

Workgroup: DMM Working Group

Internet-Draft:

draft-ietf-dmm-srv6-mobile-uplane-22

Published: 22 November 2022

Intended Status: Standards Track

Expires: 26 May 2023

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

## **Segment Routing IPv6 for Mobile User Plane**

### **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 26 May 2023.

### **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.	<a href="#">Introduction</a>
2.	<a href="#">Conventions and Terminology</a>
2.1.	<a href="#">Terminology</a>
2.2.	<a href="#">Conventions</a>
2.3.	<a href="#">Predefined SRv6 Endpoint Behaviors</a>
3.	<a href="#">Motivation</a>
4.	<a href="#">3GPP Reference Architecture</a>
5.	<a href="#">User-plane modes</a>
5.1.	<a href="#">Traditional mode</a>
5.1.1.	<a href="#">Packet flow - Uplink</a>
5.1.2.	<a href="#">Packet flow - Downlink</a>
5.2.	<a href="#">Enhanced mode</a>
5.2.1.	<a href="#">Packet flow - Uplink</a>
5.2.2.	<a href="#">Packet flow - Downlink</a>
5.2.3.	<a href="#">Scalability</a>
5.3.	<a href="#">Enhanced mode with unchanged gNB GTP behavior</a>
5.3.1.	<a href="#">Interworking with IPv6 GTP</a>
5.3.2.	<a href="#">Interworking with IPv4 GTP</a>
5.3.3.	<a href="#">Extensions to the interworking mechanisms</a>
5.4.	<a href="#">SRv6 Drop-in Interworking</a>
6.	<a href="#">SRv6 Segment Endpoint Mobility Behaviors</a>
6.1.	<a href="#">Args.Mob.Session</a>
6.2.	<a href="#">End.MAP</a>
6.3.	<a href="#">End.M.GTP6.D</a>
6.4.	<a href="#">End.M.GTP6.D.Di</a>
6.5.	<a href="#">End.M.GTP6.E</a>
6.6.	<a href="#">End.M.GTP4.E</a>
6.7.	<a href="#">H.M.GTP4.D</a>
6.8.	<a href="#">End.Limit: Rate Limiting behavior</a>
7.	<a href="#">SRv6 supported 3GPP PDU session types</a>
8.	<a href="#">Network Slicing Considerations</a>
9.	<a href="#">Control Plane Considerations</a>
10.	<a href="#">Security Considerations</a>
11.	<a href="#">IANA Considerations</a>
12.	<a href="#">Acknowledgements</a>
13.	<a href="#">Contributors</a>
14.	<a href="#">References</a>
14.1.	<a href="#">Normative References</a>
14.2.	<a href="#">Informative References</a>
Appendix A.	<a href="#">Implementations</a>

## 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

### 2.1. Terminology

- \*CNF: Cloud-native Network Function
- \*NFV: Network Function Virtualization
- \*PDU: Packet Data Unit
- \*PDU Session: Context of a UE connected to a mobile network.
- \*UE: User Equipment
- \*UPF: User Plane Function
- \*VNF: Virtual Network Function (including CNFs)
- \*Uplink: from the UE towards the DN
- \*Downlink: from the DN towards the UE

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 [\[RFC8754\]](#).

- \*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.camarilloelmalky-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.

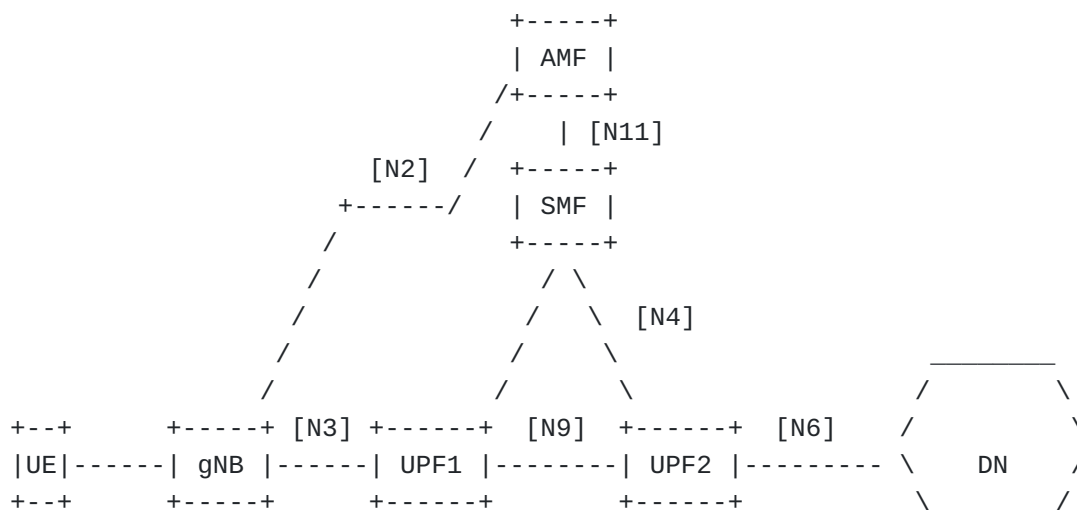


Figure 1: 3GPP 5G Reference Architecture

\*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 a 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 their 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.

