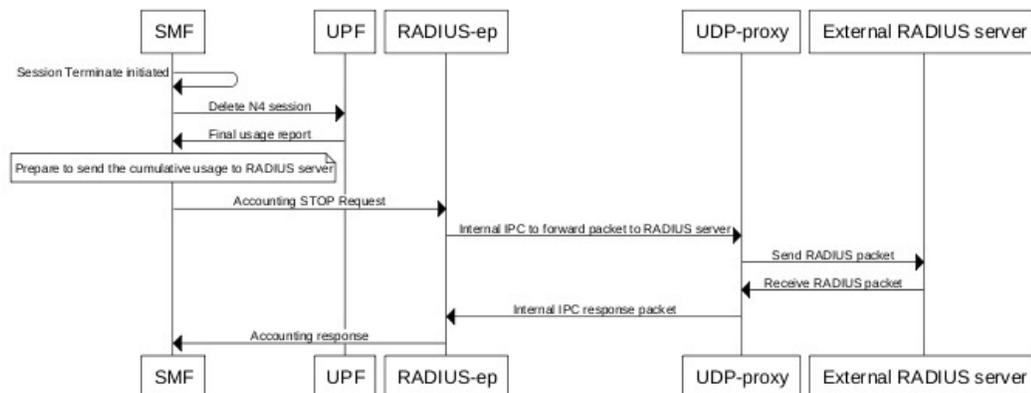
Figure 2: RADIUS Accounting Start Call Flow



### RADIUS Accounting Stop Call Flow

This section describes the call flow associated with the termination of RADIUS accounting procedure.

Figure 3: RADIUS Accounting Stop Call Flow

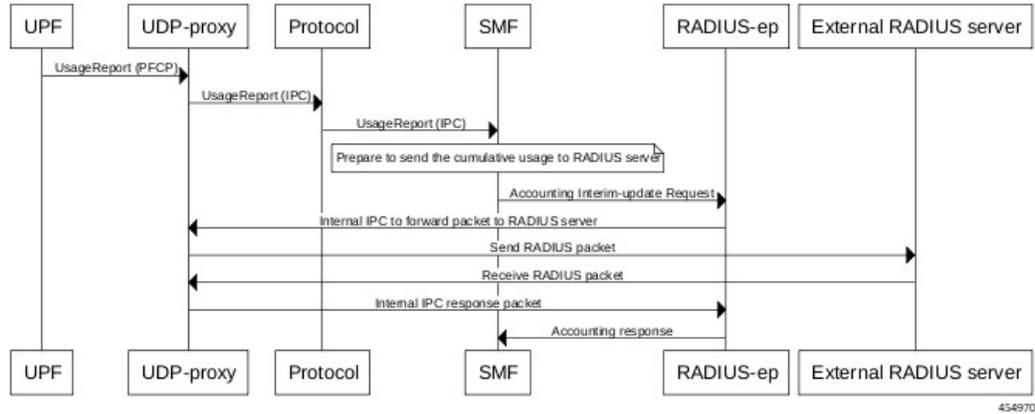


454969

### Asynchronous Accounting Interim-Update Call Flow

This section describes the call flow associated with the asynchronous interim-update request.

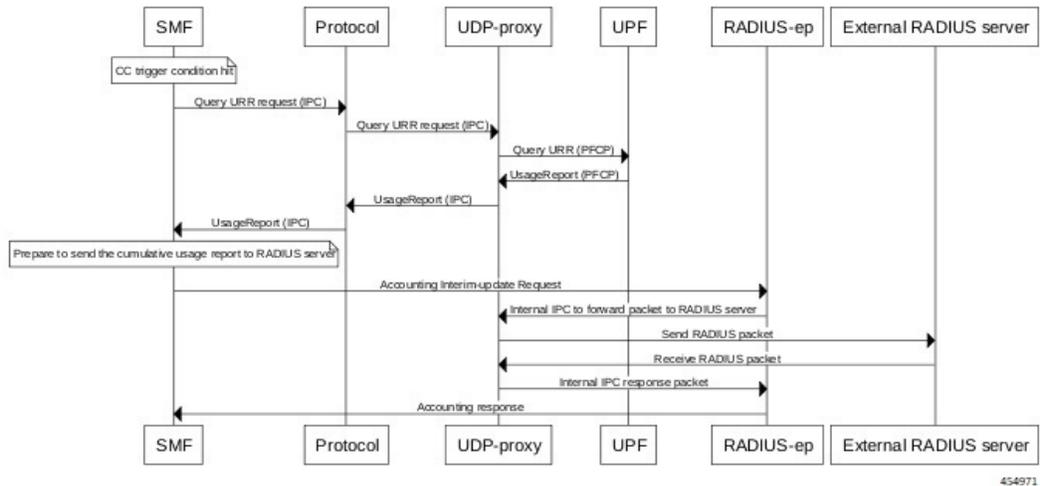
Figure 4: Asynchronous Accounting Interim-Update Call Flow



### Synchronous Accounting Interim-Update Call Flow

This section describes the call flow associated with the synchronous interim-update request.

Figure 5: Synchronous Accounting Interim-Update Call Flow



### Processing of Usage Reporting Rules

After enabling the RADIUS accounting, the SMF creates the Usage Reporting Rule (URR) and relays the rule to the UPF through the Create URR Information Element (IE). The Create URR IE is present in the N4 Session Establishment Request and it contains the volume and time limits as per the configuration.

The SMF associates the RADIUS URR only to the dynamic Packet Detection Rules (PDRs) and not for the static and predefined rules. With AAA group name in N4 session establishment request, the UPF associates the static and predefined PDRs with the RADIUS URR. The UPF sends the usage report for the RADIUS URR when the Volume limit or the Time limit is hit. Then, the SMF sends the usage in the Interim-Update Accounting-Request message to the RADIUS server.

The SMF receives the usage report for RADIUS URR in N4 Modification Response or N4 Deletion Response when any one of the following conditions are met:

- CC event condition is hit and the SMF performs Query URR
- Session Delete Response is sent

The SMF stores the values of Volume and Time thresholds reported for a previous session and reports the cumulative usage by adding the currently reported value to the stored value. The SMF sends the cumulative usage report in Accounting-Request Interim-Update and Accounting-Stop messages.

On receiving the usage report from UPF, the SMF identifies the URR IDs that are to be sent to the CHF server and to the RADIUS server. For example, if the URR ID is associated to “0x80 00 00 09”, then the SMF sends this URR ID to the RADIUS server, and the other URR IDs to the CHF server.

## Dynamic Configuration Update

The SMF allows you to dynamically change the RADIUS accounting configuration without impacting the existing sessions.

The following table identifies the impact of dynamic update to the various RADIUS accounting configurations.

