

DMM Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

U. Chunduri, Ed.
R. Li
Futurewei
S. Bhaskaran
Altiostar
J. Tantsura
Apstra, Inc.
L. Contreras
Telefonica
P. Muley
Nokia
J. Kaippallimalil
Futurewei
July 8, 2019

Transport Network aware Mobility for 5G
draft-clt-dmm-tn-aware-mobility-04

Abstract

This document specifies a framework and a mapping function for 5G mobile user plane with transport network slicing, integrated with Mobile Radio Access and a Virtualized Core Network. The integrated approach is specified in a way to fit into the 5G core network architecture defined in [TS23.501].

It focuses on an optimized mobile user plane functionality with various transport services needed for some of the 5G traffic needing low and deterministic latency, real-time, mission-critical services. This document describes, how this objective is achieved agnostic to the transport underlay used (IPv6, MPLS, IPv4) in various deployments and with a new transport network underlay routing, called Preferred Path Routing (PPR).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Problem Statement	3
1.2. Solution Approach	4
1.3. Acronyms	4
2. Transport Network and Slice aware Mobility on N3/N9	5
2.1. Integrated Approach with TNF in SBI	6
2.1.1. Transport Network Function and Interfaces	7
2.1.2. Functionality for E2E Management	7
2.2. TNF as part of existing 5G Control Function	9
2.2.1. Mobile Transport Network Context and Scalability	11
2.2.2. MTNC Identifier in the Data Packet	12
3. Using PPR as TN Underlay	12
3.1. PPR with Transport Awareness for 5GS on N3/N9 Interfaces	13
3.2. Path Steering Support to native IP user planes	15
3.3. Service Level Guarantee in Underlay	15
4. Other TE Technologies Applicability	15
5. Acknowledgements	16
6. IANA Considerations	16
7. Security Considerations	16
8. Contributing Authors	16
9. References	16

9.1. Normative References	16
9.2. Informative References	16
Appendix A. New Control Plane and User Planes	19
A.1. Slice aware Mobility: Discrete Approach	19
Appendix B. PPR with various 5G Mobility procedures	20
B.1. SSC Model1	20
B.2. SSC Mode2	21
B.3. SSC Mode3	22
Authors' Addresses	23

1. Introduction

3GPP Release 15 for 5GC is defined in [TS.23.501-3GPP], [TS.23.502-3GPP] and [TS.23.503-3GPP]. User Plane Functions (UPF) are the data forwarding entities in the 5GC architecture. The architecture allows the placement of Branching Point (BP) and Uplink Classifier (ULCL) UPFs closer to the access network (5G-AN). The 5G-AN can be a radio access network or any non-3GPP access network, for example, WLAN. The IP address is anchored by a PDU session anchor UPF (PSA UPF).

N3, N9 Interfaces: The interface between the BP/ULCL UPF and the PSA UPF is called N9 [TS.23.501-3GPP]. While in REL15, 3GPP has adopted GTP-U for the N9 interface, new user plane protocols along with GTP-U are being investigated for N9 interface in REL16, as part of [CT4SID]. Concerning to this document another relevant interface is N3, which is between the 5G-AN and the UPF. N3 interface is similar to the user plane interface S1U in LTE [TS.23.401-3GPP]. This document:

- o Do not need architectural change to [TS.23.501-3GPP] to provide 3GPP slice, QoS support in transport plane
- o and can work with any encapsulation (including GTP-U) for the N9 interface.

1.1. Problem Statement

[TS.23.501-3GPP] and [TS.23.502-3GPP] define network slicing as one of the core capability of 5GC with slice awareness from Radio and 5G Core (5GC) network. The 5G System (5GS) as defined, do not consider the resources and functionalities needed from transport network for the selection of UPF. This is seen as independent functionality and currently not part of 5GS.

However, the lack of underlying Transport Network (TN) awareness may lead to selection of sub-optimal UPF(s) and/or 5G-AN during 5GS procedures. This could also lead to inability to meet SLAs for real-

time, mission-critical or latency sensitive services. 5GS procedures including but not limited to Service Request, PDU Session Establishment, or User Equipment (UE) mobility need same service level characteristics from the TN for the Protocols Data Unit (PDU) session, similar to as provided in Radio and 5GC for the various Slice Service Types (SST) and 5QI's defined in [TS.23.501-3GPP].

1.2. Solution Approach

This document specifies 2 approaches to fulfil the needs of 5GS to transport user plane traffic from 5G-AN to UPF for all service continuity modes [TS.23.501-3GPP] in an optimized fashion. This is done by, keeping mobility procedures aware of underlying transport network along with slicing requirements.

Section 2 describes in detail on how TN aware mobility can be built irrespective of underlying TN technology used. Using Preferred Path Routing (PPR), applicable to any transport network underlay (IPv6, MPLS and IPv4) is detailed in Section 3. How other IETF TE technologies applicable for this draft is specified in Section 4. At the end, Appendix B further describes the applicability and procedures of PPR with 5G SSC modes on N3 and N9 interfaces.

1.3. Acronyms

5QI	-	5G QoS Indicator
5G-AN	-	5G Access Network
AMF	-	Access and Mobility Management Function (5G)
BP	-	Branch Point (5G)
CSR	-	Cell Site Router
DN	-	Data Network (5G)
eMBB	-	enhanced Mobile Broadband (5G)
FRR	-	Fast ReRoute
gNB	-	5G NodeB
GBR	-	Guaranteed Bit Rate (5G)
IGP	-	Interior Gateway Protocols (e.g. IS-IS, OSPFv2, OSPFv3)
LFA	-	Loop Free Alternatives (IP FRR)

mIOT	-	Massive IOT (5G)
MPLS	-	Multi Protocol Label Switching
QFI	-	QoS Flow ID (5G)
PPR	-	Preferred Path Routing
PDU	-	Protocol Data Unit (5G)
PW	-	Pseudo Wire
RQI	-	Reflective QoS Indicator (5G)
SBI	-	Service Based Interface (5G)
SID	-	Segment Identifier
SMF	-	Session Management Function (5G)
SSC	-	Session and Service Continuity (5G)
SST	-	Slice and Service Types (5G)
SR	-	Segment Routing
TE	-	Traffic Engineering
ULCL	-	Uplink Classifier (5G)
UPF	-	User Plane Function (5G)
URLLC	-	Ultra reliable and low latency communications (5G)

2. Transport Network and Slice aware Mobility on N3/N9

Currently specified Control Plane (CP) functions - the Access and Mobility Management Function (AMF), the Session Management Function (SMF) and the User plane (UP) components gNB, User Plane Function (UPF) with N2, N3, N4, N6 and N9 interfaces are relevant to this document. Other Virtualized 5G control plane components NRF, AUSF, PCF, AUSF, UDM, NEF, and AF are not directly relevant for the discussion in this document and one can see the functionalities of these in [TS.23.501-3GPP].

From encapsulation perspective, N3 interface is similar to S1U in 4G/LTE [TS.23.401-3GPP] network and uses GTP-U [TS.29.281-3GPP] to transport any UE PDUs (IPv4, IPv6, IPv4v6, Ethernet or Unstructured).

Unlike S1U, N3 has some additional aspects as there is no bearer concept and no per bearer GTP-U tunnels. Instead, QoS information is carried in the PDU Session Container GTP-U extension header.

TN Aware Mobility with optimized transport network functionality is explained below with approaches specified in Section 2.1 and Section 2.2. How PPR fits in this framework in detail along with other various TE technologies briefly are in Section 3 and Section 4 respectively.

2.1. Integrated Approach with TNF in SBI

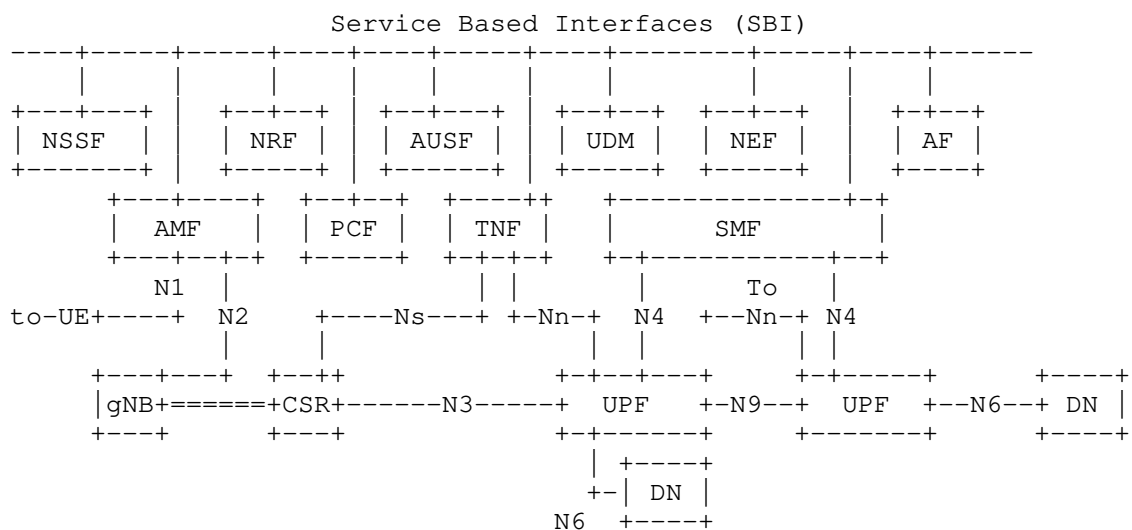


Figure 1: 5G Service Based Architecture

The above diagram depicts one of the scenarios of the 5G network specified in [TS.23.501-3GPP] and with a new and virtualized control component Transport Network Function (TNF). A Cell Site Router (CSR) is shown connecting to gNB. gNB is an entity in 5G-AN. Though it is shown as a separate block from gNB, in some cases both of these can be co-located. This document concerns with backhaul TN, from CSR to UPF on N3 interface or from Staging UPF to Anchor UPF on N9 interface.

Network Slice Selection Function (NSSF) as defined in [TS.23.501-3GPP] concerns with multiple aspects including selecting a

network slice instance when requested by AMF based on the requested SNSSAI, current location of UE, roaming indication etc. It also notifies NF service consumers (e.g AMF) whenever the status about the slice availability changes. However, the scope is only in 5GC (both control and user plane) and NG Radio Access network including the N3IWF for the non-3GPP access. The network slice instance(s) selected by the NSSF are applicable at a per PDU session granularity. An SMF and UPF are allocated from the selected slice instance during the PDU session establishment procedure.

2.1.1. Transport Network Function and Interfaces

To assuage the above situation, TNF is described (Figure 1) as part of control plane. This has the view of the underlying transport network with all links and nodes as well as various possible underlay paths with different characteristics. TNF can be seen as supporting PCE functionality [RFC5440] and optionally BGP-LS [RFC7752] to get the TE and topology information of the underlying IGP network.

A south bound interface Ns is shown which interacts with the 5G Access Network (e.g. gNB/CSR). 'Ns' can use one or more mechanism available today (PCEP [RFC5440], NETCONF [RFC6241], RESTCONF [RFC8040] or gNMI) to provision the L2/L3 VPNs along with TE underlay paths from gNB to UPF. Ns and Nn interfaces can be part of the integrated 3GPP architecture, but the specification/ownership of these interfaces SHOULD be left out of scope of 3GPP.

A north bound interface 'Nn' is shown from one or more of the transport network nodes (or ULCL/BP UPF, Anchor Point UPF) to TNF as shown in Figure 1. It would enable learning the TE characteristics of all links and nodes of the network continuously (through BGP-LS [RFC7752] or through a passive IGP adjacency and PCEP [RFC5440]).

These VPNs and/or underlay TE paths MUST be similar on all 5G-AN/CSRs and UPFs concerned to allow mobility of UEs while associated with one of the Slice/Service Types (SSTs) as defined in [TS.23.501-3GPP].

Proposed TNF as part of the 5GC shown in Figure 1 can be realized using Abstraction and Control of TE Networks (ACTN). ACTN architecture, underlying topology abstraction methods and manageability considerations of the same are detailed in [RFC8453].

2.1.2. Functionality for E2E Management

With the TNF in 5GS Service Based Interface, the following additional functionalities are required for end-2-end slice management including the transport network:

- o The Specific Network Slice Selection Assistance Information (SNSSAI) of PDU session's SHOULD be mapped to the assigned transport VPN and the TE path information for that slice.
- o For transport slice assignment for various SSTs (eMBB, URLLC, MIIOT) corresponding underlay paths need to be created and monitored from each transport end point (gNB/CSR and UPF).
- o During PDU session creation, apart from radio and 5GC resources, transport network resources needed to be verified matching the characteristics of the PDU session traffic type.
- o The TNF MUST provide an API that takes as input the source and destination 3GPP user plane element address, required bandwidth, latency and jitter characteristics between those user plane elements and returns as output a particular TE path's identifier, that satisfies the requested requirements.
- o Mapping of PDU session parameters to underlay SST paths need to be done. One way to do this to let the SMF install a Forwarding Action Rule (FAR) in the UPF via N4 with the FAR pointing to a "Network Instance" in the UPF. A "Network Instance" is a logical identifier for an underlying network. The "Network Instance" pointed by the FAR can be mapped to a transport path (through L2/L3 VPN). FARs are associated with Packet Detection Rule (PDR). PDRs are used to classify packets in the uplink (UL) and the downlink (DL) direction. For UL GTP-U TEID and/or the QFI marked in the GTPU packet can be used for classifying a packet belonging to a particular slice characteristics. For DL, at a PSA UPF, the UE IP address is used to identify the PDU session, and hence the slice a packet belongs to and the IP 5 tuple can be used for identifying the flow and QoS characteristics to be applied on the packet.
- o If any other form of encapsulation (other than GTP-U) either on N3 or N9 corresponding QFI information MUST be there in the encapsulation header.
- o In some SSC modes Appendix B, if segmented path (gNB to staging/ULCL/BP-UPF to anchor-point-UPF) is needed, then corresponding path characteristics MUST be used. This includes a path from gNB/CSR to UL-CL/BP UPF [TS.23.501-3GPP] and UL-CL/BP UPF to eventual UPF access to DN.
- o Continuous monitoring of transport path characteristics and reassignment at the endpoints MUST be performed. For all the affected PDU sessions, degraded transport paths need to be updated dynamically with similar alternate paths.

- o During UE mobility event similar to 4G/LTE i.e., gNB mobility (Xn based or N2 based), for target gNB selection, apart from radio resources, transport resources MUST be factored. This enables handling of all PDU sessions from the UE to target gNB and this require co-ordination of gNB, AMF, SMF with the TNF module.

Integrating the TNF as part of the 5GS Service Based Interfaces, provides the flexibility to control the allocation of required characteristics from the TN during a 5GS signaling procedure (e.g. PDU Session Establishment). If TNF is seen as part of management plane, this real time flexibility is lost. Changes to detailed signaling to integrate the above for various 5GS procedures as defined in [TS.23.502-3GPP] is beyond the scope of this document.

2.2. TNF as part of existing 5G Control Function

Another solution approach with TNF in Section 2.1 and transport provisioning for an engineered IP transport that supports 3GPP slicing and QoS requirements in [TS.23.501-3GPP] is described in this section.

During a PDU session setup, the 3GPP AMF using input from the NSSF selects a network slice and SMF. The SMF with user policy from Policy Control Function (PCF) sets 5QI (QoS parameters) and the UPF on the path of the PDU session. While QoS and slice selection for the PDU session can be applied across the 3GPP control and user plane functions as outlined above, the transport underlay across N3 and N9 segments do not have enough information to apply the resource constraints represented by the slicing and QoS classification. Current guidelines for interconnection with transport networks [IR.34-GSMA] provide an application mapping into DSCP. However, apart from problems with classification of encrypted packets, these recommendations do not take into consideration other aspects in slicing like isolation, protection and replication.

Transport networks have their own slice and QoS configuration based on domain policies and the underlying network capability. Transport networks can enter into an agreement for virtual network services (VNS) with client domains using the ACTN [RFC8453] framework. The 3GPP mobile network, on the other side, defines a Network Slice Selection Management Function (NSSMF) [TS 28.533] that interacts with a TN domain manager (that is out of scope of 3GPP).

The ACTN VN service can be used across the 3GPP and transport networks to provision and map between slices, QoS in the two domains. An abstraction that represents QoS and slice information in the mobile domain and mapped to ACTN VN service in the transport domain is represented here as MTNC (Mobile Transport Network Context)

identifiers. Details of how the 3GPP domain derives the MTNC identifiers and how it programs it across its control and user plane functions are for 3GPP standards to define. For completeness, some minimal outlines are provided in the description below.

When the 3GPP user plane function (gNB, UPF) does not terminate the transport underlay protocol (e.g., MPLS), it needs to be carried in the IP protocol header from end-to-end of the mobile transport connection (N3, N9). [I-D.ietf-dmm-5g-uplane-analysis] discusses these scenarios in detail.

Figure 2 shows a view of the functions and interfaces for provisioning the MTNC identifiers. The focus is on provisioning between the 3GPP management plane (NSSMF), transport network (SDN-C) and carrying the MTNC identifiers in PDU packets for the transport network to grant the provisioned resources.

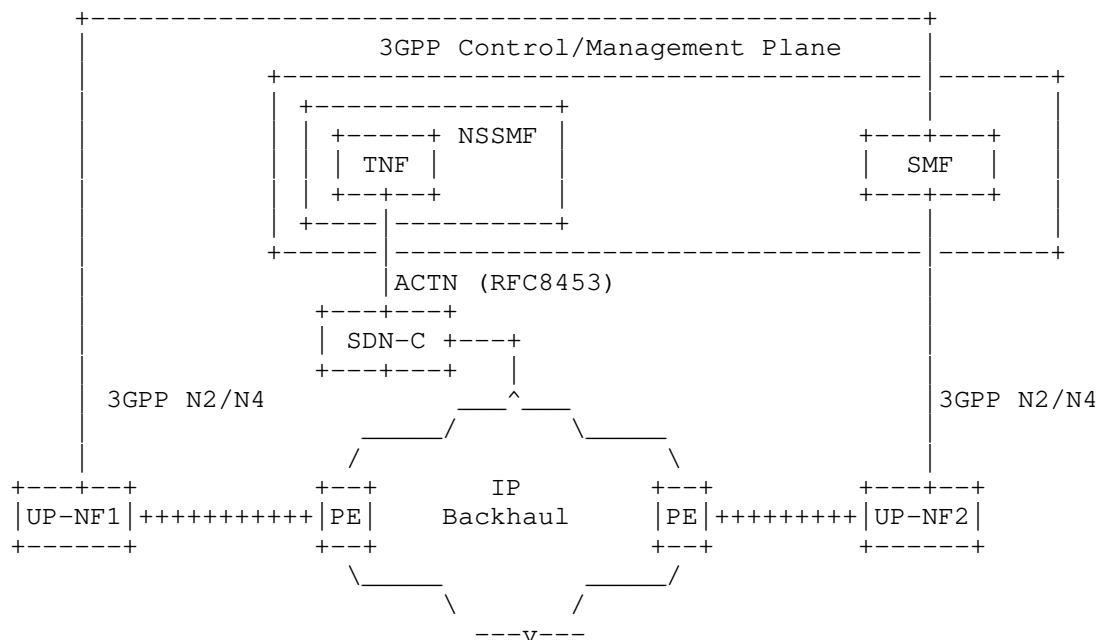


Figure 2: 5G Transport Plane Provisioning

In Figure 2, the TNF (logical functionality within the NSSMF) requests the SDN-C in the transport domain to program the TE path using ACTN [RFC 8453]. The SDN-C programs the Provider Edge (PE) routers and internal routers according to the underlay transport technology (e.g., PPR, MPLS, SRv6). The PE router inspects incoming

PDU data packets for the MTNC identifier, classifies and provides the VN service provisioned across the transport network.

The detailed mechanisms by which the NSSMF provides the MTNC identifiers to the control plane and user plane functions are for 3GPP to specify. Two possible options are outlined below for completeness. The NSSMF may provide the MTNC identifiers to the 3GPP control plane by either providing it to the Session Management Function (SMF), and the SMF in turn provisions the user plane functions (UP-NF1, UP-NF2) during PDU session setup. Alternatively, the user plane functions may request the MTNC identifiers directly from the NSSMF.

In this approach, TNF can be seen as a logical entity that can be part of NSSMF in the 3GPP management plane [TS.28.533-3GPP]. The NSSMF may use network configuration, policies, history, heuristics or some combination of these to derive traffic estimates that the TNF would use. How these estimates are derived are not in the scope of this document. The focus here is only in terms of how the TNF and SDN-C are programmed given that slice and QoS characteristics across a transport path can be represented by an MTNC identifier. The TNF requests the SDN-C in the transport network to provision paths in the transport domain based on the MTNC identifier. The TNF is capable of providing the MTNC identifier provisioned to control and user plane functions in the 3GPP domain. Detailed mechanisms for programming the MTNC identifier should be part of the 3GPP specifications.

2.2.1. Mobile Transport Network Context and Scalability

The MTNC (Mobile Transport Network Context) represents a slice, QoS configuration for a transport path between two 3GPP user plane functions. The Mobile-Transport Network Context Identifier (MTNC-ID) is generated by the TNF to be unique for each path and per traffic class (including QoS and slice aspects). Thus, there may be more than one MTNC-ID for the same QoS and path if there is a need to provide isolation (slice) of the traffic. It should be noted that MTNC are per class/path and not per user session (nor is it per data path entity). The MTNC identifiers are configured by the TNF to be unique within a provisioning domain.

Since the MTNC identifiers are not generated per user flow or session, there is no need for unique MTNC identifiers per flow/session. In addition, since the traffic estimation not performed at the time of session establishment, there is no provisioning delay experienced during session setup. The MTNC identifier space scales as a square of the number sites between which 3GPP user plane functions require paths. If there are T traffic classes across N sites, the number of MTNC identifiers in a fully meshed network is

$(N*(N-1)/2) * T$. For example, if there are 3 traffic classes between 25 sites, there would be at most 900 MTNC identifiers required. Multiple slices for the same QoS class that need to be fully isolated, will add to the MTNC provisioning. An MTNC identifier space of 16 bits (65K+ identifiers) can be expected to be sufficient.

2.2.2. MTNC Identifier in the Data Packet

When the 3GPP user plane function (gNB, UPF) and transport provider edge are on different nodes, the PE router needs to have the means by which to classify the PDU packet. IP header fields such as DSCP (DiffServ Code Point) or the IPv6 Flow Label do not satisfy the requirement as they are not immutable.

Different options for carrying the MTNC identifier in the IP data packet or in the existing user plane overlay like GTP-U [TS.29.281-3GPP] or a new overlay like GUE [I-D.ietf-intarea-gue-extensions] are possible. There are various trade-offs in terms of packet overhead, support in IPv4 and IPv6 networks as well as working across legacy and evolving transport networks that need to be considered. These considerations will be addressed in future revisions.

3. Using PPR as TN Underlay

In a network implementing source routing, packets may be transported through the use of Segment Identifiers (SIDs), where a SID uniquely identifies a segment as defined in [I-D.ietf-spring-segment-routing]. Section 5.3 [I-D.bogineni-dmm-optimized-mobile-user-plane] lays out all SRv6 features along with a few concerns in Section 5.3.7 of the same document. Those concerns are addressed by a new backhaul routing mechanism called Preferred Path Routing (PPR), of which this section provides an overview.

The label/PPR-ID refer not to individual segments of which the path is composed, but to the identifier of a path that is deployed on network nodes. The fact that paths and path identifiers can be computed and controlled by a controller, not a routing protocol, allows the deployment of any path that network operators prefer, not just shortest paths. As packets refer to a path towards a given destination and nodes make their forwarding decision based on the identifier of a path, not the identifier of a next segment node, it is no longer necessary to carry a sequence of labels. This results in multiple benefits including significant reduction in network layer overhead, increased performance and hardware compatibility for carrying both path and services along the path.

Details of the IGP extensions for PPR are provided here:

- o IS-IS - [I-D.chunduri-lsr-isis-preferred-path-routing]
- o OSPF - [I-D.chunduri-lsr-ospf-preferred-path-routing]

3.1. PPR with Transport Awareness for 5GS on N3/N9 Interfaces

PPR does not remove GTP-U, unlike some other proposals laid out in [I-D.bogineni-dmm-optimized-mobile-user-plane]. Instead, PPR works with the existing cellular user plane (GTP-U) for both N3 and any approach selected for N9 (encapsulation or no-encapsulation). In this scenario, PPR will only help providing TE benefits needed for 5G slices from transport domain perspective. It does so without adding any additional overhead to the user plane, unlike SR-MPLS or SRv6. This is achieved by:

- o For 3 different SSTs, 3 PPR-IDs can be signaled from any node in the transport network. For Uplink traffic, the 5G-AN will choose the right PPR-ID of the UPF based on the S-NSSAI the PDU Session belongs to and/or the QFI (e.g. 5QI) marking on the GTP-U encapsulation header. Similarly in the Downlink direction matching PPR-ID of the 5G-AN is chosen based on the S-NSSAI the PDU Session belongs to. The table below shows a typical mapping:

QFI (Ranges)	SST in S-NSSAI	Transport Path Info	Transport Path Characteristics
Range Xx - Xy X1, X2 (discrete values)	MIOT (massive IOT)	PW ID/VPN info, PPR-ID-A	GBR (Guaranteed Bit Rate) Bandwidth: Bx Delay: Dx Jitter: Jx
Range Yx - Yy Y1, Y2 (discrete values)	URLLC (ultra-low latency)	PW ID/VPN info, PPR-ID-B	GBR with Delay Req. Bandwidth: By Delay: Dy Jitter: Jy
Range Zx - Zy Z1, Z2 (discrete values)	EMBB (broadband)	PW ID/VPN info, PPR-ID-C	Non-GBR Bandwidth: Bx

Figure 3: QFI Mapping with PPR-IDs on N3/N9

- o It is possible to have a single PPR-ID for multiple input points through a PPR tree structure separate in UL and DL direction.
- o Same set of PPRs are created uniformly across all needed 5G-ANs and UPFs to allow various mobility scenarios.
- o Any modification of TE parameters of the path, replacement path and deleted path needed to be updated from TNF to the relevant ingress points. Same information can be pushed to the NSSF, and/or SMF as needed.
- o PPR can be supported with any native IPv4 and IPv6 data/user planes (Section 3.2) with optional TE features (Section 3.3) . As this is an underlay mechanism it can work with any overlay encapsulation approach including GTP-U as defined currently for N3 interface.

3.2. Path Steering Support to native IP user planes

PPR works in fully compatible way with SR defined user planes (SR-MPLS and SRv6) by reducing the path overhead and other challenges as listed in [I-D.chunduri-lsr-isis-preferred-path-routing] or Section 5.3.7 of [I-D.bogineni-dmm-optimized-mobile-user-plane]. PPR also expands the source routing to user planes beyond SR-MPLS and SRv6 i.e., native IPv6 and IPv4 user planes.

This helps legacy transport networks to get the immediate path steering benefits and helps in overall migration strategy of the network to the desired user plane. It is important to note, these benefits can be realized with no hardware upgrade except control plane software for native IPv6 and IPv4 user planes.

3.3. Service Level Guarantee in Underlay

PPR also optionally allows to allocate resources that are to be reserved along the preferred path. These resources are required in some cases (for some 5G SSTs with stringent GBR and latency requirements) not only for providing committed bandwidth or deterministic latency, but also for assuring overall service level guarantee in the network. This approach does not require per-hop provisioning and reduces the OPEX by minimizing the number of protocols needed and allows dynamism with Fast-ReRoute (FRR) capabilities.

4. Other TE Technologies Applicability

RSVP-TE [RFC3209] provides a lean transport overhead for the TE path for MPLS user plane. However, it is perceived as less dynamic in some cases and has some provisioning overhead across all the nodes in N3 and N9 interface nodes. Also it has another drawback with excessive state refresh overhead across adjacent nodes and this can be mitigated with [RFC8370].

SR-TE [I-D.ietf-spring-segment-routing] does not explicitly signal bandwidth reservation or mechanism to guarantee latency on the nodes/links on SR path. But, SR allows path steering for any flow at the ingress and particular path for a flow can be chosen. Some of the issues around path overhead/tax, MTU issues are documented at Section 5.3 of [I-D.bogineni-dmm-optimized-mobile-user-plane]. SR-MPLS allows reduction of the control protocols to one IGP (with out needing for LDP and RSVP-TE).

However, as specified above with PPR (Section 3), in the integrated transport network function (TNF) a particular RSVP-TE path for MPLS

or SR path for MPLS and IPv6 with SRH user plane, can be supplied to SMF for mapping a particular PDU session to the transport path.

5. Acknowledgements

Thanks to Young Lee for discussions on this document including ACTN applicability for the proposed TNF. Thanks to Sri Gundavelli and 3GPP delegates who provided detailed feedback on this document.

6. IANA Considerations

This document has no requests for any IANA code point allocations.

7. Security Considerations

This document does not introduce any new security issues.

8. Contributing Authors

The following people contributed substantially to the content of this document and should be considered co-authors.

Xavier De Foy
InterDigital Communications, LLC
1000 Sherbrooke West
Montreal
Canada

Email: Xavier.Defoy@InterDigital.com

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

9.2. Informative References

[I-D.bashandy-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-05 (work in progress), October 2018.

- [I-D.bogineni-dmm-optimized-mobile-user-plane]
Bogineni, K., Akhavain, A., Herbert, T., Farinacci, D., Rodriguez-Natal, A., Carofiglio, G., Auge, J., Muscariello, L., Camarillo, P., and S. Homma, "Optimized Mobile User Plane Solutions for 5G", draft-bogineni-dmm-optimized-mobile-user-plane-01 (work in progress), June 2018.
- [I-D.chunduri-lsr-isis-preferred-path-routing]
Chunduri, U., Li, R., White, R., Tantsura, J., Contreras, L., and Y. Qu, "Preferred Path Routing (PPR) in IS-IS", draft-chunduri-lsr-isis-preferred-path-routing-03 (work in progress), May 2019.
- [I-D.chunduri-lsr-ospf-preferred-path-routing]
Chunduri, U., Qu, Y., White, R., Tantsura, J., and L. Contreras, "Preferred Path Routing (PPR) in OSPF", draft-chunduri-lsr-ospf-preferred-path-routing-03 (work in progress), May 2019.
- [I-D.farinacci-lisp-mobile-network]
Farinacci, D., Pillay-Esnault, P., and U. Chunduri, "LISP for the Mobile Network", draft-farinacci-lisp-mobile-network-05 (work in progress), March 2019.
- [I-D.ietf-dmm-5g-uplane-analysis]
Homma, S., Miyasaka, T., Matsushima, S., and d. daniel.voyer@bell.ca, "User Plane Protocol and Architectural Analysis on 3GPP 5G System", draft-ietf-dmm-5g-uplane-analysis-02 (work in progress), July 2019.
- [I-D.ietf-dmm-srv6-mobile-uplane]
Matsushima, S., Filsfils, C., Kohno, M., Camarillo, P., daniel.voyer@bell.ca, d., and C. Perkins, "Segment Routing IPv6 for Mobile User Plane", draft-ietf-dmm-srv6-mobile-uplane-05 (work in progress), July 2019.
- [I-D.ietf-intarea-gue-extensions]
Herbert, T., Yong, L., and F. Templin, "Extensions for Generic UDP Encapsulation", draft-ietf-intarea-gue-extensions-06 (work in progress), March 2019.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[IR.34-GSMA]

GSM Association (GSMA), "Guidelines for IPX Provider Networks (Previously Inter-Service Provider IP Backbone Guidelines, Version 14.0", August 2018.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<https://www.rfc-editor.org/info/rfc6830>>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

[RFC8370] Beeram, V., Ed., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP-TE Deployments", RFC 8370, DOI 10.17487/RFC8370, May 2018, <<https://www.rfc-editor.org/info/rfc8370>>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

[TS.23.401-3GPP]
3rd Generation Partnership Project (3GPP), "Procedures for 4G/LTE System; 3GPP TS 23.401, v15.4.0", June 2018.

[TS.23.501-3GPP]
3rd Generation Partnership Project (3GPP), "System Architecture for 5G System; Stage 2, 3GPP TS 23.501 v2.0.1", December 2017.

[TS.23.502-3GPP]
3rd Generation Partnership Project (3GPP), "Procedures for 5G System; Stage 2, 3GPP TS 23.502, v2.0.0", December 2017.

[TS.23.503-3GPP]
3rd Generation Partnership Project (3GPP), "Policy and Charging Control System for 5G Framework; Stage 2, 3GPP TS 23.503 v1.0.0", December 2017.

[TS.28.533-3GPP]
3rd Generation Partnership Project (3GPP), "Management and Orchestration Architecture Framework (Release 15)", June 2018.

[TS.29.281-3GPP]
3rd Generation Partnership Project (3GPP), "GPRS Tunneling Protocol User Plane (GTPv1-U), 3GPP TS 29.281 v15.1.0", December 2018.

Appendix A. New Control Plane and User Planes

A.1. Slice aware Mobility: Discrete Approach

In this approach transport network functionality from the 5G-AN to UPF is discrete and 5GS is not aware of the underlying transport network and the resources available. Deployment specific mapping function is used to map the GTP-U encapsulated traffic at the 5G-AN (e.g. gNB) in UL and UPF in DL direction to the appropriate transport slice or transport Traffic Engineered (TE) paths. These TE paths can be established using RSVP-TE [RFC3209] for MPLS underlay, SR [I-D.ietf-spring-segment-routing] for both MPLS and IPv6 underlay or PPR [I-D.chunduri-lsr-isis-preferred-path-routing] with MPLS, IPv6 with SRH, native IPv6 and native IPv4 underlays.

As per [TS.23.501-3GPP] and [TS.23.502-3GPP] the SMF controls the user plane traffic forwarding rules in the UPF. The UPFs have a concept of a "Network Instance" which logically abstracts the underlying transport path. When the SMF creates the packet detection rules (PDR) and forwarding action rules (FAR) for a PDU session at the UPF, the SMF identifies the network instance through which the packet matching the PDR has to be forwarded. A network instance can be mapped to a TE path at the UPF. In this approach, TNF as shown in Figure 1 need not be part of the 5G Service Based Interface (SBI). Only management plane functionality is needed to create, monitor, manage and delete (life cycle management) the transport TE paths/transport slices from the 5G-AN to the UPF (on N3/N9 interfaces). The management plane functionality also provides the mapping of such TE paths to a network instance identifier to the SMF. The SMF uses this mapping to install appropriate FARs in the UPF. This approach provide partial integration of the transport network into 5GS with some benefits.

One of the limitations of this approach is the inability of the 5GS procedures to know, if underlying transport resources are available for the traffic type being carried in PDU session before making certain decisions in the 5G CP. One example scenario/decision could be, a target gNB selection during a N2 mobility event, without knowing if the target gNB is having a underlay transport slice resource for the S-NSSAI and 5QI of the PDU session. The Integrated approach specified below can mitigate this.

Appendix B. PPR with various 5G Mobility procedures

PPR fulfills the needs of 5GS to transport the user plane traffic from 5G-AN to UPF in all 3 SSC modes defined [TS.23.501-3GPP]. This is done in keeping the backhaul network at par with 5G slicing requirements that are applicable to Radio and virtualized core network to create a truly end-to-end slice path for 5G traffic. When UE moves across the 5G-AN (e.g. from one gNB to another gNB), there is no transport network reconfiguration required with the approach above.