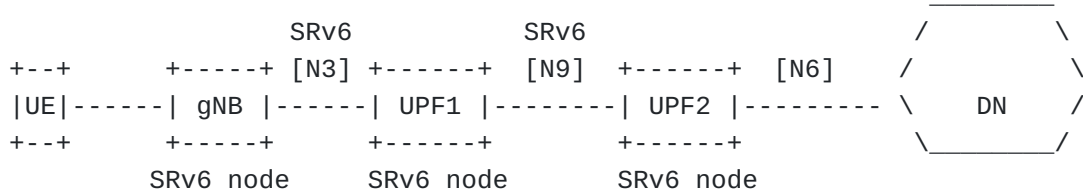


Figure 2: Traditional mode - example topology

#### 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.

The gNB/UPF control-plane (N2/N4 interface) is unchanged, specifically a single IPv6 address is provided to the gNB. A local policy instructs the gNB to use SRv6.

The gNB MAY resolve the IP address received via the control plane into a SID list using a mechanism like PCEP, DNS-lookup, LISP control-plane or others. The resolution mechanism is out of the scope of this document.

Note that the SIDs MAY use the arguments [Args.Mob.Session](#) ([Section 6.1](#)) if required by the UPFs.

[Figure 3](#) shows an Enhanced mode topology. The gNB and the UPF are SR-aware. The Figure shows two service segments, S1 and C1. S1 represents a VNF in the network, and C1 represents an intermediate router used for Traffic Engineering purposes to enforce a low-latency path in the network. Note that neither S1 nor C1 are required to have an N4 interface.

