

UPF Local Configuration

This chapter covers the following topics:

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- Configuring the Local Configuration Support for UPF, on page 3
- TACACS+ TLS support, on page 4

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-UPF
Applicable Platform(s)	VPC-SI 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
First Introduced.	2020.02.0

Feature Description

The support for processing static and predefined rules in Control and User Plane Separation of EPC nodes (CUPS) architecture is dependent on the ruledef, rulebase, and charging action. For processing L3/L4 static and predefined rules, this information is made available at the control-plane in CUPS architecture. The control plane sends all these information to the associated user-plane using the PFD management message. The UPF cannot use the PFD management message to work with CN-SNF. With this feature, the local configuration support for the User Plane Function (UPF) is enabled, which allows the UPF to work with CN-SNF.

How it Works

The Access Control System (ACS) command line interface (CLI) is configured on the user-plane and the CLI module sends it to the ACS Controller (ACSCtrl). The ACSCtrl verifies the CLI and sends it to the Session Controller (SessCtrl). The SessCtrl stores the configuration in the SCT.

The SessCtrl maintains and stores different configuration types in a skiplist. When the length of the skiplist reaches the maximum (BULK configuration length) for a particular configuration type, the entire list is pushed in BULK from the Sessctrl to the Session Manager (SessMgr). As a result, the number of messenger event/message transactions between proclets is greatly reduced since the configurations are sent in BULK in a single message. On the expiry of the bulk configuration timeout (2 seconds), the Bulk Configuration timer – which runs constantly at the Session Controller – pushes the different types of configurations to the SessMgrs.

- The following configuration types are supported for the Bulk Configuration push:
- Ruledef
- Charging Action
- Action Priority Lines
- Group of Ruledef Configuration
- Rule in Group of Ruledef Configuration
- Rulebase L3/L4/L7 Info Configuration
- APN Configuration
- ACS service Configuration

The configurations are pushed only through the bulk push mechanism for configurations that are either added or modified. On the other hand, when configurations are deleted, it is removed immediately without waiting for any response from the Bulk configuration push timer. The deleted configuration is removed from the SCT and other SessMgrs immediately.



Note The Bulk configuration timeout function is invoked forcefully to push all the pending configurations to the SessMgrs before pushing the configuration delete to avoid any race conditions.

- The configuration changes applied to all the new and existing calls are listed in Table as follows.

Table 3: Configuration Changes on New and Existing Call Flows

Change in Configuration	Impact on Existing Calls Current Flows	Impact on Existing Calls New Flows	Impact on New Calls
Existing ruledef contents/New rule addition	Rule match is not enforced on existing flows after configuration change. TRM is not disengaged on existing flows. This may lead to billing issues if ruledef contents were changed for ongoing flows.	The configuration changes apply on new flows. For new flows, anyways fresh rule match would happen and the ruledef changes are applied on new flows for existing calls.	The configuration changes apply on new calls. For new flows, anyways fresh rule match would happen and the ruledef changes are applied on flows for new calls.
No Ruledef	Rule in use cannot be deleted.	Rule in use cannot be deleted	Rule in use cannot be deleted
New Group of Ruledefs/Changes to existing Group of Ruledefs contents (Add or Delete Rule in Group of Ruledefs)	Rule match is not enforced on existing flows after configuration change. TRM is not disengaged on existing flows. This may lead to billing issues if Group of Ruledefs contents were changed for ongoing flows.	The configuration changes apply on new flows. For new flows, anyways fresh rule match would happen and the Group of Ruledefs changes are applied on new flows for existing calls.	The configuration changes apply on new calls. For new flows, anyways fresh rule match happens and the Group of Ruledefs changes are applied on flows for new calls.
No Group of Ruledefs	Group of Ruledefs in use cannot be deleted	Group of Ruledefs in use cannot be deleted	Group of Ruledefs in use cannot be deleted
No Rule in GoR	Flows continue to match the ruledef defined in Group of Ruledefs unless the ruledef itself is deleted	New flows go through a fresh rule match and configuration change takes effect.	New flows go through a fresh rule match and configuration change takes effect.
Action Priority Changes/Action Priority addition	TRM is not disengaged for ongoing flows. configuration changes do not apply on existing flows	Configuration changes apply on new flows.	Configuration changes apply on new calls.
No Action Priority	No Impact on existing flows	Configuration changes apply on new flows.	Configuration changes apply on new calls.
Rulebase parameters change	Some parameter changes apply on existing calls	Some parameter changes apply on existing calls	Configuration changes apply on new calls
No Rulebase	No Rulebase is not supported	No Rulebase is not supported	No Rulebase is not supported
No APN	No APN is not supported	No APN is not supported	No APN is not supported
IP source violation	No impact on existing calls	No impact on existing calls	Configuration changes apply on new calls.

