

VPP Support

Vector Packet Processing (VPPMOB) is a mobility-centric solution based on fd.io's VPP, an open source solution. It leverages fd.io development, particularly in the areas of IP forwarding, routing, and protocols.

- Revision History, on page 1
- Charging Support, on page 2
- Delay-Charging Via Rule Base, on page 2
- Flow Idle-time Out, on page 3
- HTTP Support, on page 3
- IP Readdressing, on page 3
- DNS Readdress Server List, on page 3
- LTE Handover, on page 5
- Next Hop, on page 5
- PDN Update, on page 5
- Policing, on page 6
- Pure-S Support, on page 7
- Response-based Charging via Service Schema, on page 7
- Response-based TRM via Service Schema, on page 7
- ToS Marking, on page 7
- Volume-based Offload, on page 8
- Supported Functionality, on page 8
- Limitations, on page 9
- Enabling Fast Path in User Plane Service, on page 9
- Enabling VPP on SI Platform, on page 9
- Monitoring and Troubleshooting VPP Fast Path, on page 10
- Support for VPP Configuration Parameters Override, on page 10

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
With this release, support has been added for VPP ctrl for per flow Fast Path information in CUPS.	21.27.x
With this release, support has been added for DNS Readdress Server List.	21.25.4
First introduced	Pre 21.24

Charging Support

Usage Reports are notified to the billing server on call deletion or volume/time threshold breach.

When a stream is created on the User Plane, flows – that involve Charging, are associated with charging-specific operations that are set during the stream-creation. The charging counters for all flows – both offloaded and non-offloaded, are maintained on the Fast Path.

During an overflow in the volume threshold, the Fast Path sends a notification with bucket counters (PUSH mode) and in the case of time threshold hit, Applications reads charging counters from Fast Path (PULL mode). The User Plane aggregates these counters with its respective URRs and triggers usage reports over the Sx interface.



Important In this release, the URR support is there for both Volume and Time Threshold. Multiple SDF and one bearer level URRs are supported.

Delay-Charging Via Rule Base

The flavors of delay-charging supported are as follows:

- Charge-to-application all-packets – All control packets (Handshake, midsession, and tear-down) on flow are charged to the application packet matched charging-action.
- Charge-to-application initial-packets – Handshake packets on flow are charged to the application packet matched charging-action.
- Charge-to-application tear-down-packets – Tear-down packets on flow are charged to the application packet matched charging-action.
- Charge-separate-from-application – All control packets are rule-matched and charged to the highest priority rule.

In all the preceding scenarios only the charging is delayed, but the rule-matching occurs on the packet contents.



Important

- Charge-separate-from-application mid session packets are not supported. For offloaded flow, they continue to match the last matched rule.

When you enable the delay-charging feature, the TCP handshake packet hits the rule when it arrives. The TCP handshake packet hits the IP or TCP rule that is based on the configuration. The **show active-charging** CLI command still sees the TCP handshake packet hitting the default rule. This rule is not considered for charging until the first L7 packet arrives. Once the first L7 packet hits the L7 rule, while sending the quota request, the L7 packet and the TCP handshake packet get included in the same L7 RG.

Flow Idle-time Out

Configurable idle-time out is supported for the maximum duration of 24 hours. In earlier releases, support was available only for specific set of values.

HTTP Support

Analysis of HTTP traffic and policy matching of such HTTP-based rules is supported in this release. Offloading for HTTP flows is supported only for WebSocket, CONNECT method, or if content is present in request/response.

IP Readdressing

IP readdressing for IPv4 and IPv6 is supported in this release.

IP readdressing is configurable using the charging-rule or post-processing rule associated with the charging-action.

Streams are created on Fast Path for flows that match these rules along with the IP Readdressing operation set. All these flows - both offloaded and non-offloaded – will have IPv4/IPv6 address set in the Fast path.