SSC mode would be specified/defaulted by SMF. No change in the mode once connection is initiated and this property is not altered here.

B.1. SSC Model

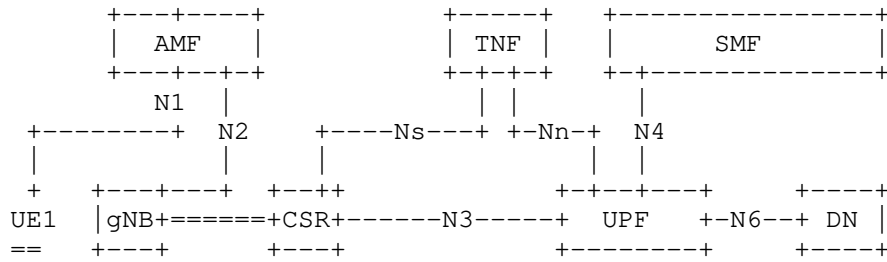


Figure 4: SSC Model with integrated Transport Slice Function

After UE1 moved to another gNB in the same UPF serving area

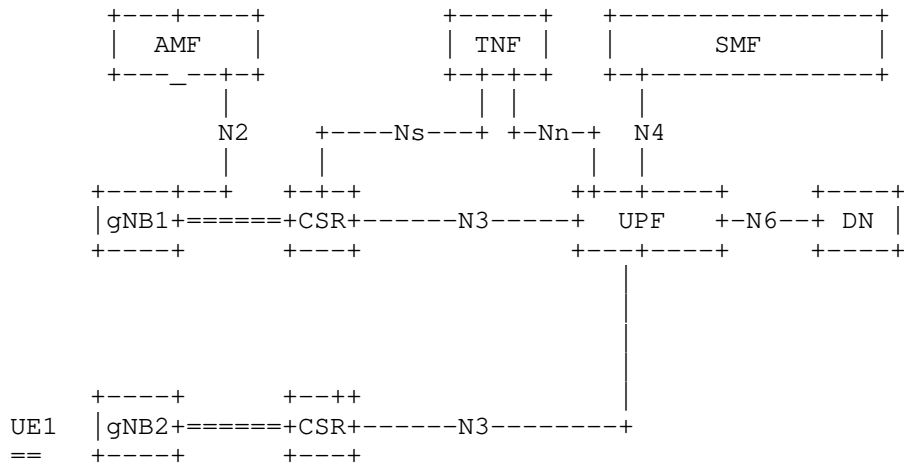


Figure 5: SSC Model with integrated Transport Slice Function

In this mode, IP address at the UE is preserved during mobility events. This is similar to 4G/LTE mechanism and for respective slices, corresponding PPR-ID (TE Path) has to be assigned to the packet at UL and DL direction. During Xn mobility as shown above, source gNB has to additionally ensure transport path's resources from TNF are available at the target gNB apart from radio resources check (at decision and request phase of Xn/N2 mobility scenario).

B.2. SSC Mode2

In this case, if IP Address is changed during mobility (different UPF area), then corresponding PDU session is released. No session continuity from the network is provided and this is designed as an

application offload and application manages the session continuity, if needed. For PDU Session, Service Request and Mobility cases mechanism to select the transport resource and the PPR-ID (TE Path) is similar to SSC Model.

B.3. SSC Mode3

In this mode, new IP address may be assigned because of UE moved to another UPF coverage area. Network ensures UE suffers no loss of 'connectivity'. A connection through new PDU session anchor point is established before the connection is terminated for better service continuity. There are two ways in which this happens.

- o Change of SSC Mode 3 PDU Session Anchor with multiple PDU Sessions.
- o Change of SSC Mode 3 PDU Session Anchor with IPv6 multi-homed PDU Session.

In the first mode, from user plane perspective, the two PDU sessions are independent and the use of PPR-ID by gNB and UPFs is exactly similar to SSC Mode 1 described above. The following paragraphs describe the IPv6 multi-homed PDU session case for SSC Mode 3.

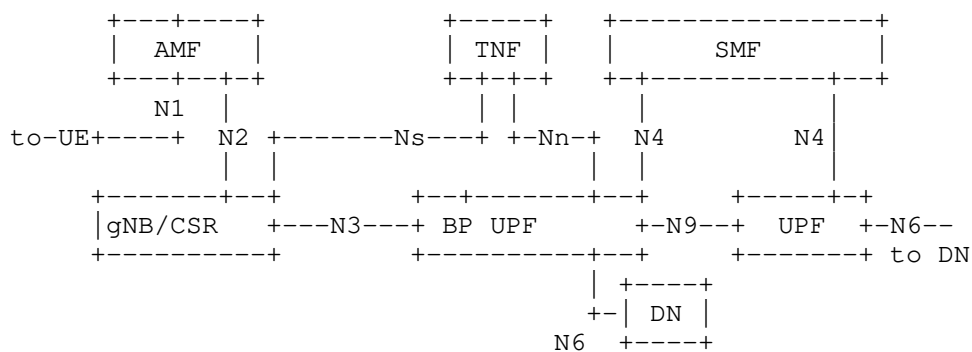


Figure 6: SSC Mode3 and Service Continuity

In the uplink direction for the traffic offloading from the Branching Point UPF, packet has to reach to the right exit UPF. In this case packet gets re-encapsulated by the BP UPF (with either GTP-U or the chosen encapsulation) after bit rate enforcement and LI, towards the anchor UPF. At this point packet has to be on the appropriate VPN/PW

to the anchor UPF. This mapping is done based on the S-NSSAI the PDU session belongs to and/or the QFI marking in the GTPU encapsulation header (e.g. 5QI value) to the PPR-ID of the exit node by selecting the respective TE PPR-ID (PPR path) of the UPF. If it's a non-MPLS underlay, destination IP address of the encapsulation header would be the mapped PPR-ID (TE path).

In the downlink direction for the incoming packet, UPF has to encapsulate the packet (with either GTP-U or the chosen encapsulation) to reach the BP UPF. Here mapping is done based on the S-NSSAI the PDU session belongs, to the PPR-ID (TE Path) of the BP UPF. If it's a non-MPLS underlay, destination IP address of the encapsulation header would be the mapped PPR-ID (TE path). In summary:

- o Respective PPR-ID on N3 and N9 has to be selected with correct transport characteristics from TNF.
- o For N2 based mobility SMF has to ensure transport resources are available for N3 Interface to new BP UPF and from there the original anchor point UPF.
- o For Service continuity with multi-homed PDU session same transport network characteristics of the original PDU session (both on N3 and N9) need to be observed for the newly configured IPv6 prefixes.

Authors' Addresses

Uma Chunduri (editor)
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: umac.ietf@gmail.com

Richard Li
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: richard.li@futurewei.com

Sridhar Bhaskaran
Altiostar

Email: sridhar.bhaskaran@gmail.com

Jeff Tantsura
Apstra, Inc.

Email: jefftant.ietf@gmail.com

Luis M. Contreras
Telefonica
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Praveen Muley
Nokia
440 North Bernardo Ave
Mountain View, CA 94043
USA

Email: praveen.muley@nokia.com

John Kaippallimalil
Futurewei
5700 Tennyson Parkway, Suite 600
Plano, TX 75024
USA

Email: john.kaippallimalil@futurewei.com

DMM Working Group
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

S. Homma
NTT
T. Miyasaka
KDDI Research
S. Matsushima
SoftBank
D. Voyer
Bell Canada
November 2, 2020

User Plane Protocol and Architectural Analysis on 3GPP 5G System
draft-ietf-dmm-5g-uplane-analysis-04

Abstract

This document analyzes the mobile user plane protocol and the architecture specified in 3GPP 5G documents. The analysis work is to clarify those specifications, extract protocol and architectural requirements and derive evaluation aspects for user plane protocols on IETF side. This work is corresponding to the User Plane Protocol Study work on 3GPP side.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 6, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Current Status of Mobile User Plane for 5G	3
1.2. Our Way of Analysis Work	4
2. Terms and Abbreviations	4
3. GTP-U Specification and Observation	6
3.1. GTP-U Tunnel	6
3.2. GTP-U Header Format	9
3.3. Control Plane Protocol for GTP-U	12
3.4. GTP-U message	13
3.5. Packet Format	14
3.6. Observations Summary	16
4. 5GS Architectural Requirements for User Plane Protocols	16
4.1. Overview of 5G System Architecture	16
4.1.1. UPF Functionalities	18
4.1.2. UP Traffic Detection	19
4.1.3. User Plane Configuration	21
4.2. Architectural Requirements for User Plane Protocols	22
4.2.1. Fundamental Functionalities	23
4.2.2. Supporting 5G Services	26
5. Evaluation Aspects	32
5.1. Supporting PDU Session Type Variations	32
5.2. Nature of Data Path	33
5.3. Supporting Transport Variations	33
5.4. Data Path Management	34
5.5. QoS Control	35
5.6. Traffic Detection and Flow Handling	35
5.7. Supporting Network Slicing Diversity	35
5.8. Reliable Communication support	36
6. Conclusion	37
7. Security Consideration	37
8. Acknowledgement	37
9. Informative References	38
Authors' Addresses	42

1. Introduction

This document analyzes the mobile user plane protocol and the architecture specified by 3GPP 5G documents. The background of the work is that 3GPP requests through a liaison statement that the IETF

to provide any information for the User Plane Protocol Study work in 3GPP [CP-180116-3GPP]. Justification and the objectives of the study can be found from [CP-173160-3GPP].

We understand that the current user plane protocol, GTP-U [TS.29.281-3GPP], has been well developed in 3GPP, and deployed very widely as the successor of legacy network technologies, such as TDM circuit, or ATM virtual circuit. That GTP-U success seems based on IP overlay technique that is dramatically scaled compare to the previous ones because it successfully isolates mobile session states from the user plane transport network.

Even after that big success, it is definitely worth that 3GPP has decided to revisit user plane which seems to response to IPv6 deployment growth and [IAB-Statement] that encourages the industry to develop strategies for IPv6-only operation. It can be seen from the justification section in [CP-173160-3GPP].

The study description mentions that the study would be based on Release 16 requirement while only Release 15 specifications has been available now. However we believe that to provide adequate information for 3GPP, we need to clearly understand what the current user plane protocol is in Release 15, and architectural requirements for the user plane.

As the liaison statement indicates 3GPP specifications related to user plane, those documents should be a good start point to clarify their specifications and to extract protocol and architectural requirements from them.

1.1. Current Status of Mobile User Plane for 5G

3GPP RAN and CT4 decided to use GTP-U as the 5G user plane encapsulation protocol over N3 and N9 that respectively described in [TS.38.300-3GPP] and [TR.29.891-3GPP]. N3 is an interface between RAN and UPF and N9 is an interface between different UPFs [TS.23.501-3GPP].

In [TR.29.891-3GPP], it captured user plane requirements and concluded that GTP-U is adopted for the user plane protocol. It seems that GTP-U was only option to be chose and it focused on how to carry 5G specific QoS information between UPF and access networks. That is described in section 5.2 and 11.2 of [TR.29.891-3GPP]. Another aspects of user plane requirements couldn't be found.

1.2. Our Way of Analysis Work

First, we analyze [TS.29.281-3GPP] for clarifying it as the current user plane protocol in the 5G system. [TR.29.891-3GPP] describes how GTP-U is selected as the user plane protocol for 5G in 3GPP. Clarified characteristics of the protocol are described in Section 3.

Then, to clarify what are required to the user plane protocol in architecture level, we analyze [TS.23.501-3GPP] as the 5G system architecture specification. [TS.23.502-3GPP] is the specification of system procedures that helps us to understand how the system works in the architecture. [TS.23.503-3GPP] is also helpful to find the role of user plane in the architecture that influences user plane protocol. Extracted architectural requirements are described in Section 4.

Based on the results of above, we identify some aspects where there might be gap between the current user plane protocol and the architectural requirements on which [TR.29.891-3GPP] does not discuss. That aspects are discussed Section 5. That's what we intend to be as a part of the reply to 3GPP. CT4 WG in 3GPP can utilize it as an input to evaluate the candidate protocols for user plane to the 5G system including the current protocol.

2. Terms and Abbreviations

This section describes terms of functions and interfaces relevant to user plane protocol which we extract from the 3GPP specifications since this document focuses on user plane.

In those specifications, there are so many unique terms and abbreviations in the 3GPP context which IETF community seems not familiar with. We will try to bring those terms with brief explanations to make sure common understanding for them.

GTP: GPRS Tunneling Protocol

GTP-U: User Plane part of GTP

Noted that GTP version 1 (GTPv1-U) is the user-plane protocol specification which is defined in [TS.29.281-3GPP]. Unless there is no specific annotation, we refer GTP-U to GTPv1-U in this document.

PDU: Protocol Data Unit of end-to-end user protocol packet.

Noted that the PDU in 3GPP includes IP header in case that PDU session type is IPv4 or IPv6. In contrast, in IETF it is supposed

that PDU is the payload of IP packet so that it doesn't include IP/TCP/UDP header in end-to-end.

T-PDU: Transport PDU.

G-PDU: GTP encapsulated user Plane Data Unit.

GTP-U has above two notions on PDU. T-PDU is a PDU that GTP-U header encapsulates. G-PDU is a PDU that includes GTP-U header. A G-PDU may include a T-PDU. G-PDU can be sent without T-PDU, but just with extension headers or TLV elements. It can be used for OAM related operations.

PDU session: Association between the UE and a Data Network that provides a PDU connectivity service.

Data Network (DN): The network of operator services, Internet access or 3rd party services.

User Plane (UP): Encapsulating user end-to-end PDU.

In fact, we can't find exact text that defines UP in the architecture specification. However when we see the figure 8.3.1-1 in [TS.23.501-3GPP], we specify UP as the layer right under PDU that directly encapsulates PDU. Underneath layers of UP are UP transport, such as IP/UDP, L2 and L1.

However 3GPP is consistent to use the term user plane when they indicate that layer. In IETF, we can see the terms data plane, or forwarding plane as variations which often makes us tend to be confused in terminology.

QFI: QoS Flow Identifier

UPF: User Plane Function

SMF: Session Management Function

SMF is a control plane function which provides session management service that handling PDU sessions in the control plane. SMF allocates tunnels corresponding to the PDU sessions and configure the tunnel to the UPF.

PFCP: Packet Forwarding Control Protocol

PFCP is used on N4 interface between SMF and UPF to configure the rules of packet detection, forwarding action, QoS enforcement, usage report and buffering for each PDU session.

PDR: Packet Detection Rule

FAR: Forwarding Action Rule

RAN: Radio Access Network

Noted that UP protocol provides a RAN to connect UPF. But the UP protocol is not appeared on the air in the RAN.

3. GTP-U Specification and Observation

In this section we analyze the GTP-U specification and summarize clarified characteristic of GTP-U to see if GTP-U meets the requirements of 5G architecture for user plane in later section.

3.1. GTP-U Tunnel

GTP-U is a tunneling protocol between given a pair of GTP-U tunnel endpoint nodes and encapsulates T-PDU from/to UE on top of IP/UDP. A Tunnel Endpoint Identifier (TEID) value allocated on each end point indicates which tunnel a particular T-PDU belongs to.

The receiving endpoint individually allocate a TEID and the sender tunnel endpoint node encapsulates the IP packet from/to UE with the TEID which is present in GTP-U header on top of IPv4 or IPv6, and UDP. That is described in section 4.2.1 of [TS.29.281-3GPP].

[GTP-U-1]: GTP-U is an unidirectional Point-to-Point tunneling protocol.

Figure 1 shows an example of GTP-U protocol stack for uplink (UL) and downlink (DL) traffic flow. Two GTP-U tunnels are required to form one bi-directional tunnel.

UL: From RAN to UPF1 (TEID=1), and from UPF1 to UPF2 (TEID=2)

DL: From UPF2 to UPF1 (TEID=3), and from UPF1 to RAN (TEID=4)

In 5GS, GTP-U tunnel is established at following interfaces to provide PDU Session between UE and 5GC.

N3: Between RAN and UPF

N9: Between different UPFs

GTP-U allows one tunnel endpoint node to send out a G-PDU to be received by multiple tunnel endpoints by utilizing IP multicast capability of underlay IP networks. That is described in section

4.2.6 of [TS.29.281-3GPP]. It looks GTP-U has Point-to-Multipoint (P2MP) tunneling capability. The P2MP tunneling is used for MBMS (Multimedia Broadcast Multicast Service) through GTP-U tunnel.

[GTP-U-2]: GTP-U supports Point-to-Multipoint tunneling.

UDP is utilized for GTP-U encapsulation and UDP destination port is 2152 which is assigned by IANA. Allocation of UDP source port depends on sender tunnel endpoint node and GTP-U supports dynamic allocation of UDP source port for load balancing objective. The specification of this dynamic allocation is described in section 4.4.2.0 of [TS.29.281-3GPP], however specific procedure, e.g., 5-tuple hashing, is not described in the document and depends on the implementation of GTP-U tunnel endpoint node.

[GTP-U-3]: GTP-U supports load balancing by using dynamic UDP source port allocation.

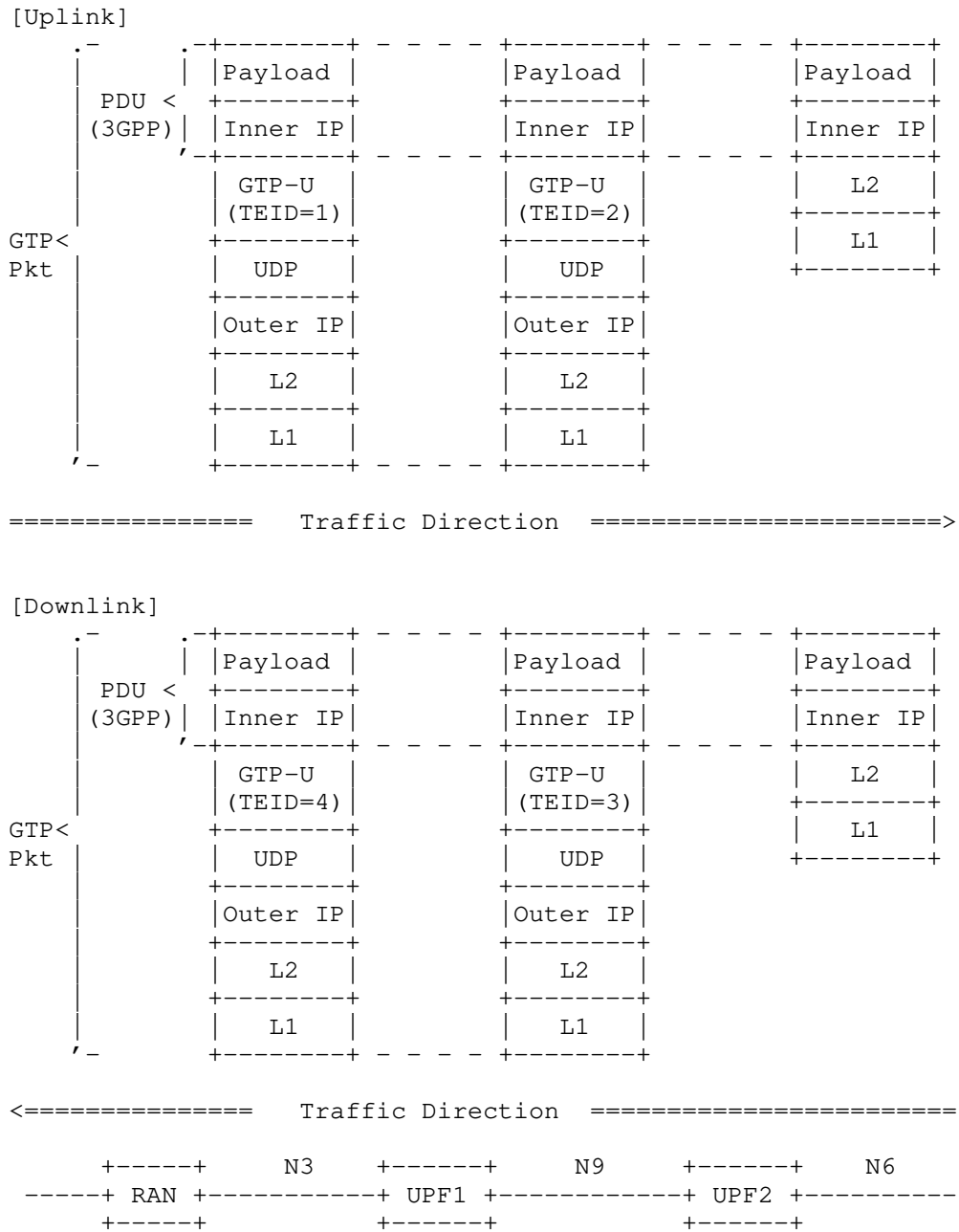


Figure 1: Protocol Stack by GTPv1-U for Uplink and Downlink Traffic Flow

IPv6 flow label [RFC6437] is also candidate method for load balancing especially for IP-in-IPv6 tunnel [RFC6438] like GTP-U. GTP-U also supports dynamic allocation of IPv6 flow label for load balancing objective. The specification of this dynamic allocation is described in section 4.4.2.0 of [TS.29.281-3GPP], however specific procedure, e.g., 5-tuple hashing, is not described in the document and depends on the implementation of GTP-U tunnel endpoint node.

[GTP-U-4]: GTP-U supports load balancing by using dynamic IPv6 flow label allocation.

GTP-U supports both IPv4 and IPv6 as the underlying network layer protocol. From Release 16, GTP-U updates their reference to IPv6 specification from [RFC2460] to [RFC8200] which allows UDP zero checksum for the protocols that use UDP as a tunnel encapsulation, such as GTP-U. As a result of the update, GTP-U over IPv6 also supports the UDP zero checksum if the sender and receiver tunnel endpoint node support the UDP zero checksum, which is described in section 4.4.2.0 of [TS.29.281-3GPP].

[GTP-U-5]: GTP-U supports UDP zero checksum.

"Unnecessary fragmentation should be avoided" is recommended and to avoid the fragmentation operator should configure MTU size at UE [TS.29.281-3GPP]. However, there's no reference and specification of Path MTU Discovery for IPv6 transport. If encapsulated IPv6 packet is too big on a network link between tunnel endpoint nodes, UE may not receive ICMPv6 Packet Too Big message and causes Path MTU Discovery black hole.

[GTP-U-6]: GTP-U does not support to response ICMP PTB for Path MTU Discovery.

Section 9.3 of [TS.23.060-3GPP] specifies advertisement of inner IPv6 link MTU size for UE by IPv6 RA message [RFC4861]. However, this document doesn't specify a procedure to measure MTU size in mobile network system and mobile network operator need to calculate MTU size for UE like Annex C of [TS.23.060-3GPP]. If link MTU of a router in a transport network is accidentally modified, UE cannot detect the event and send packet with initial MTU size, which may cause service disruption due to MTU exceed in the router link.

3.2. GTP-U Header Format

Figure 2 shows general and mandatory GTP-U header and Figure 3 shows extension GTP-U header.

[GTP-U-7]: GTP-U supports sequence number option in the header, but it is not recommended to be used by almost GTP-U entities.

GTP-U header has Sequence Number field to reorder incoming packets based on the sequence number. If Sequence Number Flag is set to '1' it indicates that Sequence Number Field exists in GTP-U header and examined at receiving tunnel endpoint node to reorder incoming packets. However, the sequence number flag is set to '1' only for RAT HO procedure and sequence number flag should be set to '0' in normal case. Therefore, in normal case receiver tunnel endpoint node doesn't examine sequence number and can't reorder GTP-U packets based on the sequence number. This specification is described in section 5.1 of [TS.29.281-3GPP]. In 3GPP, sequential delivery is required only during handover procedure and is used by only RAN entities.

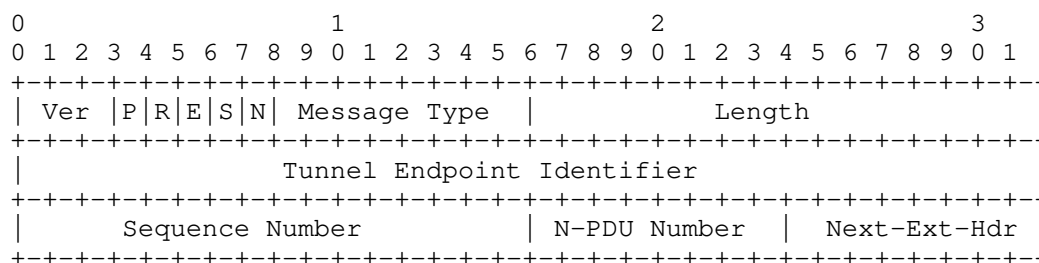


Figure 2: GTP-U Header

- o Ver: Version field (Set to '1')
- o P: Protocol Type (Set to '1')
- o R: Reserved bit (Set to '0')
- o E: Extension Header Flag (Set to '1' if extension header exists)
- o S: Sequence Number Flag (Set to '1' if sequence number exists)
- o N: N-PDU Number Flag (Set to '1' if N-PDU number exists)
- o Message Type: Indicates the type of GTP-U message
- o Length: Indicates the length in octets of the payload
- o Tunnel Endpoint Identifier (TEID)
- o Sequence Number: Indicates increasing sequence number for T-PDUs is transmitted via GTP-U tunnels

- o N-PDU Number: It is used only for inter SGSN, 2G-3G handover case, etc.
- o Next-Ext-Hdr: Indicates following extension header type

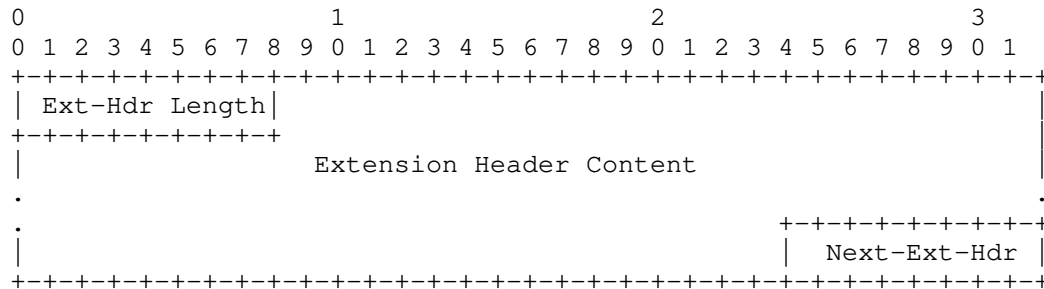


Figure 3: Extension GTP-U Header

- o Ext-Hdr Length: Represents the length of the Extension header in units of 4 octets
- o Extension Header Content: Contains 3GPP related information
- o Next-Ext-Hdr: Indicates following extension header type

The extension GTP-U header is a variable-length and extendable header and contains 3GPP specific information. Following list summarizes every extension header which is used for user plane protocol. These extension headers are defined in [TS.29.281-3GPP]. In this list Next-Ext-Hdr is represented in binary.

- o No more extension headers (Next-Ext-Hdr = 00000000)
- o Service Class Indicator (Next-Ext-Hdr = 00100000)
- o UDP Port (Next-Ext-Hdr = 01000000)
- o RAN Container (Next-Ext-Hdr = 10000001)
- o Long PDCP PDU Number (Next-Ext-Hdr = 10000010)
- o Xw RAN Container (Next-Ext-Hdr = 10000011)
- o NR RAN Container (Next-Ext-Hdr = 10000100)
- o PDU Session Container (Next-Ext-Hdr = 10000101)
- o PDCP PDU Number (Next-Ext-Hdr = 11000000)

[GTP-U-8]: GTP-U supports carrying QoS Identifiers transparently for Access Networks in an extension header.

GTP-U is designed to carry 3GPP specific information with extension headers. 3GPP creates PDU Session Container extension header for NGRAN of 5G to carry QFI. It is described in section 5.2.2.7 of [TS.29.281-3GPP].

[GTP-U-9]: GTP-U supports DSCP marking based on the QFI.

DSCP marking on outer IPv4 or IPv6 shall be set by sender tunnel endpoint node based on the QFI. This specification is described in section 4.4.1 of [TS.29.281-3GPP].

[GTP-U-10]: GTP-U does not specify extension header order.

In general, multiple GTP-U extension headers are able to contained in one GTP-U packet and the order of those extension headers is not specified by [TS.29.281-3GPP]. Thereby the receiving endpoint can't predict exact position where the target extension headers are. This could impact on header lookup performance on the node.

As for PDU Session Container extension header, there is a note in [TS.29.281-3GPP] as "For a G-PDU with several Extension Headers, the PDU Session Container should be the first Extension Header". This note was added at the version 15.3.0 of [TS.29.281-3GPP] which is published on June 2018 in order to accelerate the processing of GTP-U packet at UPF and RAN. It is only one rule regarding the extension header order.

[GTP-U-11]: GTP-U does not support to indicate next protocol type.

When Next-Ext-Hdr is set to 0x00 it indicates that no more extension headers follow. As GTP is designed to indicate protocol types for T-PDU by control-plane signaling, GTP-U doesn't have Next-Protocol-Header field to indicate the T-PDU type in the header.

3.3. Control Plane Protocol for GTP-U

Control plane protocol for GTP-U signals TEID between tunnel endpoint nodes. GTPv2-C [TS.29.274-3GPP] is the original control plane protocol tied with GTP-U in previous generation architectures before CUPS (Control and User Plane Separation).

3GPP decided to use extended PFCP (Packet Forwarding Control Protocol) [TS.29.244-3GPP] for N4 interface [TR.29.891-3GPP] to signal tunnel states from SMF to UPF.

3.4. GTP-U message

GTP-U supports in-band messaging to signal OAM. Currently GTP-U supports following messages [TS.29.281-3GPP].

- o Echo Request
- o Echo Response
- o Supported Extension Headers Notification
- o Error Indication
- o End Marker

[GTP-U-12]: GTP-U supports active OAM as a path management message "Echo Request/Response".

A GTP-U tunnel endpoint node sends a Echo Request message to another nodes for keep-alive and received node sends a Echo Response message to sender node as acknowledgment. Echo Request message and Echo Response message are described in section 7.2.1 and section 7.2.2 of [TS.29.281-3GPP] respectively. [TR.29.891-3GPP] recommends not to send Echo Request message more often than 60s on each path.

Supported Extension Headers Notification message indicates a list of supported that tunnel endpoint node can support. This message is sent only in case a tunnel endpoint node receives GTP-U packet with unsupported extension header.

[GTP-U-13]: GTP-U supports tunnel management messages "Error Indication".

GTP-U has Error Indication message to notify that the receiving endpoint discard packets of which no session exist to the sending endpoint. Error Indication message is described in section 7.3.1 of [TS.29.281-3GPP].

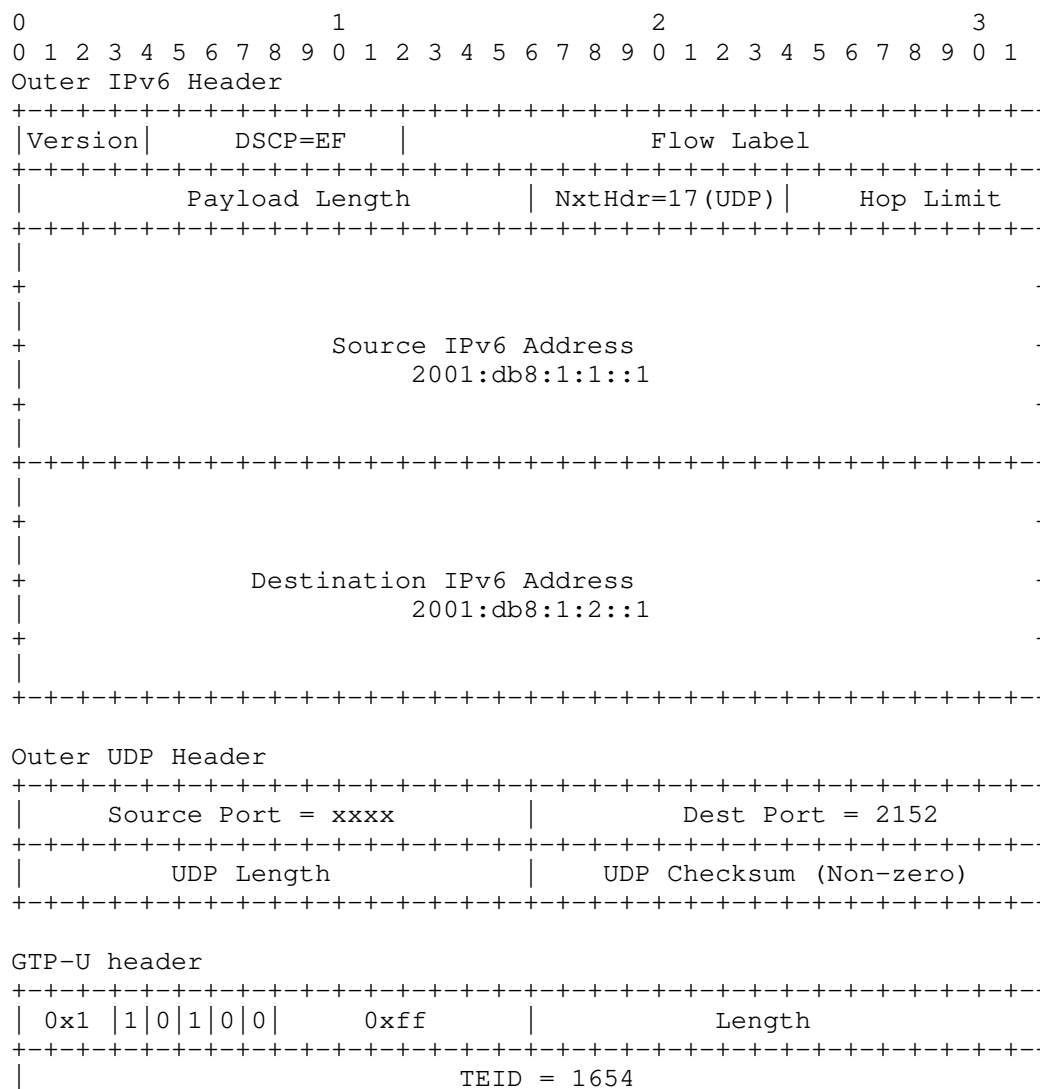
[GTP-U-14]: GTP-U supports tunnel management messages "End Marker".

GTP-U has End Marker message to indicate the end of the payload stream that needs to be sent on a GTP-U tunnel. End Marker message is described in section 7.3.2 of [TS.29.281-3GPP].

3.5. Packet Format

Figure 4 shows a packet format example of GTP-U over IPv6 that carries an extension header for QFI and an IPv6 PDU. All values in the example are illustration purpose only. The encoding of PDU Session Container for QFI refers to [TS.38.415-3GPP].

Outer IPv6 Header's DSCP value(EF) in Figure 4 is marked at sender tunnel endpoint node based on QFI value which is contained in GTP-U Extension Header (PDU Session Container).



