



# Virtual Routing and Forwarding

- [Feature Summary and Revision History, on page 1](#)
- [Feature Description, on page 2](#)
- [Configuring VRF, on page 4](#)
- [Monitoring and Troubleshooting, on page 6](#)

## Feature Summary and Revision History

### Summary Data

*Table 1: Summary Data*

Applicable Product(s) or Functional Area	5G-UPF
Applicable Platform(s)	VPC-SI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	<i>UCC 5G UPF Configuration and Administration Guide</i>

### Revision History

*Table 2: Revision History*

Revision Details	Release
Support is added for the following functionality: <ul style="list-style-type: none"><li>• Overlapping IP Pools</li><li>• Removal of mandatory VRF ordering between SMF and UPF.</li></ul>	2021.01.0
First introduced.	2020.02.0

## Feature Description

Virtual Routing and Forwarding (VRF) is a technology that allows multiple instances of a routing table to co-exist within the same router at the same time. As the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other.

In UPF, the feature enables association of IP address pools with VRF. These IP pools are chunked like any pools. The chunks from this pool are allocated to the UPFs that are configured to use these pools. VRF-associated pools in UPF can only be of type—STATIC or PRIVATE.

The chunks from the PRIVATE VRF pool are allocated when the UPF comes up for registration, similar to the normal private pools. For STATIC VRF pool, the SMF does chunk allocation to UPF during the time of configuration. An SX-Route-Update message is sent for pre-allocated static-chunks during UPF registration.

## Overlapping IP Pool

Overlapping pools share and use an IP address range. Overlapping pools can either be of type STATIC or PRIVATE. No public pools can be configured as overlapping pools. Each overlapping pool is part of different VRF (routing domain) and pool-group. Since an APN can use only one pool-group, overlapping pools are part of different APN as well.

Without this functionality, overlapping pools can be configured at SMF. However, chunks from two overlapping pools can't be sent to the same UPF. That is, the UP can't handle chunks from two different overlapping pools. So, same number of UPFs and overlapping pools are required for sharing same IP address range.

With this functionality, UPF can handle chunks from two different overlapping pools. So, a single UP can handle any number of overlapping pools sharing the same IP range.

The functionality of overlapping pools in same UPF includes:

- When a chunk from particular pool is installed on an UP, its corresponding vrf-name is sent along with the chunk.
- The UPs are made VRF-aware of chunks and therefore, UPs install chunks on the corresponding VRFs and the chunk database is populated under the VRFs.
- During call allocation, release, recovery, or any communication towards VPNMgr, the corresponding SessMgr at UP includes vrf-id. This enables VPNMgr to pick the correct chunk for that IP under the provided vrf-id for processing.

A custom IE, UE IP VRF, is introduced to encapsulate VRF name of UE IP in N4 SESSION ESTABLISHMENT REQUEST message.

SessMgr in UPF converts the received vrf- name into UP VRF CONTEXT ID and passes it on to UPFs VPNMgr in IP allocation request. This UP VRF CONTEXT ID is also used when release request is sent to UPFs VPNMgr. UPFs VPNMgr is made aware of the VRF to which that chunk belongs to by sending vrf-name in each chunk related communication between SMF and UPF. This enables UPFs VPNMgr to create database of chunks under each VRF enabling support of overlapping pools in same UPF.

### UE IP VRF Information Element

The following is the IE format of the private UE IP VRF.

Table 3: UE IP VRF Format

	Bits							
Octets	8	7	6	5	4	3	2	1
1 to 2	Type = 242 (decimal)							
3 to 4	Length = n							
5	Spare					Identical VRF flag	IPv6 VRF Valid	IPv4 VRF Valid flag
m to m+1	VRF-1 Name Length = p							
m+1 to m+1+p	VRF-1 Name							
n to n+1	VRF-2 Name Length = q							
m+1 to m+1+q	VRF-2 Name							

The following table shows the possible values of the "UE IP VRF" fields.

Cases	UE IP VRF	Value (binary)		
		Bit 3	Bit 2	Bit 1
1	None of the IPv4 and IPv6 UE IP address are associated to VRF.	0	0	0
2	Only IPv4 UE IP address is associated to an VRF	0	0	1
3	Only IPv6 UE IP address is associated to VRF	0	1	0
4	Both IPv4 and IPv6 UE IP address are associated to VRF and are different VRF.	0	1	1
5	Both IPv4 and IPv6 UE IP address are associated to VRF and is common VRF.	1	1	1

## VRF Name as Identifier

Prior to this feature, the communication between SMF and UPF, related to VRF, was done through vrf-id. This required the operator to have all the VRFs configured in both SMF and UPF, and also in the same order.

With this feature, vrf-name is used as identifier in all the communication between SMF and UPF with respect to VRFs, eliminating the requirement of configuring all the VRFs in UPF. Operator can configure VRFs in different order at SMF and UPF, and still can identify the VRF since vrf-name is same in both the nodes.