**Table 4: Dynamic Update of RADIUS Accounting Configuration**

Configuration	Dynamic Change	Impact on Existing Sessions
Enabling and disabling of RADIUS accounting configuration	Allowed at the system level	The existing sessions continue to use the old value.
CC trigger updates	Allowed as per current pod replica	The existing session uses the new value.
Volume and time limit changes	Allowed at the system level	The existing sessions continue to use the old value.

## RADIUS Attribute Definition

The detailed description of each attribute is as follows:

- USER-NAME

**Description:** String value encoded as per RFC 2865.

- 5G call: GPSI value is used, with stripped-off "msisdn-"
- 4G call: MSISDN value is used, with stripped-off "msisdn-"



**Note** PAP, CHAP, and MSCHAP authentication methods are not supported in releases prior to 2020.02.x.

In release 2020.02.x and beyond, the PAP, CHAP, and MSCHAP authentication methods are supported.

- PASSWORD

**Description:** Encrypted string value encoded as per RFC 2865.

For both 5G and 4G calls, selected RADIUS server's "secret" is set as user-password.

- CALLING-STATION-ID

**Description:** String value encoded as per RFC 2865.

5G call: GPSI value is used, with stripped of "msisdn-"

4G call: MSISDN value is used, with stripped of "msisdn-"

- CALLED-STATION-ID

**Description:** String value encoded as per RFC 2865.

For both 5G and 4G calls, DNN value is set as called-station-id.

- NAS-IP-ADDRESS

**Description:** IPv4 address value encoded as per RFC 2865.

For both 5G and 4G calls, user-configured RADIUS Client interface-type's VIP-IP is used.

- NAS-IDENTIFIER

**Description:** String value encoded as per RFC 2865.

For both 5G and 4G calls, user-configured nas-identifier attribute value is used.

- SERVICE-TYPE

**Description:** 4-byte octet (int) value encoded as per RFC 2865.

For both 5G and 4G calls, "FRAMED (2)" value is set.

- FRAMED-PROTOCOL

**Description:** 4-byte octet (int) value encoded as per RFC 2865.

For both 5G and 4G calls, "GPRS-PDP-CONTEXT (7)" value is set.

- NAS-PORT-TYPE

**Description:** 4-byte octet (int) value encoded as per RFC 2865.

For both 5G and 4G calls, "WIRELESS-OTHER (18)" value is set.

- NAS-PORT

**Description:** 4-byte octet (int) value encoded as per RFC 2865.

For both 5G and 4G calls, the base value of respective instance is used. That is:

0x4000... 0x407F is set for replica-0

0x4080... 0x40FF is set for replica-1

- 3GPP-IMSI

**Description:** String value encoded as per *3GPP TS 29.061*.

5G call: SUPI value is used.

4G call: IMSI value is used.

- 3GPP-CHARGING-ID

**Description:** 4-byte octet (int) value encoded as per *3GPP TS 29.061*.

For both 5G and 4G calls, charging-ID is set.

- 3GPP-PDP-TYPE

**Description:** 4-byte octet (int) value encoded as per *3GPP TS 29.061*.

For both 5G and 4G calls, pdp-type is set as follows:

- 0 = IPv4
- 2 = IPv6
- 3 = IPv4v6

- 3GPP-CHARGING-GATEWAY-ADDR

**Description:** 4-byte octet (IPv4-address) value encoded as per *3GPP TS 29.061*.

For both 5G and 4G calls, charging gateway address is set.

- 3GPP-GPRS-NEG-QOS-PROFILE

**Description:** Octets (special encoding) value encoded as per *3GPP TS 29.061* and *29.274*.

For 5G call, the values from default-qos profile of the system are used and the encoding is performed as follows:

**Table 5: Non-GBR case**

1-2	<Release indicator>- = "15" (UTF-8 encoded)
3	"-" (UTF-8 encoded)
4-5	ARP (UTF-8 encoded)
6-7	5QI (UTF-8 encoded)
8-9	UL Session-AMBR length (UTF-8 encoded)
10-m	UL Session-AMBR (UTF-8 encoded)
(m+1) - (m+2)	DL Session-AMBR length (UTF-8 encoded)
(m+3) – n	DL Session-AMBR (UTF-8 encoded)

**Table 6: GBR case**

1-2	<Release indicator> = "15" (UTF-8 encoded)
3	"-" (UTF-8 encoded)
4-5	ARP (UTF-8 encoded)
6-7	5QI (UTF-8 encoded)
8-9	UL MFBR length (UTF-8 encoded)
10-m	UL MFBR (UTF-8 encoded)

(m+1)-(m+2)	DL MFBR length (UTF-8 encoded)
(m+3)-n	DL MFBR (UTF-8 encoded)
(n+1)-(n+2)	UL GFBR length (UTF-8 encoded)
(n+3)-o	UL GFBR (UTF-8 encoded)
(o+1) – (o+2)	UL GFBR length (UTF-8 encoded)
(o+3) - p	DL GFBR (UTF-8 encoded)

For 4G call, the values from the default-qos profile of the system are used and the encoding is performed as follows:

**Table 7: Non-GBR case**

1-2	<Release indicator>- = "08" (UTF-8 encoded)
3	"-" (UTF-8 encoded)
4-5	ARP (UTF-8 encoded)
6-7	5QI (UTF-8 encoded)
8-11	UL Session-AMBR (UTF-8 encoded)
12-15	DL Session-AMBR (UTF-8 encoded)

**Table 8: GBR case**

1-2	<Release indicator> = "08" (UTF-8 encoded)
3	"-" (UTF-8 encoded)
4-5	ARP (UTF-8 encoded)
6-7	5QI (UTF-8 encoded)
8-11	UL MBR (UTF-8 encoded)
12-15	DL MBR (UTF-8 encoded)
16-19	UL GBR (UTF-8 encoded)
20-23	DL GBR (UTF-8 encoded)

- 3GPP-SGSN-ADDRESS

**Description:** 4-byte octet (IPv4-address) value encoded as per *3GPP TS 29.061*.

For 5G call, the AMF address is set.

For 4G call, the S-GW address is set.

- 3GPP-GGSN-ADDRESS

**Description:** 4-byte octet (IPv4-address) value encoded as per *3GPP TS 29.061*.

For both 5G and 4G calls, the SMF-Service IP is set.

- 3GPP-IMSI-MCC-MNC

**Description:** String value encoded as per *3GPP TS 29.061*.

For 5G call, SUPIs MCC and MNC values are set.

For 4G call, IMSIs MCC and MNC values are set.

MCC is first 3 bytes, MNC is next 2 or 3 bytes.

If MCC value is any of the following, then MNC will be of 3 bytes, else MNC will be of 2 bytes.

300 302 310 311 312 313 316 334 338 342 344 346 348 354 356 358 360 365 376 405 708 722 732

- 3GPP-GGSN-MCC-MNC

**Description:** String value encoded as per *3GPP TS 29.061*.

For both 5G and 4G calls, configured MCC and MNC value of SMF is used.