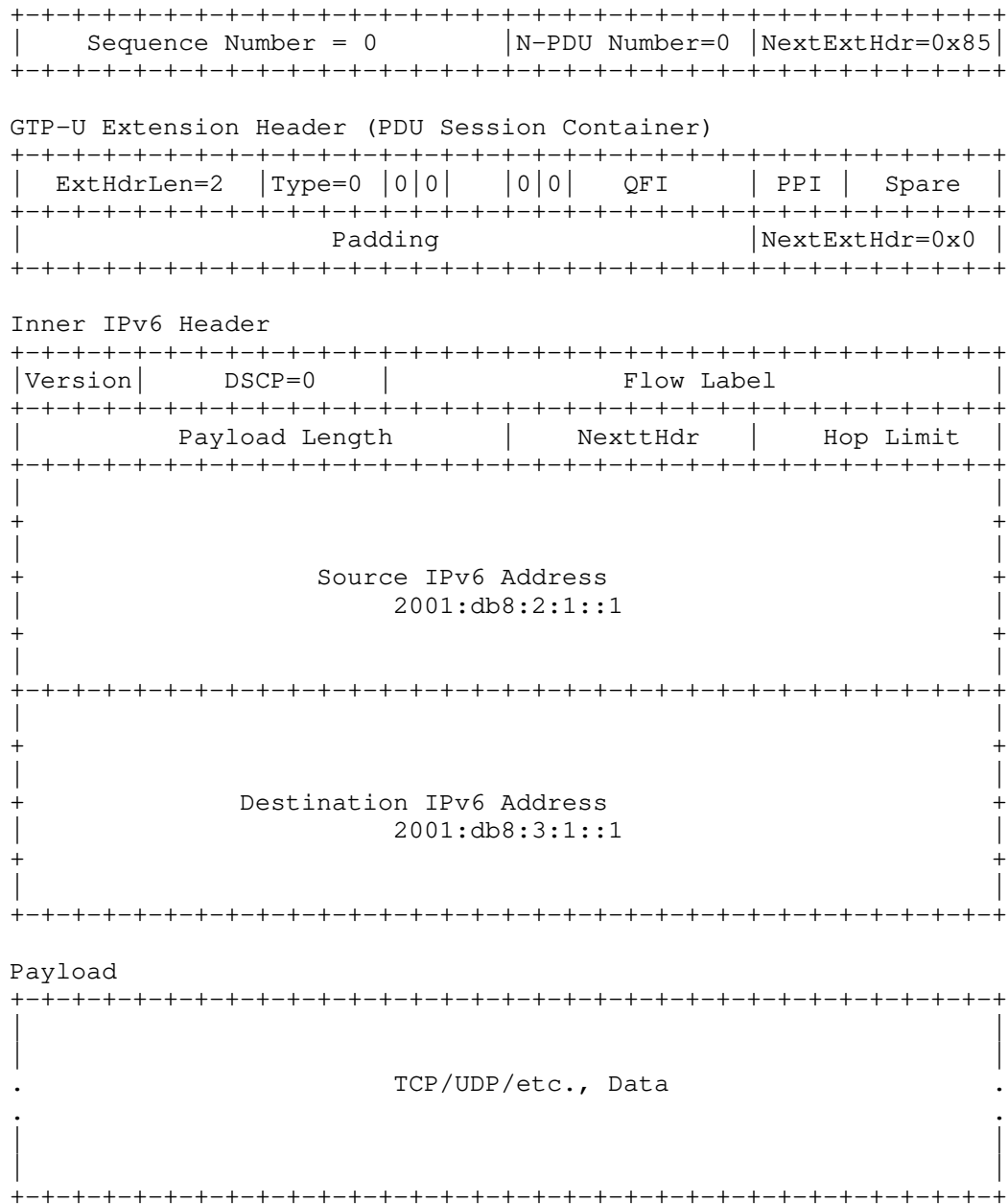


Figure 4: GTP-U Protocol Stack Example

3.6. Observations Summary

- [GTP-U-1]: An unidirectional Point-to-Point tunneling protocol.
- [GTP-U-2]: Supports Point-to-Multipoint tunneling.
- [GTP-U-3]: Supports load balancing by using dynamic UDP port allocation.
- [GTP-U-4]: Does not support IPv6 flow label for load balancing in case of IPv6 transport.
- [GTP-U-5]: UDP zero checksum is not available in case of IPv6 transport.
- [GTP-U-6]: Does not support to response ICMP PTB for Path MTU Discovery.
- [GTP-U-7]: Supports sequence number option and sequence number flag in the header, but it is not recommended to be used by almost GTP-U entities.
- [GTP-U-8]: Supports carrying QoS Identifiers transparently for Access Networks in extension headers.
- [GTP-U-9]: Supports DSCP marking based on the QFI.
- [GTP-U-10]: Does not specify the rule for the extension header order.
- [GTP-U-11]: Does not support an indication of next-header type.
- [GTP-U-12]: Supports active OAM as a path management message "Echo Request/Response".
- [GTP-U-13]: Supports tunnel management messages "Error Indication".
- [GTP-U-14]: Supports tunnel management messages "End Marker".

4. 5GS Architectural Requirements for User Plane Protocols

4.1. Overview of 5G System Architecture

The 5G system is designed for applying to diverse devices and services due to factors such as the diffusion of IoT devices, and the UP protocol is required to have capabilities for satisfying their requirements.

As a principle of the 5G system, User Plane (UP) functions are separated from the Control Plane (CP) functions for allowing independent scalability, evolution and flexible deployments.

Network slicing is also one of the fundamental concepts of the 5G system, and it provides logical network separation. In terms of user plane, multiple network slices can be comprised of UPFs on top of same physical network resources. Allocated resources and structures may be differentiated among the slices by which the required features or capabilities.

The 3GPP 5G architecture [TS.23.501-3GPP] defines slice types which are eMBB, URLLC and MIoT from Rel-15. In addition to that, V2X slice type is defined from Rel-16.

The architecture overview is shown in Figure 5. The details of functions are described in [TS.23.501-3GPP]. A UPF handles UP paths on N3, N9 and N6 interface, and the setup is controlled by SMF via N4 interface. A UP path will be manipulated based on application requirements for the PDU session corresponding to the path. An SMF is also capable to receive information regarding routing path with API from AF via NEF, PCF, and SMF.

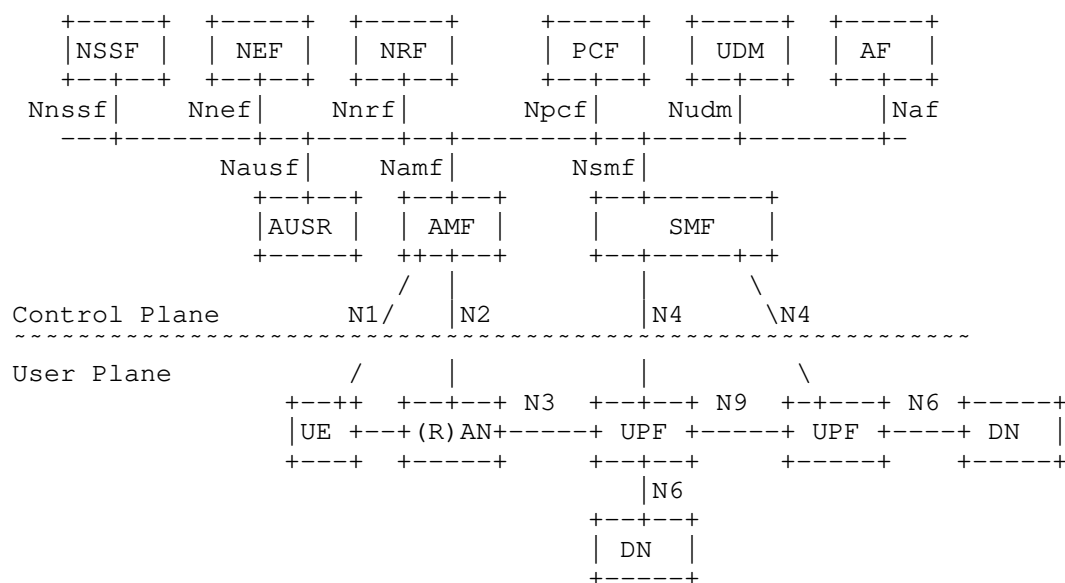


Figure 5: 5GS Architecture and Service-based Interfaces

This document mainly focuses on requirements for N9 interface as relevant to UP protocol of 5G system.

4.1.1. UPF Functionalities

UPF has a role to handle UP traffic, and provides functionalities to look up user data traffic and enforce the appropriate policies to it.

The followings are defined as UPF functionalities defined in the section 6.2.3 of [TS.23.501-3GPP]

- o Anchor point for Intra-/Inter-RAT mobility (when applicable).
- o External PDU Session point of interconnect to Data Network.
- o Packet routing and forwarding (e.g. support of Uplink classifier to route traffic flows to an instance of a data network, support of Branching point to support multi-homed PDU Session).
- o Packet inspection (e.g. Application detection based on service data flow template and the optional PFDs received from the SMF in addition).

- o User Plane part of policy rule enforcement, e.g. Gating, Redirection, Traffic steering).
- o Lawful intercept (UP collection).
- o Traffic usage reporting.
- o QoS handling for user plane, e.g. UL/DL rate enforcement, Reflective QoS marking in DL.
- o Uplink Traffic verification (SDF to QoS Flow mapping).
- o Transport level packet marking in the uplink and downlink.
- o Downlink packet buffering and downlink data notification triggering.
- o Sending and forwarding of one or more "end marker" to the source NG-RAN node.
- o ARP proxying and / or IPv6 Neighbour Solicitation Proxying for the Ethernet PDUs.
- o Packet duplication in downlink direction and elimination in uplink direction in UP protocol layer.
- o TSN Translator functionality to hold and forward user plane packets for de-jittering when 5G System is integrated as a bridge with the TSN network.

4.1.2. UP Traffic Detection

The traffic detection is described in the section 5.8.2.4 of [TS.23.501-3GPP]. In 3GPP UP packet forwarding model, UPF detects UP traffic flow which belong to a N4 session configured by SMF.

The protocol of N4 interface, PFCP, brings a set of traffic detection information from SMF to UPF as Packet Detection Information (PDI) in a PDR to establish/modify the N4 PFCP session. It is defined in section 7.5.2.2 of [TS.29.244-3GPP].

Combination of the following information is used for the traffic detection:

- o For IPv4 or IPv6 PDU Session type
 - * CN tunnel info (Tunnel ID and the endpoint IP address of 5G Core)

- * Network instance
 - * QFI
 - * IP Packet Filter Set
 - * Application Identifier: The Application ID is an index to a set of application detection rules configured in UPF
- o For Ethernet PDU Session type
 - * CN tunnel info(Tunnel ID and the endpoint IP address of 5G Core)
 - * Network instance
 - * QFI
 - * Ethernet Packet Filter Set

It is noted that Network Instance is encoded as Octet String in PFCP, and is NOT appeared in UP packet over the wire. It is expected like an attribute of the receiving IP interface of the UPF. It supports UPF to be able to connect to different IP domains of N3, N9 or N6, which run each independent policy in routing and addressing. The UPF detects traffic flow with Network Instance which the receiving interface attributed to.

The IP Packet Filter Set and Ethernet Packet Filter Set defined in clause 5.7.6 of [TS.23.501-3GPP] are following:

- o IP Packet Filter Set:
 - * Source/destination IP address or IPv6 prefix
 - * Source/destination port number
 - * Protocol ID of the protocol above IP/Next header type
 - * Type of Service (TOS) (IPv4) / Traffic class (IPv6) and Mask.
 - * Flow Label (IPv6)
 - * Security parameter index
 - * Packet filter direction
- o Ethernet Packet Filter Set:

- * Source/destination MAC address
- * Ethertype as defined in IEEE 802.3
- * Customer-VLAN tag(C-TAG) and/or Service-VLAN tag(S-TAG) VID fields as defined in IEEE 802.1Q
- * Customer-VLAN tag(C-TAG) and/or Service-VLAN tag(S-TAG) PCP/DEI fields as defined in IEEE 802.1Q
- * IP Packet Filter Set, in case Ethertype indicates IPv4/IPv6 payload
- * Packet filter direction

4.1.3. User Plane Configuration

User Plane configuration on a UPF is managed by an SMF through PFCP [TS.29.244-3GPP]. The SMF establishes PFCP sessions on the UPF per PDU session basis. The UPF maintains each configured PFCP session states during the sessions exist.

A PFCP session consists of the rules of packet detection, forwarding action, QoS enforcement, usage reporting and buffering action. Figure 6 depicts overview of the PFCP session state structure.

The listed information in Section 4.1.2 indicates packet detection information of packet detection rule for that the rest of related rules within the PFCP session to be derived. All rules are per session unique and no rules are shared with other sessions.

```

PFCP-Session* [F-SEID]
+- F-SEID(Full Qualified Session Endpoint ID)      uint64
+- PDU-Session-Type                               [IPv4|IPv6|IPv4v6|Ether|Unstrct]
+- DNN(Data Network Name)
+- PDR(Packet Detection Rule)* [PDR-ID]
|   +- PDR-ID      uint16
|   +- PDI (Packet Detection Information)
|   |   +- Traffic-Endpoint-ID?  -> Traffic-Endpoint-ID reference
|   |   +- ....
|   +- FAR/URR/QER-ID            -> FAR/URR/QER-ID references
+- FAR(Forwarding Action Rule)* [FAR-ID]
|   +- FAR-ID              uint32
|   +- Forwarding-Parameters
|   |   +- Network-Instance?      Octet String
|   |   +- Outer-Header-Creation
|   |   +- Outer-Hdr-Creation-Desc [GTPoUDP/IPv4|IPv6, etc.,]

```

```

|   |   |   +- TEID, outer IP-Address for N3/N9
|   |   |   +- C/S-TAG, UDP Port-number for N6
|   |   +- Forwarding-Policy-ID?   Octet String
|   |   +- ....
|   +- Duplicating-Parameters
|   |   +- ....
|   +- BAR-ID?                      -> BAR-ID reference
+- QER(QoS Enforcement Rule)* [QER-ID]
|   +- QER-ID                      uint32
|   +- MBR(Maximum Bit Rate)
|   |   +- UL/DL-MBR?   bitrate_in_kbps (0..10000000)
|   +- GBR(Guaranteed Bit Rate)
|   |   +- UL/DL-GBR?   bitrate_in_kbps (0..10000000)
|   +- QoS-flow-identifier?        QFI value(6-bits)
|   +- Reflective-QoS?              boolean
|   +- Paging-Policy-Indicator?     PPI value(3-bits)
|   +- ....
+- URR(Usage Reporting Rule)* [URR-ID]
|   +- URR-ID                      uint32
|   +- Measurement-Method, Period, Reporting-Triggers?
|   +- Volume/Event/Time Threshold, Quota?
|   +- Quota-Holding-Time?
|   +- FAR-ID for Quota action?      -> FAR-ID reference
|   +- ....
+- BAR(Buffering Action Rule)* [BAR-ID]
|   +- BAR-ID                      uint8
|   +- Suggested-Buffering-Packets-Count
+- Traffic-Endpoint* [Traffic-Endpoint-ID]
|   +- Traffic-Endpoint-ID          uint8
|   +- TEID, Tunnel IP Address, UE Address...?

```

Figure 6: User Plane Configuration Model

4.2. Architectural Requirements for User Plane Protocols

This section lists the requirements for the UP protocol on the 5G system. The requirements are picked up from [TS.23.501-3GPP]. In addition, some of service requirements described in [TS.22.261-3GPP] are referred to clarify the originations of architectural requirements.

According to [TS.23.501-3GPP], the specifications potentially have assumptions that the UP protocol is a tunnel representing a single TEID between a pair of UPFs and it is corresponding to a single PDU session. In short, the UP protocol is a tunnel and it is assumed to be managed under per PDU session handling. Also, it should be a stateful tunnel in the UPFs along with the PDU session.

4.2.1. Fundamental Functionalities

The fundamental requirements for UP protocols are described below:

ARCH-Req-1: Supporting IPv4, IPv6, Ethernet and Unstructured PDU

The 5G system defines four types of PDU session as IPv4, IPv6, Ethernet, and Unstructured. Therefore, UP protocol must support to convey all of these PDU session types. This is described in [TS.23.501-3GPP].

Note: In TS 23.501 v15.2.0, IPv4v6 is added as a PDU session type.

ARCH-Req-2: Supporting IP connectivity for N3, N6, and N9 interfaces

The 5G system requires IP connectivity for N3, N6, and N9 interfaces. The IP connectivity is assumed that it comprises of IP routing and L1/L2 transport networks which are outside of 3GPP specifications.

It is desirable that the IP connectivity built on IPv6 networks when it comes to address space for end-to-end user plane coverage. But it is expected to take certain time. During the IPv6 networks are not deployed for all the coverage, UP protocol should support RANs and DNS running on IPv4 transport connect to UPF running on IPv6 transport.

Furthermore, on N6 interface, point-to-point tunneling based on UDP/IPv6 may be used to deliver unstructured PDU type data. Then, the content information of the PDU may be mapped into UDP port number, and the UDP port numbers is pre-configured in the UPF and DN. This is described in the section 9.2 of [TS.29.561-3GPP].

ARCH-Req-3: Supporting deployment of multiple UPFs as anchors for a single PDU session

The 5G system allows to deploy multiple UPFs as anchors for a single PDU session, and supports multihoming of a single PDU session for such anchor UPFs.

Multihoming is provided with Branching Point (BP). BP provides forwarding of UL traffic towards the different PDU Session Anchors based on the source IPv6 prefixes and merge of DL traffic to the UE. IPv6 multihoming only means multiple source IPv6 prefixes are used for a PDU session. It is identical to one classified as scenario 1 in [RFC7157].

Up link classifier (UL CL) is to forward uplink packets to multiple anchor UPFs based on the destination IP of the T-PDU regardless of

the source IP address. Noted that single source IP address/prefix PDU session is not defined as multihoming PDU session in 5GCS even though a PDU session has multiple anchor UPFs.

On UL side, P2P tunnels are established per destination anchor UPFs basis from one UL CL UPF to the anchor UPFs for the PDU session.

On DL side, one single multipoint-to-point (MP2P) tunnel exists from the source anchor UPFs to the destination BP UPF for the PDU session. It means that the paths from the anchor UPFs are merged into just one tunnel state at the destination BP UPF.

Multiple P2P paths on DL could also be used for multihoming. However it should be the multiple PDU sessions multihoming case where the destination gNB or UPF needs to maintain multiple tunnel states under the one PDU session to one UP tunnel architectural principle. It causes increase of load on tunnel states management in UPF due to increment of the anchor UPF for the PDU session.

However, P2P tunneling could increase explosively the number of states in UPF as the anchor UPF/DN incremented to the PDU session. Thereby single PDU session multihoming with MP2P path should be a better option for multihoming in terms of reducing total number of tunnel states.

SSC mode 3 for session continuity in hand-over case uses a single PDU multihoming with BP to make sure make-before-break. It is described in the section 5.6.4 and 5.6.9 of [TS.23.501-3GPP].

Multihoming is also assumed to be used for edge computing scenario. Edge computing enables some services to be hosted close to the UE's access point of attachment, and achieves an efficient service delivery through the reduced end-to-end latency and load on the transport network. In edge computing, local user's traffic is routed or steered to application in the local DN by UPF. This refers the section 5.13 of [TS.23.501-3GPP].

ARCH-Req-4: Supporting flexible UPF selection for PDU

The appropriate UPFs are selected for a PDU session based on parameters and information such as UPF's dynamic load or UE location information. Examples of parameters and information are described in the section 6.3.3 of [TS.23.501-3GPP].

This means that it is possible to make routing on user plane more efficient in the 5GS. For example, in case that UPFs are distributed geographically, decision of the destination UPF based on locations of end hosts (e.g., UE or NF in DN) enables to forward PDUs with a route

connecting between UPFs nearby the hosts directly. This would be useful UE-to-UE or UE-to-local_DN communication, and such usage is described in the section 6.5 of [TS.22.261-3GPP].

The 5GS allows operators to select parameters used for UPF selection. (In other words, any specific schemes on UPF selection are not defined in the current 3GPP documents.)

ARCH-Req-5: No limitation for number of UPFs in a data path

The number of UPF in the data path is not constrained by 3GPP specifications. This specification is described in the section 8.3.1 of [TS.23.501-3GPP].

Putting multiple UPFs, which provides specific function, in a data path enables flexible function deployment to make sure load distribution optimizations, etc.

Meanwhile, each UPF in a data path shall be controlled by an SMF via N4 interface. Thus putting an excess of UPF for data paths might cause increase of load of an SMF. Pragmatically, the number of UPF put in a data path is one or two (e.g., for MEC or roaming cases), and, at most, it would be three (e.g., for case where UE moves during a session).

It is expected that multiple UPFs with per session tunnel handling for a PDU session becomes complicated task more and more for a SMF by increasing number of UPFs.

ARCH-Req-6: Supporting aggregation of multiple QoS Flow indicated with QFI into a PDU Session

Against to the previous generation, 5G enables UPF to multiplex QoS Flows, equivalent with IP-CAN bearers in the previous generation, into one single PDU session. That means that a single tunnel includes multiple QFIs contrast to just one QoS Flow (a bearer) to one tunnel before 5G.

In even the 5GS, each flow is forwarded based on the appropriate QoS rules. QoS rules are configured by SMF as QoS profiles to UP components and these components perform QoS controls to PDUs based on rules. In downlink, a UPF pushes QFI into an extension header, and transmits the PDU to RAN or another UPF. Then, such UPF may perform transport level QoS packet marking (e.g., DSCP marking in the outer header). In uplink, each UE obtains the QoS rule from SMF, and transmit PDUs with QFI containing the QoS rules to the RAN. The following RAN and UPFs perform enforcement of QoS control and charging based on the QFI.

This specification is described in 5.7.1 of [TS.23.501-3GPP].

ARCH-Req-7: Supporting network slicing

The 5GS fundamentally supports network slicing for provision the appropriate end-to-end communication to various services. In the relevant documents (e.g., [TS.23.501-3GPP], [TS.28.530-3GPP]), a network slice is defined as virtual network and it is structured with 5GS NF instances, such as SMF, UPF including IP transport connectivity between RANs and DNS. Each network slice is independent and its user plane (including network functions and links) should be noninteractive against the others.

The 5G architecture specification has been updated with that Network Instance is defined as the glue of network slice between 5G slice and corresponding IP transport slice in addition to the original role of separating IP domains, which is described in Section 4.1.2.

It has been appeared from version 15.2.0 of [TS.23.501-3GPP] in section 5.6.12.

UP underlay transport networks and UPFs may be shared by 5G slices, as described in section 4 of [TS.28.530-3GPP]. The data model defined in [TS.29.510-3GPP] allows that a Network Instance, a UPF and its interfaces can belong to multiple slices as same as other type of NFs. UP endpoint IP prefix/address of an interface can also be shared with multiple interfaces on the UPF as the model doesn't make them slice unique.

The slice lifecycle managements is described in the relevant documents: [TS.28.531-3GPP], [TS.28.532-3GPP], and [TS.28.533-3GPP].

ARCH-Req-8: End Marker support

The construction of End Marker packets specified in [TS.23.501-3GPP] may either be done in the CP/UP functions for indicating the end of the payload stream on a given UP tunnel. PDU packets arrive after an End Marker message on the tunnel may be silently discarded. For example, End Marker is used for handover procedures, and it can prevent reordering of arriving packets due to switch of anchor UPFs.

4.2.2. Supporting 5G Services

In the release 16 [TS.23.501-3GPP], some specifications have been added to support 5G specific services and communications. This section describes overviews of the specifications relevant to use plane functionalities.

ARCHI-Req-9: URLLC Support

The 5GS supports Ultra-Reliable Low Latency Communication (URLLC) for mission critical applications. The User Plane features are described below.

o Redundant UP transmission for URLLC

The 5G is expected to support services which are latency sensitive and require high reliability. Communication to realize such services is called Ultra-Reliable and Low-Latency Communication or URLLC. In URLLC, redundancy of QoS flows is required for providing highly reliable communication. For instance, a set of UP NFs (e.g., UPF or gNB) and interfaces between UE and DN are redundant, and packets are replicated and forwarded via each route. UEs and DN support dual connectivity and drop duplicated received packets. The scheme of packet dropping at UE is out of responsibility of 3GPP. The overview is shown in Figure 7.

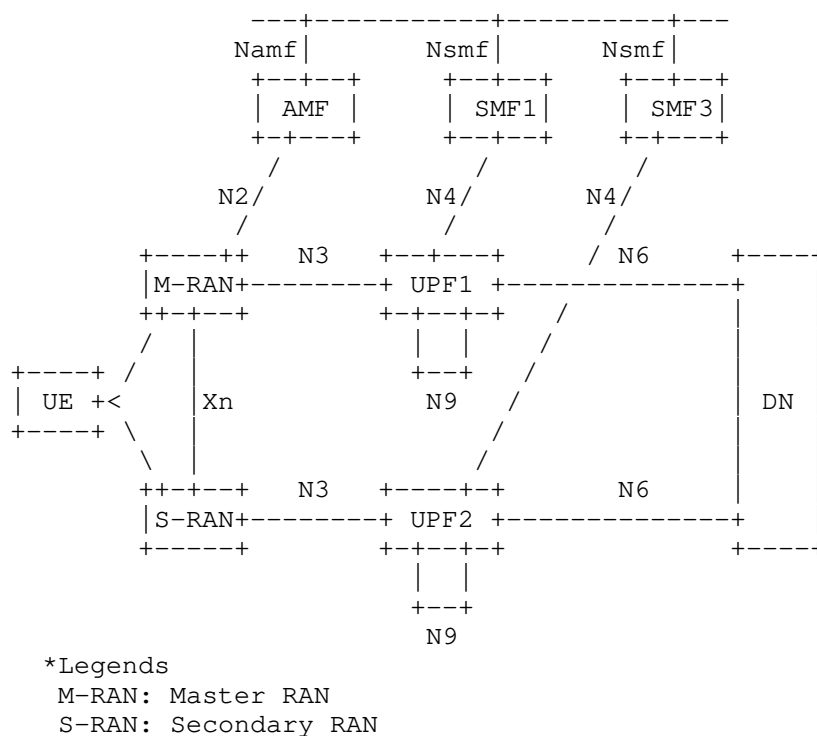


Figure 7: Redundant UP paths using dual connectivity

Otherwise, in case that RAN nodes and UPFs have enough reliability and they are not redundant by dual devices, reliable connectivity of QoS flows is provided by dual N3 tunnels between RAN and UPFs. Such tunnels are treated as individual ones, but they have the same sequence number. UP NFs identifies the duplication of PDU packets based on sequence number content in the UP tunnel headers. For uplink packets, a RAN node replicates each packet from a UE. An anchor UPF receives the duplicated packets, and drops ones which reach later in each duplicated packet pare. On the other hand, for downlink packets, a UPF replicates packets received from DN, and a RAN node drops the duplicated packets as well. The overviews of the ways are shown in Figure 8.

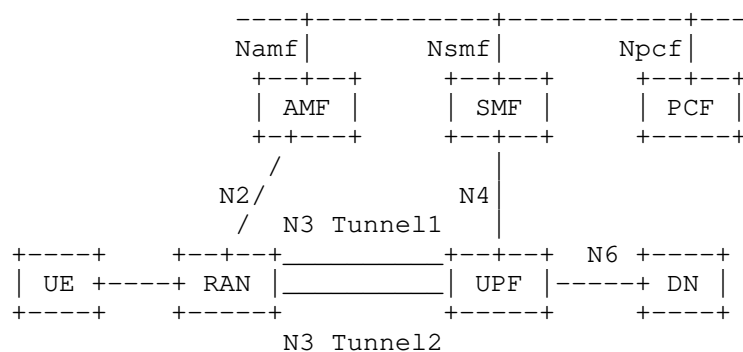


Figure 8: Redundant UP transmission with two N3 tunnels

In addition, there is a case that two intermediate UPFs (I-UPFs) between anchor UPF and RAN are used to support the redundant transmission based on two N3 and N9 tunnels between single anchor UPF and RAN node. The RAN node and anchor UPF support the packet replication and dropping of duplicated packets as described above. As described above, anchor UPF and RAN node detect packet duplication with sequence number of UP tunnels, and thus I-UPFs would forward the packets with the same sequence number on N3 and N9 tunnels. The overview is shown in Figure 9.

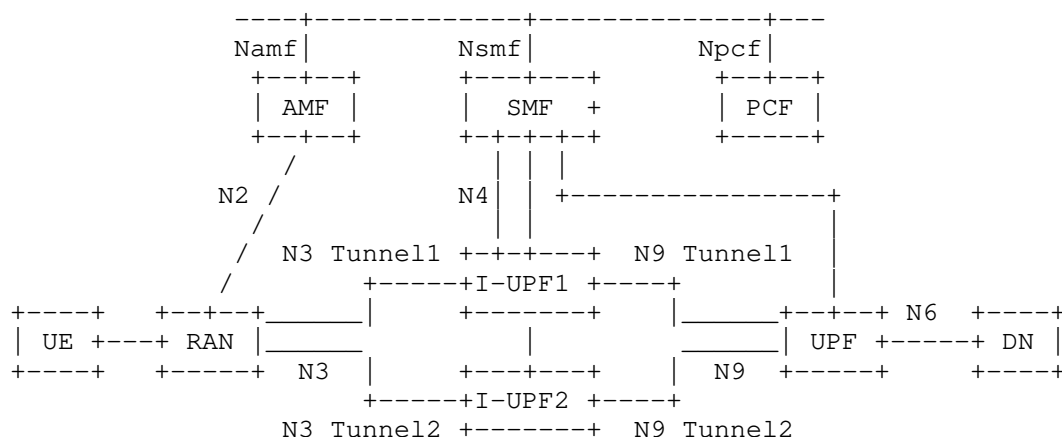


Figure 9: Redundant UP transmission with two I-UPF and N3/N9 tunnels

o Supporting QoS Monitoring for URLLC

QoS monitoring is also required for URLLC. It means that the user plane should be able to measure packet delay between anchor UPF and UE. The measurement would be in various granularities, in the basis of per QoS Flow per UE, or per UP path for example.

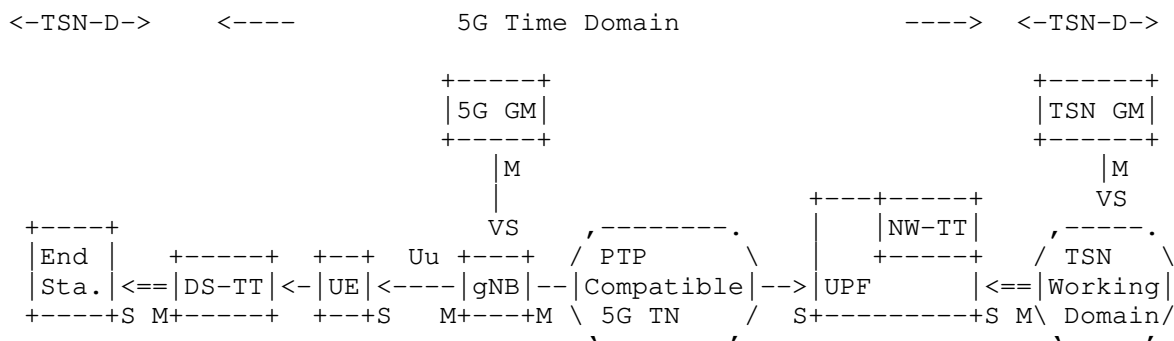
To help the measurement at anchor UPF and RAN, UP protocol requires to have capability to convey necessary information to do that; such as time information at sending or reception of a measurement packet. That information should exist in per F-TEID and QFI basis which indicates QoS Flow of the packet. UP protocol should also be able to indicate which packets include the corresponding information for each measurement.

The QoS monitoring requirement has been appeared in section 5.33.3 of [TS.23.501-3GPP] from Rel-16, version 16.2.0.

ARCHI-Req-10: Time Sensitive Communication Support

The 5GS supports Time Sensitive Communications (TSC) for realtime applications, and it can be integrated transparently as a bridge in an IEEE 802.1 TSN network. For TSN time synchronization, the E2E 5GS can be considered as a "time-aware system (ref [IEEE-Std-802.1AS])". The TSN Translators (TTs) at the edges of the 5GS need to support the [IEEE-Std-802.1AS] operations. For instance, UE, gNB, NW-TT (Network-side TSN Translator) and DS-TTs (Device-side TSN Translators) are synchronized with the Grandmaster (GM) located in the 5GS. In addition, the TTs fulfill some functions related to [IEEE-Std-802.1AS] (e.g., gPTP support, timestamping, rateRatio,

etc.). An overview of the 5G and TSN GM clock distribution model via the 5GS is shown in Figure 10.



Legend

TSN-D : Non-3GPP TSN Domain
 TN : Transport Network
 End Sta.: End Station
 <-- : 5GS timing direction
 <== : TSN timing direction
 M : Master
 S : Slave

Figure 10: An overview of the 5G and TSN GM clock distribution model

In this model, two independent synchronizations are processing, and gNB only needs to be synchronized to the 5G GM clock. To enable TSN domain synchronization, the 5GS calculates and adds the measured residence time between the DS-TT and NW-TT into the Correction Field (CF) of the synchronization packet of the TSN working domain. The details are described in section 5.27 in [TS.23.501-3GPP].

From this feature, UP functions and protocol are needed to support TSN specified in [IEEE-Std-802.1AS] .

ARCHI-Req-11: Cellular IoT Support

For supporting Cellular IoT (CIoT) (ref. [TS.22.261-3GPP]), optimizations of functionalities of the 5GS is needed. CIoT is in earlier 3GPP release also referred to as Machine Type Communication (MTC). Some of CIoT functionalities relevant to user plane are described in this section. The details of CIoT support is described in section 5.31 in [TS.23.501-3GPP].

- o Non-IP Data Delivery (NIDD)

The 5GS may support Non-IP Data Delivery (NIDD) to handle Mobile Originated (MO) and Mobile Terminated (MT) communication for unstructured data. Thus, User Plane Protocol should be conveyable such unstructured data units.

- o Reliable Data Service (RDS)

Reliable Data Service (RDS) may be used for a PDU session of unstructured type. The service provides a mechanism for the NEF or UPF to determine if the data was successfully delivered to the UE and for the UE to determine if the data was successfully delivered to the NEF or UPF.

When the service is enabled, a protocol that uses a packet header to identify the requested acknowledgement from peered end-point may be used between end-points of the PDU session. In addition, port numbers in the header are used to identify the applications on the originator and receiver. The UE, NEF and the UPF may support reservation of the source and destination port numbers for their use and subsequent release of the reserved port numbers.

Therefore, UP protocol is required to have fields for containing information to determine normality of unstructured PDU sessions and used applications.

- o High Latency Communication

Functions for High Latency Communication may be used to handle mobile terminated (MT) communication with UEs being unreachable while using power saving functions. "High latency" refers to the initial response time before normal exchange of packets is established. High latency communication is supported by extended buffering of downlink data in the UPF, SMF or NEF when a UE is using power saving functions in CM-IDL state and the UE is not reachable.

- o Small Data Rate Control

The SMF may apply Small Data Rate Control for PDU sessions based on, for example, operator policy, DNN, S-NSSAI, RAT type etc. The rate control may indicate following parameters in each of uplink and downlink.

- an integer number of packets per time unit

- an integer number of additional allowed exception report packets per time unit once the rate control limit has been reached

The UE shall comply with this uplink rate control instruction. If the UE exceeds the uplink number of packet per time, the UE may still send uplink exception report if allowed and the number exception reports per time unit has not been exceeded.