## Limitations and Restrictions

The following are the known limitations/restrictions of the feature in UPF:

- VRF limit per UPF is 205.

- Only VRF-based overlapping pools are supported in UPF. Other flavors of overlapping pools, like NH-based, VLAN-based, and so on, are not supported.
- PDN Type IPv4v6-based call is not permitted on static IP pools with multiple UP in same UP-group.
- Dynamic update of the VRF is not supported.

## Configuring VRF

Follow these steps to implement VRF support in UPF.

**At SMF:**

1. Create APN/DNN profile.
2. Create overlapping IP pools and associate the respective APN/DNN and VRF at context-level.
3. Associate APN/DNN to UPF profile.

The following is an example of the SMF configuration:

```
profile dnn intershat1
.
.
.
    upf apn mpls1.com
exit
profile dnn intershat2
.
.
.
    upf apn mpls2.com
exit
profile network-element upf upf1
.
.
.
    dnn-list [ intershat1 intershat2 ]
exit
profile network-element upf upf2
.
.
.
    dnn-list [ intershat1 intershat2 ]
exit
ipam
source local
address-pool pool-intershat1
    vrf-name mpls-vrf-1@isp
    tags
        dnn intershat1
    exit
    ipv4
        address-range 192.0.0.1 255.0.0.1
    exit
exit
address-pool pool-intershat2
    vrf-name mpls-vrf-2@isp
    tags
        dnn intershat2
```

```

        exit
    ipv4
        address-range 192.0.0.1 255.0.0.1
    exit
exit
exit

```

### At UPF:

It's recommended to configure VRF in UPF before chunk is pushed from SMF. Else, it leads to the failure of complete IP pool transaction (including chunks that don't belong to the VRF), and retry attempt by SMF after some time.

The following is an example of the UPF configurations:

### UPF 1:

```

config
context EPC2
    sx-service sx
        instance-type userplane
        bind ipv4-address 192.0.0.2 ipv6-address bbbb:aaaa::4
    exit
user-plane-service up
    associate gtpu-service pgw-gtpu pgw-ingress
    associate gtpu-service sgw-ingress-gtpu sgw-ingress
    associate gtpu-service sgw-engress-gtpu sgw-egress
    associate gtpu-service saegw-sxu cp-tunnel
    associate sx-service sx
    associate fast-path service
    associate control-plane-group g1
    exit

context isp
    ip vrf mpls-vrf-1
    #exit
    ip vrf mpls-vrf-2
    #exit
    apn mpls1.com
    pdp-type ipv4 ipv6
    bearer-control-mode mixed
    selection-mode sent-by-ms
    ip context-name isp
    exit
exit
control-plane-group g1
    peer-node-id ipv4-address 192.0.0.10
    #exit
user-plane-group default

```

### UPF 2:

```

config
context EPC2
    sx-service sx
        instance-type userplane
        bind ipv4-address 192.0.0.3 ipv6-address bbbb:aaaa::5
    exit
user-plane-service up
    associate gtpu-service pgw-gtpu pgw-ingress
    associate gtpu-service sgw-ingress-gtpu sgw-ingress
    associate gtpu-service sgw-engress-gtpu sgw-egress
    associate gtpu-service saegw-sxu cp-tunnel
    associate sx-service sx
    associate fast-path service

```

```

        associate control-plane-group g1
    exit
exit

context isp
    ip vrf mpls-vrf-1
    #exit
    ip vrf mpls-vrf-2
    #exit
    apn mpls2.com
        pdp-type ipv4 ipv6
        bearer-control-mode mixed
        selection-mode sent-by-ms
        ip context-name isp
    exit
exit

control-plane-group g1
    peer-node-id ipv4-address 192.0.0.10
    #exit
    user-plane-group default

```

## Monitoring and Troubleshooting

This section provides information regarding the CLI commands available for monitoring and troubleshooting the feature.

### Show Commands and/or Outputs

This section provides information regarding show commands and their outputs in support of this feature.

#### show ip chunks

The output of this CLI command displays all the chunks in that context.

With Overlapping IP Pools functionality, VRF option is introduced in the output of **show ip chunks vrf** *vrf\_name* CLI command that displays only the chunks under that VRF.

- chunk-id
- chunk-size
- vrf-name
- start-addr
- end-addr
- used-addr
- Peer Address

#### show ipv6 chunks

The output of this CLI command displays all the chunks in that context.

With Overlapping IP Pools functionality, VRF option is introduced in the output of **show ipv6 chunks vrf** *vrf\_name* CLI command that displays only the chunks under that VRF.

- chunk-id
- chunk-size
- vrf-name
- start-prefix
- end-prefix
- used-prefixes
- Peer Address

show ipv6 chunks