MCC is first 3 bytes, and MNC is next 2 or 3 bytes.

- 3GPP-SGSN-MCC-MNC

**Description:** String value encoded as per *3GPP TS 29.061*.

For 5G call, AMFs MCC and MNC values are set.

For 4G call, SGWs MCC and MNC values are set.

MCC is first 3 bytes, and MNC is next 2 or 3 bytes.

- 3GPP-NSAPI

**Description:** String value encoded as per *3GPP TS 29.061*.

For 5G call, QFI value from the defaultQos profile is set.

For 4G call, EPS bearer ID is set.

- 3GPP-SELECTION-MODE

**Description:** String value encoded as per *3GPP TS 29.061*.

For both 4G and 5G calls, the value is set to "0".

- 3GPP-CHARGING-CHARACTERISTICS

**Description:** String value encoded as per *3GPP TS 29.061*.

For both 4G and 5G calls, generic charging character is set.

- 3GPP-IMEISV

**Description:** String value encoded as per *3GPP TS 29.061*.

For 5G call, PEI value is set.

For 4G call, IMEI value is set.

- 3GPP-RAT-TYPE

**Description:** 1-byte octet encoded as per *3GPP TS 29.061*.

For 5G call, value "NR (51)" is set.

For 4G call, value "EUTRAN (6)" is set.

For WLAN call, value "WLAN (3)" is set.

- 3GPP-USER-LOCATION

**Description:** Special octet value encoded as per *3GPP TS 29.061*.

For 5G call, the following encoding logic is used:

1	Location-Type Only TAI = 136 Only NCGI = 135 Both TAI + NCGI =137
2-7	TAI-Encoding (if present)
8-15	NCGI-Encoding (if present)

TAI Encoding header:

1	MCC digit 2	MCC digit 1
2	MNC digit 3	MCC digit 3
3	MNC digit 2	MNC digit 1
4-6	TAC value	

NCGI Encoding header:

1	MCC digit 2	MCC digit 1
2	MNC digit 3	MCC digit 3
3	MNC digit 2	MNC digit 1
4	SPARE	NCI
5-8	NR Cell Identifier (NCI)	

For 4G call, the following encoding logic is used:

1	Location-Type
	Only TAI = 128
	Only ECGI = 129
	Both TAI + ECGI =130
2-6	TAI-Encoding (if present)
7-13	ECGI-Encoding (if present)

TAI Encoding header:

1	MCC digit 2	MCC digit 1
2	MNC digit 3	MCC digit 3
3	MNC digit 2	MNC digit 1
4-5	TAC value	

ECGI Encoding header:

1	MCC digit 2	MCC digit 1
2	MNC digit 3	MCC digit 3
3	MNC digit 2	MNC digit 1
4	Spare	ECI
5-7	EUTRAN Cell Identifier (ECI)	

- 3GPP-MS-TIMEZONE

**Description:** Special octet value encoded as per *3GPP TS 29.061*.

Timezone string (for example: -07:00+1) is encoded as two-byte value as mentioned in the following table.

1	TIMEZONE The first byte timezone is encoded as per 3GPP 29.061, 3GPP 29.274, 3GPP 24.008, and 3GPP 23.040 (section 9.2.3.11).
2	DAYLIGHT SAVING 0, or +1 or +2 The second byte daylight consists of two bits used (00-0, 01-+1, 10-+2, 11 – Unused).

- 3GPP-NEGOTIATED-DSCP

**Description:** 1-byte octet encoded as per *3GPP TS 29.061*

For both 5G and 4G calls, DSCP configuration from DNN qos-profile configuration is used.

Sub -> DNN profile -> QosProfile -> DSCPMap -> Qi5 value check -> ARP priority check

- Acct-Status-Type

**Description:** Enum value encoded as per RFC 2866. The value of this attribute can be one of the following:

- 1 - Start
- 2 - Stop
- 3 - Interim Update

- Acct-Delay-Time

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of time client is trying to send the accounting record.

- Acct-Input-Octets

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of bytes received. This attribute contains 4 bytes.

The SMF wraps values when the number crosses the maximum value.

- Acct-Output-Octets

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of bytes transmitted. This attribute contains 4 bytes.

The SMF wraps values when the number crosses the maximum value.

- Acct-Input-Packets

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of packets received. This attribute contains 4 bytes.

The SMF wraps values when the number crosses the maximum value.

- Acct-Output-Packets

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of packets transmitted. This attribute contains 4 bytes.

The SMF wraps values when the number crosses the maximum value.

- Acct-Input-Gigawords

**Description:** Integer value encoded as per RFC 2869. This attribute indicates how many times the Acct-Input-Octets counter has wrapped around  $2^{32}$  over the course of this service being provided. This value is incremented whenever Acct-Input-Octets is wrapped.

- Acct-Output-Gigawords

**Description:** Integer value encoded as per RFC 2869. This attribute indicates how many times the Acct-Output-Octets counter has wrapped around  $2^{32}$  over the course of this service being provided. This value is incremented whenever Acct-Output-Octets is wrapped.

- Acct-Session-Id

**Description:** String value encoded as per RFC 2866. This attribute represents the unique accounting ID of subscriber. The accounting ID is unique to make it easy to match start and stop records in a log file. The start and stop records for a given session **MUST** have the same Acct-Session-Id. An Accounting-Request packet **MUST** have an Acct-Session-Id.

An Access-Request packet **MAY** have an Acct-Session-Id; if it does, then the NAS **MUST** use the same Acct-Session-Id in the Accounting-Request packets for that session. The Acct-Session-Id contains UTF-8 encoded 10646 characters.

- Acct-Session-Time

**Description:** Integer value encoded as per RFC 2866. This attribute represents the amount of time the subscriber is active.

- Framed-MTU

**Description:** This attribute indicates the Maximum Transmission Unit to be configured for the user, when it is not negotiated by some other means (such as PPP). The default value is 1500.

It **MAY** be used in Access-Accept packets. It **MAY** be used in an Access-Request packet as a hint by the NAS to the server that it would prefer that value, but the server is not required to honour the hint.

- Acct-Terminate-cause

**Description:** Enum value encoded as per RFC 2866. This attribute represents the reason for termination of subscriber.

- FRAMED-IP

The IPv4 address value decoded as per RFC 2865.

For both 4G and 5G calls, the received value is set as the IPv4 address for the subscriber.

- FRAMED-IPv6-PREFIX

The IPv6 Prefix + Length value decoded as per RFC 3162.

For both 4G and 5G calls, the received value is set as the IPv6 prefix for the subscriber.



---

**Important** If the received prefix-length is !=64, the SMF overrides to 64.

---

- IDLE-TIMEOUT

The 4-byte octet (integer) value encoded as per RFC 2865. This attribute is supported in the inbound RADIUS packet.