For the UPF and NEF, Small Data Rate Control is based on a maximum allowed rate per direction. The UPF or NEF may enforce the uplink rate by discarding or delaying packets that exceed the maximum allowed rate. The UPF or NEF shall enforce the downlink rate by discarding or delaying packets that exceed the downlink part of the maximum allowed rate.

o User Plane CIoT 5GS Optimisation

User Plane CIoT 5GS Optimization enables transfer of user plane data from CM-IDLE without the need for using the Service Request procedure by negotiation between UE and AMF in advance. In case that there are many devices being CM-IDLE state for long time, it would be better that User Plane Protocol is session less.

5. Evaluation Aspects

This section provides UP protocol evaluation aspects that are mainly derived from the architectural requirements described in Section 4. Those aspects are not prioritized by the order here. Expected deployment scenarios explain the evaluations purpose in the corresponding aspects.

As we were noticed that the gaps between GTP-U specifications and 5G architectural requirements through the analysis, those each gap are briefly described in the evaluation aspect associated to it.

Since it is obvious that 5G system should be able to interwork with existing previous generation based systems, any aspects from coexisting and interworking point of view are not particularly articulated here. It may be described in a next version.

5.1. Supporting PDU Session Type Variations

Given that UP protocol is required to support all PDU session types: IPv4, IPv6, Ethernet, and Unstructured. However, it is expected that some deployment cases allow candidate protocol to adopt only one or few PDU session type(s) for simplicity of operations. As we can expect that IPv4 connectivity services will be available through IPv6-only PDU session that enabled by bunch of IPv6 transition solutions already available in the field.

For this, the expected evaluation points from this aspect should be whether there is substitutional means to cover other PDU session types. And how much it makes simple the system than deploying original PDU session types.

5.2. Nature of Data Path

As it is described in Section 4.2, the single PDU session multi-homing case requires multipoint-to-point (MP2P) data path. It should be much scalable than multi-homing with multiple PDU sessions because number of required path states in the UPFs are reduced as closed to egress endpoint. Against that point-to-point (P2P) protocol requires same number of states in each UPF throughout the path, and it could increase explosively the load on management of tunnel states.

From this point of view, the expected evaluation points from this aspect is whether the nature of candidate UP protocols are to utilize MP2P data path. Supporting MP2P data path by GTP-U could be a gap since GTP-U is a point-to-point tunneling protocol as it is described in Section 3.

Noted that 3GPP CT WG4 pointed out GTP-U was already required to allow one single tunnel endpoint to receive packets from multiple source endpoints ([C4-185491-3GPP]). It was an architectural requirement of 3GPP system from a previous generation. It means that MP2P data path requirement for UP protocol has been existed before the 5G system.

5.3. Supporting Transport Variations

The 5G system will be expected that the new radio spectrums in high frequency bands require operators to deploy their base stations much dense for much wider areas compare to previous generation footprints. To make sure that density and coverage, all available types of transport in the field must be employed between RAN to UPF, or UPF to UPF.

It is also expected that MTU size of each transport could be varied. Because one could be own fiber which the operator configure the MTU size as they like while others are third-party provided L2/L3 VPN lines which MTU size can't be controlled by the operators.

The MTU between RAN and UPF can be discovered by offline means and the operator takes into account the MTU that is transferable on the radio interface and based on this the operator configures the right MTU to be used. That is then signaled to the UE either via PCO (for IPv4 case) or the IPv6 RA message (for IPv6 case).

In addition, for cases that third-parties provide VPN lines, it would be recommended MTU size discovery for each data path and dynamic MTU size adjustment mechanisms, while GTP-U does not support those mechanisms.

As the study item in 3GPP mentioned, IPv6 is preferable address family not only for UEs, but also for the UP transport, in terms of size of available address space to support dense and wide footprint of base stations. However it increases header size from 20bytes to 40bytes compare to IPv4. It could be a problem if the MTU size is uncontrollable, or only limited MTU size available to carry committed PDU size on the user plane.

The expected evaluation points from this aspect should be that the candidate protocols are able to dynamically adjust path MTU size with appropriate MTU size discovery mechanism. It also should be that how the candidate protocols leverage IPv6 to deal with header size increasing.

5.4. Data Path Management

As Section 4.2 described, the 5G systems allows user plane that flexible UPF selection, multiple anchor UPFs, and no limit on how many UPFs chained for the data path of the PDU session. UPF deployments in the field will thereby be distributed to be able to optimize the data path based on various logics and service scenarios.

That powerful user plane capability could make data path management through the control plane, or operation support systems (OSS) be complicated and difficult. Perhaps it could be the case where the UP protocol nature is P2P and it only supports per session base data path handling. Therefore it would be better that UP protocol could support to aggregate several PDU sessions into a tunnel or shall be a session-less tunnel.

Because it increases data path states by number of sessions, and number of endpoints of UPFs that makes data path handling much hectic and the control plane tend to be overloaded by not only usual attach/detach/hand-over operations, but also existing session manipulation triggered by UPF and transport nodes/paths restoration, etc.,

The expected evaluation points from this aspect should be that how much the candidate protocols can reduce data path management loads both on the control plane NFs and UPFs compare to the per session based handling for P2P paths. It could possibly include N3 and N6 in addition to N9 while it supports flexible user plane data path optimizations for some example scenarios.

5.5. QoS Control

The QoS model is based on QoS flows to which QFI indicates in the 5G system that allows multiple QoS flows are aggregated into a single PDU session. So that it is given that the UP protocol should convey QFIs for a PDU session and the UPF needs to lookup them. It makes sure that reflects QoS policy in the 5G system to corresponding forwarding policy in the user plane IP transports.

The expected evaluation points from this aspect should be whether the candidate protocols can provide stable ID space for QFI which shouldn't be a big deal since QFI just requires 6-bits space.

As we pointed out in Section 3.2, the lookup process could impact UPF performance if the QFI container position in the header is unpredictable. It could happen many times along the path because the each UPFs should do it again and again in case that various different QoS policies are deployed in the networks under the UP as we discussed in Section 5.3.

As [TS.29.281-3GPP] updated in version 15.3.0, it is recommended that the first extension header is the PDU session container in which QFI is present.

5.6. Traffic Detection and Flow Handling

As described in Section 4.1.1, UPF need to detect traffic flow specified by SMF within a PDU session, and enforce some processes to the PDU based on the pre-configured policy rule.

As similar with QoS flow lookup described in Section 5.5, UPFs along the path are repeatedly detecting an specified traffic flow in inner PDU. It could increase redundant flow detection load on every UPFs that could be avoided if the upstream UPF put some identifier which abstracts the detected flow into the packets. It enables following UPFs just find the ID to detect the indicated flow from the packet.

The expected evaluation points from this aspect should be whether the candidate protocols can provide means to reduce that redundant flow detection that could be enough bits space on stable ID space to put abstracted detected flow identifier.

5.7. Supporting Network Slicing Diversity

Network Instance has been defined as the glue of network slice between 5G and IP transport in addition to IP domain separation, as described in Section 4.1.2. It is expected that SMF is able to configure UPF to send UP packet to corresponding transport slice by

indicating Network Instance in FAR so that UPF can determine outgoing interface for the UP packet.

It is assumed that IP transport networks are Network Instance agnostic, i.e., transport slices are independently instantiated and not bound to specific IP address space in the 5GC, for preventing increase of routing table size.

As a transport slice may be shared with multiple IP domains, Network Instance could be instantiated for all combination of IP domains and transport slice. To indicate those combination in UP packet over the wire, the 5G architecture expects VPN solutions as described in section 5.6.12 of [TS.23.501-3GPP].

Binding Network Instance with corresponding VPN would be varied per VPN solutions and FAR is not able to do. Hence it is out of scope of 3GPP and it may be covered by IETF, or other SDOs.

Apart from binding, if it is the case where MPLS based VPNs, such as [RFC4364] and [RFC4664] are expected as the existing VPN solution which bound to Network Instance, there are some available deployment options, such as 1). PE router integrates UPF, 2). CE router integrates UPF, 3). UPF connects to the VPN behind the CE router.

Option 1 could work since all legacy MPLS or Segment Routing [RFC8402] based solution are available for both VPN and transport slicing at the UPF integrated PE router. However it is hard to expect it in multi-vendor deployment case, where the PE routers providing vendor is different from the vendor who provides UPFs, for example.

Option 2 and 3 are expected as IP domain separation, but it is hard to see that it is able to indicate transport slice in addition to the IP domain. Other L2 and tunneling solutions should be same with those options.

The expected evaluation points from this aspect should be whether the candidate protocols can contain forwarding information associated to the assigned IP domain and transport slice for all possible deployment cases.

5.8. Reliable Communication support

As Section 4.2 described, more than two UP paths are required for a QoS flow of a PDU session between the anchor UPF and gNB. Those UP paths are to convey redundant duplicated packets.

To support reliable communication with above requirements, UPF and gNB must replicate the sending UP packets and eliminate the received duplicated UP packets. Not to mention that UP protocol should be able to make sure that the paths are not in fate sharing, the UP packet must have sequence number to indicate duplicate packets per QoS flow basis.

The expected evaluation points from this aspect should be whether the candidate protocols can indicate packet sequence and diversified paths in the context of QoS flow, not in UP tunnel context. The packet sequence information should be transparent through I-UPF(s) exist in the middle of the path even in case that the UP tunnels are terminated at the I-UPF(s).

6. Conclusion

We analyzed the 3GPP specifications of the 5G architecture in terms of user plane and the current protocol adopted to the user plane. After the analysis work, we believe that the results described in this document shows that we reach at certain level of understanding on the 5G systems and ready to provide our inputs to 3GPP.

We clarified GTP-U through the analysis and listed observed characteristics in Section 3.6. We also clarified the architectural requirements for UP protocol described in Section 4.2.

Our conclusion here is that it is hopefull if the evaluation aspects described in Section 5 help for the study progress. It is worth to study possible candidate UP protocols for the 5G system including current one based from the aspects.

7. Security Consideration

TBD

8. Acknowledgement

The authors would like to thank Tom Herbert, Takashi Ito, John Leddy, Pablo Camarillo, Daisuke Yokota, Satoshi Watanabe, Koji Tsubouchi and Miya Kohno for their detailed reviews, comments, and contributions.

A special thank you goes to Arashmid Akhavain for his technical review and feedback.

Lastly, the authors would like to thank 3GPP CT WG4 folks for their review and feedback.

9. Informative References

- [C4-185491-3GPP]
3rd Generation Partnership Project (3GPP), "LS OUT on User Plane Analysis", July 2018,
<http://www.3gpp.org/ftp/tsg_ct/WG4_protocollars_ex-CN4/TSGCT4_85bis_Sophia_Antipolis/Docs/C4-185491.zip>.
- [CP-173160-3GPP]
3rd Generation Partnership Project (3GPP), "New Study Item on User Plane Protocol in 5GC", December 2017,
<http://www.3gpp.org/ftp/tsg_ct/TSG_CT/TSGC_78_Lisbon/Docs/CP-173160.zip>.
- [CP-180116-3GPP]
3rd Generation Partnership Project (3GPP), "LS on user plane protocol study", March 2018,
<http://www.3gpp.org/ftp/tsg_ct/TSG_CT/TSGC_79_Chennai/Docs/CP-180116.zip>.
- [IAB-Statement]
Internet Architecture Board (IAB), "IAB Statement on IPv6", November 2016,
<<https://www.iab.org/2016/11/07/iab-statement-on-ipv6/>>.
- [IEEE-Std-802.1AS]
Institute of Electrical and Electronics Engineers (IEEE), "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", March 2011,
<<https://www.ieee802.org/1/pages/802.1as.html>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.

- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <<https://www.rfc-editor.org/info/rfc6935>>.
- [RFC7157] Troan, O., Ed., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, DOI 10.17487/RFC7157, March 2014, <<https://www.rfc-editor.org/info/rfc7157>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8402] Filts, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [TR.29.891-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TR 29.891 (V15.0.0): 5G System Phase 1, CT WG4 Aspects", December 2017, <http://www.3gpp.org/FTP/Specs/2017-12/Rel-15/29_series/29891-f00.zip>.
- [TS.22.261-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 22.261 (V15.7.0): Service requirements for 5G system stage 1", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/22_series/22261-f70.zip>.
- [TS.23.060-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 23.060 (V15.3.0): General Packet Radio Service (GPRS); Service description; Stage 2", June 2018, <http://www.3gpp.org/ftp/Specs/archive/23_series/23.060/23060-f30.zip>.

[TS.23.501-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V16.2.0): System Architecture for 5G System; Stage 2", September 2019, <http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip>.

[TS.23.502-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.502 (V15.4.0): Procedures for 5G System; Stage 2", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23_series/23502-f40.zip>.

[TS.23.503-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.503 (V15.4.0): Policy and Charging Control System for 5G Framework; Stage 2", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23_series/23503-f40.zip>.

[TS.28.530-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.530 (V15.1.0): Management and orchestration of networks and network slicing; Concepts, use cases and requirements (work in progress)", December 2018, <http://ftp.3gpp.org//Specs/archive/28_series/28.530/28530-f10.zip>.

[TS.28.531-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.531 (V15.1.0): Management and orchestration of networks and network slicing; Provisioning; Stage 1 (Release 15)", December 2018, <http://ftp.3gpp.org//Specs/archive/28_series/28.531/28531-f10.zip>.

[TS.28.532-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.532 (V15.1.0): Management and orchestration of networks and network slicing; Provisioning; Stage 2 and stage 3 (Release 15)", December 2018, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.532/28532-f10.zip>.

[TS.28.533-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.533 (V15.1.0): Management and orchestration of networks and network slicing; Management and orchestration architecture (Release 15)", December 2018, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.533/28533-f10.zip>.

[TS.29.244-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.244 (V15.1.0): Interface between the Control Plane and the User Plane Nodes; Stage 3", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/29_series/29244-f40.zip>.

[TS.29.274-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.274 (V15.4.0): 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunneling Protocol for Control plane (GTPv2-C); Stage 3", June 2018, <http://www.3gpp.org/ftp//Specs/archive/29_series/29.274/29274-f40.zip>.

[TS.29.281-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.281 (V16.1.0): GPRS Tunneling Protocol User Plane (GTPv1-U)", September 2020, <https://www.3gpp.org/ftp//Specs/archive/29_series/29.281/29281-g10.zip>.

[TS.29.510-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.510 (V15.2.0): 5G System; Network Function Repository Services; Stage 3", December 2018, <http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29_series/29510-f20.zip>.

[TS.29.561-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.561 (V15.1.0): 5G System; Interworking between 5G Network and external Data Networks; Stage 3", September 2018, <http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29_series/29561-f10.zip>.

[TS.38.300-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.300 (v15.4.0): NR and NG-RAN Overall Description; Stage 2", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38_series/38300-f40.zip>.

[TS.38.401-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.401 (v15.4.0): NG-RAN; Architecture Description", December 2018, <http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38_series/38401-f40.zip>.

[TS.38.415-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.415 (v16.2.0): NG-RAN; PDU Session User Plane protocol", October 2020, <https://www.3gpp.org/ftp//Specs/archive/38_series/38.415/38415-g20.zip>.

Authors' Addresses

Shunsuke Homma
NTT

Email: homma.shunsuke@lab.ntt.co.jp

Takuya Miyasaka
KDDI Research

Email: ta-miyasaka@kddi-research.jp

Satoru Matsushima
SoftBank

Email: satoru.matsushima@g.softbank.co.jp

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 13, 2019

S. Matsushima
SoftBank
L. Bertz
Sprint
M. Liebsch
NEC
S. Gundavelli
Cisco
D. Moses
Intel Corporation
C. Perkins
Futurewei
April 11, 2019

YANG for Protocol for Forwarding Policy Configuration (FPC)
draft-ietf-dmm-fpc-yang-00

Abstract

This document provides YANG modules to exhibit a data model for the information models defining Forwarding Policy Configuration (FPC) to manage the separation of data-plane and control-plane. FPC defines a flexible mobility management system using FPC agent and FPC client functions. A FPC agent provides an abstract interface to the data-plane. The FPC client configures data-plane nodes by using the functions and abstractions provided by the FPC agent for the data-plane nodes.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 13, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. YANG Data Model for the FPC protocol	3
3.1. FPC YANG Model	5
3.2. FPC YANG Settings and Extensions Model	27
3.3. PMIP QoS Model	39
3.4. Traffic Selectors YANG Model	46
3.5. RFC 5777 Classifier YANG Model	54
4. FPC YANG Tree Structure	62
5. Work Team Participants	80
6. References	80
6.1. Normative References	80
6.2. Informative References	81
Authors' Addresses	81

1. Introduction

This document provides YANG modules to exhibit a data model for the information models defining Forwarding Policy Configuration (FPC) [I-D.ietf-dmm-fpc-cpdp] to manage the separation of data-plane and control-plane. FPC defines a flexible mobility management system using FPC agent and FPC client functions. A FPC agent provides an abstract interface to the data-plane. The FPC client configures data-plane nodes by using the functions and abstractions provided by the FPC agent for the data-plane nodes.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. The

following FPC-specific terms used in this document are defined in [I-D.ietf-dmm-fpc-cpdp].

- o Domain
- o DPN
- o FPC Agent
- o FPC Client
- o Mobility Context
- o Monitor
- o Policy
- o Template
- o Tenant
- o Topology

3. YANG Data Model for the FPC protocol

This section provides a type mapping for FPC structures in YANG. When being mapped to a specific information such as YANG the data type MAY change.

Action and Descriptor Templates are mapped as choices. This was done to ensure no duplication of Types and avoid use of identityref for typing.

Policy Expressions are provided as default values. NOTE that a static value CANNOT be supported in YANG.

Mapping of templates to YANG are performed as follows:

Value is defined as a choice statement for extensibility and therefore a type value is not necessary to discriminated types.

Generic attributes are distinguished by the "Settings" type and holds ANY value. It is in a data node under configurations.

The CONFIGURE and CONFIGURE-RESULT-NOTIFICATION use the yang-patch-status which is a container for edits. This was done to maximize YANG reuse.

In the configure rpc, operation-id is mapped to patch-id and in an edit the edit-type is mapped to operation.

The Result-Status attribute is mapped to the 'ok' (empty leaf) or errors structure.

The Policy-Status is mapped to entity-state to reduce YANG size.

Five modules are defined:

- o ietf-dmm-fpc (fpc) - Defines the base model and messages for FPC that are meant to be static in FPC.
- o ietf-dmm-fpc-settingsext - A FPC module that defines the information model elements that are likely to be extended in FPC.
- o ietf-pmip-qos (pmip-qos) - Defines proxy mobile IPv6 QoS parameters per [RFC7222]
- o ietf-trafficselectors-types (traffic-selectors) - Defines Traffic Selectors per [RFC6088]
- o ietf-diam-trafficclassifier (diamclassifier) - Defines the Classifier per [RFC5777]

All modules defined in this specification make use of (import) ietf-inet-types as defined in [RFC6991].

ietf-dmm-fpc-settingsext and ietf-diam-trafficclassifier make use of (imports) ietf-yang-types as defined in [RFC6991].

ietf-dmm-fpc imports the restconf (ietf-restconf) [RFC8040] and yang patch (ietf-yang-patch) [RFC8072] modules.

ietf-pmip-qos and ietf-dmm-fpc-settings import the trafficselector from the ietf-traffic-selector-types module.

ietf-dmm-fpc-settings also imports the qosattribute (ietf-pmip-qos) and classifier (ietf-diam-trafficclassifier).

ietf-dmm-fpc-settingsext groups various settings, actions and descriptors and is used by the fpc module (ietf-dmm-fpc).

The following groupings are intended for reuse (import) by other modules.

- o qosoption (ietf-qos-pmip module)
- o qosattribute (ietf-qos-pmip module)
- o qosoption (ietf-qos-pmip module)
- o Allocation-Retention-Priority-Value (ietf-qos-pmip module)
- o trafficselector (ietf-traffic-selector-types)
- o classifier (ietf-diam-trafficclassifier)
- o packet-filter (ietf-dmm-fpc-settingsext)
- o instructions (ietf-dmm-fpc-settingsext)
- o fpc-descriptor-value (ietf-dmm-fpc-settingsext)
- o fpc-action-value (ietf-dmm-fpc-settingsext)

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

A DPN conformant to NMDA MAY only have policies, installed policies, topology, domains and mobility session information that has been assigned to it in its intended and operational datastores.

ServiceGroups are not expected to appear in operational datastores of DPNs as they remain in and are used by FPC Agents and Clients. They MAY be operationally present in DNS when using the Dynamic Delegation and Discovery System (DDDS) as defined in [RFC3958] or the operational datastore of systems that provide equivalent functionality.

3.1. FPC YANG Model

This module defines the information model and protocol elements specified in this document.

This module references [RFC6991], [RFC8040] and the fpc-settingsext module defined in this document.

```
<CODE BEGINS> file "ietf-dmm-fpc@2018-05-17.yang"
module ietf-dmm-fpc {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpc";
  prefix fpc;

  import ietf-inet-types { prefix inet;
    revision-date 2013-07-15; }
  import ietf-dmm-fpc-settingsext { prefix fpcbase;
    revision-date 2018-05-17; }
  import ietf-diam-trafficclassifier { prefix rfc5777;
    revision-date 2018-05-17; }
  import ietf-restconf { prefix rc;
    revision-date 2017-01-26; }
  import ietf-yang-patch { prefix ypatch;
    revision-date 2017-02-22; }

  organization "IETF Distributed Mobility Management (DMM)
    Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    WG Chair: Dapeng Liu
              <mailto:maxpassion@gmail.com>
```

WG Chair: Jouni Korhonen
<mailto:jouni.nospam@gmail.com>

Editor: Satoru Matsushima
<mailto:satoru.matsushima@g.softbank.co.jp>

Editor: Lyle Bertz
<mailto:lylebe551144@gmail.com>;

description

"This module contains YANG definition for
Forwarding Policy Configuration Protocol (FPCP).

Copyright (c) 2016 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal
Provisions Relating to IETF Documents
(<http://trustee.ietf.org/license-info>) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with
respect to this document. Code Components extracted from this
document must include Simplified BSD License text as described
in Section 4.e of the Trust Legal Provisions and are provided
without warranty as described in the Simplified BSD License.";

```
revision 2018-05-17 {  
  description "Initial Revision."  
  reference "draft-ietf-dmm-fpc-cdp-10";  
}
```

```
//General Structures
```

```
grouping templatedef {  
  leaf extensible {  
    type boolean;  
    description "Indicates if the template is extensible";  
  }  
  leaf-list static-attributes {  
    type string;  
    description "Attribute (Name) whose value cannot  
      change";  
  }  
  leaf-list mandatory-attributes {  
    type string;  
    description "Attribute (Name) of optional attributes  
      that MUST be present in instances of this template.";  
  }  
  leaf entity-state {
```



```
    type enumeration {
      enum initial {
        description "Initial Configuration";
      }
      enum partially-configured {
        description "Partial Configuration";
      }
      enum configured {
        description "Configured";
      }
      enum active {
        description "Active";
      }
    }
    default initial;
    description "Entity State";
  }
  leaf version {
    type uint32;
    description "Template Version";
  }
  description "Teemplate Definition";
}
typedef fpc-identity {
  type union {
    type uint32;
    type instance-identifier;
    type string;
  }
  description "FPC Identity";
}
grouping index {
  leaf index {
    type uint16;
    description "Index";
  }
  description "Index Value";
}

// Policy Structures
grouping descriptor-template-key {
  leaf descriptor-template-key {
    type fpc:fpc-identity;
    mandatory true;
    description "Descriptor Key";
  }
  description "Descriptor-Template Key";
}
```

```
    grouping action-template-key {
      leaf action-template-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Action Key";
      }
      description "Action-Template Key";
    }
    grouping rule-template-key {
      leaf rule-template-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Rule Identifier";
      }
      description "Rule Key";
    }
    grouping policy-template-key {
      leaf policy-template-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Rule Identifier";
      }
      description "Rule Key";
    }
  }

  grouping fpc-setting-value {
    anydata setting;
    description "FPC Setting Value";
  }
  // Configuration / Settings
  grouping policy-configuration-choice {
    choice policy-configuration-value {
      case descriptor-value {
        uses fpcbase:fpc-descriptor-value;
        description "Descriptor Value";
      }
      case action-value {
        uses fpcbase:fpc-action-value;
        description "Action Value";
      }
      case setting-value {
        uses fpc:fpc-setting-value;
        description "Setting";
      }
      description "Policy Attributes";
    }
    description "Policy Configuration Value Choice";
  }
}
```

```
grouping policy-configuration {
  list policy-configuration {
    key index;
    uses fpc:index;
    uses fpc:policy-configuration-choice;
    description "Policy Configuration";
  }
  description "Policy Configuration Value";
}
grouping ref-configuration {
  uses fpc:policy-template-key;
  uses fpc:policy-configuration;
  uses fpc:templatedef;
  description "Policy-Configuration Entry";
}

// FPC Policy
grouping policy-information-model {
  list action-template {
    key action-template-key;
    uses fpc:action-template-key;
    uses fpcbase:fpc-action-value;
    uses fpc:templatedef;
    description "Action Template";
  }
  list descriptor-template {
    key descriptor-template-key;
    uses fpc:descriptor-template-key;
    uses fpcbase:fpc-descriptor-value;
    uses fpc:templatedef;
    description "Descriptor Template";
  }
  list rule-template {
    key rule-template-key;
    uses fpc:rule-template-key;
    leaf descriptor-match-type {
      type enumeration {
        enum or {
          value 0;
          description "OR logic";
        }
        enum and {
          value 1;
          description "AND logic";
        }
      }
    }
    mandatory true;
    description "Type of Match (OR or AND) applied";
  }
}
```

```
        to the descriptor-configurations";
    }
    list descriptor-configuration {
        key "descriptor-template-key";
        uses fpc:descriptor-template-key;
        leaf direction {
            type rfc5777:direction-type;
            description "Direction";
        }
        list attribute-expression {
            key index;
            uses fpc:index;
            uses fpcbase:fpc-descriptor-value;
            description "Descriptor Attributes";
        }
        uses fpc:fpc-setting-value;
        description "A set of Descriptor references";
    }
    list action-configuration {
        key "action-order";
        leaf action-order {
            type uint32;
            mandatory true;
            description "Action Execution Order";
        }
        uses fpc:action-template-key;
        list attribute-expression {
            key index;
            uses fpc:index;
            uses fpcbase:fpc-action-value;
            description "Action Attributes";
        }
        uses fpc:fpc-setting-value;
        description "A set of Action references";
    }
    uses fpc:templatedef;
    list rule-configuration {
        key index;
        uses fpc:index;
        uses fpc:policy-configuration-choice;
        description "Rule Configuration";
    }
    description "Rule Template";
}
list policy-template {
    key policy-template-key;
    uses fpc:policy-template-key;
    list rule-template {
```

```
        key "precedence";
        unique "rule-template-key";
        leaf precedence {
            type uint32;
            mandatory true;
            description "Rule Precedence";
        }
        uses fpc:rule-template-key;
        description "Rule Entry";
    }
    uses fpc:templatedef;
    uses fpc:policy-configuration;
    description "Policy Template";
}
description "FPC Policy Structures";
}

// Topology Information Model
identity role {
    description "Role";
}
grouping dpn-key {
    leaf dpn-key {
        type fpc:fpc-identity;
        description "DPN Key";
    }
    description "DPN Key";
}
grouping role-key {
    leaf role-key {
        type identityref {
            base "fpc:role";
        }
        mandatory true;
        description "Access Technology Role";
    }
    description "Access Technology Role key";
}
grouping interface-key {
    leaf interface-key {
        type fpc:fpc-identity;
        mandatory true;
        description "interface identifier";
    }
    description "Interface Identifier key";
}
identity interface-protocols {
    description "Protocol supported by the interface";
}
```

```
    }
    identity features {
        description "Protocol features";
    }

// Mobility Context
grouping mobility-context {
    leaf mobility-context-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Mobility Context Key";
    }
    leaf-list delegating-ip-prefix {
        type inet:ip-prefix;
        description "IP Prefix";
    }
    leaf parent-context {
        type fpc:fpc-identity;
        description "Parent Mobility Context";
    }
    leaf-list child-context {
        type fpc:fpc-identity;
        description "Child Mobility Context";
    }
}
container mobile-node {
    leaf-list ip-address {
        type inet:ip-address;
        description "IP Address";
    }
    leaf imsi {
        type fpcbase:imsi-type;
        description "IMSI";
    }
    list mn-policy-configuration {
        key policy-template-key;
        uses fpc:ref-configuration;
        description "MN Policy Configuration";
    }
    description "Mobile Node";
}
container domain {
    leaf domain-key {
        type fpc:fpc-identity;
        description "Domain Key";
    }
    list domain-policy-settings {
        key policy-template-key;
        uses fpc:ref-configuration;
    }
}
```

```
        description "MN Policy Configuration";
    }
    description "Domain";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    list dpn-policy-configuration {
        key policy-template-key;
        uses fpc:ref-configuration;
        description "DPN Policy Configuration";
    }
    leaf role {
        type identityref {
            base "fpc:role";
        }
        description "Role";
    }
    list service-data-flow {
        key identifier;
        leaf identifier {
            type uint32;
            description "Generic Identifier";
        }
        leaf service-group-key {
            type fpc:fpc-identity;
            description "Service Group Key";
        }
        list interface {
            key interface-key;
            uses fpc:interface-key;
            description "interface assigned";
        }
        list service-data-flow-policy-configuration {
            key policy-template-key;
            uses fpc:ref-configuration;
            description "Flow Policy Configuration";
        }
        description "Service Dataflow";
    }
    description "DPN";
}
description "Mobility Context";
}

// Events, Probes & Notifications
identity event-type {
    description "Base Event Type";
}
```

```
}
typedef event-type-id {
  type uint32;
  description "Event ID Type";
}
grouping monitor-key {
  leaf monitor-key {
    type fpc:fpc-identity;
    mandatory true;
    description "Monitor Key";
  }
  description "Monitor Id";
}
grouping monitor-config {
  uses fpc:templatedef;
  uses fpc:monitor-key;
  leaf target {
    type string;
    description "target";
  }
  leaf deferrable {
    type boolean;
    description "Indicates reports related to this
      config can be delayed.";
  }
  choice configuration {
    mandatory true;
    leaf period {
      type uint32;
      description "Period";
    }
    case threshold-config {
      leaf low {
        type uint32;
        description "low threshold";
      }
      leaf hi {
        type uint32;
        description "high threshold";
      }
      description "Threshold Config Case";
    }
    leaf schedule {
      type uint32;
      description "Reporting Time";
    }
  }
  leaf-list event-identities {
    type identityref {
```



```
        base "fpc:event-type";
    }
    description "Event Identities";
}
leaf-list event-ids {
    type uint32;
    description "Event IDs";
}
description "Event Config Value";
}
description "Monitor Configuration";
}

// Top Level Structures
list tenant {
    key "tenant-key";
    leaf tenant-key {
        type fpc:fpc-identity;
        description "Tenant Key";
    }
}
container topology-information-model {
    config false;
    list service-group {
        key "service-group-key role-key";
        leaf service-group-key {
            type fpc:fpc-identity;
            mandatory true;
            description "Service Group Key";
        }
        leaf service-group-name {
            type string;
            description "Service Group Name";
        }
    }
    uses fpc:role-key;
    leaf role-name {
        type string;
        mandatory true;
        description "Role Name";
    }
}
leaf-list protocol {
    type identityref {
        base "interface-protocols";
    }
    min-elements 1;
    description "Supported protocols";
}
leaf-list feature {
    type identityref {
```

```
        base "interface-protocols";
    }
    description "Supported features";
}
list service-group-configuration {
    key index;
    uses fpc:index;
    uses fpc:policy-configuration-choice;
    description "Settings";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    min-elements 1;
    list referenced-interface {
        key interface-key;
        uses fpc:interface-key;
        leaf-list peer-service-group-key {
            type fpc:fpc-identity;
            description "Peer Service Group";
        }
        description "Referenced Interface";
    }
    description "DPN";
}
description "Service Group";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    leaf dpn-name {
        type string;
        description "DPN name";
    }
    leaf dpn-resource-mapping-reference {
        type string;
        description "Reference to underlying DPN resource(s)";
    }
    leaf domain-key {
        type fpc:fpc-identity;
        description "Domains";
    }
    leaf-list service-group-key {
        type fpc:fpc-identity;
        description "Service Group";
    }
    list interface {
        key "interface-key";
```

```
    uses fpc:interface-key;
    leaf interface-name {
        type string;
        description "Service Endpoint Interface Name";
    }
    leaf role {
        type identityref {
            base "fpc:role";
        }
        description "Roles supported";
    }
    leaf-list protocol {
        type identityref {
            base "interface-protocols";
        }
        description "Supported protocols";
    }
    list interface-configuration {
        key index;
        uses fpc:index;
        uses fpc:policy-configuration-choice;
        description "Interface settings";
    }
    description "DPN interfaces";
}
list dpn-policy-configuration {
    key policy-template-key;
    uses fpc:ref-configuration;
    description "DPN Policy Configuration";
}
description "Set of DPNs";
}
list domain {
    key domain-key;
    leaf domain-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Domain Key";
    }
    leaf domain-name {
        type string;
        description "Domain displayname";
    }
    list domain-policy-configuration {
        key policy-template-key;
        uses fpc:ref-configuration;
        description "Domain Configuration";
    }
}
```

```
    description "List of Domains";
  }
  container dpn-checkpoint {
    uses fpc:basename-info;
    description "DPN Checkpoint information";
  }
  container service-group-checkpoint {
    uses fpc:basename-info;
    description "Service Group Checkpoint information";
  }
  container domain-checkpoint {
    uses fpc:basename-info;
    description "Domain Checkpoint information";
  }
  description "FPC Topology grouping";
}
container policy-information-model {
  config false;
  uses fpc:policy-information-model;
  uses fpc:basename-info;
  description "Policy";
}
list mobility-context {
  key "mobility-context-key";
  config false;
  uses fpc:mobility-context;
  description "Mobility Context";
}
list monitor {
  key monitor-key;
  config false;
  uses fpc:monitor-config;
  description "Monitor";
}
description "Tenant";
}

typedef agent-identifier {
  type fpc:fpc-identity;
  description "Agent Identifier";
}
typedef client-identifier {
  type fpc:fpc-identity;
  description "Client Identifier";
}
grouping basename-info {
  leaf basename {
    type fpc:fpc-identity;
  }
}
```

```
        description "Rules Basename";
    }
    leaf base-checkpoint {
        type string;
        description "Checkpoint";
    }
    description "Basename Information";
}

// RPCs
grouping client-id {
    leaf client-id {
        type fpc:client-identifier;
        mandatory true;
        description "Client Id";
    }
    description "Client Identifier";
}
grouping execution-delay {
    leaf execution-delay {
        type uint32;
        description "Execution Delay (ms)";
    }
    description "Execution Delay";
}
typedef ref-scope {
    type enumeration {
        enum none {
            value 0;
            description "no references";
        }
        enum op {
            value 1;
            description "All references are intra-operation";
        }
        enum bundle {
            value 2;
            description "All references in exist in bundle";
        }
        enum storage {
            value 3;
            description "One or more references exist in storage.";
        }
        enum unknown {
            value 4;
            description "The location of the references are unknown.";
        }
    }
}
```

```
    description "Search scope for references in the operation.";
  }
  rpc configure {
    description "Configure RPC";
    input {
      uses client-id;
      uses execution-delay;
      uses ypatch:yang-patch;
    }
    output {
      uses ypatch:yang-patch-status;
    }
  }
  augment "/configure/input/yang-patch/edit" {
    leaf reference-scope {
      type fpc:ref-scope;
      description "Reference Scope";
    }
    uses fpcbase:instructions;
    description "yang-patch edit augments for configure rpc";
  }
  grouping subsequent-edits {
    list subsequent-edit {
      key edit-id;
      ordered-by user;

      description "Edit list";

      leaf edit-id {
        type string;
        description "Arbitrary string index for the edit.";
      }

      leaf operation {
        type enumeration {
          enum create {
            description "Create";
          }
          enum delete {
            description "Delete";
          }
          enum insert {
            description "Insert";
          }
          enum merge {
            description "Merge";
          }
          enum move {
```

```
        description "Move";
    }
    enum replace {
        description "Replace";
    }
    enum remove {
        description
            "Delete the target node if it currently exists.";
    }
}
mandatory true;
description
    "The datastore operation requested";
}

leaf target {
    type ypatch:target-resource-offset;
    mandatory true;
    description
        "Identifies the target data node";
}

leaf point {
    when "../operation = 'insert' or ../operation = 'move'"
    + "and ../where = 'before' or ../where = 'after'" {
        description
            "This leaf only applies for 'insert' or 'move'
            operations, before or after an existing entry.";
    }
    type ypatch:target-resource-offset;
    description
        "The absolute URL path for the data node";
}

leaf where {
    when "../operation = 'insert' or ../operation = 'move'" {
        description
            "This leaf only applies for 'insert' or 'move'
            operations.";
    }
    type enumeration {
        enum before {
            description
                "Insert or move a data node before.";
        }
        enum after {
            description
                "Insert or move a data node after.";
        }
    }
}
```

```
    }
    enum first {
        description
            "Insert or move a data node so it becomes ordered
            as the first entry.";
    }
    enum last {
        description
            "Insert or move a data node so it becomes ordered
            as the last entry.";
    }
}
default last;
description
    "Identifies where a data resource will be inserted
    or moved.";
}

anydata value {
    when "../operation = 'create' "
        + "or ../operation = 'merge' "
        + "or ../operation = 'replace' "
        + "or ../operation = 'insert' " {
        description
            "The anydata 'value' is only used for 'create',
            'merge', 'replace', and 'insert' operations.";
    }
    description
        "Value used for this edit operation.";
}
}
description "Subsequent Edits";
}
augment "/configure/output yang-patch-status/edit-status/edit/"
    + "edit-status-choice/ok" {
    leaf notify-follows {
        type boolean;
        description "Notify Follows Indication";
    }
    uses fpc:subsequent-edits;
    description "Configure output augments";
}

grouping op-header {
    uses client-id;
    uses execution-delay;
    leaf operation-id {
        type uint64;
```



```
        mandatory true;
        description "Operation Identifier";
    }
    description "Common Operation header";
}
grouping monitor-response {
    leaf operation-id {
        type uint64;
        mandatory true;
        description "Operation Identifier";
    }
    choice edit-status-choice {
        description
            "A choice between different types of status
            responses for each 'edit' entry.";
        leaf ok {
            type empty;
            description
                "This 'edit' entry was invoked without any
                errors detected by the server associated
                with this edit.";
        }
        case errors {
            uses rc:errors;
            description
                "The server detected errors associated with the
                edit identified by the same 'edit-id' value.";
        }
    }
    description "Monitor Response";
}

// Common RPCs
rpc register_monitor {
    description "Used to register monitoring of parameters/events";
    input {
        uses fpc:op-header;
        list monitor {
            key monitor-key;
            uses fpc:monitor-config;
            description "Monitor Configuration";
        }
    }
    output {
        uses fpc:monitor-response;
    }
}
rpc deregister_monitor {
```

```
description "Used to de-register monitoring of
  parameters/events";
input {
  uses fpc:op-header;
  list monitor {
    key monitor-key;
    uses fpc:monitor-key;
    min-elements 1;
    leaf send_data {
      type boolean;
      description "Indicates if NOTIFY with final data
        is desired upon deregistration";
    }
    description "Monitor Identifier";
  }
}
output {
  uses fpc:monitor-response;
}
}

rpc probe {
  description "Probe the status of a registered monitor";
  input {
    uses fpc:op-header;
    list monitor {
      key monitor-key;
      uses fpc:monitor-key;
      min-elements 1;
      description "Monitor";
    }
  }
  output {
    uses fpc:monitor-response;
  }
}

// Notification Messages & Structures
notification config-result-notification {
  uses ypatch:yang-patch-status;
  description "Configuration Result Notification";
}
augment "/config-result-notification" {
  uses fpc:subsequent-edits;
  description "config-result-notificatio augment";
}

identity notification-cause {
  description "Notification Cause";
}
```

```
}
identity subscribed-event-occurred {
  base "notification-cause";
  description "Subscribed Event Occurrence";
}
identity low-threshold-crossed {
  base "notification-cause";
  description "Subscribed Event Occurrence";
}
identity high-threshold-crossed {
  base "notification-cause";
  description "Subscribed Event Occurrence";
}
identity periodic-report {
  base "notification-cause";
  description "Periodic Report";
}
identity scheduled-report {
  base "notification-cause";
  description "Scheduled Report";
}
identity probe {
  base "notification-cause";
  description "Probe";
}
identity deregistration-final-value {
  base "notification-cause";
  description "Probe";
}
identity monitoring-suspension {
  base "notification-cause";
  description "Indicates monitoring suspension";
}
identity monitoring-resumption {
  base "notification-cause";
  description "Indicates that monitoring has resumed";
}
identity dpn-available {
  base "notification-cause";
  description "DPN Candidate Available";
}
identity dpn-unavailable {
  base "notification-cause";
  description "DPN Unavailable";
}
notification notify {
  leaf notification-id {
    type uint32;
  }
}
```

```
        description "Notification Identifier";
    }
    leaf timestamp {
        type uint32;
        description "timestamp";
    }
    list report {
        key monitor-key;
        uses fpc:monitor-key;
        min-elements 1;
        leaf trigger {
            type identityref {
                base "notification-cause";
            }
            description "Notification Cause";
        }
        choice value {
            case dpn-candidate-available {
                leaf node-id {
                    type inet:uri;
                    description "Topology URI";
                }
                list supported-interface-list {
                    key role-key;
                    uses fpc:role-key;
                    description "Support Intefaces";
                }
                description "DPN Candidate Information";
            }
            case dpn-unavailable {
                leaf dpn-id {
                    type fpc:fpc-identity;
                    description "DPN Identifier for DPN Unavailable";
                }
                description "DPN Unavailable";
            }
            anydata report-value {
                description "Any non integer report";
            }
            description "Report Value";
        }
        description "Report";
    }
    description "Notify Message";
}
<CODE ENDS>
```

