UDP Proxy for SMF

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	SMF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	Pre-2020.02.0

Feature Description

The SMF has UDP interfaces toward the UPF (N3) and SGW (s5/s8 for EPS interworking). With the help of the protocol layer pods (smf-protocol and gtp-ep), the messages are encoded and decoded and exchanged on these UDP interfaces. For achieving the functionalities mentioned on the 3GPP specifications:

- It is mandatory for the protocol layer pods to receive the original source and destination IP address and port number. But the original IP and UDP header is not preserved when the incoming packets arrive at the UDP service in the Kubernetes (K8s) cluster.
- Similarly, for the outgoing messages, the source IP set to the external IP address of the UDP service (published to the peer node) is mandatory. But the source IP is selected as per the egress interface, when different instances of protocol layer pods send outgoing messages from different nodes of the K8s cluster.

The protocol layer POD spawns on the node, which has the physical interface configured with the external IP address to achieve the conditions mentioned earlier. However, spawning the protocol layer pods has the following consequences:

- It is not possible to achieve the node level HA (High Availability) because the protocol pods are spawned on the same node of the K8s cluster. Any failure to that node may result in loss of service.
- The protocol pods (smf-protocol, gtp-ep, and radius-ep) must include their own UDP client and server functionalities. In addition, each protocol layer pod may require labeling of the K8s nodes with the affinity rules. This restricts the scaling requirements of the protocol layer pods.

The SMF addresses these issues with the introduction of a new K8s POD called "smf-udp-proxy." The primary objectives of this POD are:

- The "smf-udp-proxy" POD acts as a proxy for all kinds of UDP messages. It also owns the UDP client and server functionalities.
- The protocol pods perform the individual protocol (PFCP, GTP, Radius) encoding and decoding and provide the UDP payload to the "smf-udp-proxy" POD. The "smf-udp-proxy" POD sends the UDP payload out after it receives the payload from the protocol pods.
- The "smf-udp-proxy" POD opens the UDP sockets on a virtual IP (VIP) instead of a physical IP. This ensures that the "smf-udp-proxy" POD does not have any strict affinity to a specific K8s node (VM). Thus, enabling node level HA for the UDP proxy.

Note One instance of the "smf-udp-proxy" POD is spawned by default in all the worker nodes in the K8s cluster. There are no changes to the configurations in this release.

Relationships

The UDP Proxy for SMF feature has functional relationship with the Virtual IP Address feature.

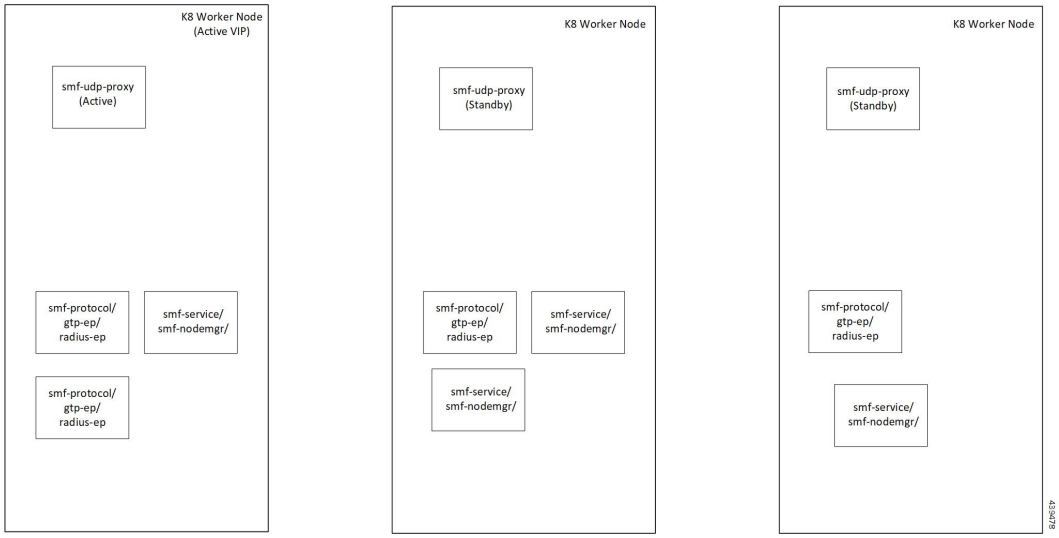
Architecture

The "smf-udp-proxy" POD is placed in the worker nodes in the K8s cluster.

1. Each of the K8s worker node contains one instance of the "smf-udp-proxy" POD. However, only one of the K8s worker node owns the virtual IP at any time. The worker node that owns the virtual IP remains in the active mode while all the other worker nodes remain in the standby mode.
2. The active "smf-udp-proxy" POD binds to the virtual IP and the designated ports for listening to the UDP messages from the peer nodes (UPF and SGW).