For both 4G and 5G calls, the received value is used as the maximum number of consecutive seconds of idle time that the user is permitted before being disconnected by the NAS.

- SESSION-TIMEOUT

The 4-byte octet (integer) value encoded as per RFC 2865. This attribute is supported in the inbound RADIUS packet.

For both 4G and 5G calls, the received value is used as the maximum number of seconds that the user is allowed to remain connected by the NAS.



---

**Note** The WiFi call attributes are the same as the 4G call.

---

## Standards Compliance

The RADIUS Client feature complies with the following standards:

- RFC 2865: Remote Authentication Dial in User Service (RADIUS)
- RFC 2866: RADIUS Accounting
- RFC 3162: RADIUS and IPv6
- 3GPP TS 29.061
- 3GPP TS 29.274

## Limitations and Restrictions

The SMF has the following limitations:

- The SMF supports only single RADIUS attribute profile, and does not support dictionary selection.
- If RADIUS accounting is enabled and server-group is configured within DNN profile, the SMF sends server-group as AAA group in charging-params in N4 session establishment. The UPF displays an error if there is a server group mismatch between SMF and UPF.

In this scenario, static and predefined usage are not accounted in the RADIUS URR. However, the dynamic rules traffic is accounted in the RADIUS URR.

- Currently, VLAN tagging for outbound RADIUS packets is based on the static routes configured on SMF. So, the overlapping AAA server addresses for different MVNOs where the VLAN tag must be different from SMF is not supported.

## Configuring the RADIUS Client

The RADIUS client provides both RADIUS authentication and accounting functionalities. For using these functionalities, it is important to enable the RADIUS authentication and accounting framework through the associated CLI configuration.

This section describes how to configure the RADIUS client.



### Important

Configuring the VIP-IP of the RADIUS client interface is mandatory for the RADIUS client to work. Also, the VIP-IP must be the same as the IP of the UDP proxy pod.

Configuring the RADIUS Client feature involves the following:

- [Configuring RADIUS Server, on page 27](#)
- [Configuring RADIUS Server Selection Logic, on page 29](#)
- [Configuring RADIUS Attributes, on page 29](#)
- [Configuring RADIUS Detect Dead Server, on page 30](#)
- [Configuring RADIUS Dead Time, on page 30](#)
- [Configuring RADIUS Retries, on page 31](#)
- [Configuring RADIUS Timeout, on page 31](#)
- [Configuring RADIUS Pod, on page 32](#)
- [Configuring RADIUS NAS-IP, on page 32](#)
- [Configuring Secondary Authentication Method, on page 33](#)
- [Configuring PAP, CHAP, or MSCHAP-based Authentication, on page 34](#)
- [Enabling RADIUS Accounting, on page 35](#)
- [Defining RADIUS Server Group in DNN Profile, on page 36](#)
- [Configuring RADIUS Accounting Options, on page 37](#)
- [Configuring RADIUS Accounting Server Group, on page 38](#)
- [Configuring the Session Disconnect Feature, on page 38](#)

## Configuring RADIUS Server

Use the following sample configuration to configure the RADIUS server.

```

config
  profile radius
    server ipv4_address port_num
    secret secret_key
    priority priority_value
    type { acct | auth }
    commit

```

**NOTES:**

- **profile radius:** Enter the RADIUS configuration mode.
- **server *ipv4\_address port\_num*:** Specify the IPv4 address and port of the RADIUS server.
- **secret *secret\_key*:** Specify the secret key.
- **priority *priority\_value*:** Specify the server priority.
- **type { acct | auth }:** Specify the type of the RADIUS server. The server can be one of the following:
  - **acct:** RADIUS server used for the accounting requests
  - **auth:** RADIUS server used for the authentication requests
- **commit:** Commit the configuration.

**Example**

The following is an example of the RADIUS server configuration.

```

profile radius
  server 1.2.3.4 1812
  secret  $8$73a0i4G3ILj0Np+8tn2QOoWDj3QkB+oefPc2ZK6RE6A=
  priority 1
  exit
  server 1.2.5.6 1812
  secret  $8$VccEEUVou7m5ptA9WZRPR7KDmxQ/L3K1J3QqgHjexkk=
  priority 2
  exit
exit

```

**Verifying the RADIUS Configuration**

Use the **show radius** command to display information about the RADIUS servers (both accounting and authentication) that are configured in the system.

The following configuration is a sample output of the **show radius** command:

```

bng# show radius
radius
-----
Server: 10.0.0.1, port: 1812, status: up, port-type: Auth
2 requests, 0 pending, 0 retransmits
1 accepts, 1 rejects, 0 timeouts
0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 4 ms latest rtt
-----
Server: 10.0.0.1, port: 1813, status: up, port-type: Acct
3 requests, 0 pending, 0 retransmits
3 responses, 0 timeouts

```

```
0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 1 ms latest rtt
-----
```

## Configuring RADIUS Server Selection Logic

Use the following sample configuration to configure the RADIUS server selection logic.

```
config
  profile radius
    algorithm { first-server | round-robin }
  commit
```

### NOTES:

- **profile radius**: Enter the RADIUS configuration mode.
- **algorithm { first-server | round-robin }**: Define the algorithm for selecting the RADIUS server.
  - **first-server**: Set the selection logic as highest priority first. This is the default behavior.
  - **round-robin**: Set the selection logic as round-robin order of servers.
- **commit**: Commit the configuration.

### Example

The following is an example of the RADIUS server selection logic configuration.

```
config
  profile radius
    algorithm round-robin
  exit
```

## Configuring RADIUS Attributes

Use the following sample configuration to configure the RADIUS attributes for authentication and accounting.

```
config
  profile radius
    attribute { nas-identifier value | nas-ip ipv4_address }
  commit
```

### NOTES:

- **profile radius**: Enter the RADIUS configuration mode.
- **attribute { nas-identifier value | nas-ip ipv4\_address }**: Configure the RADIUS identification parameters.
  - **nas-identifier value**: Specify the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
  - **nas-ip ipv4\_address**: Specify the NAS IPv4 address. *ipv4\_address* must be an IPv4 address in dotted decimal notation.
- **commit**: Commit the configuration.

**Example**

The following is an example of the RADIUS attributes configuration.

```
config
  profile radius
    attribute
      nas-identifier CiscoSmf
    exit
  exit
```

## Configuring RADIUS Detect Dead Server

Use the following sample configuration to configure the RADIUS detect dead server.

```
config
  profile radius
    detect-dead-server response-timeout value
  commit
```

**NOTES:**

- **profile radius:** Enter the RADIUS configuration mode.
- **detect-dead-server response-timeout *value*:** Set the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.  
*value* must be an integer in the range of 1–65535. Default: 10 seconds.
- **commit:** Commit the configuration.