3.2. FPC YANG Settings and Extensions Model

This module defines the base data elements in FPC that are likely to be extended.

This module references [RFC6991], ietf-trafficselector-types and ietf-pmip-qos modules.

```
<CODE BEGINS> file "ietf-dmm-fpc-settingsext@2018-05-17.yang"
module ietf-dmm-fpc-settingsext {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpc-settingsext";
  prefix fpabase;

  import ietf-inet-types { prefix inet;
    revision-date 2013-07-15; }
  import ietf-trafficselector-types { prefix traffic-selectors;
    revision-date 2018-05-17; }
  import ietf-yang-types { prefix ytypes;
    revision-date 2013-07-15; }
  import ietf-pmip-qos { prefix pmipqos;
    revision-date 2018-05-17; }
  import ietf-diam-trafficclassifier { prefix rfc5777;
    revision-date 2018-05-17; }

  organization "IETF Distributed Mobility Management (DMM)
    Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    WG Chair: Dapeng Liu
               <mailto:maxpassion@gmail.com>

    WG Chair: Sri Gundavelli
               <mailto:sgundave@cisco.com>

    Editor:   Satoru Matsushima
               <mailto:satoru.matsushima@g.softbank.co.jp>

    Editor:   Lyle Bertz
               <mailto:lylebe551144@gmail.com>";

  description
    "This module contains YANG definition for
    Forwarding Policy Configuration Protocol (FPCP)."
```

It contains Settings definitions as well as Descriptor and Action extensions.

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.";

```
revision 2018-05-17 {
  description "Initial Revision.";
  reference "draft-ietf-dmm-fpc-cpdp-10";
}

//Tunnel Information
identity tunnel-type {
  description "Tunnel Type";
}
identity grev1 {
  base "fpcbase:tunnel-type";
  description "GRE v1";
}
identity grev2 {
  base "fpcbase:tunnel-type";
  description "GRE v2";
}
identity ipinip {
  base "fpcbase:tunnel-type";
  description "IP in IP";
}
identity gtpv1 {
  base "fpcbase:tunnel-type";
  description "GTP version 1 Tunnel";
}
identity gtpv2 {
  base "fpcbase:tunnel-type";
  description "GTP version 2 Tunnel";
}

grouping tunnel-value {
  container tunnel-info {
```

```
leaf tunnel-local-address {
    type inet:ip-address;
    description "local tunnel address";
}
leaf tunnel-remote-address {
    type inet:ip-address;
    description "remote tunnel address";
}
leaf mtu-size {
    type uint32;
    description "MTU size";
}
leaf tunnel {
    type identityref {
        base "fpcbase:tunnel-type";
    }
    description "tunnel type";
}
leaf payload-type {
    type enumeration {
        enum ipv4 {
            value 0;
            description "IPv4";
        }
        enum ipv6 {
            value 1;
            description "IPv6";
        }
        enum dual {
            value 2;
            description "IPv4 and IPv6";
        }
    }
    description "Payload Type";
}
leaf gre-key {
    type uint32;
    description "GRE_KEY";
}
container gtp-tunnel-info {
    leaf local-tunnel-identifier {
        type uint32;
        description "Tunnel Endpoint Identifier (TEID)";
    }
    leaf remote-tunnel-identifier {
        type uint32;
        description "Tunnel Endpoint Identifier (TEID)";
    }
}
```

```

        leaf sequence-numbers-enabled {
            type boolean;
            description "Sequence No. Enabled";
        }
        description "GTP Tunnel Information";
    }
    leaf ebi {
        type fpcbase:ebi-type;
        description "EPS Bearier Identifier";
    }
    leaf lbi {
        type fpcbase:ebi-type;
        description "Linked Bearier Identifier";
    }
    description "Tunnel Information";
}
description "Tunnel Value";
}

////////////////////////////////////
// DESCRIPTOR DEFINITIONS

// From 3GPP TS 24.008 version 13.5.0 Release 13
typedef packet-filter-direction {
    type enumeration {
        enum preRel7Tft {
            value 0;
            description "Pre-Release 7 TFT";
        }
        enum uplink {
            value 1;
            description "uplink";
        }
        enum downlink {
            value 2;
            description "downlink";
        }
        enum bidirectional {
            value 3;
            description "bi-direcitonal";
        }
    }
    description "Packet Filter Direction";
}
typedef component-type-id {
    type uint8 {
        range "16 | 17 | 32 | 33 | 35 | 48 | 64 | 65 | "
        + " 80 | 81 | 96 | 112 | 128";
    }
}

```



```
    }
    description "Specifies the Component Type";
}
grouping packet-filter {
  leaf direction {
    type fpcbase:packet-filter-direction;
    description "Filter Direction";
  }
  leaf identifier {
    type uint8 {
      range "1..15";
    }
    description "Filter Identifier";
  }
  leaf evaluation-precedence {
    type uint8;
    description "Evaluation Precedence";
  }
  list contents {
    key component-type-identifier;
    description "Filter Contents";
    leaf component-type-identifier {
      type fpcbase:component-type-id;
      description "Component Type";
    }
  }
  choice value {
    leaf ipv4-local {
      type inet:ipv4-address;
      description "IPv4 Local Address";
    }
    leaf ipv6-prefix-local {
      type inet:ipv6-prefix;
      description "IPv6 Local Prefix";
    }
    leaf ipv4-ipv6-remote {
      type inet:ip-address;
      description "Ipv4 Ipv6 remote address";
    }
    leaf ipv6-prefix-remote {
      type inet:ipv6-prefix;
      description "IPv6 Remote Prefix";
    }
    leaf next-header {
      type uint8;
      description "Next Header";
    }
    leaf local-port {
      type inet:port-number;
    }
  }
}
```

```
        description "Local Port";
    }
    case local-port-range {
        leaf local-port-lo {
            type inet:port-number;
            description "Local Port Min Value";
        }
        leaf local-port-hi {
            type inet:port-number;
            description "Local Port Max Value";
        }
    }
    leaf remote-port {
        type inet:port-number;
        description "Remote Port";
    }
    case remote-port-range {
        leaf remote-port-lo {
            type inet:port-number;
            description "Remote Por Min Value";
        }
        leaf remote-port-hi {
            type inet:port-number;
            description "Remote Port Max Value";
        }
    }
    leaf ipsec-index {
        type traffic-selectors:ipsec-spi;
        description "IPSec Index";
    }
    leaf traffic-class {
        type inet:dscp;
        description "Traffic Class";
    }
    case traffic-class-range {
        leaf traffic-class-lo {
            type inet:dscp;
            description "Traffic Class Min Value";
        }
        leaf traffic-class-hi {
            type inet:dscp;
            description "Traffic Class Max Value";
        }
    }
    leaf-list flow-label {
        type inet:ipv6-flow-label;
        description "Flow Label";
    }
}
```

```
        description "Component Value";
    }
}
description "Packet Filter";
}

grouping prefix-descriptor {
    leaf destination-ip {
        type inet:ip-prefix;
        description "Rule of destination IP";
    }
    leaf source-ip {
        type inet:ip-prefix;
        description "Rule of source IP";
    }
    description "Traffic descriptor based upon source/
        destination as IP prefixes";
}

grouping fpc-descriptor-value {
    choice descriptor-value {
        mandatory true;
        leaf all-traffic {
            type empty;
            description "admit any";
        }
        leaf no-traffic {
            type empty;
            description "deny any";
        }
    }
    case prefix-descriptor {
        uses fpcbase:prefix-descriptor;
        description "IP Prefix descriptor";
    }
    case pmip-selector {
        uses traffic-selectors:traffic-selector;
        description "PMIP Selector";
    }
    container rfc5777-classifier-template {
        uses rfc5777:classifier;
        description "RFC 5777 Classifier";
    }
    container packet-filter {
        uses fpcbase:packet-filter;
        description "Packet Filter";
    }
    case tunnel-info {
        uses fpcbase:tunnel-value;
    }
}
```

```
        description "Tunnel Descriptor (only
            considers source info)";
    }
    description "Descriptor Value";
}
description "FPC Descriptor Values";
}

// Next Hop Structures
typedef fpc-service-path-id {
    type uint32 {
        range "0..33554431";
    }
    description "SERVICE_PATH_ID";
}
typedef fpc-mpls-label {
    type uint32 {
        range "0..1048575";
    }
    description "MPLS label";
}
typedef segment-id {
    type string {
        length "16";
    }
    description "SR Segement Identifier";
}
grouping fpc-nexthop {
    choice next-hop-value {
        leaf ip-address {
            type inet:ip-address;
            description "IP Value";
        }
        leaf mac-address {
            type ytypes:mac-address;
            description "MAC Address Value";
        }
        leaf service-path {
            type fpcbase:fpc-service-path-id;
            description "Service Path Value";
        }
        leaf mpls-path {
            type fpcbase:fpc-mpls-label;
            description "MPLS Value";
        }
        leaf nsh {
            type string {
                length "16";
            }
        }
    }
}
```

```

        }
        description "Network Service Header";
    }
    leaf interface {
        type uint16;
        description "If (interface) Value";
    }
    leaf segment-identifier {
        type fpcbase:segment-id;
        description "Segment Id";
    }
    leaf-list mpls-label-stack {
        type fpcbase:fpc-mpls-label;
        description "MPLS Stack";
    }
    leaf-list mpls-sr-stack {
        type fpcbase:fpc-mpls-label;
        description "MPLS SR Stack";
    }
    leaf-list srv6-stack {
        type fpcbase:segment-id;
        description "Segment Id";
    }
    case tunnel-info {
        uses fpcbase:tunnel-value;
        description "Tunnel Descriptor (only
            considers source info)";
    }
    description "Value";
}
description "Nexthop Value";
}

////////////////////////////////////
// PMIP Integration           //
typedef pmip-commandset {
    type bits {
        bit assign-ip {
            position 0;
            description "Assign IP";
        }
        bit assign-dpn {
            position 1;
            description "Assign DPN";
        }
        bit session {
            position 2;
            description "Session Level";
        }
    }
}

```

```
    }
    bit uplink {
        position 3;
        description "Uplink";
    }
    bit downlink {
        position 4;
        description "Downlink";
    }
}
description "PMIP Instructions";
}
////////////////////////////////////
// 3GPP Integration          //

// Type Defs
typedef fpc-qos-class-identifier {
    type uint8 {
        range "1..9";
    }
    description "QoS Class Identifier (QCI)";
}
typedef ebi-type {
    type uint8 {
        range "0..15";
    }
    description "EUTRAN Bearere Identifier (EBI) Type";
}
typedef imsi-type {
    type uint64;
    description
        "International Mobile Subscriber Identity (IMSI)
        Value Type";
}
// Instructions
typedef threegpp-instr {
    type bits {
        bit assign-ip {
            position 0;
            description "Assign IP Address/Prefix";
        }
        bit assign-fteid-ip {
            position 1;
            description "Assign FTEID-IP";
        }
        bit assign-fteid-teid {
            position 2;
            description "Assign FTEID-TEID";
        }
    }
}
```

```
    }
    bit session {
        position 3;
        description "Commands apply to the Session Level";
    }
    bit uplink {
        position 4;
        description "Commands apply to the Uplink";
    }
    bit downlink {
        position 5;
        description "Commands apply to the Downlink";
    }
    bit assign-dpn {
        position 6;
        description "Assign DPN";
    }
}
description "Instruction Set for 3GPP R11";
}

////////////////////////////////////
// ACTION VALUE AUGMENTS
grouping fpc-action-value {
    choice action-value {
        mandatory true;
        leaf drop {
            type empty;
            description "Drop Traffic";
        }
        container rewrite {
            choice rewrite-value {
                case prefix-descriptor {
                    uses fpcbase:prefix-descriptor;
                    description "IP Prefix descriptor";
                }
                case pmip-selector {
                    uses traffic-selectors:traffic-selector;
                    description "PMIP Selector";
                }
                container rfc5777-classifier-template {
                    uses rfc5777:classifier;
                    description "RFC 5777 Classifier";
                }
            }
            description "Rewrite Choice";
        }
        description "Rewrite/NAT value";
    }
}
```

```
    container copy-forward-nexthop {
        uses fpcbase:fpc-nexthop;
        description "Copy Forward Value";
    }
    container nexthop {
        uses fpcbase:fpc-nexthop;
        description "NextHop Value";
    }
    case qos {
        leaf trafficclass {
            type inet:dscp;
            description "Traffic Class";
        }
        uses pmipqos:qosattribute;
        leaf qci {
            type fpcbase:fpc-qos-class-identifier;
            description "QCI";
        }
        leaf ue-agg-max-bitrate {
            type uint32;
            description "UE Aggregate Max Bitrate";
        }
        leaf apn-ambr {
            type uint32;
            description
                "Access Point Name Aggregate Max Bit Rate";
        }
        description "QoS Attributes";
    }
    description "Action Value";
}
description "FPC Action Value";
}

// Instructions
grouping instructions {
    container command-set {
        choice instr-type {
            leaf instr-3gpp-mob {
                type fpcbase:threegpp-instr;
                description "3GPP GTP Mobility Instructions";
            }
            leaf instr-pmip {
                type pmip-commandset;
                description "PMIP Instructions";
            }
        }
        description "Instruction Value Choice";
    }
}
```



```
        description "Instructions";
    }
    description "Instructions Value";
}
}
<CODE ENDS>
```

3.3. PMIP QoS Model

This module defines the base protocol elements specified in this document.

This module references [RFC6991].

```
<CODE BEGINS> file "ietf-pmip-qos@2018-05-17.yang"
module ietf-pmip-qos {
    yang-version 1.1;

    namespace
        "urn:ietf:params:xml:ns:yang:ietf-pmip-qos";

    prefix "qos-pmip";

    import ietf-inet-types {
        prefix inet;
        revision-date 2013-07-15;
    }
    import ietf-trafficselector-types { prefix traffic-selectors;
        revision-date 2018-05-17; }

    organization "IETF Distributed Mobility Management (DMM)
        Working Group";

    contact
        "WG Web:    <http://tools.ietf.org/wg/netmod/>
        WG List:    <mailto:netmod@ietf.org>

        WG Chair: Dapeng Liu
                  <mailto:maxpassion@gmail.com>

        WG Chair: Sri Gundavelli
                  <mailto:sgundave@cisco.com>

        Editor:    Satoru Matsushima
                  <mailto:satoru.matsushima@g.softbank.co.jp>

        Editor:    Lyle Bertz
                  <mailto:lylebe551144@gmail.com>";
```

description

"This module contains a collection of YANG definitions for quality of service parameters used in Proxy Mobile IPv6.

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.";

```
revision 2018-05-17 {
  description "Initial Revision.";
  reference "RFC 6088: Traffic Selectors for Flow Bindings";
}

// Type Definitions

// QoS Option Field Type Definitions
typedef sr-id {
  type uint8;
  description
    "An 8-bit unsigned integer used for identifying the QoS
    Service Request.";
}

typedef traffic-class {
  type inet:dscp;
  description
    "Traffic Class consists of a 6-bit DSCP field followed by a
    2-bit reserved field.";
  reference
    "RFC 3289: Management Information Base for the
    Differentiated Services Architecture
    RFC 2474: Definition of the Differentiated Services Field
    (DS Field) in the IPv4 and IPv6 Headers
    RFC 2780: IANA Allocation Guidelines For Values In
    the Internet Protocol and Related Headers";
}

typedef operational-code {
  type enumeration {
```

```
enum RESPONSE {
    value 0;
    description "Response to a QoS request";
}
enum ALLOCATE {
    value 1;
    description "Request to allocate QoS resources";
}
enum DE-ALLOCATE {
    value 2;
    description "Request to de-Allocate QoS resources";
}
enum MODIFY {
    value 3;
    description "Request to modify QoS parameters for a
        previously negotiated QoS Service Request";
}
enum QUERY {
    value 4;
    description "Query to list the previously negotiated QoS
        Service Requests that are still active";
}
enum NEGOTIATE {
    value 5;
    description "Response to a QoS Service Request with a
        counter QoS proposal";
}
}
description
    "The type of QoS request. Reserved values: (6) to (255)
        Currently not used. Receiver MUST ignore the option
        received with any value in this range.";
}

//Value definitions
typedef Per-MN-Agg-Max-DL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for all the mobile node's IP flows.
        The measurement units are bits per second.";
}

typedef Per-MN-Agg-Max-UL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum uplink bit rate that is
        requested/allocated for the mobile node's IP flows. The
```

```
        measurement units are bits per second.";
    }

    // Generic Structure for the uplink and downlink
    grouping Per-Session-Agg-Max-Bit-Rate-Value {
        leaf max-rate {
            type uint32;
            mandatory true;
            description
                "The aggregate maximum bit rate that is requested/allocated
                for all the IP flows associated with that mobility session.
                The measurement units are bits per second.";
        }
        leaf service-flag {
            type boolean;
            mandatory true;
            description
                "This flag is used for extending the scope of the
                target flows for Per-Session-Agg-Max-UL/DL-Bit-Rate
                from(UL)/to(DL) the mobile node's other mobility sessions
                sharing the same Service Identifier.";
            reference
                "RFC 5149 - Service Selection mobility option";
        }
        leaf exclude-flag {
            type boolean;
            mandatory true;
            description
                "This flag is used to request that the uplink/downlink
                flows for which the network is providing
                Guaranteed-Bit-Rate service be excluded from the
                target IP flows for which
                Per-Session-Agg-Max-UL/DL-Bit-Rate is measured.";
        }
        description "Per-Session-Agg-Max-Bit-Rate Value";
    }

    grouping Allocation-Retention-Priority-Value {
        leaf priority-level {
            type uint8 {
                range "0..15";
            }
            mandatory true;
            description
                "This is a 4-bit unsigned integer value. It is used to decide
                whether a mobility session establishment or modification
                request can be accepted; this is typically used for
                admission control of Guaranteed Bit Rate traffic in case of
```

```
        resource limitations.";
    }
    leaf preemption-capability {
        type enumeration {
            enum enabled {
                value 0;
                description "enabled";
            }
            enum disabled {
                value 1;
                description "disabled";
            }
            enum reserved1 {
                value 2;
                description "reserved1";
            }
            enum reserved2 {
                value 3;
                description "reserved2";
            }
        }
        mandatory true;
        description
        "This is a 2-bit unsigned integer value. It defines whether a
        service data flow can get resources tha were already
        assigned to another service data flow with a lower priority
        level.";
    }
    leaf preemption-vulnerability {
        type enumeration {
            enum enabled {
                value 0;
                description "enabled";
            }
            enum disabled {
                value 1;
                description "disabled";
            }
            enum reserved1 {
                value 2;
                description "reserved1";
            }
            enum reserved2 {
                value 3;
                description "reserved2";
            }
        }
        mandatory true;
    }
```

```
        description
        "This is a 2-bit unsigned integer value.  It defines whether a
        service data flow can lose the resources assigned to it in
        order to admit a service data flow with a higher priority
        level.";
    }
    description "Allocation-Retention-Priority Value";
}

typedef Aggregate-Max-DL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for downlink IP flows.  The measurement
        units are bits per second.";
}

typedef Aggregate-Max-UL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for downlink IP flows.  The measurement
        units are bits per second.";
}

typedef Guaranteed-DL-Bit-Rate-Value {
    type uint32;
    description
        "The guaranteed bandwidth in bits per second for downlink
        IP flows.  The measurement units are bits per second.";
}

typedef Guaranteed-UL-Bit-Rate-Value {
    type uint32;
    description
        "The guaranteed bandwidth in bits per second for uplink
        IP flows.  The measurement units are bits per second.";
}

grouping QoS-Vendor-Specific-Attribute-Value-Base {
    leaf vendorid {
        type uint32;
        mandatory true;
        description
            "The Vendor ID is the SMI (Structure of Management
            Information) Network Management Private Enterprise Code of
            the IANA-maintained 'Private Enterprise Numbers'
            registry.";
    }
}
```

```
reference
  "'PRIVATE ENTERPRISE NUMBERS', SMI Network Management
  Private Enterprise Codes, April 2014,
  <http://www.iana.org/assignments/enterprise-numbers>";
}
leaf subtype {
  type uint8;
  mandatory true;
  description
    "An 8-bit field indicating the type of vendor-specific
    information carried in the option. The namespace for this
    sub-type is managed by the vendor identified by the
    Vendor ID field.";
}
description
  "QoS Vendor-Specific Attribute.";
}

//Primary Structures (groupings)
grouping qosattribute {
  leaf per-mn-agg-max-dl {
    type qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value;
    description "Per-MN-Agg-Max-DL-Bit-Rate Value";
  }
  leaf per-mn-agg-max-ul {
    type qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value;
    description "Per-MN-Agg-Max-UL-Bit-Rate Value";
  }
  container per-session-agg-max-dl {
    uses qos-pmip:Per-Session-Agg-Max-Bit-Rate-Value;
    description "Per-Session-Agg-Max-Bit-Rate Value";
  }
  container per-session-agg-max-ul {
    uses qos-pmip:Per-Session-Agg-Max-Bit-Rate-Value;
    description "Per-Session-Agg-Max-Bit-Rate Value";
  }
  uses qos-pmip:Allocation-Retention-Priority-Value;
  leaf agg-max-dl {
    type qos-pmip:Aggregate-Max-DL-Bit-Rate-Value;
    description "Aggregate-Max-DL-Bit-Rate Value";
  }
  leaf agg-max-ul {
    type qos-pmip:Aggregate-Max-UL-Bit-Rate-Value;
    description "Aggregate-Max-UL-Bit-Rate Value";
  }
  leaf gbr-dl {
    type qos-pmip:Guaranteed-DL-Bit-Rate-Value;
    description "Guaranteed-DL-Bit-Rate Value";
  }
}
```

```

    }
    leaf gbr-ul {
        type qos-pmip:Guaranteed-UL-Bit-Rate-Value;
        description "Guaranteed-UL-Bit-Rate Value";
    }
    description "PMIP QoS Attributes. Note Vendor option
    is not a part of this grouping";
}

grouping qosoption {
    leaf srid {
        type sr-id;
        mandatory true;
        description "Service Request Identifier";
    }
    leaf trafficclass {
        type traffic-class;
        mandatory true;
        description "Traffic Class";
    }
    leaf operationcode {
        type operational-code;
        mandatory true;
        description "Operation Code";
    }
    uses qos-pmip:qosattribute;
    uses qos-pmip:QoS-Vendor-Specific-Attribute-Value-Base;
    container traffic-selector {
        uses traffic-selectors:traffic-selector;
        description "traffic selector";
    }
    description "PMIP QoS Option";
}
}
<CODE ENDS>

```

3.4. Traffic Selectors YANG Model

This module defines traffic selector types commonly used in Proxy Mobile IP (PMIP).

This module references [RFC6991].

```

<CODE BEGINS> file "ietf-trafficselector-types@2018-05-17.yang"
module ietf-trafficselector-types {
    yang-version 1.1;

    namespace

```



```
"urn:ietf:params:xml:ns:yang:ietf-trafficselector-types";

prefix "traffic-selectors";

import ietf-inet-types {
  prefix inet;
  revision-date 2013-07-15;
}

organization "IETF Distributed Mobility Management (DMM)
Working Group";

contact
  "WG Web: <http://tools.ietf.org/wg/netmod/>
  WG List: <mailto:netmod@ietf.org>

  WG Chair: Dapeng Liu
  <mailto:maxpassion@gmail.com>

  WG Chair: Sri Gundavelli
  <mailto:sgundave@cisco.com>

  Editor: Satoru Matsushima
  <mailto:satoru.matsushima@g.softbank.co.jp>

  Editor: Lyle Bertz
  <mailto:lylebe551144@gmail.com>";

description
  "This module contains a collection of YANG definitions for
  traffic selectors for flow bindings.

  Copyright (c) 2016 IETF Trust and the persons identified as the
  document authors. All rights reserved.

  This document is subject to BCP 78 and the IETF Trust's Legal
  Provisions Relating to IETF Documents
  (http://trustee.ietf.org/license-info) in effect on the date of
  publication of this document. Please review these documents
  carefully, as they describe your rights and restrictions with
  respect to this document. Code Components extracted from this
  document must include Simplified BSD License text as described
  in Section 4.e of the Trust Legal Provisions and are provided
  without warranty as described in the Simplified BSD License.";

  revision 2018-05-17 {
    description
      "Initial Revision.";
```

```
    reference
      "RFC 6088: Traffic Selectors for Flow Bindings";
  }

// Identities
identity traffic-selector-format {
  description
    "The base type for Traffic-Selector Formats";
}

identity ipv4-binary-selector-format {
  base traffic-selector-format;
  description
    "IPv4 Binary Traffic Selector Format";
}

identity ipv6-binary-selector-format {
  base traffic-selector-format;
  description
    "IPv6 Binary Traffic Selector Format";
}

// Type definitions and groupings
typedef ipsec-spi {
  type uint32;
  description
    "The first 32-bit IPsec Security Parameter Index (SPI)
    value on data. This field is defined in [RFC4303].";
  reference
    "RFC 4303: IP Encapsulating Security
    Payload (ESP)";
}

grouping traffic-selector-base {
  description "A grouping of the common leaves between the
    v4 and v6 Traffic Selectors";
  container ipsec-spi-range {
    presence "Enables setting ipsec spi range";
    description
      "Inclusive range representing IPsec Security Parameter
      Indices to be used. When only start-spi is present, it
      represents a single spi.";
  }
  leaf start-spi {
    type ipsec-spi;
    mandatory true;
    description
      "The first 32-bit IPsec SPI value on data.";
  }
}
```

```
leaf end-spi {
    type ipsec-spi;
    must ". >= ../start-spi" {
        error-message
            "The end-spi must be greater than or equal
             to start-spi";
    }
    description
        "If more than one contiguous SPI value needs to be matched,
         then this field indicates the end value of a range.";
}
}
container source-port-range {
    presence "Enables setting source port range";
    description
        "Inclusive range representing source ports to be used.
         When only start-port is present, it represents a single
         port. These value(s) are from the range of port numbers
         defined by IANA (http://www.iana.org).";
    leaf start-port {
        type inet:port-number;
        mandatory true;
        description
            "The first 16-bit source port number to be matched";
    }
    leaf end-port {
        type inet:port-number;
        must ". >= ../start-port" {
            error-message
                "The end-port must be greater than or equal to start-port";
        }
        description
            "The last 16-bit source port number to be matched";
    }
}
}
container destination-port-range {
    presence "Enables setting destination port range";
    description
        "Inclusive range representing destination ports to be used.
         When only start-port is present, it represents a single
         port.";
    leaf start-port {
        type inet:port-number;
        mandatory true;
        description
            "The first 16-bit destination port number to be matched";
    }
    leaf end-port {
```

```
        type inet:port-number;
        must ".. >= ../start-port" {
            error-message
                "The end-port must be greater than or equal to
                start-port";
        }
        description
            "The last 16-bit destination port number to be matched";
    }
}

grouping ipv4-binary-traffic-selector {
    container source-address-range-v4 {
        presence "Enables setting source IPv4 address range";
        description
            "Inclusive range representing IPv4 addresses to be used. When
            only start-address is present, it represents a single
            address.";
        leaf start-address {
            type inet:ipv4-address;
            mandatory true;
            description
                "The first source address to be matched";
        }
        leaf end-address {
            type inet:ipv4-address;
            description
                "The last source address to be matched";
        }
    }
    container destination-address-range-v4 {
        presence "Enables setting destination IPv4 address range";
        description
            "Inclusive range representing IPv4 addresses to be used.
            When only start-address is present, it represents a
            single address.";
        leaf start-address {
            type inet:ipv4-address;
            mandatory true;
            description
                "The first destination address to be matched";
        }
        leaf end-address {
            type inet:ipv4-address;
            description
                "The last destination address to be matched";
        }
    }
}
```

```
}
container ds-range {
  presence "Enables setting dscp range";
  description
    "Inclusive range representing DiffServ Codepoints to be used.
    When only start-ds is present, it represents a single
    Codepoint.";
  leaf start-ds {
    type inet:dscp;
    mandatory true;
    description
      "The first differential service value to be matched";
  }
  leaf end-ds {
    type inet:dscp;
    must ". >= ../start-ds" {
      error-message
        "The end-ds must be greater than or equal to start-ds";
    }
    description
      "The last differential service value to be matched";
  }
}
container protocol-range {
  presence "Enables setting protocol range";
  description
    "Inclusive range representing IP protocol(s) to be used. When
    only start-protocol is present, it represents a single
    protocol.";
  leaf start-protocol {
    type uint8;
    mandatory true;
    description
      "The first 8-bit protocol value to be matched.";
  }
  leaf end-protocol {
    type uint8;
    must ". >= ../start-protocol" {
      error-message
        "The end-protocol must be greater than or equal to
        start-protocol";
    }
    description
      "The last 8-bit protocol value to be matched.";
  }
}
description "ipv4 binary traffic selector";
}
```

```
grouping ipv6-binary-traffic-selector {
  container source-address-range-v6 {
    presence "Enables setting source IPv6 address range";
    description
      "Inclusive range representing IPv6 addresses to be used.
      When only start-address is present, it represents a
      single address.";
    leaf start-address {
      type inet:ipv6-address;
      mandatory true;
      description
        "The first source address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
    leaf end-address {
      type inet:ipv6-address;
      description
        "The last source address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
  }
  container destination-address-range-v6 {
    presence "Enables setting destination IPv6 address range";
    description
      "Inclusive range representing IPv6 addresses to be used.
      When only start-address is present, it represents a
      single address.";
    leaf start-address {
      type inet:ipv6-address;
      mandatory true;
      description
        "The first destination address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
    leaf end-address {
      type inet:ipv6-address;
      description
        "The last destination address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
  }
}
container flow-label-range {
  presence "Enables setting Flow Label range";
  description
    "Inclusive range representing IPv4 addresses to be used. When
    only start-flow-label is present, it represents a single
    flow label.";
  leaf start-flow-label {
```

```
    type inet:ipv6-flow-label;
    description
      "The first flow label value to be matched";
  }
  leaf end-flow-label {
    type inet:ipv6-flow-label;
    must ". >= ../start-flow-label" {
      error-message
        "The end-flow-label must be greater than or equal to
        start-flow-label";
    }
    description
      "The first flow label value to be matched";
  }
}
container traffic-class-range {
  presence "Enables setting the traffic class range";
  description
    "Inclusive range representing IPv4 addresses to be used. When
    only start-traffic-class is present, it represents a single
    traffic class.";
  leaf start-traffic-class {
    type inet:dscp;
    description
      "The first traffic class value to be matched";
    reference
      "RFC 3260: New Terminology and Clarifications for Diffserv
      RFC 3168: The Addition of Explicit Congestion Notification
      (ECN) to IP";
  }
  leaf end-traffic-class {
    type inet:dscp;
    must ". >= ../start-traffic-class" {
      error-message
        "The end-traffic-class must be greater than or equal to
        start-traffic-class";
    }
    description
      "The last traffic class value to be matched";
  }
}
container next-header-range {
  presence "Enables setting Next Header range";
  description
    "Inclusive range representing Next Headers to be used. When
    only start-next-header is present, it represents a
    single Next Header.";
  leaf start-next-header {
```

```

        type uint8;
        description
            "The first 8-bit next header value to be matched.";
    }
    leaf end-next-header {
        type uint8;
        must ". >= ../start-next-header" {
            error-message
                "The end-next-header must be greater than or equal to
                start-next-header";
        }
        description
            "The last 8-bit next header value to be matched.";
    }
}
description "ipv6 binary traffic selector";
}

grouping traffic-selector {
    leaf ts-format {
        type identityref {
            base traffic-selector-format;
        }
        description "Traffic Selector Format";
    }
    uses traffic-selectors:traffic-selector-base;
    uses traffic-selectors:ipv4-binary-traffic-selector;
    uses traffic-selectors:ipv6-binary-traffic-selector;
    description
        "The traffic selector includes the parameters used to match
        packets for a specific flow binding.";
    reference
        "RFC 6089: Flow Bindings in Mobile IPv6 and Network
        Mobility (NEMO) Basic Support";
}
}
<CODE ENDS>

```

3.5. RFC 5777 Classifier YANG Model

This module defines the RFC 5777 Classifier.