3. The UDP payload received from the peer nodes are forwarded to one instance of the smf-protocol, gtp-ep, or radius-ep pods. The payload is forwarded either on the same node or different node for further processing.
4. The response message from the smf-protocol, gtp-ep, or radius-ep pods is forwarded back to the active instance of the "smf-udp-proxy" POD. The "smf-udp-proxy" POD sends the response message back to the corresponding peer nodes.
5. The SMF-initiated messages are encoded at the smf-protocol, gtp-ep, or radius-ep pods. In addition, the UDP payload is sent to the "smf-udp-proxy" POD. Eventually, the "smf-udp-proxy" POD comprises of the complete IP payload and sends the message to the peer. When the response from the peer is received, the UDP payload is sent back to the same smf-protocol, gtp-ep, or radius-ep POD from which the message originated.

Figure 1: UDP Proxy Architecture



How it Works

The following sections describe the UDP proxy working principles.

Port and Sequence Number Selection

The duplicate messages are detected based on the source IP, source address, and sequence number for all UDP-based protocols. Each message from the peer includes a unique combination of these three parameters (source IP, source address, and sequence number).

In this release, only the sequence number changes for each new SMF initiated message. The SMF initiated messages use the same source IP (Virtual IP) and same port (fixed to 8809) number.

The sequence number (PFCP) is a 24-bit value. The 8 MSBs (Most Significant Bit) specify the smf-protocol pod's instance number. The 16 LSBs (Least Significant Bit) are incrementing counters, which generate a unique number for each instance of the smf-protocol POD.

Note

- The UDP proxy uses the smf-protocol POD instance sequence number to determine the smf-protocol instance to which the response message must be forwarded to.
- Message retransmission and duplicate detection are not supported in this release.

Protocol POD Selection for Peer Initiated Messages

When the "smf-udp-proxy" POD receives the peer node (for instance UPF) initiated messages, it is load balanced across the smf-protocol instances to select any instance of the smf-protocol POD. An entry of this instance number is stored along with the source IP and source port number of the peer node. This ensures that the messages form the same source IP and source port are sent to the same instance that was selected earlier.

Note This release does not support the landing of retransmitted messages (from the peer nodes) on the same instance of the smf-protocol where duplicate messages are detected.

High Availability for the UDP Proxy

The UDP proxy's HA model is based on the keepalived virtual IP concepts. A VIP is designated to the N4 interface during deployment. Also, a keepalived instance manages the VIP and ensures that the IP address of the VIP is created as the secondary address of an interface in one of the worker nodes of the K8s cluster.

The "smf-udp-proxy" instance on this worker node binds to the VIP and assumes the role of the active "smf-udp-proxy" POD. All "smf-udp-proxy" instances in other worker nodes remain in the standby mode.

When the worker node hosting the VIP fails, the keepalived instance moves the VIP to another worker node in the K8s cluster. The "smf-udp-proxy" instance in that worker node assumes the active role now.

Note In this release, the VIP support for the "smf-udp-proxy" is enabled with single instance of "smf-udp-proxy." The failover scenario is not supported in this release.

Call Flows

This section describes the call flow defined for the UDP proxy feature.

UDP Proxy for SMF Call Flow

The following call flow describes the flow of messages for the SMF initiated messages (applicable to messages initiated by peer nodes as well).

Figure 2: UDP Proxy for SMF Call Flow

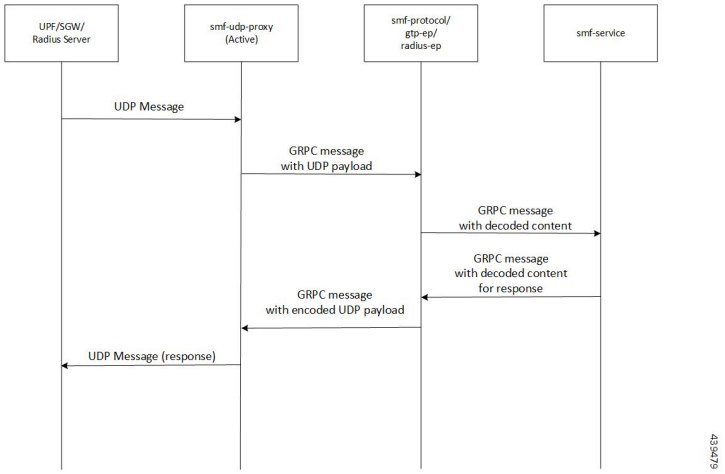


Table 3: UDP Proxy for SMF Call Flow

Step	Description
1	The peer nodes send the UDP messages to the VIP address. The active instance of "smf-udp-proxy" receives the UDP messages.
2	The UDP message's IP and the UDP header is stripped and the UDP payload is sent to the selected smf-protocol, gtp-ep, or radius-ep POD. The meta-data contains the source IP and source port number. With the help of the internal GRPC based IPC, the message is forwarded to the smf-protocol, gtp-ep, or radius-ep POD.
3	Based on the protocol, the smf-protocol, gtp-ep, or radius-ep POD decodes the message and loads the contents in proto-encoded buffer. The smf-protocol, gtp-ep, or radius-ep POD forwards the message to the smf-service POD for further processing over GRPC.
4	The smf-service POD generates the response message and sends it back to the smf-protocol, gtp-ep, or radius-ep POD in proto-encoded buffer.
5	The smf-protocol, gtp-ep, or radius-ep POD encodes the message and creates the UDP payload. The UDP payload is sent to the active "smf-udp-proxy" in a GRPC message.
6	The active "smf-udp-proxy" sends the message to the peer nodes on the UDP socket.