**Example**

The following is an example of the RADIUS detect dead server configuration.

```
config
  profile radius
    detect-dead-server response-timeout 100
  exit
```

## Configuring RADIUS Dead Time

Use the following sample configuration to configure the RADIUS dead time.

```
config
  profile radius
    deadtime value
  commit
```

**NOTES:**

- **profile radius:** Enter the RADIUS configuration mode.
- **deadtime *value*:** Set the time to elapse between RADIUS server marked unreachable and when we can reattempt to connect.  
*value* must be an integer in the range of 1–65535. Default: 10 minutes.
- **commit:** Commit the configuration.

### Example

The following is an example of the RADIUS dead time configuration.

```
config
  profile radius
    deadtime 15
  exit
```

## Configuring RADIUS Retries

Use the following sample configuration to configure the maximum RADIUS retries.

```
config
  profile radius
    max-retry value
  commit
```

### NOTES:

- **profile radius:** Enter the RADIUS configuration mode.
- **max-retry *value*:** Set the maximum number of times that the system will attempt retry with the RADIUS server.  
*value* must be an integer in the range of 0–65535. Default: 2
- **commit:** Commit the configuration.

### Example

The following is an example of the RADIUS retries configuration.

```
config
  profile radius
    max-retry 2
  exit
```

## Configuring RADIUS Timeout

Use the following sample configuration to configure the RADIUS timeout.

```
config
  profile radius
    timeout value_in_seconds
  commit
```

### NOTES:

- **profile radius:** Enter the RADIUS configuration mode.
- **timeout *value\_in\_seconds*:** Set the time to wait for response from the RADIUS server before retransmitting.  
*value\_in\_seconds* must be an integer in the range of 1–65535. Default: 2 seconds.
- **commit:** Commit the configuration.

**Example**

The following is an example of the RADIUS timeout configuration.

```
config
  profile radius
    timeout 4
  exit
```

## Configuring RADIUS Pod

Use the following sample configuration to configure the RADIUS pod.

```
config

  endpoint radius
    replicas number_of_replicas
  commit
```

**NOTES:**

- **endpoint radius:** Enter the RADIUS endpoint configuration mode.
- **replicas *number\_of\_replicas*:** Set the number of replicas required.
- **commit:** Commit the configuration.

**Example**

The following is an example of the RADIUS pod configuration.

```
config

  endpoint radius
    replicas 3
  exit
```

## Configuring RADIUS NAS-IP

This section describes how to configure the RADIUS NAS-IP.

**Multiple RADIUS NAS-IP Configuration**

Use the following sample configuration to configure multiple RADIUS NAS-IP addresses at various levels.

```
config
  profile radius
    attribute nas-ip-address ipv4_address
    accounting attribute nas-ip-address ipv4_address
    server-group group_name attribute nas-ip-address ipv4_address
    server-group group_name accounting attribute nas-ip-address ipv4_address

  commit
```

**NOTES:**

- **profile radius:** Enter the RADIUS configuration mode.

- **attribute nas-ip-address *ipv4\_address***: Set the global NAS-IP address value.
- **accounting attribute nas-ip-address *ipv4\_address***: Set the global accounting NAS-IP address value.
- **server-group *group\_name* attribute nas-ip-address *ipv4\_address***: Set the per server-group common NAS-IP address value.
- **server-group *group\_name* accounting attribute nas-ip-address *ipv4\_address***: Set the per server-group accounting NAS-IP address value.
- **commit**: Commit the configuration.

### Example:

The following is an example of the multiple RADIUS NAS-IP configuration.

```

config
profile radius
attribute
nas-ip-address 1.2.3.4
exit
accounting
attribute
nas-ip-address 1.2.3.5
exit
exit
server-group grp1
attribute
nas-ip-address 1.2.3.6
exit
accounting
attribute
nas-ip-address 1.2.3.7
exit
exit
server-group grp2
attribute
nas-ip-address 1.2.3.78
exit
accounting
attribute
nas-ip-address 1.2.3.9
exit
exit
exit
exit

```

## Configuring Secondary Authentication Method

Use the following sample configuration to configure the secondary authentication method.

```

config
profile dnn dnn_name
authentication secondary radius [ group group_name ]
commit

```

### NOTES:

- **profile dnn *dnn\_name***: Enter the DNN Profile configuration mode.

- **authentication secondary radius [ group *group\_name* ]**: Enable secondary authentication under the DNN profile and sets method as RADIUS.  
**group *group\_name***: This keyword is optional. This keyword defines the RADIUS server group name.
- **commit**: Commit the configuration.

### Example

The following is a configuration example of the secondary authentication method.

```
config
  profile dnn intershat
  ...
  authentication secondary radius
exit
```

## Verifying the RADIUS Authentication Configuration

Use the **show radius auth-server** command to display detailed statistics for RADIUS authentication server and port.

The following configuration is a sample output of the **show radius auth-server** command:

```
bng# show radius auth-server
-----
Server: 10.0.0.1, port: 1812, status: up, port-type: Auth
2 requests, 0 pending, 0 retransmits
1 accepts, 1 rejects, 0 timeouts
0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 4 ms latest rtt
-----
```

## Configuring PAP, CHAP, or MSCHAP-based Authentication

This section provides the configuration to enable the PAP, CHAP, and MSCHAP-based RADIUS authentication. This configuration aids in converting the CHAP Challenge and Response received in PCO IE as MSCHAP Challenge and Response.

### Defining Priority for Authentication Algorithm

Use the following sample configuration to define the priority for different authentication algorithms (PAP or CHAP or MSCHAP) for RADIUS-based authentication in SMF.

```
config
  profile dnn profile_name
    authentication { { secondary radius [ group group_name ] | { algorithm
  { pap priority_value [ password-use-pco ] | chap priority_value [
convert-to-mschap ] | mschap priority_value } }
    end
```

#### NOTES:

- **password-use-pco**: This keyword overrides the DNN configured password with PCO password. The default setting is disabled.

If the host level password is not configured at DNN, then the SMF uses the UE given password for PAP-based authentication even though this configuration is disabled.

- **convert-to-mschap**: This keyword converts the received CHAP Challenge and Response to MSCHAP if the CHAP Response length is 49 bytes. Otherwise, the SMF sends as CHAP only even though this configuration is explicitly enabled.
- The default priority for PAP, CHAP, and MSCHAP algorithms is 0 which means that the configuration is disabled. The valid values are 1, 2, and 3. Lower the value, higher is the priority. It is used to resolve conflicts if the UE sends multiple authentication parameters in the PCO, EPCO, or APCO IE.