This module references [RFC5777].

```

<CODE BEGINS> file "ietf-diam-trafficclassifier@2018-05-17.yang"
module ietf-diam-trafficclassifier {
    yang-version 1.1;

```



```
namespace
"urn:ietf:params:xml:ns:yang:ietf-diam-trafficclassifier";

prefix "diamclassifier";

import ietf-inet-types {
  prefix inet;
  revision-date 2013-07-15;
}
import ietf-yang-types { prefix yang-types; }

organization "IETF Distributed Mobility Management (DMM)
Working Group";

contact
"WG Web: <http://tools.ietf.org/wg/netmod/>
WG List: <mailto:netmod@ietf.org>

WG Chair: Dapeng Liu
<mailto:maxpassion@gmail.com>

WG Chair: Sri Gundavelli
<mailto:sgundave@cisco.com>

Editor: Satoru Matsushima
<mailto:satoru.matsushima@g.softbank.co.jp>

Editor: Lyle Bertz
<mailto:lylebe551144@gmail.com>";

description
"This module contains a collection of YANG definitions for
traffic classification and QoS Attributes for Diameter.

Copyright (c) 2018 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with
respect to this document. Code Components extracted from this
document must include Simplified BSD License text as described
in Section 4.e of the Trust Legal Provisions and are provided
without warranty as described in the Simplified BSD License.";

revision 2018-05-17 {
```

```
description
    "Initial";
reference
    "RFC 5777: Traffic Classification and Quality of Service (QoS)
    Attributes for Diameter";
}

typedef eui64-address-type {
    type string {
        length "6";
    }
    description
        "specifies a single layer 2 address in EUI-64 format.
        The value is an 8-octet encoding of the address as
        it would appear in the frame header.";
}
typedef direction-type {
    type enumeration {
        enum IN {
            value 0;
            description
                "Applies to flows from the managed terminal.";
        }
        enum OUT {
            value 1;
            description
                "Applies to flows to the managed terminal.";
        }
        enum BOTH {
            value 2;
            description
                "Applies to flows both to and from the managed
                terminal.";
        }
    }
    description
        "Specifies in which direction to apply the classifier.";
}
typedef negated-flag-type {
    type enumeration {
        enum False { value 0;
            description "false"; }
        enum True { value 1;
            description "True"; }
    }
    description
        "When set to True, the meaning of the match is
        inverted and the classifier will match addresses
```

other than those specified by the From-Spec or To-Spec AVP.

Note that the negation does not impact the port comparisons.";

```
}
grouping index {
  leaf index {
    type uint16;
    mandatory true;
    description "Identifier used for referencing";
  }
  description "Index Value";
}
grouping to-from-spec-value {
  leaf-list ip-address {
    type inet:ip-address;
    description "IP address";
  }
  list ip-address-range {
    key index;
    uses diamclassifier:index;
    leaf ip-address-start {
      type inet:ip-address;
      description "IP Address Start";
    }
    leaf ip-address-end {
      type inet:ip-address;
      description "IP Address End";
    }
    description "IP Address Range";
  }
  leaf-list ip-address-mask {
    type inet:ip-prefix;
    description "IP Address Mask";
  }
  leaf-list mac-address {
    type yang-types:mac-address;
    description "MAC address";
  }
  list mac-address-mask {
    key mac-address;
    leaf mac-address {
      type yang-types:mac-address;
      mandatory true;
      description "MAC address";
    }
    leaf macaddress-mask-pattern {
```

```
        type yang-types:mac-address;
        mandatory true;
        description
            "The value specifies the bit positions of a
             MAC address that are taken for matching.";
    }
    description "MAC Address Mask";
}
leaf-list eui64-address {
    type diamclassifier:eui64-address-type;
    description "EUI64 Address";
}
list eui64-address-mask {
    key eui64-address;
    leaf eui64-address {
        type diamclassifier:eui64-address-type;
        mandatory true;
        description "eui64 address";
    }
    leaf eui64-address-mask-pattern {
        type diamclassifier:eui64-address-type;
        mandatory true;
        description
            "The value is 8 octets specifying the bit
             positions of a EUI64 address that are taken
             for matching.";
    }
    description "EUI64 Address Mask";
}
leaf-list port {
    type inet:port-number;
    description "Port Number";
}
list port-range {
    key index;
    uses diamclassifier:index;
    leaf ip-address-start {
        type inet:port-number;
        description "Port Start";
    }
    leaf ip-address-end {
        type inet:port-number;
        description "Port End";
    }
    description "Port Range";
}
leaf negated {
    type diamclassifier:negated-flag-type;
```

```
        description "Negated";
    }
    leaf use-assigned-address {
        type boolean;
        description "Use Assigned Address";
    }
    description
        "Basic traffic description value";
}

grouping option-type-group {
    leaf option-type {
        type uint8;
        mandatory true;
        description "Option Type";
    }
    leaf-list ip-option-value {
        type string;
        description "Option Value";
    }
    leaf negated {
        type diamclassifier:negated-flag-type;
        description "Negated";
    }
    description "Common X Option Pattern";
}

typedef vlan-id {
    type uint32 {
        range "0..4095";
    }
    description "VLAN ID";
}

grouping classifier {
    leaf protocol {
        type uint8;
        description "Protocol";
    }
    leaf direction {
        type diamclassifier:direction-type;
        description "Direction";
    }
    list from-spec {
        key index;
        uses diamclassifier:index;
        uses diamclassifier:to-from-spec-value;
        description "from specification";
    }
}
```

```
list to-spec {
    key index;
    uses diamclassifier:index;
    uses diamclassifier:to-from-spec-value;
    description "to specification";
}
leaf-list disffserv-code-point {
    type inet:dscp;
    description "DSCP";
}
leaf fragmentation-flag {
    type enumeration {
        enum DF {
            value 0;
            description "Don't Fragment";
        }
        enum MF {
            value 1;
            description "More Fragments";
        }
    }
    description "Fragmenttation Flag";
}
list ip-option {
    key option-type;
    uses diamclassifier:option-type-group;
    description "IP Option Value";
}
list tcp-option {
    key option-type;
    uses diamclassifier:option-type-group;
    description "TCP Option Value";
}
list tcp-flag {
    key tcp-flag-type;
    leaf tcp-flag-type {
        type uint32;
        mandatory true;
        description "TCP Flag Type";
    }
    leaf negated {
        type diamclassifier:negated-flag-type;
        description "Negated";
    }
    description "TCP Flags";
}
list icmp-option {
    key option-type;
```

```
        uses diamclassifier:option-type-group;
        description "ICMP Option Value";
    }
    list eth-option {
        key index;
        uses diamclassifier:index;
        container eth-proto-type {
            leaf-list eth-ether-type {
                type string {
                    length "2";
                }
                description "value of ethertype field";
            }
            leaf-list eth-sap {
                type string {
                    length "2";
                }
                description "802.2 SAP";
            }
            description "Ether Proto Type";
        }
    }
    list vlan-id-range {
        key index;
        uses diamclassifier:index;
        leaf-list s-vlan-id-start {
            type diamclassifier:vlan-id;
            description "S-VID  VLAN ID Start";
        }
        leaf-list s-vlan-id-end {
            type diamclassifier:vlan-id;
            description "S-VID  VLAN ID End";
        }
        leaf-list c-vlan-id-start {
            type diamclassifier:vlan-id;
            description "C-VID  VLAN ID Start";
        }
        leaf-list c-vlan-id-end {
            type diamclassifier:vlan-id;
            description "C-VID  VLAN ID End";
        }
        description "VLAN ID Range";
    }
    list user-priority-range {
        key index;
        uses diamclassifier:index;
        leaf-list low-user-priority {
            type uint32 {
                range "0..7";
            }
        }
    }
```

```

    }
    description "Low User Priority";
  }
  leaf-list high-user-priority {
    type uint32 {
      range "0..7";
    }
    description "High User Priority";
  }
  description "User priority range";
}
description "Ether Option";
}
description "RFC 5777 Classifier";
}
}
<CODE ENDS>

```

4. FPC YANG Tree Structure

This section only shows the structure for FPC YANG model. NOTE, it does NOT show the settings, Action values or Descriptor Value.

```

descriptor_value:
+--rw (descriptor-value)
|   +--:(all-traffic)
|   |   +--rw all-traffic?                empty
|   +--:(no-traffic)
|   |   +--rw no-traffic?                 empty
|   +--:(prefix-descriptor)
|   |   +--rw destination-ip?             inet:ip-prefix
|   |   +--rw source-ip?                  inet:ip-prefix
|   +--:(pmip-selector)
|   |   +--rw ts-format?                   identityref
|   |   +--rw ipsec-spi-range!
|   |   |   +--rw start-spi                ipsec-spi
|   |   |   +--rw end-spi?                 ipsec-spi
|   |   +--rw source-port-range!
|   |   |   +--rw start-port                inet:port-number
|   |   |   +--rw end-port?                 inet:port-number
|   |   +--rw destination-port-range!
|   |   |   +--rw start-port                inet:port-number
|   |   |   +--rw end-port?                 inet:port-number
|   |   +--rw source-address-range-v4!
|   |   |   +--rw start-address              inet:ipv4-address
|   |   |   +--rw end-address?              inet:ipv4-address
|   |   +--rw destination-address-range-v4!
|   |   |   +--rw start-address              inet:ipv4-address

```



```

|   +---rw end-address?      inet:ipv4-address
+---rw ds-range!
|   +---rw start-ds         inet:dscp
|   +---rw end-ds?         inet:dscp
+---rw protocol-range!
|   +---rw start-protocol    uint8
|   +---rw end-protocol?    uint8
+---rw source-address-range-v6!
|   +---rw start-address     inet:ipv6-address
|   +---rw end-address?     inet:ipv6-address
+---rw destination-address-range-v6!
|   +---rw start-address     inet:ipv6-address
|   +---rw end-address?     inet:ipv6-address
+---rw flow-label-range!
|   +---rw start-flow-label? inet:ipv6-flow-label
|   +---rw end-flow-label?  inet:ipv6-flow-label
+---rw traffic-class-range!
|   +---rw start-traffic-class? inet:dscp
|   +---rw end-traffic-class?  inet:dscp
+---rw next-header-range!
|   +---rw start-next-header? uint8
|   +---rw end-next-header?  uint8
+---:(rfc5777-classifier-template)
+---rw rfc5777-classifier-template
|   +---rw protocol?          uint8
|   +---rw direction?         diamclassifier:direction-type
|   +---rw from-spec* [index]
|   |   +---rw index          uint16
|   |   +---rw ip-address*    inet:ip-address
|   |   +---rw ip-address-range* [index]
|   |   |   +---rw index      uint16
|   |   |   +---rw ip-address-start? inet:ip-address
|   |   |   +---rw ip-address-end?  inet:ip-address
|   |   +---rw ip-address-mask*  inet:ip-prefix
|   |   +---rw mac-address*      yang-types:mac-address
|   |   +---rw mac-address-mask* [mac-address]
|   |   |   +---rw mac-address      yang-types:mac-address
|   |   |   +---rw macaddress-mask-pattern yang-types:mac-address
|   +---rw eui64-address*
|   |   diamclassifier:eui64-address-type
|   +---rw eui64-address-mask* [eui64-address]
|   |   +---rw eui64-address
|   |   |   diamclassifier:eui64-address-type
|   |   +---rw eui64-address-mask-pattern
|   |   |   diamclassifier:eui64-address-type
|   +---rw port*              inet:port-number
|   +---rw port-range* [index]
|   |   +---rw index          uint16

```

```

| | | +--rw ip-address-start?   inet:port-number
| | | +--rw ip-address-end?     inet:port-number
| | | +--rw negated?
| | |     diamclassifier:negated-flag-type
| | | +--rw use-assigned-address?  boolean
+--rw to-spec* [index]
| | | +--rw index                uint16
| | | +--rw ip-address*          inet:ip-address
+--rw ip-address-range* [index]
| | | +--rw index                uint16
| | | +--rw ip-address-start?    inet:ip-address
| | | +--rw ip-address-end?      inet:ip-address
+--rw ip-address-mask*          inet:ip-prefix
+--rw mac-address*              yang-types:mac-address
+--rw mac-address-mask* [mac-address]
| | | +--rw mac-address          yang-types:mac-address
| | | +--rw macaddress-mask-pattern  yang-types:mac-address
+--rw eui64-address*
| | |     diamclassifier:eui64-address-type
+--rw eui64-address-mask* [eui64-address]
| | | +--rw eui64-address
| | |     diamclassifier:eui64-address-type
+--rw eui64-address-mask-pattern
| | |     diamclassifier:eui64-address-type
+--rw port*                     inet:port-number
+--rw port-range* [index]
| | | +--rw index                uint16
| | | +--rw ip-address-start?    inet:port-number
| | | +--rw ip-address-end?      inet:port-number
+--rw negated?
| | |     diamclassifier:negated-flag-type
| | | +--rw use-assigned-address?  boolean
+--rw disffserv-code-point*    inet:dscp
+--rw fragmentation-flag?      enumeration
+--rw ip-option* [option-type]
| | | +--rw option-type          uint8
| | | +--rw ip-option-value*     string
| | | +--rw negated?             diamclassifier:negated-flag-type
+--rw tcp-option* [option-type]
| | | +--rw option-type          uint8
| | | +--rw ip-option-value*     string
| | | +--rw negated?             diamclassifier:negated-flag-type
+--rw tcp-flag* [tcp-flag-type]
| | | +--rw tcp-flag-type        uint32
| | | +--rw negated?             diamclassifier:negated-flag-type
+--rw icmp-option* [option-type]
| | | +--rw option-type          uint8
| | | +--rw ip-option-value*     string

```

```

|   +--rw negated?                diamclassifier:negated-flag-type
+--rw eth-option* [index]
|   +--rw index                    uint16
|   +--rw eth-proto-type
|   |   +--rw eth-ether-type*      string
|   |   +--rw eth-sap*             string
|   +--rw vlan-id-range* [index]
|   |   +--rw index                uint16
|   |   +--rw s-vlan-id-start*     diamclassifier:vlan-id
|   |   +--rw s-vlan-id-end*       diamclassifier:vlan-id
|   |   +--rw c-vlan-id-start*     diamclassifier:vlan-id
|   |   +--rw c-vlan-id-end*       diamclassifier:vlan-id
|   +--rw user-priority-range* [index]
|   |   +--rw index                uint16
|   |   +--rw low-user-priority*   uint32
|   |   +--rw high-user-priority*  uint32
+--:(packet-filter)
|   +--rw packet-filter
|   |   +--rw direction?           fpcbase:packet-filter-direction
|   |   +--rw identifier?          uint8
|   |   +--rw evaluation-precedence? uint8
|   |   +--rw contents* [component-type-identifier]
|   |   |   +--rw component-type-identifier fpcbase:component-type-id
|   |   |   +--rw (value)?
|   |   |   |   +--:(ipv4-local)
|   |   |   |   |   +--rw ipv4-local?          inet:ipv4-address
|   |   |   |   +--:(ipv6-prefix-local)
|   |   |   |   |   +--rw ipv6-prefix-local?    inet:ipv6-prefix
|   |   |   |   +--:(ipv4-ipv6-remote)
|   |   |   |   |   +--rw ipv4-ipv6-remote?     inet:ip-address
|   |   |   |   +--:(ipv6-prefix-remote)
|   |   |   |   |   +--rw ipv6-prefix-remote?    inet:ipv6-prefix
|   |   |   |   +--:(next-header)
|   |   |   |   |   +--rw next-header?          uint8
|   |   |   |   +--:(local-port)
|   |   |   |   |   +--rw local-port?           inet:port-number
|   |   |   |   +--:(local-port-range)
|   |   |   |   |   +--rw local-port-lo?        inet:port-number
|   |   |   |   |   +--rw local-port-hi?        inet:port-number
|   |   |   |   +--:(remote-port)
|   |   |   |   |   +--rw remote-port?          inet:port-number
|   |   |   |   +--:(remote-port-range)
|   |   |   |   |   +--rw remote-port-lo?       inet:port-number
|   |   |   |   |   +--rw remote-port-hi?       inet:port-number
|   |   |   |   +--:(ipsec-index)
|   |   |   |   |   +--rw ipsec-index?          traffic-selectors:ipsec-spi
|   |   |   |   +--:(traffic-class)
|   |   |   |   |   +--rw traffic-class?        inet:dscp

```

```

|         +---:(traffic-class-range)
|         |   +---rw traffic-class-lo?          inet:dscp
|         |   +---rw traffic-class-hi?          inet:dscp
|         +---:(flow-label)
|         |   +---rw flow-label*      inet:ipv6-flow-label
+---:(tunnel-info)
  +---rw tunnel-info
    +---rw tunnel-local-address?    inet:ip-address
    +---rw tunnel-remote-address?   inet:ip-address
    +---rw mtu-size?                uint32
    +---rw tunnel?                  identityref
    +---rw payload-type?            enumeration
    +---rw gre-key?                 uint32
    +---rw gtp-tunnel-info
    |   +---rw local-tunnel-identifier?    uint32
    |   +---rw remote-tunnel-identifier?   uint32
    |   +---rw sequence-numbers-enabled?   boolean
    +---rw ebi?                     fpcbase:ebi-type
    +---rw lbi?                     fpcbase:ebi-type

action_value:
+---:(action-value)
|   +---rw (action-value)
|   |   +---:(drop)
|   |   |   +---rw drop?                empty
|   |   +---:(rewrite)
|   |   |   +---rw rewrite
|   |   |   |   +---rw (rewrite-value)?
|   |   |   |   |   +---:(prefix-descriptor)
|   |   |   |   |   |   +---rw destination-ip?    inet:ip-prefix
|   |   |   |   |   |   +---rw source-ip?        inet:ip-prefix
|   |   |   |   |   +---:(pmip-selector)
|   |   |   |   |   |   +---rw ts-format?          identityref
|   |   |   |   |   +---rw ipsec-spi-range!
|   |   |   |   |   |   +---rw start-spi      ipsec-spi
|   |   |   |   |   |   +---rw end-spi?      ipsec-spi
|   |   |   |   +---rw source-port-range!
|   |   |   |   |   +---rw start-port    inet:port-number
|   |   |   |   |   +---rw end-port?    inet:port-number
|   |   |   +---rw destination-port-range!
|   |   |   |   +---rw start-port    inet:port-number
|   |   |   |   +---rw end-port?    inet:port-number
|   |   +---rw source-address-range-v4!
|   |   |   +---rw start-address    inet:ipv4-address
|   |   |   +---rw end-address?    inet:ipv4-address
|   |   +---rw destination-address-range-v4!
|   |   |   +---rw start-address    inet:ipv4-address
|   |   |   +---rw end-address?    inet:ipv4-address

```

```

+--rw ds-range!
|   +--rw start-ds      inet:dscp
|   +--rw end-ds?      inet:dscp
+--rw protocol-range!
|   +--rw start-protocol  uint8
|   +--rw end-protocol?  uint8
+--rw source-address-range-v6!
|   +--rw start-address   inet:ipv6-address
|   +--rw end-address?   inet:ipv6-address
+--rw destination-address-range-v6!
|   +--rw start-address   inet:ipv6-address
|   +--rw end-address?   inet:ipv6-address
+--rw flow-label-range!
|   +--rw start-flow-label?  inet:ipv6-flow-label
|   +--rw end-flow-label?   inet:ipv6-flow-label
+--rw traffic-class-range!
|   +--rw start-traffic-class?  inet:dscp
|   +--rw end-traffic-class?    inet:dscp
+--rw next-header-range!
|   +--rw start-next-header?  uint8
|   +--rw end-next-header?   uint8
+--:(rfc5777-classifier-template)
+--rw rfc5777-classifier-template
|   +--rw protocol?          uint8
|   +--rw direction?
|       diamclassifier:direction-type
+--rw from-spec* [index]
|   +--rw index              uint16
|   +--rw ip-address*        inet:ip-address
|   +--rw ip-address-range* [index]
|       +--rw index          uint16
|       +--rw ip-address-start?  inet:ip-address
|       +--rw ip-address-end?    inet:ip-address
|   +--rw ip-address-mask*    inet:ip-prefix
|   +--rw mac-address*        yang-types:mac-address
|   +--rw mac-address-mask* [mac-address]
|       +--rw mac-address
|           yang-types:mac-address
|   +--rw macaddress-mask-pattern
|       yang-types:mac-address
+--rw eui64-address*
|   diamclassifier:eui64-address-type
+--rw eui64-address-mask* [eui64-address]
|   +--rw eui64-address
|       diamclassifier:eui64-address-type
|   +--rw eui64-address-mask-pattern
|       diamclassifier:eui64-address-type
+--rw port*                  inet:port-number

```

```

+---rw port-range* [index]
|   +---rw index                               uint16
|   +---rw ip-address-start?                   inet:port-number
|   +---rw ip-address-end?                     inet:port-number
+---rw negated?
|   diamclassifier:negated-flag-type
+---rw use-assigned-address?   boolean
+---rw to-spec* [index]
|   +---rw index                               uint16
|   +---rw ip-address*                   inet:ip-address
+---rw ip-address-range* [index]
|   +---rw index                               uint16
|   +---rw ip-address-start?               inet:ip-address
|   +---rw ip-address-end?                 inet:ip-address
+---rw ip-address-mask*               inet:ip-prefix
+---rw mac-address*
|   yang-types:mac-address
+---rw mac-address-mask* [mac-address]
|   +---rw mac-address
|   yang-types:mac-address
|   +---rw macaddress-mask-pattern
|   yang-types:mac-address
+---rw eui64-address*
|   diamclassifier:eui64-address-type
+---rw eui64-address-mask* [eui64-address]
|   +---rw eui64-address
|   diamclassifier:eui64-address-type
|   +---rw eui64-address-mask-pattern
|   diamclassifier:eui64-address-type
+---rw port*                           inet:port-number
+---rw port-range* [index]
|   +---rw index                               uint16
|   +---rw ip-address-start?                   inet:port-number
|   +---rw ip-address-end?                     inet:port-number
+---rw negated?
|   diamclassifier:negated-flag-type
|   +---rw use-assigned-address?   boolean
+---rw disffserv-code-point*   inet:dscp
+---rw fragmentation-flag?     enumeration
+---rw ip-option* [option-type]
|   +---rw option-type           uint8
|   +---rw ip-option-value*      string
+---rw negated?
|   diamclassifier:negated-flag-type
+---rw tcp-option* [option-type]
|   +---rw option-type           uint8
|   +---rw ip-option-value*      string
+---rw negated?

```

```

|                                     diamclassifier:negated-flag-type
| +--rw tcp-flag* [tcp-flag-type]
| |   +--rw tcp-flag-type      uint32
| |   +--rw negated?
| |       diamclassifier:negated-flag-type
| +--rw icmp-option* [option-type]
| |   +--rw option-type          uint8
| |   +--rw ip-option-value*     string
| |   +--rw negated?
| |       diamclassifier:negated-flag-type
| +--rw eth-option* [index]
| |   +--rw index                  uint16
| |   +--rw eth-proto-type
| | |   +--rw eth-ether-type*      string
| | |   +--rw eth-sap*            string
| | +--rw vlan-id-range* [index]
| | |   +--rw index                  uint16
| | |   +--rw s-vlan-id-start*
| | | |       diamclassifier:vlan-id
| | |   +--rw s-vlan-id-end*
| | | |       diamclassifier:vlan-id
| | |   +--rw c-vlan-id-start*
| | | |       diamclassifier:vlan-id
| | |   +--rw c-vlan-id-end*
| | | |       diamclassifier:vlan-id
| | +--rw user-priority-range* [index]
| | |   +--rw index                  uint16
| | |   +--rw low-user-priority*     uint32
| | |   +--rw high-user-priority*    uint32
+--:(copy-forward-nexthop)
|   +--rw copy-forward-nexthop
|   |   +--rw (next-hop-value)?
|   |   |   +--:(ip-address)
|   |   |   |   +--rw ip-address?      inet:ip-address
|   |   |   +--:(mac-address)
|   |   |   |   +--rw mac-address?      ytypes:mac-address
|   |   |   +--:(service-path)
|   |   |   |   +--rw service-path?    fpcbase:fpc-service-path-id
|   |   |   +--:(mpls-path)
|   |   |   |   +--rw mpls-path?        fpcbase:fpc-mpls-label
|   |   |   +--:(nsh)
|   |   |   |   +--rw nsh?              string
|   |   |   +--:(interface)
|   |   |   |   +--rw interface?        uint16
|   |   |   +--:(segment-identifier)
|   |   |   |   +--rw segment-identifier? fpcbase:segment-id
|   |   |   +--:(mpls-label-stack)
|   |   |   |   +--rw mpls-label-stack* fpcbase:fpc-mpls-label

```

```

+---:(mpls-sr-stack)
|  +---rw mpls-sr-stack*          fpcbase:fpc-mpls-label
+---:(srv6-stack)
|  +---rw srv6-stack*            fpcbase:segment-id
+---:(tunnel-info)
  +---rw tunnel-info
    +---rw tunnel-local-address?   inet:ip-address
    +---rw tunnel-remote-address?  inet:ip-address
    +---rw mtu-size?               uint32
    +---rw tunnel?                 identityref
    +---rw payload-type?           enumeration
    +---rw gre-key?                uint32
    +---rw gtp-tunnel-info
      +---rw local-tunnel-identifier?  uint32
      +---rw remote-tunnel-identifier? uint32
      +---rw sequence-numbers-enabled? boolean
    +---rw ebi?                    fpcbase:ebi-type
    +---rw lbi?                    fpcbase:ebi-type
+---:(nexthop)
  +---rw nexthop
    +---rw (next-hop-value)?
      +---:(ip-address)
      |  +---rw ip-address?         inet:ip-address
      +---:(mac-address)
      |  +---rw mac-address?        ytypes:mac-address
      +---:(service-path)
      |  +---rw service-path?       fpcbase:fpc-service-path-id
      +---:(mpls-path)
      |  +---rw mpls-path?          fpcbase:fpc-mpls-label
      +---:(nsh)
      |  +---rw nsh?                string
      +---:(interface)
      |  +---rw interface?          uint16
      +---:(segment-identifier)
      |  +---rw segment-identifier?  fpcbase:segment-id
      +---:(mpls-label-stack)
      |  +---rw mpls-label-stack*    fpcbase:fpc-mpls-label
      +---:(mpls-sr-stack)
      |  +---rw mpls-sr-stack*       fpcbase:fpc-mpls-label
      +---:(srv6-stack)
      |  +---rw srv6-stack*         fpcbase:segment-id
      +---:(tunnel-info)
      +---rw tunnel-info
        +---rw tunnel-local-address?  inet:ip-address
        +---rw tunnel-remote-address?  inet:ip-address
        +---rw mtu-size?              uint32
        +---rw tunnel?                identityref
        +---rw payload-type?          enumeration

```



```

        +--rw gre-key?                               uint32
        +--rw gtp-tunnel-info
        |   +--rw local-tunnel-identifier?           uint32
        |   +--rw remote-tunnel-identifier?         uint32
        |   +--rw sequence-numbers-enabled?         boolean
        +--rw ebi?                                    fpcbase:ebi-type
        +--rw lbi?                                    fpcbase:ebi-type
+--:(qos)
  +--rw trafficclass?                                inet:dscp
  +--rw per-mn-agg-max-dl?
    qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value
  +--rw per-mn-agg-max-ul?
    qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
  +--rw per-session-agg-max-dl
    |   +--rw max-rate                               uint32
    |   +--rw service-flag                           boolean
    |   +--rw exclude-flag                           boolean
  +--rw per-session-agg-max-ul
    |   +--rw max-rate                               uint32
    |   +--rw service-flag                           boolean
    |   +--rw exclude-flag                           boolean
  +--rw priority-level                               uint8
  +--rw preemption-capability                         enumeration
  +--rw preemption-vulnerability                     enumeration
  +--rw agg-max-dl?
    qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
  +--rw agg-max-ul?
    qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
  +--rw gbr-dl?
    qos-pmip:Guaranteed-DL-Bit-Rate-Value
  +--rw gbr-ul?
    qos-pmip:Guaranteed-UL-Bit-Rate-Value
  +--rw qci?
    fpcbase:fpc-qos-class-identifier
  +--rw ue-agg-max-bitrate?                           uint32
  +--rw apn-ambr?                                     uint32

policy-configuration-value:
  +--rw (policy-configuration-value)?
    +--:(descriptor-value)
    |   ...
    +--:(action-value)
    |   ...
    +--:(setting-value)
    |   +--rw setting?                                <anydata>

policy-configuration:
  +--rw policy-configuration* [index]

```

```

| | | +--rw index                               uint16
| | | +--rw extensible?                         boolean
| | | +--rw static-attributes*                  string
| | | +--rw mandatory-attributes*              string
| | | +--rw entity-state?                      enumeration
| | | +--rw version?                           uint32
| | | +--rw (policy-configuration-value)?
| | | ...
module: ietf-dmm-fpc
+--rw tenant* [tenant-key]
|   +--rw tenant-key                          fpc:fpc-identity
|   +--rw topology-information-model
|   |   +--rw service-group* [service-group-key role-key]
|   |   |   +--rw service-group-key          fpc:fpc-identity
|   |   |   +--rw service-group-name?       string
|   |   |   +--rw role-key                   identityref
|   |   |   +--rw role-name?                 string
|   |   |   +--rw protocol*                  identityref
|   |   |   +--rw feature*                   identityref
|   |   |   +--rw service-group-configuration* [index]
|   |   |   |   +--rw index                     uint16
|   |   |   |   +--rw (policy-configuration-value)?
|   |   |   |   ...
|   |   +--rw dpn* [dpn-key]
|   |   |   +--rw dpn-key                     fpc:fpc-identity
|   |   |   +--rw referenced-interface* [interface-key]
|   |   |   |   +--rw interface-key           fpc:fpc-identity
|   |   |   |   +--rw peer-service-group-key* fpc:fpc-identity
|   |   +--rw dpn* [dpn-key]
|   |   |   +--rw dpn-key                     fpc:fpc-identity
|   |   |   +--rw dpn-name?                   string
|   |   |   +--rw dpn-resource-mapping-reference? string
|   |   |   +--rw domain-key                  fpc:fpc-identity
|   |   |   +--rw service-group-key*          fpc:fpc-identity
|   |   +--rw interface* [interface-key]
|   |   |   +--rw interface-key               fpc:fpc-identity
|   |   |   +--rw interface-name?             string
|   |   |   +--rw role?                       identityref
|   |   |   +--rw protocol*                   identityref
|   |   |   +--rw interface-configuration* [index]
|   |   |   |   +--rw (policy-configuration-value)?
|   |   |   |   ...
|   |   +--rw dpn-policy-configuration* [policy-template-key]
|   |   |   +--rw policy-template-key         fpc:fpc-identity
|   |   |   +--rw policy-configuration* [index]
|   |   |   |   +--rw index                     uint16
|   |   |   |   +--rw (policy-configuration-value)?