DNS Readdress Server List

Whenever you use an unauthorized DNS server, the request is modified to readdress the DNS IPs to use the authorized servers. **Ruledef** determines if a packet belongs to a DNS query and if the DNS query belongs to a set of authorized DNS servers or not. If the DNS query does not belong to the authorized DNS servers, the flow action is to pick up DNS servers from the **readdress-server-list**.

A **readdress-server-list** is configured under the active charging server. When the flow matches a **ruledef**, the flow action can be configured to use the servers from the **readdress-server-list**.

Configure the **readdress-server-list** under **active-charging service** as follows:

```
configure
  active-charging service service_name
    readdress-server-list name_of_list
      server ipv4_address [ port ]
      server ipv6_address [ port ]
```

Note A maximum of 10 servers can be configured in a **readdress-server-list** and a maximum of 10 **readdress-server-lists** can be configured under active-charging service. Both IPv4 and IPv6 addresses can be configured in the same **readdress-server-list**.

Select the **readdress-server-list** from the list using one of the following two ways:

- Round-robin—Server selection occurs in a round-robin manner for every new flow. Inactive servers in the list are not considered during the selection.
- Hierarchy—The servers that are tagged in this approach are primary, secondary, tertiary, and so on, depending on the order they are defined in the **readdress-server-list**. All flows are readdressed to the primary server as long as it is available. If the Primary server goes down, then flows are readdressed to the secondary server and the same logic recurs. Once, the primary server is active then flows switch back to the primary server for readdressing.

The following CLI command defines the approach for a server selection.

```
charging-action action_name
   flow action readdress-server-list name_of_list [ hierarchy | round-robin
 ]
```

The **round-robin** option is considered as the default option, when no option is provided in the CLI command that is mentioned in the preceding code.

Configure the following CLI command under active-charging service.

```
configure
   active-charging service service_name
      readdress-server-list name_of_list
         server ipv4_address [ port ]
         server ipv6_address [ port ]
         consecutive-failures integer_value
         response-timeout integer_value
         reactivation-time integer_value

         charging-action action_name
            flow action readdress server-list name_of_list
      exit
```

Note Consider the following values to configure the CLI command mentioned in the preceding code.

- **consecutive-failures**—Integer value must range between 1–10. The default value is *5*.
- **response-timeout**—Integer value must range between 1–10000 milliseconds. The default value is *1000*.
- **reactivation-time**—Integer value must range between 1–1800 seconds. The default value is *300*.

Readdress Server States

The readdress server states are described as follows:

- Active state—Once configured, all servers are marked as Active.
- Inactive state—If no response is received from the readdressed server, then the server is marked as Inactive.
- Active-Pending state—Once the server is in Active-Pending state, it is available to accept the requests for readdressing. In this state, if a request is readdressed to this server and response is returned from it, then the server-state is changed to Active. Otherwise, it is moved back to Inactive state.

LTE Handover

The following types of handovers are supported:

- S-GW Relocation for X2 based handovers (OI set to 1).
- S-GW Relocation for S1 based handovers (OI set to 0).
- eNodeB F-TEIDu Update.

For S-GW relocation, the following combinations are supported:

- P-GW anchored call.
- P-GW anchored call to Collapsed call.
- Collapsed call to P-GW anchored call.

Next Hop

Next hop address for IPv4 and IPv6 is supported in this release.

The Next-Hop address is configurable using the charging-rule or post-processing rule associated with the charging-action.

Streams are created on Fast Path for flows that match these rules along with the Next Hop operation set. All these flows - both offloaded and non-offloaded – will have Next Hop address set in the Fast path.

PDN Update

PDN Update procedures are supported with VPP in this release.

All flows are onloaded to SM-U whenever Rule Addition/Modify/Removal is received through any Gx procedures. All the packets on these onloaded flows are then sent to SM-U. The flows are also onloaded when transport level marking and charging parameters changes for the flow. These flows are again offloaded on the packet for which rule-match changes, or Transaction Rule Matching (TRM) engages again.