## Configuring Host Password

Use the following sample configuration to specify the host password at DNN level which is used as a password for PAP-based authentication.

```
config
  profile dnn profile_name
    outbound password password
  end
```

### NOTES:

- **profile dnn *profile\_name***: Specify the DNN profile name as an alphanumeric string to enter the DNN configuration mode.
- **outbound password *password***: Specify the DNN host password for authentication. By default, the SMF sends this password in PAP user-password if it is not explicitly overridden using the **password-use-pco** option.

By default, the SMF encrypts the given password using AES-128-CFB encryption algorithm.

## Enabling RADIUS Accounting

Use the following sample configuration to enable RADIUS accounting on SMF and configure the RADIUS accounting specific parameters.

```
config
  profile charging charging_profile_name
    accounting limit { duration value | volume { downlink value | total
value | uplink value } }
    accounting triggers [ ambr-change | plmn-change | qos-change |
rat-change | serv-node-change | ue-time-change | user-loc-change ]
    commit
```

### NOTES:

- **profile charging *charging\_profile\_name***: Specify the charging profile name. *charging\_profile\_name* must be an alphanumeric string.
- **accounting**: Specify this option to enable RADIUS accounting on SMF for the subscribers.
- **limit { duration *value* | volume { downlink *value* | total *value* | uplink *value* } }**: Specify the volume and time limits for RADIUS accounting.
  - duration *value***: Specify the time duration value as an integer in the range of 0–2147483647.
  - downlink *value***: Specify the downlink volume limit for interim generation in bytes, as an integer in the range of 100000–4000000000.

**total value:** Specify the total volume limit for interim generation in bytes, as an integer in the range of 100000–4000000000.

**uplink value:** Specify the uplink volume limit for interim generation in bytes, as an integer in the range of 100000–4000000000.

- **accounting triggers** [ **ambr-change** | **plmn-change** | **qos-change** | **rat-change** | **serv-node-change** | **ue-time-change** | **user-loc-change** ]: Enable the appropriate RADIUS accounting triggers according to the following conditions:
  - AMBR change
  - PLMN change
  - Quality of Service change
  - Routing Area Information change
  - Serving node change
  - Traffic Flow Template (TFT) change
  - UE time change
  - User Location Information change - applicable only for P-GW and GGSN.




---

**Important** Enabling any one of these triggers turns off the remaining triggers.

---

- **commit:** Commit the configuration.

## Defining RADIUS Server Group in DNN Profile

Use the following sample configuration to set RADIUS server-group to use for accounting in DNN profile. All subscribers under the specified DNN will have RADIUS accounting enabled.

```
configure
  profile dnn dnn_profile_name
    accounting server-group group_name
  commit
```

### NOTES:

- **profile dnn** *dnn\_profile\_name*: Specify the DNN profile name to enter the DNN configuration mode. *dnn\_profile\_name* must be an alphanumeric string.
- **accounting server-group** *group\_name*: Specify the RADIUS server-group to use for accounting in the configured DNN profile. *group\_name* must be an alphanumeric string.
- **commit:** Commit the configuration.

## Configuring RADIUS Accounting Options

Use the following sample configuration to configure the RADIUS accounting options.

```
config
  profile radius accounting
    algorithm { first-server | round-robin }
    attribute { nas-identifier value | nas-ip ipv4_address }
    deadtime value
    detect-dead-server response-timeout value
    max-retry value
    timeout value
  commit
```

### NOTES:

- **profile radius accounting:** Enter the RADIUS accounting configuration mode.
- **algorithm { first-server | round-robin }:** Define the algorithm for selecting the RADIUS server.
  - **first-server:** Set the selection logic as highest priority first. This is the default behavior.
  - **round-robin:** Set the selection logic as round-robin order of servers.
- **attribute { nas-identifier value | nas-ip ipv4\_address }:** Configure the RADIUS identification parameters.
  - **nas-identifier value:** Specify the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
  - **nas-ip ipv4\_address:** Specify the NAS IPv4 address. *ipv4\_address* must be an IPv4 address in dotted decimal notation.
- **deadtime value:** Set the time to elapse between RADIUS server marked unreachable and when we can re-attempt to connect.  
*value* must be an integer from 0 through 65535. Default: 10 minutes.
- **detect-dead-server response-timeout value:** Set the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.  
*value* must be an integer from 1 through 65535. Default: 10 seconds.
- **max-retry value:** Set the maximum number of times that the system will attempt retry with the RADIUS server.  
*value* must be an integer in the range of 0–65535. Default: 2
- **timeout value:** Set the time to wait for response from the RADIUS server before retransmitting.  
*value* must be an integer in the range of 1–65535. Default: 2 seconds.
- **commit:** Commit the configuration.
- All the keyword options under the RADIUS accounting configuration mode are also available within the RADIUS configuration mode.

## Configuring RADIUS Accounting Server Group

Use the following sample configuration to configure the RADIUS server group.

```
config
  profile radius
    server-group group_name
  commit
```

### NOTES:

- **profile radius:** Enter the RADIUS configuration mode.
- **server group *group\_name*:** Specify the name of server group for use in RADIUS accounting. *group\_name* must be an alphanumeric string.
- **commit:** Commit the configuration.

## Verifying the RADIUS Accounting Configuration

Use the **show radius acct-server** command to display statistics for RADIUS accounting server and port.

The following configuration is a sample output of the **show radius acct-server** command:

```
bng# show radius acct-server
-----
Server: 10.0.0.1, port: 1813, status: up, port-type: Acct
3 requests, 0 pending, 0 retransmits
3 responses, 0 timeouts
0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 1 ms latest rtt
-----
```

## Configuring the Session Disconnect Feature

This section describes how to configure the Session Disconnect feature.

Configuring the Session Disconnect feature in SMF involves the following steps:

- [Configuring the Dynamic Authorization Service, on page 38](#)
- [Configuring the CoA-NAS Interface, on page 39](#)

## Configuring the Dynamic Authorization Service

Use the following sample configuration to enable the NAS as an authentication, authorization, and accounting (AAA) server for the dynamic authorization service. This service supports the RADIUS Disconnect and Change of Authorization (CoA) functionality.

```
config
  profile radius-dynamic-author
    client ipv4_address [ secret shared_secret ]
    nas-identifier value
    secret shared_secret
  end
```

### NOTES:

- **profile radius-dynamic-author**: Enter the dynamic authorization configuration mode.
- **client ipv4\_address [ secret shared\_secret ]**: Specify the IP address of the Dynamic Authorization Client. *ipv4\_address* must be in standard IPv4 dotted decimal notation.

You can add a list of client IPs from which the Disconnect message is accepted.

**secret shared\_secret**: This is an optional keyword. Specify the secret key at the client level.




---

**Important** Configuring the server key at the client level overrides the server key configured at the global level.

---

- **nas-identifier value**: Specify the dynamic authorization specific NAS-Identifier value. *value* must be an alphanumeric string of 1 to 64 characters.

If this keyword is configured, it is validated against the value received in DM request. If this keyword is not configured, the input value is silently ignored. That is, the DM requests from unlisted or unauthenticated clients are silently discarded.