```

```

|         | ...
+--rw domain* [domain-key]
|   +--rw domain-key          fpc:fpc-identity
|   +--rw domain-name?       string
|   +--rw domain-policy-configuration* [policy-template-key]
|       +--rw policy-template-key    fpc:fpc-identity
|       +--rw policy-configuration* [index]
|           | ...
+--rw dpn-checkpoint
|   +--rw basename?          fpc:fpc-identity
|   +--rw base-checkpoint?   string
+--rw service-group-checkpoint
|   +--rw basename?          fpc:fpc-identity
|   +--rw base-checkpoint?   string
+--rw dpn-checkpoint
|   +--rw basename?          fpc:fpc-identity
|   +--rw base-checkpoint?   string
+--rw policy-information-model
|   +--rw action-template* [action-template-key]
|       +--rw action-template-key    fpc:fpc-identity
|       +--rw (action-value)
|           | ...
|       +--rw extensible?            boolean
|       +--rw static-attributes*     string
|       +--rw mandatory-attributes*  string
|       +--rw entity-state?          enumeration
|       +--rw version?               uint32
+--rw descriptor-template* [descriptor-template-key]
|   +--rw descriptor-template-key    fpc:fpc-identity
|   +--rw (descriptor-value)
|       | ...
|       +--rw extensible?            boolean
|       +--rw static-attributes*     string
|       +--rw mandatory-attributes*  string
|       +--rw entity-state?          enumeration
|       +--rw version?               uint32
+--rw rule-template* [rule-template-key]
|   +--rw rule-template-key          fpc:fpc-identity
|   +--rw descriptor-match-type      enumeration
|   +--rw descriptor-configuration* [descriptor-template-key]
|       +--rw descriptor-template-key    fpc:fpc-identity
|       +--rw direction?                rfc5777:direction-type
|       +--rw setting?                  <anydata>
|       +--rw attribute-expression* [index]
|           +--rw index                  uint16
|           +--rw (descriptor-value)
|               | ...
+--rw action-configuration* [action-order]

```

```

    +--rw action-order                uint32
    +--rw action-template-key         fpc:fpc-identity
    +--rw setting?                    <anydata>
    +--rw attribute-expression* [index]
        +--rw index                  uint16
        +--rw (action-value)
            | ...
    +--rw extensible?                 boolean
    +--rw static-attributes*          string
    +--rw mandatory-attributes*       string
    +--rw entity-state?               enumeration
    +--rw version?                   uint32
    +--rw rule-configuration* [index]
        +--rw index                  uint16
        +--rw (policy-configuration-value)?
            | ...
    +--rw policy-template* [policy-template-key]
        +--rw policy-template-key     fpc:fpc-identity
        +--rw rule-template* [precedence]
            +--rw precedence           uint32
            +--rw rule-template-key    fpc:fpc-identity
        +--rw extensible?             boolean
        +--rw static-attributes*      string
        +--rw mandatory-attributes*   string
        +--rw entity-state?           enumeration
        +--rw version?                uint32
        +--rw policy-configuration* [index]
            ...
    +--rw basename?                  fpc:fpc-identity
    +--rw base-checkpoint?           string
    +--rw mobility-context* [mobility-context-key]
        +--rw mobility-context-key    fpc:fpc-identity
        +--rw delegating-ip-prefix*   inet:ip-prefix
        +--rw parent-context?         fpc:fpc-identity
        +--rw child-context*          fpc:fpc-identity
    +--rw mobile-node
        +--rw ip-address*             inet:ip-address
        +--rw imsi?                   fpcbase:imsi-type
        +--rw mn-policy-configuration* [policy-template-key]
            +--rw policy-template-key  fpc:fpc-identity
            +--rw policy-configuration* [index]
                ...
    +--rw domain
        +--rw domain-key?             fpc:fpc-identity
        +--rw domain-policy-configuration* [policy-template-key]
            +--rw policy-template-key  fpc:fpc-identity
            +--rw policy-configuration* [index]
                ...

```

```

    +--rw dpn* [dpn-key]
      +--rw dpn-key          fpc:fpc-identity
      +--rw dpn-policy-configuration* [policy-template-key]
        +--rw policy-template-key  fpc:fpc-identity
        +--rw policy-configuration* [index]
        ...
      +--rw role?              identityref
      +--rw service-data-flow* [identifier]
        +--rw identifier          uint32
        +--rw service-group-key?  fpc:fpc-identity
        +--rw interface* [interface-key]
          +--rw interface-key    fpc:fpc-identity
        +--rw service-data-flow-policy-
            configuration* [policy-template-key]
          +--rw policy-template-key  fpc:fpc-identity
          +--rw policy-configuration* [index]
          ...
+--rw monitor* [monitor-key]
  +--rw extensible?          boolean
  +--rw static-attributes*   string
  +--rw mandatory-attributes* string
  +--rw entity-state?        enumeration
  +--rw version?             uint32
  +--rw monitor-key          fpc:fpc-identity
  +--rw target?              string
  +--rw deferrable?          boolean
  +--rw (configuration)
    +--:(period)
      +--rw period?          uint32
    +--:(threshold-config)
      +--rw low?              uint32
      +--rw hi?               uint32
    +--:(schedule)
      +--rw schedule?        uint32
    +--:(event-identities)
      +--rw event-identities* identityref
    +--:(event-ids)
      +--rw event-ids*        uint32

rpcs:
  +---x configure
    +---w input
      +---w client-id          fpc:client-identifier
      +---w execution-delay?   uint32
      +---w yang-patch
        +---w patch-id        string
        +---w comment?         string
        +---w edit* [edit-id]

```

```

+---w edit-id          string
+---w operation        enumeration
+---w target           target-resource-offset
+---w point?           target-resource-offset
+---w where?           enumeration
+---w value?           <anydata>
+---w reference-scope? fpc:ref-scope
+---w command-set
+---w (instr-type)?
+---:(instr-3gpp-mob)
|   +---w instr-3gpp-mob? fpcbase:threegpp-instr
+---:(instr-pmip)
|   +---w instr-pmip?      pmip-commandset
+--ro output
+--ro yang-patch-status
+--ro patch-id          string
+--ro (global-status)?
+---:(global-errors)
+---ro errors
+---ro error*
+---ro error-type       enumeration
+---ro error-tag        string
+---ro error-app-tag?   string
+---ro error-path?      instance-identifier
+---ro error-message?   string
+---ro error-info?      <anydata>
+---:(ok)
+---ro ok?              empty
+--ro edit-status
+--ro edit* [edit-id]
+--ro edit-id           string
+--ro (edit-status-choice)?
+---:(ok)
+---ro ok?              empty
+---ro notify-follows?  boolean
+---ro subsequent-edit* [edit-id]
+---ro edit-id          string
+---ro operation         enumeration
+---ro target            ypatch:target-resource-offset
+---ro point?           ypatch:target-resource-offset
+---ro where?           enumeration
+---ro value?           <anydata>
+---:(errors)
+---ro errors
+---ro error*
+---ro error-type       enumeration

```

```

|                                     +--ro error-tag          string
|                                     +--ro error-app-tag?       string
|                                     +--ro error-path?
|                                         instance-identifier
|                                     +--ro error-message?       string
|                                     +--ro error-info?          <anydata>
+---x register_monitor
|   +---w input
|   |   +---w client-id          fpc:client-identifier
|   |   +---w execution-delay?   uint32
|   |   +---w operation-id       uint64
|   |   +---w monitor* [monitor-key]
|   |   |   +---w extensible?    boolean
|   |   |   +---w static-attributes*    string
|   |   |   +---w mandatory-attributes* string
|   |   |   +---w entity-state?    enumeration
|   |   |   +---w version?         uint32
|   |   |   +---w monitor-key      fpc:fpc-identity
|   |   |   +---w target?         string
|   |   |   +---w deferrable?      boolean
|   |   |   +---w (configuration)
|   |   |   |   +--:(period)
|   |   |   |   |   +---w period?          uint32
|   |   |   |   +--:(threshold-config)
|   |   |   |   |   +---w low?            uint32
|   |   |   |   |   +---w hi?            uint32
|   |   |   |   +--:(schedule)
|   |   |   |   |   +---w schedule?       uint32
|   |   |   |   +--:(event-identities)
|   |   |   |   |   +---w event-identities* identityref
|   |   |   |   +--:(event-ids)
|   |   |   |   |   +---w event-ids*      uint32
|   |   +---ro output
|   |   |   +--ro operation-id      uint64
|   |   |   +--ro (edit-status-choice)?
|   |   |   |   +--:(ok)
|   |   |   |   |   +--ro ok?        empty
|   |   |   |   +--:(errors)
|   |   |   |   |   +--ro errors
|   |   |   |   |   |   +--ro error*
|   |   |   |   |   |   |   +--ro error-type    enumeration
|   |   |   |   |   |   |   +--ro error-tag      string
|   |   |   |   |   |   |   +--ro error-app-tag? string
|   |   |   |   |   |   |   +--ro error-path?    instance-identifier
|   |   |   |   |   |   |   +--ro error-message? string
|   |   |   |   |   |   |   +--ro error-info?    <anydata>
+---x deregister_monitor
|   +---w input

```

```

|      +---w client-id          fpc:client-identifier
|      +---w execution-delay?   uint32
|      +---w operation-id       uint64
|      +---w monitor* [monitor-key]
|          +---w monitor-key     fpc:fpc-identity
|          +---w send_data?      boolean
+---ro output
+---ro operation-id             uint64
+---ro (edit-status-choice)?
+---: (ok)
|   +---ro ok?                  empty
+---: (errors)
+---ro errors
+---ro error*
+---ro error-type               enumeration
+---ro error-tag                string
+---ro error-app-tag?           string
+---ro error-path?              instance-identifier
+---ro error-message?           string
+---ro error-info?              <anydata>
+---x probe
+---w input
|      +---w client-id          fpc:client-identifier
|      +---w execution-delay?   uint32
|      +---w operation-id       uint64
|      +---w monitor* [monitor-key]
|          +---w monitor-key     fpc:fpc-identity
+---ro output
+---ro operation-id             uint64
+---ro (edit-status-choice)?
+---: (ok)
|   +---ro ok?                  empty
+---: (errors)
+---ro errors
+---ro error*
+---ro error-type               enumeration
+---ro error-tag                string
+---ro error-app-tag?           string
+---ro error-path?              instance-identifier
+---ro error-message?           string
+---ro error-info?              <anydata>

notifications:
+---n config-result-notification
|   +---ro yang-patch-status
|   |   +---ro patch-id         string
|   |   +---ro (global-status)?
|   |   |   +---: (global-errors)

```



```

    +--ro errors
      +--ro error*
        +--ro error-type      enumeration
        +--ro error-tag       string
        +--ro error-app-tag?   string
        +--ro error-path?     instance-identifier
        +--ro error-message?   string
        +--ro error-info?     <anydata>
      +--:(ok)
        +--ro ok?             empty
    +--ro edit-status
      +--ro edit* [edit-id]
        +--ro edit-id         string
        +--ro (edit-status-choice)?
          +--:(ok)
            +--ro ok?         empty
          +--:(errors)
            +--ro errors
              +--ro error*
                +--ro error-type      enumeration
                +--ro error-tag       string
                +--ro error-app-tag?   string
                +--ro error-path?     instance-identifier
                +--ro error-message?   string
                +--ro error-info?     <anydata>
    +--ro subsequent-edit* [edit-id]
      +--ro edit-id         string
      +--ro operation       enumeration
      +--ro target          ypatch:target-resource-offset
      +--ro point?         ypatch:target-resource-offset
      +--ro where?         enumeration
      +--ro value?         <anydata>
+---n notify
  +--ro notification-id?   uint32
  +--ro timestamp?        uint32
  +--ro report* [monitor-key]
    +--ro monitor-key      fpc:fpc-identity
    +--ro trigger?         identityref
    +--ro (value)?
      +--:(dpn-candidate-available)
        +--ro node-id?     inet:uri
        +--ro supported-interface-list* [role-key]
          +--ro role-key    identityref
      +--:(dpn-unavailable)
        +--ro dpn-id?      fpc:fpc-identity
      +--:(report-value)
        +--ro report-value? <anydata>

```

Figure 1: YANG FPC Agent Tree

5. Work Team Participants

Participants in the FPSM work team discussion include Satoru Matsushima, Danny Moses, Sri Gundavelli, Marco Liebsch, Pierrick Seite, Alper Yegin, Carlos Bernardos, Charles Perkins and Fred Templin.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<https://www.rfc-editor.org/info/rfc5777>>.
- [RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, DOI 10.17487/RFC6088, January 2011, <<https://www.rfc-editor.org/info/rfc6088>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8072] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", RFC 8072, DOI 10.17487/RFC8072, February 2017, <<https://www.rfc-editor.org/info/rfc8072>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

6.2. Informative References

- [I-D.ietf-dmm-fpc-cpdp]
Matsushima, S., Bertz, L., Liebsch, M., Gundavelli, S.,
Moses, D., and C. Perkins, "Protocol for Forwarding Policy
Configuration (FPC) in DMM", draft-ietf-dmm-fpc-cpdp-12
(work in progress), June 2018.
- [RFC3958] Daigle, L. and A. Newton, "Domain-Based Application
Service Location Using SRV RRs and the Dynamic Delegation
Discovery Service (DDDS)", RFC 3958, DOI 10.17487/RFC3958,
January 2005, <<https://www.rfc-editor.org/info/rfc3958>>.
- [RFC7222] Liebsch, M., Seite, P., Yokota, H., Korhonen, J., and S.
Gundavelli, "Quality-of-Service Option for Proxy Mobile
IPv6", RFC 7222, DOI 10.17487/RFC7222, May 2014,
<<https://www.rfc-editor.org/info/rfc7222>>.

Authors' Addresses

Satoru Matsushima
SoftBank
1-9-1, Higashi-Shimbashi, Minato-Ku
Tokyo 105-7322
Japan

Email: satoru.matsushima@g.softbank.co.jp

Lyle Bertz
6220 Sprint Parkway
Overland Park KS, 66251
USA

Email: lylebe551144@gmail.com

Marco Liebsch
NEC Laboratories Europe
NEC Europe Ltd.
Kurfuersten-Anlage 36
D-69115 Heidelberg
Germany

Phone: +49 6221 4342146
Email: liebsch@neclab.eu

Sri Gundavelli
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: sgundave@cisco.com

Danny Moses

Email: danny.moses@intel.com

Charles E. Perkins
Futurewei Inc.
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1-408-330-4586
Email: charliep@computer.org

DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 November 2022

S. Matsushima, Ed.
SoftBank
C. Filsfils
M. Kohno
P. Camarillo, Ed.
Cisco Systems, Inc.
D. Voyer
Bell Canada
C.E. Perkins
Lupin Lodge
9 May 2022

Segment Routing IPv6 for Mobile User Plane
draft-ietf-dmm-srv6-mobile-uplane-21

Abstract

This document specifies the applicability of SRv6 (Segment Routing IPv6) to the user-plane of mobile networks. The network programming nature of SRv6 accomplishes mobile user-plane functions in a simple manner. The statelessness of SRv6 and its ability to control both service layer path and underlying transport can be beneficial to the mobile user-plane, providing flexibility, end-to-end network slicing, and SLA control for various applications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 10 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Conventions and Terminology	3
2.1. Terminology	3
2.2. Conventions	4
2.3. Predefined SRv6 Endpoint Behaviors	4
3. Motivation	5
4. 3GPP Reference Architecture	5
5. User-plane modes	6
5.1. Traditional mode	7
5.1.1. Packet flow - Uplink	8
5.1.2. Packet flow - Downlink	9
5.2. Enhanced mode	9
5.2.1. Packet flow - Uplink	10
5.2.2. Packet flow - Downlink	11
5.2.3. Scalability	12
5.3. Enhanced mode with unchanged gNB GTP behavior	12
5.3.1. Interworking with IPv6 GTP	12
5.3.2. Interworking with IPv4 GTP	15
5.3.3. Extensions to the interworking mechanisms	18
5.4. SRv6 Drop-in Interworking	18
6. SRv6 Segment Endpoint Mobility Behaviors	19
6.1. Args.Mob.Session	20
6.2. End.MAP	20
6.3. End.M.GTP6.D	21
6.4. End.M.GTP6.D.Di	22
6.5. End.M.GTP6.E	23
6.6. End.M.GTP4.E	24
6.7. H.M.GTP4.D	25
6.8. End.Limit: Rate Limiting behavior	26
7. SRv6 supported 3GPP PDU session types	27
8. Network Slicing Considerations	27
9. Control Plane Considerations	27
10. Security Considerations	28
11. IANA Considerations	28
12. Acknowledgements	29
13. Contributors	29
14. References	29

14.1. Normative References	29
14.2. Informative References	30
Appendix A. Implementations	32
Authors' Addresses	32

1. Introduction

In mobile networks, mobility systems provide connectivity over a wireless link to stationary and non-stationary nodes. The user-plane establishes a tunnel between the mobile node and its anchor node over IP-based backhaul and core networks.

This document specifies the applicability of SRv6 (Segment Routing IPv6) to mobile networks.

Segment Routing [RFC8402] is a source routing architecture: a node steers a packet through an ordered list of instructions called "segments". A segment can represent any instruction, topological or service based.

SRv6 applied to mobile networks enables a source-routing based mobile architecture, where operators can explicitly indicate a route for the packets to and from the mobile node. The SRv6 Endpoint nodes serve as mobile user-plane anchors.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.1. Terminology

- * CNF: Cloud-native Network Function
- * NFV: Network Function Virtualization
- * PDU: Packet Data Unit
- * PDU Session: Context of a UE connects to a mobile network.
- * UE: User Equipment
- * UPF: User Plane Function
- * VNF: Virtual Network Function (including CNFs)

The following terms used within this document are defined in [RFC8402]: Segment Routing, SR Domain, Segment ID (SID), SRv6, SRv6 SID, Active Segment, SR Policy, Prefix SID, Adjacency SID and Binding SID.

The following terms used within this document are defined in [RFC8754]: SRH, SR Source Node, Transit Node, SR Segment Endpoint Node and Reduced SRH.

The following terms used within this document are defined in [RFC8986]: NH, SL, FIB, SA, DA, SRv6 SID behavior, SRv6 Segment Endpoint Behavior.

2.2. Conventions

An SR Policy is resolved to a SID list. A SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit, and S3 is the last SID to visit along the SR path.

(SA,DA) (S3, S2, S1; SL) represents an IPv6 packet with:

- * Source Address is SA, Destination Address is DA, and next-header is SRH
- * SRH with SID list <S1, S2, S3> with Segments Left = SL
- * Note the difference between the <> and () symbols: <S1, S2, S3> represents a SID list where S1 is the first SID and S3 is the last SID to traverse. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID. When referring to an SR policy in a high-level use-case, it is simpler to use the <S1, S2, S3> notation. When referring to an illustration of the detailed packet behavior, the (S3, S2, S1; SL) notation is more convenient.
- * The payload of the packet is omitted.

SRH[n]: A shorter representation of Segment List[n], as defined in [RFC8754]. SRH[SL] can be different from the DA of the IPv6 header.

- * gNB::1 is an IPv6 address (SID) assigned to the gNB.
- * U1::1 is an IPv6 address (SID) assigned to UPF1.
- * U2::1 is an IPv6 address (SID) assigned to UPF2.
- * U2:: is the Locator of UPF2.

2.3. Predefined SRv6 Endpoint Behaviors

The following SRv6 Endpoint Behaviors are defined in [RFC8986].

- * End.DT4: Decapsulation and Specific IPv4 Table Lookup
- * End.DT6: Decapsulation and Specific IPv6 Table Lookup
- * End.DT46: Decapsulation and Specific IP Table Lookup
- * End.DX4: Decapsulation and IPv4 Cross-Connect
- * End.DX6: Decapsulation and IPv6 Cross-Connect
- * End.DX2: Decapsulation and L2 Cross-Connect

* End.T: Endpoint with specific IPv6 Table Lookup

This document defines new SRv6 Segment Endpoint Behaviors in Section 6.

3. Motivation

Mobile networks are becoming more challenging to operate. On one hand, traffic is constantly growing, and latency requirements are tighter; on the other-hand, there are new use-cases like distributed NFVi that are also challenging network operations.

The current architecture of mobile networks does not take into account the underlying transport. The user-plane is rigidly fragmented into radio access, core and service networks, connected by tunneling according to user-plane roles such as access and anchor nodes. These factors have made it difficult for the operator to optimize and operate the data-path.

In the meantime, applications have shifted to use IPv6, and network operators have started adopting IPv6 as their IP transport. SRv6, the IPv6 dataplane instantiation of Segment Routing [RFC8402], integrates both the application data-path and the underlying transport layer into a single protocol, allowing operators to optimize the network in a simplified manner and removing forwarding state from the network. It is also suitable for virtualized environments, like VNF/CNF to VNF/CNF networking. SRv6 has been deployed in dozens of networks [I-D.matsushima-spring-srv6-deployment-status].

SRv6 defines the network-programming concept [RFC8986]. Applied to mobility, SRv6 can provide the user-plane behaviors needed for mobility management. SRv6 takes advantage of the underlying transport awareness and flexibility together with the ability to also include services to optimize the end-to-end mobile dataplane.

The use-cases for SRv6 mobility are discussed in [I-D.camarilloelmalaky-springdmm-srv6-mob-usecases], and the architectural benefits are discussed in [I-D.kohno-dmm-srv6mob-arch].

4. 3GPP Reference Architecture

This section presents a reference architecture and possible deployment scenarios.

Figure 1 shows a reference diagram from the 5G packet core architecture [TS.23501].

The user plane described in this document does not depend on any specific architecture. The 5G packet core architecture as shown is based on the latest 3GPP standards at the time of writing this draft.

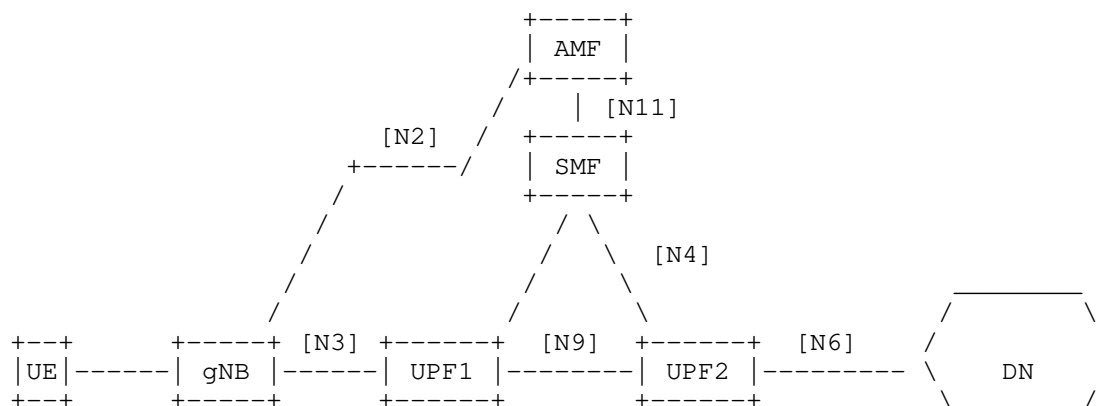


Figure 1: 3GPP 5G Reference Architecture

- * UE: User Endpoint
- * gNB: gNodeB with N3 interface towards packet core (and N2 for control plane)
- * UPF1: UPF with Interfaces N3 and N9 (and N4 for control plane)
- * UPF2: UPF with Interfaces N9 and N6 (and N4 for control plane)
- * SMF: Session Management Function
- * AMF: Access and Mobility Management Function
- * DN: Data Network e.g. operator services, Internet access

This reference diagram does not depict a UPF that is only connected to N9 interfaces, although the mechanisms defined in this document also work in such case.

Each session from a UE gets assigned to a UPF. Sometimes multiple UPFs may be used, providing richer service functions. A UE gets its IP address from the DHCP block of its UPF. The UPF advertises that IP address block toward the Internet, ensuring that return traffic is routed to the right UPF.

5. User-plane modes

This section introduces an SRv6 based mobile user-plane.

In order to simplify the adoption of SRv6, we present two different "modes" that vary with respect to the use of SRv6. The first one is the "Traditional mode", which inherits the current 3GPP mobile architecture. In this mode GTP-U protocol [TS.29281] is replaced by

SRv6, however the N3, N9 and N6 interfaces are still point-to-point interfaces with no intermediate waypoints as in the current mobile network architecture.

The second mode is the "Enhanced mode". This is an evolution from the "Traditional mode". In this mode the N3, N9 or N6 interfaces have intermediate waypoints -SIDs- that are used for Traffic Engineering or VNF purposes transparent to 3GPP functionalities. This results in optimal end-to-end policies across the mobile network with transport and services awareness.

In both, the Traditional and the Enhanced modes, we assume that the gNB as well as the UPFs are SR-aware (N3, N9 and -potentially- N6 interfaces are SRv6).

In addition to those two modes, we introduce two mechanisms for interworking with legacy access networks (those where the N3 interface is unmodified). In this document we introduce them as a variant to the Enhanced mode, however they are equally applicable to the Traditional mode.

One of these mechanisms is designed to interwork with legacy gNBs using GTP/IPv4. The second mechanism is designed to interwork with legacy gNBs using GTP/IPv6.

This document uses SRv6 Segment Endpoint Behaviors defined in [RFC8986] as well as new SRv6 Segment Endpoint Behaviors designed for the mobile user plane that are defined in this document in Section 6.

Note that the modes discussed throughout this section (with the exception of Section 5.4) only have informational purpose to implementors as well as operators deploying this technology. Indeed, it is expected that the operator defines his own operational model that best suits their needs.

5.1. Traditional mode

In the traditional mode, the existing mobile UPFs remain unchanged with the sole exception of the use of SRv6 as the data plane instead of GTP-U. There is no impact to the rest of the mobile system.

In existing 3GPP mobile networks, a PDU Session is mapped 1-for-1 with a specific GTP tunnel (TEID). This 1-for-1 mapping is mirrored here to replace GTP encapsulation with the SRv6 encapsulation, while not changing anything else. There will be a unique SRv6 SID associated with each PDU Session, and the SID list only contains a single SID.

The traditional mode minimizes the changes required to the mobile system; hence it is a good starting point for forming a common ground.

The gNB/UPF control-plane (N2/N4 interface) is unchanged, specifically a single IPv6 address is provided to the gNB. The same control plane signalling is used, and the gNB/UPF decides to use SRv6 based on signaled GTP-U parameters per local policy. The only information from the GTP-U parameters used for the SRv6 policy is the TEID, QFI, and the IPv6 Destination Address.

Our example topology is shown in Figure 2. The gNB and the UPFs are SR-aware. In the descriptions of the uplink and downlink packet flow, A is an IPv6 address of the UE, and Z is an IPv6 address reachable within the Data Network DN. A new SRv6 Endpoint Behavior, End.MAP, defined in Section 6.2, is used.

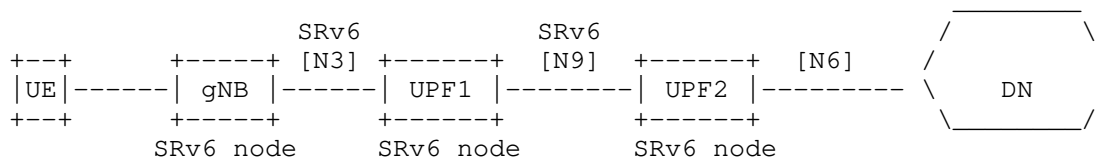


Figure 2: Traditional mode - example topology

5.1.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

UE_out   : (A,Z)
gNB_out  : (gNB, U1::1) (A,Z)    -> H.Encaps.Red <U1::1>
UPF1_out : (gNB, U2::1) (A,Z)    -> End.MAP
UPF2_out : (A,Z)                 -> End.DT4 or End.DT6

```

When the UE packet arrives at the gNB, the gNB performs a H.Encaps.Red operation. Since there is only one SID, there is no need to push an SRH. gNB only adds an outer IPv6 header with IPv6 DA U1::1. gNB obtains the SID U1::1 from the existing control plane (N2 interface). U1::1 represents an anchoring SID specific for that session at UPF1.

When the packet arrives at UPF1, the SID U1::1 is associated with the End.MAP SRv6 Endpoint Behavior. End.MAP replaces U1::1 by U2::1, that belongs to the next UPF (U2).

When the packet arrives at UPF2, the SID U2::1 corresponds to an End.DT4/End.DT6/End.DT46 SRv6 Endpoint Behavior. UPF2 decapsulates the packet, performs a lookup in a specific table associated with that mobile network and forwards the packet toward the data network (DN).

5.1.2. Packet flow - Downlink

The downlink packet flow is as follows:

```
UPF2_in : (Z,A)
UPF2_out: (U2::, U1::2) (Z,A)    -> H.Encaps.Red <U1::2>
UPF1_out: (U2::, gNB::1) (Z,A)   -> End.MAP
gNB_out  : (Z,A)                  -> End.DX4, End.DX6, End.DX2
```

When the packet arrives at the UPF2, the UPF2 maps that flow into a PDU Session. This PDU Session is associated with the segment endpoint <U1::2>. UPF2 performs a H.Encaps.Red operation, encapsulating the packet into a new IPv6 header with no SRH since there is only one SID.

Upon packet arrival on UPF1, the SID U1::2 is a local SID associated with the End.MAP SRv6 Endpoint Behavior. It maps the SID to the next anchoring point and replaces U1::2 by gNB::1, that belongs to the next hop.

Upon packet arrival on gNB, the SID gNB::1 corresponds to an End.DX4, End.DX6 or End.DX2 behavior (depending on the PDU Session Type). The gNB decapsulates the packet, removing the IPv6 header and all its extensions headers, and forwards the traffic toward the UE.

5.2. Enhanced mode

Enhanced mode improves scalability, provides traffic engineering capabilities, and allows service programming [I-D.ietf-spring-sr-service-programming], thanks to the use of multiple SIDs in the SID list (instead of a direct connectivity in between UPFs with no intermediate waypoints as in Traditional Mode).

Thus, the main difference is that the SR policy MAY include SIDs for traffic engineering and service programming in addition to the anchoring SIDs at UPFs.

Additionally in this mode the operator may choose to aggregate several devices under the same SID list (e.g., stationary residential meters connected to the same cell) to improve scalability.

When gNB transmits the packet, it contains all the segments of the SR policy. The SR policy includes segments for traffic engineering (C1) and for service programming (S1).

Nodes S1 and C1 perform their related Endpoint functionality and forward the packet.

When the packet arrives at UPF1, the active segment (U1::1) is an End.DT4/End.DT6/End.DT2U which performs the decapsulation (removing the IPv6 header with all its extension headers) and forwards toward the data network.

5.2.2. Packet flow - Downlink

The downlink packet flow is as follows:

```

UPF1_in : (Z,A)                                ->UPF1 maps the flow w/
                                                SID list <C1,S1, gNB>
UPF1_out: (U1::1, C1) (gNB::1, S1; SL=2) (Z,A) ->H.Encaps.Red
C1_out  : (U1::1, S1) (gNB::1, S1; SL=1) (Z,A)
S1_out  : (U1::1, gNB::1) (Z,A)                ->End with PSP
gNB_out : (Z,A)                                ->End.DX4/End.DX6/End.DX2

```

When the packet arrives at the UPF1, the UPF1 maps that particular flow into a UE PDU Session. This UE PDU Session is associated with the policy <C1, S1, gNB>. The UPF1 performs a H.Encaps.Red operation, encapsulating the packet into a new IPv6 header with its corresponding SRH.

The nodes C1 and S1 perform their related Endpoint processing.

Once the packet arrives at the gNB, the IPv6 DA corresponds to an End.DX4, End.DX6 or End.DX2 behavior at the gNB (depending on the underlying traffic). The gNB decapsulates the packet, removing the IPv6 header, and forwards the traffic towards the UE. The SID gNB::1 is one example of a SID associated to this service.

Note that there are several means to provide the UE session aggregation. The decision on which one to use is a local decision made by the operator. One option is to use the Args.Mob.Session (Section 6.1). Another option comprises the gNB performing an IP lookup on the inner packet by using the End.DT4, End.DT6, and End.DT2 behaviors.

5.2.3. Scalability

The Enhanced Mode improves since it allows the aggregation of several UEs under the same SID list. For example, in the case of stationary residential meters that are connected to the same cell, all such devices can share the same SID list. This improves scalability compared to Traditional Mode (unique SID per UE) and compared to GTP-U (dedicated TEID per UE).

5.3. Enhanced mode with unchanged gNB GTP behavior

This section describes two mechanisms for interworking with legacy gNBs that still use GTP: one for IPv4, and another for IPv6.

In the interworking scenarios as illustrated in Figure 4, the gNB does not support SRv6. The gNB supports GTP encapsulation over IPv4 or IPv6. To achieve interworking, an SR Gateway (SRGW) entity is added. The SRGW maps the GTP traffic into SRv6.

The SRGW is not an anchor point and maintains very little state. For this reason, both IPv4 and IPv6 methods scale to millions of UEs.

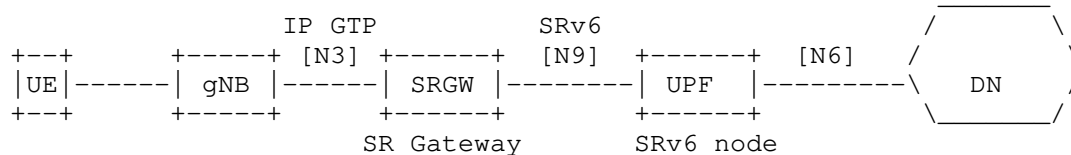


Figure 4: Example topology for interworking

Both of the mechanisms described in this section are applicable to either the Traditional Mode or the Enhanced Mode.

5.3.1. Interworking with IPv6 GTP

In this interworking mode the gNB at the N3 interface uses GTP over IPv6.

Key points:

- * The gNB is unchanged (control-plane or user-plane) and encapsulates into GTP (N3 interface is not modified).
- * The 5G Control-Plane towards the gNB (N2 interface) is unmodified, though multiple UPF addresses need to be used - one IPv6 address (i.e. a BSID at the SRGW) is needed per <SLA, PDU session type>. The SRv6 SID is different depending on the required <SLA, PDU session type> combination.

- * In the uplink, the SRGW removes GTP, finds the SID list related to the IPv6 DA, and adds SRH with the SID list.
- * There is no state for the downlink at the SRGW.
- * There is simple state in the uplink at the SRGW; using Enhanced mode results in fewer SR policies on this node. An SR policy is shared across UEs as long as they belong to the same context (i.e., tenant). A set of many different policies (i.e., different SLAs) increases the amount of state required.
- * When a packet from the UE leaves the gNB, it is SR-routed. This simplifies network slicing [I-D.ietf-lsr-flex-algo].
- * In the uplink, the SRv6 BSID steers traffic into an SR policy when it arrives at the SRGW.

An example topology is shown in Figure 5.

S1 and C1 are two service segments. S1 represents a VNF in the network, and C1 represents a router configured for Traffic Engineering.

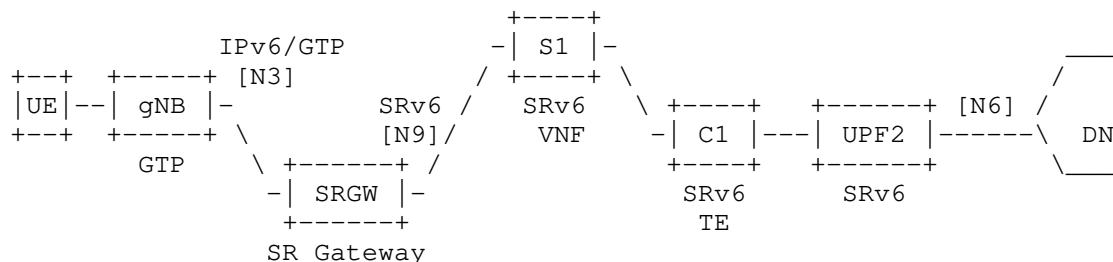


Figure 5: Enhanced mode with unchanged gNB IPv6/GTP behavior

5.3.1.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

UE_out   : (A,Z)
gNB_out  : (gNB, B) (GTP: TEID T) (A,Z)      -> Interface N3 unmodified
                                                (IPv6/GTP)
SRGW_out : (SRGW, S1) (U2::T, C1; SL=2) (A,Z) -> B is an End.M.GTP6.D
                                                SID at the SRGW
S1_out   : (SRGW, C1) (U2::T, C1; SL=1) (A,Z)
C1_out   : (SRGW, U2::T) (A,Z)                -> End with PSP
UPF2_out : (A,Z)                             -> End.DT4 or End.DT6

```

The UE sends a packet destined to Z toward the gNB on a specific bearer for that session. The gNB, which is unmodified, encapsulates the packet into IPv6, UDP, and GTP headers. The IPv6 DA B, and the GTP TEID T are the ones received in the N2 interface.