Policing

The policer configuration uses inputs from the session manager, these inputs are received either from PCRF as AMBR or from flow-level QoS information. The values received from the PCRF is always accepted for session level AMBR policing. But, the flow-level policing is prioritized, if available, and sequentially the AMBR policing is applied. In other words, the policer engine applies the hierarchical policing - first the flow-level/rule bandwidth limiting and then the session level bandwidth limiting.



Note AMBR modifications during session run-time through RAR or CCA-U is applicable.

The input values received from the session manager are pushed into a policer configuration and a policer token bucket. For each direction - uplink or downlink, a new record is created for Policer configuration and Policer token bucket.

The Policer configuration is the reference for the policer engine, and the policer token bucket is used for calculation and restoration of values.

Currently, Policing is supported for AMBR received from PCRF and Rule-level QoS information for dynamic rules. For static and predefined rules, bandwidth limiting is achieved by the bandwidth policy configuration. Extended bit rates configured in bandwidth-policy configuration in Active Charging Service Configuration mode on Control Plane is provided to the User Plane as part of the configuration push mechanism, and same is applied for policing by User Plane. The following is an example configuration of bandwidth policy:

```
configure
   active-charging service ACS
      bandwidth-policy BWP

         flow limit-for-bandwidth id 1 group-id 1

         flow limit-for-bandwidth id 2 group-id 2
         group-id 1 direction uplink peak-data-rate 256000 peak-burst-size 32000
violate-action discard
         group-id 1 direction downlink peak-data-rate 256000 peak-burst-size 32000
violate-action discard
         group-id 2 direction uplink peak-data-rate 128000 peak-burst-size 16000
violate-action discard
         group-id 2 direction downlink peak-data-rate 56000 peak-burst-size 7000
violate-action discard
         exit
```

Limitations

In this release, Policing has the following limitations:

- Modification of **bandwidth-policy** is not supported.
- Interaction with other features such as - ITC bandwidth limiting, token replenishment (both APN level and ACL level) is not supported.
- Currently, policer-based statistics are not supported.

Note As policer statistics are not yet supported, the operator can verify bandwidth limiting using network performance monitoring tools.

Pure-S Support

Pure-S default bearer VPP integration is now supported in the CUPS Architecture. Earlier, Pure-S calls on CUPS were supported using IFTASK. Now, Pure-S call data path also uses VPP.

As part of VPP integration for Pure-S calls, calls on SAEGW-UP will install one bearer stream (3 Tuple – GTPU Service IP address, TEID, VRF id) per direction and also one TEP row per direction is created.

Supported Functionality:

Supported functionality for Pure-S includes:

- Most procedures for Collision between MME and Network Initiated scenarios (MBR/CBR/UBR/DBR).
- DBCmd and BRCmd commands.
- SAEGW-UP supports movement of IP transport from IPv4 to IPv6, or IPv6 to IPv4 during IDLE to ACTIVE transition, and handover procedures on S1-u interface. Transport selected on S1-u at the time of attach is also supported. For example, eNode handover from IPv4 eNodeB to IPv6 eNodeB.

Response-based Charging via Service Schema

HTTP Request is charged to the HTTP Response matched charging-action.

Response-based TRM via Service Schema

The Transaction Rule Matching (TRM) on uplink stream is engaged only after the HTTP response is received.

ToS Marking

Feature Description

ToS Marking for IPv4 and IPv6 is supported in this release.

The inner IP ToS marking address is configurable using the charging-rule or post-processing rule associated with the charging-action. The outer IP ToS marking is performed using the QCI-DSCP marking table configured on the control plane.

Streams are created on Fast Path for flows that match these rules along with the operations set. All these flows - both offloaded and non-offloaded – will have IPv4/IPv6 ToS marking set in the Fast path.

Volume-based Offload

In case of HTTP protocol, the content in request/response (if present) gets offloaded to fastpath for each transaction in a flow. The last packet of the content switches back the stream to passive state and the packet reaches the Session Manager.