- **secret shared\_secret**: Specify the global shared secret key of the server.

## Verifying the Session Disconnect Feature Configuration

This section describes how to verify the configuration associated with the Session Disconnect feature.

To view the information about the RADIUS Dynamic Authorization Clients that are configured in the system, use the **show radius-dyn-auth** command.

The following is a sample output of the **show radius-dyn-auth** command.

```
[unknown] smf# show radius-dyn-auth
radius-dyn-auth
-----
IP: 10.0.0.2
-----
COA:
0 total-requests      0 inprocess-requests
  0 retry-request-drops 0 invalid-requests
  0 bad-authenticators 0 internal-errors
0 ack-sent            0 nak-sent
-----
DISCONNECT:
0 total-requests      0 inprocess-requests
  0 retry-request-drops 0 invalid-requests
  0 bad-authenticators 0 internal-errors
0 ack-sent            0 nak-sent
-----
UnknownTypesRcvd: 0
-----
```

## Configuring the CoA-NAS Interface

Use the following sample configuration to define Change of Authorization (CoA) NAS interface in the RADIUS endpoint.

```
config
    endpoint radius
```

```
interface coa-nas
  vip-ip ipv4_address vip-port port_number
end
```

**NOTES:**

- **endpoint radius:** Enter the RADIUS endpoint configuration mode.
- **interface coa-nas:** Enter the CoA NAS interface configuration mode. This keyword defines a new interface "coa-nas".
- **vip-ip ipv4\_address vip-port port\_number:** Specify the IP address of the host. *ipv4\_address* must be in standard IPv4 dotted decimal notation.

You can configure a list of VIP-IPs to listen to the inbound CoA or DM requests.

**vip-port port\_number:** Specify the port number of the UDP proxy. By default, the port number is 3799. This default value is used only when the VIP-IP is specified.




---

**Important** This configuration allows only port to be specified per IP.

---

The SMF (udp-pxy) listens to the inbound CoA or DM request messages on these ports, and ACK or NAK messages sent with the respective source IP and port.

## RADIUS Test CLI support

The **RADIUS test** CLI provides a mechanism for testing network connectivity with and configuration of RADIUS authentication and accounting servers.

This functionality is useful in determining the accuracy of the system RADIUS configuration, the configuration of the subscriber profile on the RADIUS server and troubleshooting the server response time.

## Testing a RADIUS Accounting Server

When used to test a RADIUS accounting server, the tool generates an accounting request message for a specific username.




---

**Note** The user name must already be configured on the RADIUS authentication server prior to executing the test.

---

To execute the RADIUS authentication test tool, enter the following command:

```
test-radius accounting { all | server-group group_name | server server_name
port server_port } { user_name client_nas_ip_address }
```

**NOTES:**

- **all:** Specify that all configured RADIUS accounting servers be tested.
- **radius group group\_name:** Specify the configured RADIUS authentication servers in a RADIUS server group named *group\_name* for server group functionality.

- *server\_name*: Specify the IP address of a specific RADIUS accounting server to test.
- *server\_port*: Specify the TCP port over that the system must use when communicating with the RADIUS accounting server to test.
- *user\_name*: Specify a username that is supplied to the RADIUS server for accounting.
- *client\_nas\_ip\_address*: Specify the IP address of the source NAS that is supplied to the RADIUS server for accounting.

### Example

The following command verifies all the RADIUS servers.

```
test-radius accounting all
```

The following command verifies the RADIUS accounting for user *user1* for the *sampleServer*.

```
test-radius accounting server sampleServer port 5000 username user1
```

The following command verifies the RADIUS accounting server group *star1* for user *user1*.

```
test-radius accounting server-group star1 username user1
```

## Testing a RADIUS Authentication Server

When used to test a RADIUS authentication server, the tool generates an authentication request message for a specific user name.



**Note** The user name must already be configured on the RADIUS authentication server prior to executing the test.

To execute the RADIUS authentication test tool, in the Exec mode, use the following command:

```
test-radius authentication { all | server-group group_name | server server_name
port server_port } { user_name password client_nas_ip_address }
```

### NOTES:

- **all**: Specify that all configured RADIUS authentication servers be tested.
- **radius group** *group\_name*: Specify the configured RADIUS authentication servers in a RADIUS server group named *group\_name* for server group functionality.
- *server\_name*: Specify the IP address of a specific RADIUS authentication server to test.
- *server\_port*: Specify the TCP port over that the system must use when communicating with the RADIUS authentication server to test.
- *user\_name*: Specify a username that is supplied to the RADIUS server for authentication.
- *password*: Specify the password associated with the username that is supplied to the RADIUS server for authentication.
- *client\_nas\_ip\_address*: Specify the IP address of the source NAS that is supplied to the RADIUS server for accounting.

### Example

The following command verifies all the RADIUS servers.

```
test-radius authentication all
```

The following command verifies the RADIUS authentication for user *user1* for the *sampleServer*.

```
test-radius authentication server sampleServer port 5000 username user1  
password dummyPwd
```

The following command verifies the RADIUS authentication server group *star1* for user *user1*.

```
test-radius authentication server-group star1 username user1
```

## RADIUS Client OA&M Support

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

### Statistics Support

#### RADIUS Authentication Statistics

This feature supports the following statistics related to RADIUS Authentication:

- SMF-Service:
  - Number of Secondary-Authentication requests sent
  - Number of Secondary-Authentication response received
- RADIUS-EP:
  - Number of Secondary-Authentication requests sent
  - Number of Secondary-Authentication response received
  - Number of RADIUS packets sent
  - Number of RADIUS packets received

#### RADIUS Accounting Statistics

The SMF maintains the following statistics to track the total number of attempted, successful, and failed RADIUS Accounting Start, Accounting Update Interim and Accounting Terminate requests and responses.

- SMF\_SERVICE\_STATS for the following procedure types:
  - radius\_initial: This counter gets incremented for Accounting Start request and response.
  - radius\_update: This counter gets incremented for Accounting Interim Update request and response.
  - radius\_terminate: This counter gets incremented for Accounting Terminate request and response.

### RADIUS Access Management Statistics

The following statistics track the number of times the AVP is received in the RADIUS Access-Accept messages at SMF.

- SmfRadiusMessageStats

INBOUND:

- radius\_access\_accept
  - radius\_avp\_session\_timeout
  - radius\_avp\_idle\_timeout

### PAP, CHAP, or MSCHAP-based Authentication Statistics

The SMF supports the following statistics to track the number of times the AVP sent in Access-Request messages.