The IPv6 address that was signaled over the N2 interface for that UE PDU Session, B, is now the IPv6 DA. B is an SRv6 Binding SID at the SRGW. Hence the packet is routed to the SRGW.

When the packet arrives at the SRGW, the SRGW identifies B as an End.M.GTP6.D Binding SID (see Section 6.3). Hence, the SRGW removes the IPv6, UDP, and GTP headers, and pushes an IPv6 header with its own SRH containing the SIDs bound to the SR policy associated with this BindingSID. There at least one instance of the End.M.GTP6.D SID per PDU type.

S1 and C1 perform their related Endpoint functionality and forward the packet.

When the packet arrives at UPF2, the active segment is (U2::T) which is bound to End.DT4/6. UPF2 then decapsulates (removing the outer IPv6 header with all its extension headers) and forwards the packet toward the data network.

5.3.1.2. Packet flow - Downlink

The downlink packet flow is as follows:

```
UPF2_in : (Z,A)                                -> UPF2 maps the flow with
                                                <C1, S1, SRGW::TEID,gNB>
UPF2_out: (U2::1, C1)(gNB, SRGW::TEID, S1; SL=3)(Z,A) -> H.Encaps.Red
C1_out   : (U2::1, S1)(gNB, SRGW::TEID, S1; SL=2)(Z,A)
S1_out   : (U2::1, SRGW::TEID)(gNB, SRGW::TEID, S1, SL=1)(Z,A)
SRGW_out : (SRGW, gNB)(GTP: TEID=T)(Z,A)      -> SRGW/96 is End.M.GTP6.E
gNB_out  : (Z,A)
```

When a packet destined to A arrives at the UPF2, the UPF2 performs a lookup in the table associated to A and finds the SID list <C1, S1, SRGW::TEID, gNB>. The UPF2 performs an H.Encaps.Red operation, encapsulating the packet into a new IPv6 header with its corresponding SRH.

C1 and S1 perform their related Endpoint processing.

Once the packet arrives at the SRGW, the SRGW identifies the active SID as an End.M.GTP6.E function. The SRGW removes the IPv6 header and all its extensions headers. The SRGW generates new IPv6, UDP, and GTP headers. The new IPv6 DA is the gNB which is the last SID in the received SRH. The TEID in the generated GTP header is an argument of the received End.M.GTP6.E SID. The SRGW pushes the headers to the packet and forwards the packet toward the gNB. There is one instance of the End.M.GTP6.E SID per PDU type.

Once the packet arrives at the gNB, the packet is a regular IPv6/GTP packet. The gNB looks for the specific radio bearer for that TEID and forward it on the bearer. This gNB behavior is not modified from current and previous generations.

5.3.1.3. Scalability

For the downlink traffic, the SRGW is stateless. All the state is in the SRH pushed by the UPF2. The UPF2 must have the UE states since it is the UE's session anchor point.

For the uplink traffic, the state at the SRGW does not necessarily need to be unique per PDU Session; the SR policy can be shared among UEs. This enables more scalable SRGW deployments compared to a solution holding millions of states, one or more per UE.

5.3.2. Interworking with IPv4 GTP

In this interworking mode the gNB uses GTP over IPv4 in the N3 interface

Key points:

- * The gNB is unchanged and encapsulates packets into GTP (the N3 interface is not modified).
- * N2 signaling is not changed, though multiple UPF addresses need to be provided – one for each PDU Session Type.
- * In the uplink, traffic is classified by SRGW's classification engine and steered into an SR policy. The SRGW may be implemented in a UPF or as a separate entity. How the classification engine rules are set up is outside the scope of this document, though one example is using BGP signaling from a Mobile User Plane Controller [I-D.mhkk-dmm-srv6mup-architecture].
- * SRGW removes GTP, finds the SID list related to DA, and adds an SRH with the SID list.

An example topology is shown in Figure 6. In this mode the gNB is an unmodified gNB using IPv4/GTP. The UPFs are SR-aware. As before, the SRGW maps the IPv4/GTP traffic to SRv6.

S1 and C1 are two service segment endpoints. S1 represents a VNF in the network, and C1 represents a router configured for Traffic Engineering.

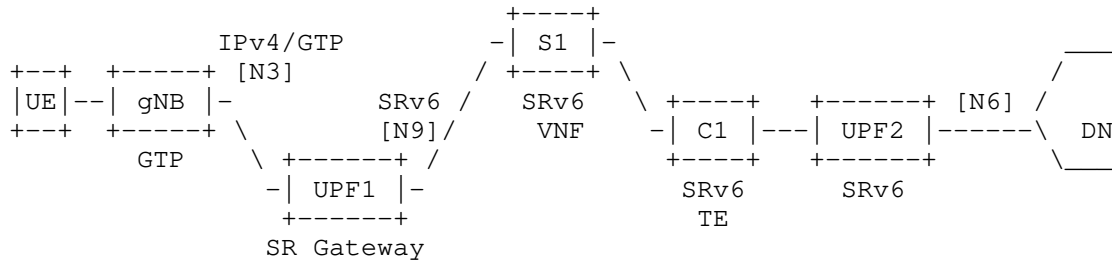


Figure 6: Enhanced mode with unchanged gNB IPv4/GTP behavior

5.3.2.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

gNB_out : (gNB, B) (GTP: TEID T) (A, Z)      -> Interface N3
                                              unchanged IPv4/GTP
SRGW_out: (SRGW, S1) (U2::1, C1; SL=2) (A, Z) -> H.M.GTP4.D function
S1_out  : (SRGW, C1) (U2::1, C1; SL=1) (A, Z)
C1_out  : (SRGW, U2::1) (A, Z)                -> PSP
UPF2_out: (A, Z)                             -> End.DT4 or End.DT6

```

The UE sends a packet destined to Z toward the gNB on a specific bearer for that session. The gNB, which is unmodified, encapsulates the packet into a new IPv4, UDP, and GTP headers. The IPv4 DA, B, and the GTP TEID are the ones received at the N2 interface.

When the packet arrives at the SRGW for UPF1, the SRGW has an classification engine rule for incoming traffic from the gNB, that steers the traffic into an SR policy by using the function H.M.GTP4.D. The SRGW removes the IPv4, UDP, and GTP headers and pushes an IPv6 header with its own SRH containing the SIDs related to the SR policy associated with this traffic. The SRGW forwards according to the new IPv6 DA.

S1 and C1 perform their related Endpoint functionality and forward the packet.

When the packet arrives at UPF2, the active segment is (U2::1) which is bound to End.DT4/6 which performs the decapsulation (removing the outer IPv6 header with all its extension headers) and forwards toward the data network.

Note that the interworking mechanisms for IPv4/GTP and IPv6/GTP differs. This is due to the fact that in IPv6/GTP we can leverage the remote steering capabilities provided by the Segment Routing BSID. In IPv4 this construct is not available, and building a similar mechanism would require a significant address consumption.

5.3.2.2. Packet flow - Downlink

The downlink packet flow is as follows:

```

UPF2_in : (Z,A)                                -> UPF2 maps flow with SID
                                                <C1, S1,GW::SA:DA:TEID>
UPF2_out: (U2::1, C1) (GW::SA:DA:TEID, S1; SL=2) (Z,A) ->H.Encaps.Red
C1_out   : (U2::1, S1) (GW::SA:DA:TEID, S1; SL=1) (Z,A)
S1_out   : (U2::1, GW::SA:DA:TEID) (Z,A)
SRGW_out: (GW, gNB) (GTP: TEID=T) (Z,A)         -> End.M.GTP4.E
gNB_out  : (Z,A)

```

When a packet destined to A arrives at the UPF2, the UPF2 performs a lookup in the table associated to A and finds the SID list <C1, S1, SRGW::SA:DA:TEID>. The UPF2 performs a H.Encaps.Red operation, encapsulating the packet into a new IPv6 header with its corresponding SRH.

The nodes C1 and S1 perform their related Endpoint processing.

Once the packet arrives at the SRGW, the SRGW identifies the active SID as an End.M.GTP4.E function. The SRGW removes the IPv6 header and all its extensions headers. The SRGW generates an IPv4, UDP, and GTP headers. The IPv4 SA and DA are received as SID arguments. The TEID in the generated GTP header is also the arguments of the received End.M.GTP4.E SID. The SRGW pushes the headers to the packet and forwards the packet toward the gNB.

When the packet arrives at the gNB, the packet is a regular IPv4/GTP packet. The gNB looks for the specific radio bearer for that TEID and forwards it on the bearer. This gNB behavior is not modified from current and previous generations.

5.3.2.3. Scalability

For the downlink traffic, the SRGW is stateless. All the state is in the SRH pushed by the UPF2. The UPF must have this UE-base state anyway (since it is its anchor point).

For the uplink traffic, the state at the SRGW is dedicated on a per UE/session basis according to a classification engine. There is state for steering the different sessions in the form of an SR Policy. However, SR policies are shared among several UE/sessions.

5.3.3. Extensions to the interworking mechanisms

In this section we presented two mechanisms for interworking with gNBs and UPFs that do not support SRv6. These mechanisms are used to support GTP over IPv4 and IPv6.

Even though we have presented these methods as an extension to the "Enhanced mode", it is straightforward in its applicability to the "Traditional mode".

5.4. SRv6 Drop-in Interworking

In this section we introduce another mode useful for legacy gNB and UPFs that still operate with GTP-U. This mode provides an SRv6-enabled user plane in between two GTP-U tunnel endpoints.

In this mode we employ two SRGWs that map GTP-U traffic to SRv6 and vice-versa.

Unlike other interworking modes, in this mode both of the mobility overlay endpoints use GTP-U. Two SRGWs are deployed in either N3 or N9 interface to realize an intermediate SR policy.

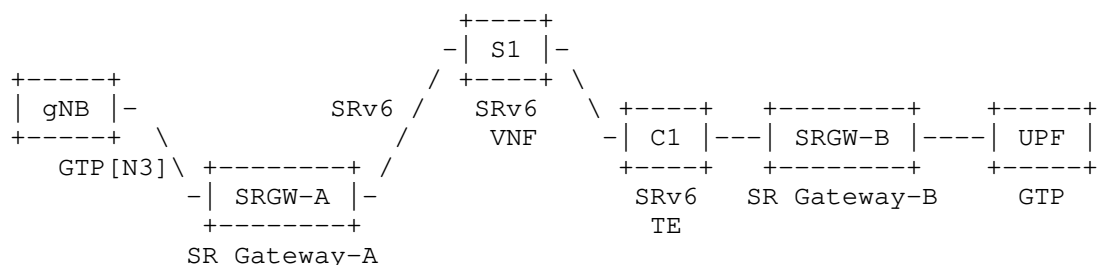


Figure 7: Example topology for SRv6 Drop-in mode

The packet flow of Figure 7 is as follows:

```

gNB_out : (gNB, U::1) (GTP: TEID T) (A,Z)
GW-A_out: (GW-A, S1) (U::1, SGB::TEID, C1; SL=3) (A,Z) ->U::1 is an
                                                    End.M.GTP6.D.Di
                                                    SID at SRGW-A
S1_out   : (GW-A, C1) (U::1, SGB::TEID, C1; SL=2) (A,Z)
C1_out   : (GW-A, SGB::TEID) (U::1, SGB::TEID, C1; SL=1) (A,Z)
GW-B_out: (GW-B, U::1) (GTP: TEID T) (A,Z) ->SGB::TEID is an
                                                    End.M.GTP6.E
                                                    SID at SRGW-B
UPF_out  : (A,Z)

```

When a packet destined to Z is sent to the gNB, which is unmodified (control-plane and user-plane remain GTP-U), gNB performs encapsulation into a new IP, UDP, and GTP headers. The IPv6 DA, U::1, and the GTP TEID are the ones received at the N2 interface.

The IPv6 address that was signaled over the N2 interface for that PDU Session, U::1, is now the IPv6 DA. U::1 is an SRv6 Binding SID at SRGW-A. Hence the packet is routed to the SRGW.

When the packet arrives at SRGW-A, the SRGW identifies U::1 as an End.M.GTP6.D.Di Binding SID (see Section 6.4). Hence, the SRGW removes the IPv6, UDP, and GTP headers, and pushes an IPv6 header with its own SRH containing the SIDs bound to the SR policy associated with this Binding SID. There is one instance of the End.M.GTP6.D.Di SID per PDU type.

S1 and C1 perform their related Endpoint functionality and forward the packet.

Once the packet arrives at SRGW-B, the SRGW identifies the active SID as an End.M.GTP6.E function. The SRGW removes the IPv6 header and all its extensions headers. The SRGW generates new IPv6, UDP, and GTP headers. The new IPv6 DA is U::1 which is the last SID in the received SRH. The TEID in the generated GTP header is an argument of the received End.M.GTP6.E SID. The SRGW pushes the headers to the packet and forwards the packet toward UPF. There is one instance of the End.M.GTP6.E SID per PDU type.

Once the packet arrives at UPF, the packet is a regular IPv6/GTP packet. The UPF looks for the specific rule for that TEID to forward the packet. This UPF behavior is not modified from current and previous generations.

6. SRv6 Segment Endpoint Mobility Behaviors

6.1. Args.Mob.Session

Args.Mob.Session provide per-session information for charging, buffering and lawful intercept (among others) required by some mobile nodes. The Args.Mob.Session argument format is used in combination with End.Map, End.DT4/End.DT6/End.DT46 and End.DX4/End.DX6/End.DX2 behaviors. Note that proposed format is applicable for 5G networks, while similar formats could be used for legacy networks.

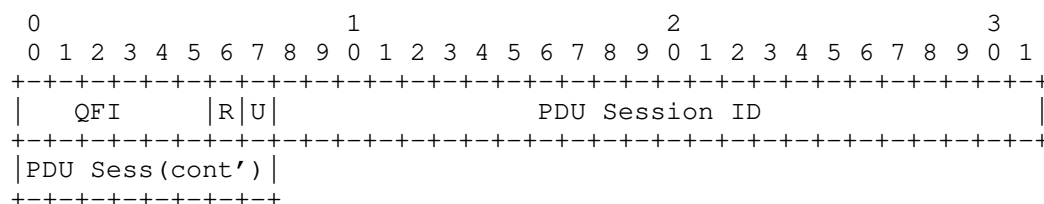


Figure 8: Args.Mob.Session format

- * QFI: QoS Flow Identifier [TS.38415]
- * R: Reflective QoS Indication [TS.23501]. This parameter indicates the activation of reflective QoS towards the UE for the transferred packet. Reflective QoS enables the UE to map UL User Plane traffic to QoS Flows without SMF provided QoS rules.
- * U: Unused and for future use. MUST be 0 on transmission and ignored on receipt.
- * PDU Session ID: Identifier of PDU Session. The GTP-U equivalent is TEID.

Arg.Mob.Session is required in case that one SID aggregates multiple PDU Sessions. Since the SRv6 SID is likely NOT to be instantiated per PDU session, Args.Mob.Session helps the UPF to perform the behaviors which require per QFI and/or per PDU Session granularity.

Note that the encoding of user-plane messages (e.g., Echo Request, Echo Reply, Error Indication and End Marker) is out of the scope of this draft. [I-D.murakami-dmm-user-plane-message-encoding] defines one possible encoding.

6.2. End.MAP

The "Endpoint behavior with SID mapping" behavior (End.MAP for short) is used in several scenarios. Particularly in mobility, End.MAP is used by the intermediate UPFs.

When node N receives a packet whose IPv6 DA is D and D is a local End.MAP SID, N does:


```
S01. If (IPv6 Hop Limit <= 1) {
S02.   Send an ICMP Time Exceeded message to the Source Address,
      Code 0 (Hop limit exceeded in transit),
      interrupt packet processing, and discard the packet.
S03. }
S04. Decrement IPv6 Hop Limit by 1
S05. Update the IPv6 DA with the new mapped SID
S06. Submit the packet to the egress IPv6 FIB lookup for
      transmission to the new destination
```

Notes: The SIDs in the SRH are not modified.

6.3. End.M.GTP6.D

The "Endpoint behavior with IPv6/GTP decapsulation into SR policy" behavior (End.M.GTP6.D for short) is used in interworking scenario for the uplink towards SRGW from the legacy gNB using IPv6/GTP. Any SID instance of this behavior is associated with an SR Policy B and an IPv6 Source Address S.

When the SR Gateway node N receives a packet destined to D and D is a local End.M.GTP6.D SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.     Send an ICMP Parameter Problem to the Source Address,
      Code 0 (Erroneous header field encountered),
      Pointer set to the Segments Left field,
      interrupt packet processing, and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an End.M.GTP6.D SID, N does:

```
S01. If (Next Header (NH) == UDP & UDP_Dest_port == GTP) {
S02.   Copy the GTP TEID and QFI to buffer memory
S03.   Pop the IPv6, UDP, and GTP Headers
S04.   Push a new IPv6 header with its own SRH containing B
S05.   Set the outer IPv6 SA to S
S06.   Set the outer IPv6 DA to the first SID of B
S07.   Set the outer Payload Length, Traffic Class, Flow Label,
       Hop Limit, and Next-Header (NH) fields
S08.   Write in the SRH[0] the Args.Mob.Session based on
       the information of buffer memory
S09.   Submit the packet to the egress IPv6 FIB lookup and
       transmission to the new destination
S10. } Else {
S11.   Process as per [RFC8986] Section 4.1.1
S12. }
```

Notes: S07. The NH is set based on the SID parameter. There is one instantiation of the End.M.GTP6.D SID per PDU Session Type, hence the NH is already known in advance. For the IPv4v6 PDU Session Type, in addition we inspect the first nibble of the PDU to know the NH value.

The last segment (S3 in above example) SHOULD be followed by an Arg.Mob.Session argument space which is used to provide the session identifiers.

6.4. End.M.GTP6.D.Di

The "Endpoint behavior with IPv6/GTP decapsulation into SR policy for Drop-in Mode" behavior (End.M.GTP6.D.Di for short) is used in SRv6 drop-in interworking scenario described in Section 5.4. The difference between End.M.GTP6.D as another variant of IPv6/GTP decapsulation function is that the original IPv6 DA of GTP packet is preserved as the last SID in SRH.

Any SID instance of this behavior is associated with an SR Policy B and an IPv6 Source Address S.

When the SR Gateway node N receives a packet destined to D and D is a local End.M.GTP6.D.Di SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.     Send an ICMP Parameter Problem to the Source Address,
        Code 0 (Erroneous header field encountered),
        Pointer set to the Segments Left field,
        interrupt packet processing, and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an End.M.GTP6.Di SID, N does:

```
S01. If (Next Header = UDP & UDP_Dest_port = GTP) {
S02.   Copy D to buffer memory
S03.   Pop the IPv6, UDP, and GTP Headers
S04.   Push a new IPv6 header with its own SRH containing B
S05.   Set the outer IPv6 SA to S
S06.   Set the outer IPv6 DA to the first SID of B
S07.   Set the outer Payload Length, Traffic Class, Flow Label,
        Hop Limit, and Next-Header fields
S08.   Prepend D to the SRH (as SRH[0]) and set SL accordingly
S09.   Submit the packet to the egress IPv6 FIB lookup and
        transmission to the new destination
S10. } Else {
S11.   Process as per [RFC8986] Section 4.1.1
S12. }
```

Notes: S07. The NH is set based on the SID parameter. There is one instantiation of the End.M.GTP6.D SID per PDU Session Type, hence the NH is already known in advance. For the IPv4v6 PDU Session Type, in addition we inspect the first nibble of the PDU to know the NH value.

S SHOULD be an End.M.GTP6.E SID instantiated at the SR gateway.

6.5. End.M.GTP6.E

The "Endpoint behavior with encapsulation for IPv6/GTP tunnel" behavior (End.M.GTP6.E for short) is used among others in the interworking scenario for the downlink toward the legacy gNB using IPv6/GTP.

The prefix of End.M.GTP6.E SID MUST be followed by the Arg.Mob.Session argument space which is used to provide the session identifiers.

When the SR Gateway node N receives a packet destined to D, and D is a local End.M.GTP6.E SID, N does the following:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 1) {
S03.     Send an ICMP Parameter Problem to the Source Address,
        Code 0 (Erroneous header field encountered),
        Pointer set to the Segments Left field,
        interrupt packet processing, and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an End.M.GTP6.E SID, N does:

```
S01.   Copy SRH[0] and D to buffer memory
S02.   Pop the IPv6 header and all its extension headers
S03.   Push a new IPv6 header with a UDP/GTP Header
S04.   Set the outer IPv6 SA to S
S05.   Set the outer IPv6 DA from buffer memory
S06.   Set the outer Payload Length, Traffic Class, Flow Label,
        Hop Limit, and Next-Header fields
S07.   Set the GTP TEID (from buffer memory)
S08.   Submit the packet to the egress IPv6 FIB lookup and
        transmission to the new destination
```

Notes: An End.M.GTP6.E SID MUST always be the penultimate SID. The TEID is extracted from the argument space of the current SID.

The source address S SHOULD be an End.M.GTP6.D SID instantiated at an SR gateway.

6.6. End.M.GTP4.E

The "Endpoint behavior with encapsulation for IPv4/GTP tunnel" behavior (End.M.GTP4.E for short) is used in the downlink when doing interworking with legacy gNB using IPv4/GTP.

When the SR Gateway node N receives a packet destined to S and S is a local End.M.GTP4.E SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.     Send an ICMP Parameter Problem to the Source Address,
        Code 0 (Erroneous header field encountered),
        Pointer set to the Segments Left field,
        interrupt packet processing, and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an End.M.GTP4.E SID, N does:

- S01. Store the IPv6 DA and SA in buffer memory
- S02. Pop the IPv6 header and all its extension headers
- S03. Push a new IPv4 header with a UDP/GTP Header
- S04. Set the outer IPv4 SA and DA (from buffer memory)
- S05. Set the outer Total Length, DSCP, Time To Live, and Next-Header fields
- S06. Set the GTP TEID (from buffer memory)
- S07. Submit the packet to the egress IPv6 FIB lookup and transmission to the new destination

Notes: The End.M.GTP4.E SID in S has the following format:

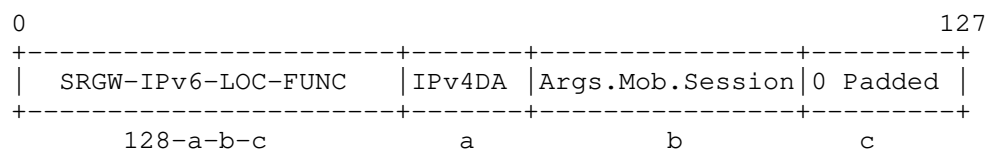


Figure 9: End.M.GTP4.E SID Encoding

The IPv6 Source Address has the following format:

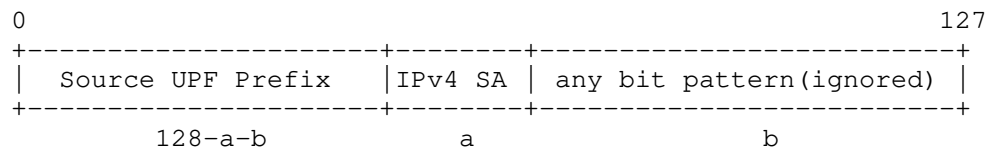


Figure 10: IPv6 SA Encoding for End.M.GTP4.E

6.7. H.M.GTP4.D

The "SR Policy Headend with tunnel decapsulation and map to an SRv6 policy" behavior (H.M.GTP4.D for short) is used in the direction from legacy IPv4 user-plane to SRv6 user-plane network.

When the SR Gateway node N receives a packet destined to a IW-IPv4-Prefix, N does:

```

S01. IF Payload == UDP/GTP THEN
S02.   Pop the outer IPv4 header and UDP/GTP headers
S03.   Copy IPv4 DA, TEID to form SID B
S04.   Copy IPv4 SA to form IPv6 SA B'
S05.   Encapsulate the packet into a new IPv6 header   ;;Ref1
S06.   Set the IPv6 DA = B
S07.   Forward along the shortest path to B
S08. ELSE
S09.   Drop the packet

```

Ref1: The NH value is identified by inspecting the first nibble of the inner payload.

The SID B has the following format:

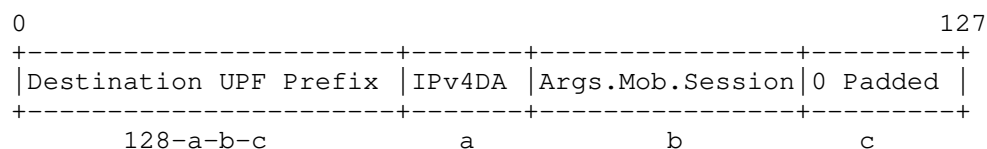


Figure 11: H.M.GTP4.D SID Encoding

The SID B MAY be an SRv6 Binding SID instantiated at the first UPF (U1) to bind an SR policy [I-D.ietf-spring-segment-routing-policy].

6.8. End.Limit: Rate Limiting behavior

The mobile user-plane requires a rate-limit feature. For this purpose, we define a new behavior "End.Limit". The "End.Limit" behavior encodes in its arguments the rate limiting parameter that should be applied to this packet. Multiple flows of packets should have the same group identifier in the SID when those flows are in the same AMBR (Aggregate Maximum Bit Rate) group. The encoding format of the rate limit segment SID is as follows:

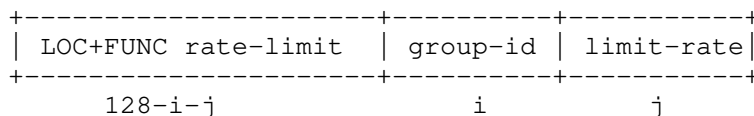


Figure 12: End.Limit: Rate limiting behavior argument format

If the limit-rate bits are set to zero, the node should not do rate limiting unless static configuration or control-plane sets the limit rate associated to the SID.

7. SRv6 supported 3GPP PDU session types

The 3GPP [TS.23501] defines the following PDU session types:

- * IPv4
- * IPv6
- * IPv4v6
- * Ethernet
- * Unstructured

SRv6 supports the 3GPP PDU session types without any protocol overhead by using the corresponding SRv6 behaviors (End.DX4, End.DT4 for IPv4 PDU sessions; End.DX6, End.DT6, End.T for IPv6 PDU sessions; End.DT46 for IPv4v6 PDU sessions; End.DX2 for L2 and Unstructured PDU sessions).

8. Network Slicing Considerations

A mobile network may be required to implement "network slices", which logically separate network resources. User-plane behaviors represented as SRv6 segments would be part of a slice.

[I-D.ietf-spring-segment-routing-policy] describes a solution to build basic network slices with SR. Depending on the requirements, these slices can be further refined by adopting the mechanisms from:

- * IGP Flex-Algo [I-D.ietf-lsr-flex-algo]
- * Inter-Domain policies
[I-D.ietf-spring-segment-routing-central-epe]

Furthermore, these can be combined with ODN/AS (On Demand Nexthop/ Automated Steering) [I-D.ietf-spring-segment-routing-policy] for automated slice provisioning and traffic steering.

Further details on how these tools can be used to create end to end network slices are documented in [I-D.ali-spring-network-slicing-building-blocks].

9. Control Plane Considerations

This document focuses on user-plane behavior and its independence from the control plane. While the SRv6 mobile user-plane behaviors may be utilized in emerging architectures, such as [I-D.gundavelli-dmm-mfa], [I-D.mhkk-dmm-srv6mup-architecture] for example, require control plane support for the user-plane, this document does not impose any change to the existent mobility control plane.

Section 11 allocates SRv6 Segment Endpoint Behavior codepoints for the new behaviors defined in this document.

10. Security Considerations

The security considerations for Segment Routing are discussed in [RFC8402]. More specifically for SRv6 the security considerations and the mechanisms for securing an SR domain are discussed in [RFC8754]. Together, they describe the required security mechanisms that allow establishment of an SR domain of trust to operate SRv6-based services for internal traffic while preventing any external traffic from accessing or exploiting the SRv6-based services.

The technology described in this document is applied to a mobile network that is within the SR Domain.

This document introduces new SRv6 Endpoint Behaviors. Those behaviors do not need any special security consideration given that it is deployed within that SR Domain.

11. IANA Considerations

The following values have been allocated within the "SRv6 Endpoint Behaviors" [RFC8986] sub-registry belonging to the top-level "Segment Routing Parameters" registry:

Value	Hex	Endpoint behavior	Reference
40	0x0028	End.MAP	[This.ID]
41	0x0029	End.Limit	[This.ID]
69	0x0045	End.M.GTP6.D	[This.ID]
70	0x0046	End.M.GTP6.Di	[This.ID]
71	0x0047	End.M.GTP6.E	[This.ID]
72	0x0048	End.M.GTP4.E	[This.ID]

Table 1: SRv6 Mobile User-plane Endpoint Behavior Types

12. Acknowledgements

The authors would like to thank Daisuke Yokota, Bart Peirens, Ryokichi Onishi, Kentaro Ebisawa, Peter Bosch, Darren Dukes, Francois Clad, Sri Gundavelli, Sridhar Bhaskaran, Arashmid Akhavain, Ravi Shekhar, Aeneas Dodd-Noble, Carlos Jesus Bernardos, Dirk v. Hugo and Jeffrey Zhang for their useful comments of this work.

13. Contributors

Kentaro Ebisawa Toyota Motor Corporation Japan

Email: ebisawa@toyota-tokyo.tech

Tetsuya Murakami Arrcus, Inc. United States of America

Email: tetsuya.ietf@gmail.com

14. References

14.1. Normative References

- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-22, 22 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-segment-routing-policy-22>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [TS.23501] 3GPP, "System Architecture for the 5G System", 3GPP TS 23.501 15.0.0, November 2017.

14.2. Informative References

- [I-D.ali-spring-network-slicing-building-blocks]
Ali, Z., Filsfils, C., Camarillo, P., and D. Voyer,
"Building blocks for Slicing in Segment Routing Network",
Work in Progress, Internet-Draft, draft-ali-spring-
network-slicing-building-blocks-04, 21 February 2021,
<<https://datatracker.ietf.org/doc/html/draft-ali-spring-network-slicing-building-blocks-04>>.
- [I-D.camarilloelmalky-springdmm-srv6-mob-usecases]
Garvia, P. C., Filsfils, C., Elmalky, H., Matsushima, S.,
Voyer, D., Cui, A., and B. Peirens, "SRv6 Mobility Use-
Cases", Work in Progress, Internet-Draft, draft-
camarilloelmalky-springdmm-srv6-mob-usecases-02, 15 August
2019, <<https://datatracker.ietf.org/doc/html/draft-camarilloelmalky-springdmm-srv6-mob-usecases-02>>.
- [I-D.gundavelli-dmm-mfa]
Gundavelli, S., Liebsch, M., and S. Matsushima, "Mobility-
aware Floating Anchor (MFA)", Work in Progress, Internet-
Draft, draft-gundavelli-dmm-mfa-01, 19 September 2018,
<<https://datatracker.ietf.org/doc/html/draft-gundavelli-dmm-mfa-01>>.
- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and
A. Gulko, "IGP Flexible Algorithm", Work in Progress,
Internet-Draft, draft-ietf-lsr-flex-algo-19, 7 April 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-lsr-flex-algo-19>>.
- [I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D.
Afanasyev, "Segment Routing Centralized BGP Egress Peer
Engineering", Work in Progress, Internet-Draft, draft-
ietf-spring-segment-routing-central-epe-10, 21 December
2017, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-segment-routing-central-epe-10>>.

- [I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C.,
Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and
S. Salsano, "Service Programming with Segment Routing",
Work in Progress, Internet-Draft, draft-ietf-spring-sr-
service-programming-05, 10 September 2021,
<<https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-05>>.
- [I-D.kohno-dmm-srv6mob-arch]
Kohno, M., Clad, F., Camarillo, P., and Z. Ali,
"Architecture Discussion on SRv6 Mobile User plane", Work
in Progress, Internet-Draft, draft-kohno-dmm-srv6mob-arch-
05, 8 November 2021,
<<https://datatracker.ietf.org/doc/html/draft-kohno-dmm-srv6mob-arch-05>>.
- [I-D.matsushima-spring-srv6-deployment-status]
Matsushima, S., Filsfils, C., Ali, Z., Li, Z., Rajaraman,
K., and A. Dhamija, "SRv6 Implementation and Deployment
Status", Work in Progress, Internet-Draft, draft-
matsushima-spring-srv6-deployment-status-15, 5 April 2022,
<<https://datatracker.ietf.org/doc/html/draft-matsushima-spring-srv6-deployment-status-15>>.
- [I-D.mhkk-dmm-srv6mup-architecture]
Matsushima, S., Horiba, K., Khan, A., Kawakami, Y.,
Murakami, T., Patel, K., Kohno, M., Kamata, T., Garvia, P.
C., Voyer, D., Zadok, S., Meilik, I., Agrawal, A.,
Perumal, K., and J. Horn, "Segment Routing IPv6 Mobile
User Plane Architecture for Distributed Mobility
Management", Work in Progress, Internet-Draft, draft-mhkk-
dmm-srv6mup-architecture-03, 20 March 2022,
<<https://datatracker.ietf.org/doc/html/draft-mhkk-dmm-srv6mup-architecture-03>>.
- [I-D.murakami-dmm-user-plane-message-encoding]
Murakami, T., Matsushima, S., Ebisawa, K., Camarillo, P.,
and R. Shekhar, "User Plane Message Encoding", Work in
Progress, Internet-Draft, draft-murakami-dmm-user-plane-
message-encoding-05, 5 March 2022,
<<https://datatracker.ietf.org/doc/html/draft-murakami-dmm-user-plane-message-encoding-05>>.
- [TS.29281] 3GPP, "General Packet Radio System (GPRS) Tunnelling
Protocol User Plane (GTPv1-U)", 3GPP TS 29.281 15.1.0,
December 2017.

[TS.38415] 3GPP, "Draft Specification for 5GS container (TS 38.415)",
3GPP R3-174510 0.0.0, August 2017.

Appendix A. Implementations

This document introduces new SRv6 Endpoint Behaviors. These behaviors have an open-source P4 implementation available in <https://github.com/ebiken/p4srv6>.

Additionally, a full implementation of this document is available in Linux Foundation FD.io VPP project since release 20.05. More information available here: https://docs.fd.io/vpp/20.05/d7/d3c/srv6_mobile_plugin_doc.html.

There are also experimental implementations in M-CORD NGIC and Open Air Interface (OAI).

Authors' Addresses

Satoru Matsushima (editor)
SoftBank
Japan
Email: satoru.matsushima@g.softbank.co.jp

Clarence Filsfils
Cisco Systems, Inc.
Belgium
Email: cf@cisco.com

Miya Kohno
Cisco Systems, Inc.
Japan
Email: mkohno@cisco.com

Pablo Camarillo Garvia (editor)
Cisco Systems, Inc.
Spain
Email: pcamaril@cisco.com

Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Charles E. Perkins
Lupin Lodge
20600 Aldercroft Heights Rd.
Los Gatos, CA 95033
United States of America
Email: charliep@computer.org