Configuring the Local Configuration Support for UPF

Use the following CLI commands to configure the User Plane Function (UPF) locally.

```
configure
  require upf
  end
```

TACACS+ TLS support

Table 4: Feature history

Feature name	Release	Feature dscription
Added TACACS+ TLS support	2026.03.0	StarOS now supports TLS-secured TACACS+ communication, enabling encrypted AAA traffic and certificate-based trust validation between StarOS nodes and TACACS+ servers. This enhancement strengthens management-plane security while maintaining compatibility with existing TACACS+ deployments. TLS support must be enabled on both the StarOS node and the TACACS+ server to establish secure communication.

TACACS+ TLS support is an optional security enhancement, disabled by default, that integrates TLS 1.3 for strong mutual authentication and encryption. Administrators may enable this feature to secure administrative traffic, effectively replacing insecure MD5 obfuscation methods with robust TLS protection

TACACS+ TLS support is a security enhancement that:

- Establishes a TLS 1.3 session to the TACACS+ server,
- Enables certificate-based mutual authentication (mTLS) between the client and server, and
- Replaces insecure MD5 obfuscation methods with robust TLS protection.

Policy-driven TLS management allows network administrators to manage certificates via the platform secret store without requiring service restarts.

Configure TACACS+ TLS

Enable TLS transport for TACACS+ to secure administrative traffic.

Before you begin

- Ensure the `.pem` files for the client and CA certificates are accessible via the configured URL.
- Verify that the TACACS+ server supports TLS 1.3.

Follow these steps to configure TACACS+ TLS:

Procedure

Step 1 Configure the client certificate and private key for mutual TLS (mTLS) authentication with the TACACS+ server

```
[local]# config
[local](config)# certificate name tls-client pem url /sftp/client_v3.crt private-key pem url
/sftp/client_v3.key
[local](config)# end
```

Step 2 Configure the CA certificate to verify the TACACS+ server certificate during the TLS handshake

```
[local]# config
[local](config)# certificate name tls-ca pem url /sftp/ca.pem private-key pem url /sftp/ca.key
[local](config)# end
```

Step 3 Associate the client certificate profile with the TACACS+ server for TLS connections. The profile name must match the certificate name configured in Step 1.

```
[local](config)# tacacs mode
[local](config-tacacs)# server priority 1 ip-address 10.88.0.1 port <6049> password <secret>
[local](config-tacacs)# tls-profile <profile-name>
```

Step 4 Associate the CA certificate profile with the TACACS+ server for server certificate verification. The profile name must match the CA certificate name configured in Step 2.

```
[local](config-tacacs)# tls-ca-profile <ca-profile-name>
```

Step 5 (Optional) Enable hostname/IP verification to prevent man-in-the-middle attacks by verifying that the server certificate Subject Alternative Name (SAN) matches the configured server IP.

```
[local](config-tacacs)# tls-verify-hostname
```

To disable hostname verification (default):

```
[local](config-tacacs)# no tls-verify-hostname
```

Step 6 Complete the TACACS+ configuration.

```
[local](config-tacacs)# exit
[local](config)# aaa tacacs+
[local](config)# end
```

The TACACS+ server is configured to use the TLS port (6049) instead of the cleartext port (49), and the TACACS+ connection is established over a secure TLS 1.3 tunnel.

TACACS+ TLS monitoring and troubleshooting

This section provides the commands and field descriptions required to monitor TLS-enabled TACACS+ sessions.

Use the `show tacacs session statistics` command to verify the status of your TLS-enabled sessions. The output confirms if TLS is active for the current session, displaying the associated profiles or 'N/A' if the feature is disabled.

Output Examples

When TACACS+/TLS is enabled:

```
tls enabled                    : yes
tls profile                    : tls-client
tls ca profile                 : tls-ca
```

When TACACS+/TLS is disabled:

```
tls enabled                    : no
tls profile                    : N/A
tls ca profile                 : N/A
```