**Group:** smf\_radius\_message\_stats

**Format:** {app\_name, cluster, data\_center, direction, instance\_id, message\_type, radius\_avp\_type, rat\_type, service\_name}

**message\_type:** radius\_access\_request

**radius\_avp\_type:**

- radius\_avp\_pap\_user\_password
- radius\_avp\_pap\_username
- radius\_avp\_chap\_challenge
- radius\_avp\_chap\_response
- radius\_avp\_mschap\_challenge
- radius\_avp\_mschap\_response

#### Example:

```
smf_radius_message_stats{app_name="SMF",cluster="Local",data_center="DC",direction="outbound",instance_id="0",message_type="radius_access_request",radius_avp_type="radius_avp_pap_user_password",rat_type="NR",service_name="smf-service"} 1
```

```
smf_radius_message_stats{app_name="SMF",cluster="Local",data_center="DC",direction="outbound",instance_id="0",message_type="radius_access_request",radius_avp_type="radius_avp_pap_username",rat_type="NR",service_name="smf-service"} 1
```

The SMF supports these additional statistics to track the number of attempted, successful and failed responses received due to PAP, CHAP, and MSCHAP authentication.

**Group:** radius\_authentication\_message\_stats

**Format:** {app\_name, cluster, data\_center, dnn, instance\_id, radius\_auth\_algorithm, rat\_type, reason, service\_name, status}

**radius\_auth\_algorithm:**

- radius\_auth\_algorithm\_default

- radius\_auth\_algorithm\_pap
- radius\_auth\_algorithm\_chap
- radius\_auth\_algorithm\_mschap

**rat\_type:**

- NR
- EUTRA
- WLAN

**status:**

- decode\_failed
- encode\_failed
- attempted
- success
- failed
- timeout

**reason:**

- parse\_error
- invalid\_code
- invalid\_option
- invalid\_pco
- invalid\_epco
- invalid\_apco
- write\_error

**Example:**

```
radius_authentication_message_stats{app_name="SMF",cluster="Local",
data_center="DC",dnn="intershat2",instance_id="0",
radius_auth_algorithm="radius_auth_algorithm_default",rat_type="NR",reason="",
service_name="smf-service",status="attempted"} 2

radius_authentication_message_stats{app_name="SMF",cluster="Local",
data_center="DC",dnn="intershat2",instance_id="0",radius_auth_algorithm="radius_auth_algorithm_default",
rat_type="NR",reason="",service_name="smf-service",status="success"} 2

radius_authentication_message_stats{app_name="SMF",cluster="Local",data_center="DC",
dnn="intershat",instance_id="0",radius_auth_algorithm="radius_auth_algorithm_chap",
rat_type="EUTRA",reason="",service_name="smf-service",status="attempted"} 2

radius_authentication_message_stats{app_name="SMF",cluster="Local",
data_center="DC",dnn="intershat",instance_id="0",radius_auth_algorithm="radius_auth_algorithm_chap",
rat_type="EUTRA",reason="",service_name="smf-service",status="failed"} 2
```

### **RADIUS Disconnect and CoA Request Related Statistics**

The RADIUS endpoint (radius-ep) pod supports the following statistics.

#### **Radius\_Server\_Status**

Description: Display the active or inactive status of RADIUS server.

Metrics-Type: Gauge

Metrics-Value: 1 – ActiveServer, 0 – Inactive Server

Labels:

- Label: radSvrIP
  - Description: Server IP Address
  - Value: <any-ip-address>
  
- Label: radSvrPort
  - Description: Server Port
  - Value: <any-port>
  
- Label: radSvrPortType
  - Description: Authentication or Accounting type
  - Value: Auth, Acct

#### **Radius\_Requests\_Current**

Description: Displays the outstanding authentication and accounting requests

Metrics-Type: Gauge

Labels:

- Label: radMsgCode
  - Description: RADIUS Message Type
  - Values: SecondaryAuthenReq, RadiusAcctReq, TestAuth, TestAcct
  
- Label: radSvrIP
  - Description: Server IP Address
  - Value: <any-ip-address>
  
- Label: radSvrPort
  - Description: Server Port
  - Value: <any-port>
  
- Label: radSvrPortType
  - Description: Authentication or Accounting type

- Value: Auth, Acct
- Label: dnn
  - Description: DNN of subscriber
  - Value: <string>
- Label: procType
  - Description: Procedure-type
  - Value: <string>
- Label: ratType
  - Description: RAT type of subscriber
  - Value: <string>
- Label: sessType
  - Description: Session-type of subscriber
  - Value: <string>

### **Radius\_Requests\_Statistics**

Description: Displays the total authentication and accounting requests transmitted, retransmitted, and responses received

Metrics-Type: Counter

Labels:

- Label: radMsgCode
  - Description: Radius Message Type
  - Values: SecondaryAuthenReq, RadiusAcctReq, TestAuth, TestAcct
- Label: radPacketType
  - Description: Direction of packet
  - Value: Tx, Rx, Retry\_Tx
- Label: radResult
  - Description: Result of operation
  - Value: Success, Failed, Timeout, Failure\_Reject, ...
- Label: radSvrIP
  - Description: Server IP Address
  - Value: <any-ip-address>

- Label: radSvrPort
  - Description: Server Port
  - Value: <any-port>
- Label: radSvrPortType
  - Description: Authentication or Accounting type
  - Value: Auth, Acct
- Label: dnn
  - Description: DNN of subscriber
  - Value: <string>
- Label: procType
  - Description: Procedure-type
  - Value: <string>
- Label: ratType
  - Description: RAT type of subscriber
  - Value: <string>
- Label: sessType
  - Description: Session-type of subscriber
  - Value: <string>

#### **Radius\_CoaDM\_Requests\_Current**

Description: Displays the outstanding CoA and DM requests being processed.

Metrics-Type: Gauge

Labels:

- Label: radMsgCode
  - Description: RADIUS Message Type
  - Values: DisconnectRequest, CoARequest
- Label: radSvrIP
  - Description: Server IP Address
  - Value: <any-ip-address>

#### **Radius\_CoaDM\_Requests\_Statistics**

Description: Displays the total CoA and DM requests received and processed.

Metrics-Type: Counter

Labels:

- Label: radMsgCode
  - Description: Radius Message Type
  - Values: DisconnectRequest, DisconnectACK, DisconnectNAK, CoARequest, CoaDMReq, CoAACK
- Label: radPacketType
  - Description: Direction of packet
  - Value: Tx, Rx
- Label: radResult
  - Description: Result of operation
  - Value: Success, Failure\_Invalid\_Request, Failure\_Drop\_Retry\_Coa, Failure\_Unknown\_Error...
- Label: radSvrIP
  - Description: Server IP Address
  - Value: <any-ip-address>
- Label: nakErrorCause
  - Description: Error-cause set during COA-NAK / DM-NAK (not applicable for other cases)
  - Value: Missing-Attribute, NAS-Identification-Mismatch, Unsupported-Service, Invalid-Attribute-Value, Session-Context-Not-Found, Internal-Error