Supported Functionality

The following call flavors are supported in this release:

- Pure-P IPv4/IPv6 calls.
- Collapsed IPv4/IPv6 calls.
- Default bearer.
- Pure-S functionality.
- Dedicated bearer.
- Handovers.

The following functionalities are supported in this release:

- ToS marking of the payload packets (Charging action) and outer GTP-U packets (QCI/QoS mapping table).
- Next hop feature (IPv4/IPv6).
- IP Readdressing feature (IPv4/IPv6).
- Post processing rules with action as discard.
- Post Processing rules with action as Next hop forwarding (IPv4/IPv6).
- Post Processing rules with action as ToS marking (UL, and DL).
- Post Processing rules with action as Readdressing (IPv4/IPv6).
- URR functionality (Gz only) - One SDF, and one bearer level URR.
- Only Gz charging is supported.
- Fragmentation and reassembly is supported in VPP.
- HTTP traffic policy match is supported. HTTP offload support is only for CONNECT and WebSocket requests.
- This release has been validated to support up to 5000 flows across all applications per subscriber. Although this limit is not imposed by the software, it is the recommended operating limit. Exceeding this limit may lead to application failures and so, it is recommended that the following CLI be configured in the Rulebase Configuration mode: **flow limit-across-applications** *5000*.

Limitations

The following functionalities are not supported in this release:

- Gy and Rf are supported independently, however, they both cannot be enabled at the same time for the same subscriber.
- Fast Path CLI can be disabled if it was previously enabled. However, User Plane must be reloaded.
- Generation of crash records and mini-core files are supported. Generation of full core files for VPP is not supported.
- When the node manager pod restarts, the ARP entries related to the UDP protocol are lost. This causes the VPP to drop the first message post recovery. Therefore, the call gets attached only after the ARP is resolved.

Enabling Fast Path in User Plane Service

Use the following CLI commands to enable Fast Path (VPP) in User Plane service.

```
configure
  context context_name
    user-plane-service service_name
      associate fast-path service
      end
```

NOTES:

- **fast-path**: Specifies the Fast Path related parameters.
- **service**: Specifies the Fast Path related configurations.

Enabling VPP on SI Platform

To launch VPP:

1. Log on to host machine, and create an ISO image that contains the file: *staros_param.cfg*
2. Create a file that has the line: FORWARDER_TYPE=vpp
3. Create an ISO file containing the *staros_param.cfg* file:

   ```
   genisoimage -l -o ssi_vpp.iso -r vppiso/
   ```

   If genisoimage is not installed, execute:

   ```
   sudo apt-get install genisoimage
   ```

4. Stop the VM if it is running:

   ```
   virsh destroy <vm_name>
   ```

5. If a disk is already attached to the VM that does not have VPP identified as the forwarder, then detach the disk.

Run the **dumpxml** command on the VM to see if there is a disk attached.

To detach the disk, execute:

```
virsh detach-disk <vm_name> hdc –config
```

6. Attach the ISO file that contains the *staros_param.cfg* file:

```
virsh attach-disk <vm_name> <Path_of_ISO_FILE> hdc –type cdrom –config
```

Monitoring and Troubleshooting VPP Fast Path

To determine if the flows are offloaded, check for Fast Path statistics in the output of the following CLI commands:

- **show subscribers user-plane-only full all**
- **show user-plane-service all**
- **show user-plane-service statistics analyzer name ip**
- **show user-plane-service statistics analyzer name ipv6**
- **show user-plane-service statistics analyzer name tcp**
- **show user-plane-service statistics analyzer name udp**
- **show user-plane-service statistics analyzer name http**

To determine the per flow Fast Path information, check for Fast Path Info fields in the output of the following CLI command:

- **show subscribers user-plane-only callid** *call_id* **flows full**

Support for VPP Configuration Parameters Override

To configure the VPP Configuration parameters, see the *VPC-SI Administration Guide*. These parameters can be overriden. Ensure that you contact your Cisco account representative to assist in identifying the override values.