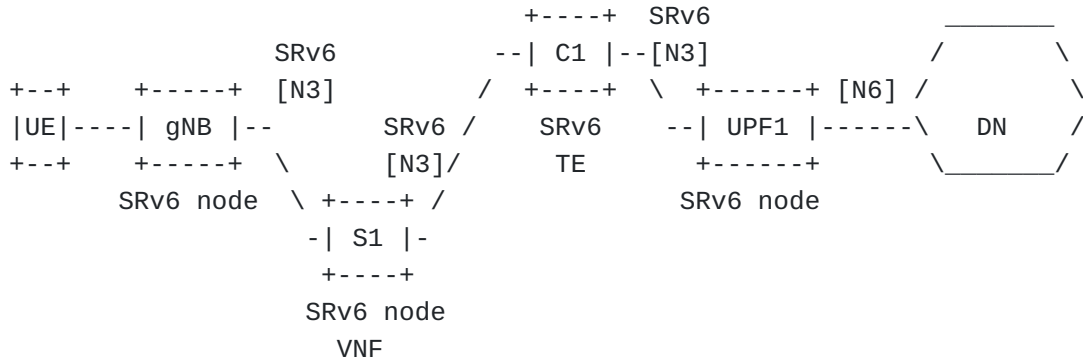


Figure 3: Enhanced mode - Example topology

#### 5.2.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

UE_out   : (A,Z)
gNB_out  : (gNB, S1)(U1::1, C1; SL=2)(A,Z)->H.Encaps.Red<S1,C1,U1::1>
S1_out   : (gNB, C1)(U1::1, C1; SL=1)(A,Z)
C1_out   : (gNB, U1::1)(A,Z)                ->End with PSP
UPF1_out : (A,Z)                            ->End.DT4,End.DT6,End.DT2U

```

UE sends its packet (A,Z) on a specific bearer to its gNB. gNB's control plane associates that session from the UE(A) with the IPv6 address B. gNB's control plane does a lookup on B to find the related SID list <S1, C1, U1::1>.

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.DT2U behaviors.

### 5.2.3. Scalability

The Enhanced Mode improves scalability 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.

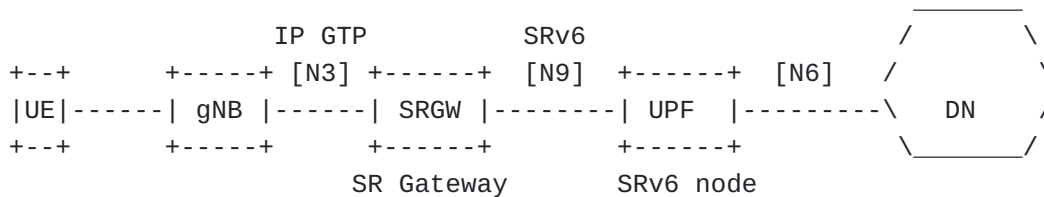


Figure 4: Example topology for interworking

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.

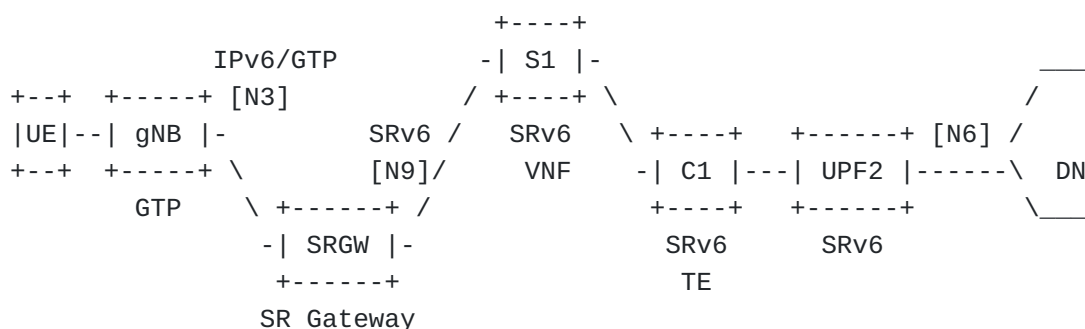


Figure 5: Enhanced mode with unchanged gNB IPv6/GTP behavior

### 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 forwards 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.

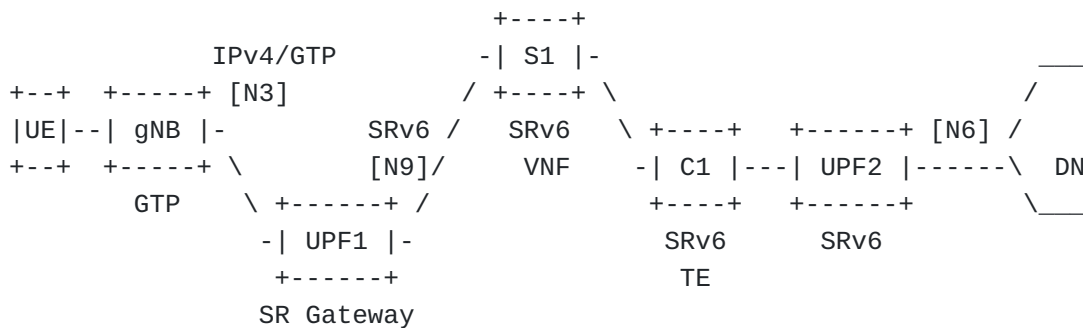


Figure 6: Enhanced mode with unchanged gNB IPv4/GTP behavior

### 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.

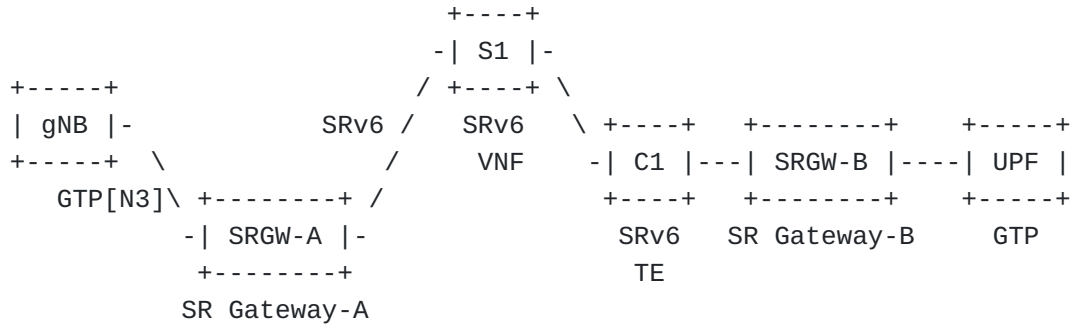


Figure 7: Example topology for SRv6 Drop-in mode

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 or other purposes 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.

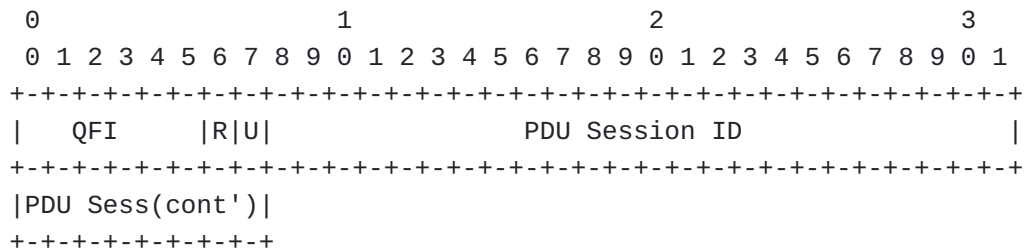


Figure 8: Args.Mob.Session format

\*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 SRH is not modified (neither the SID, nor the SL value).

## 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 SHOULD be followed by an Arg.Mob.Session argument space which is used to provide the session identifiers, as shown in line S08.

#### **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 the 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 the egress 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:

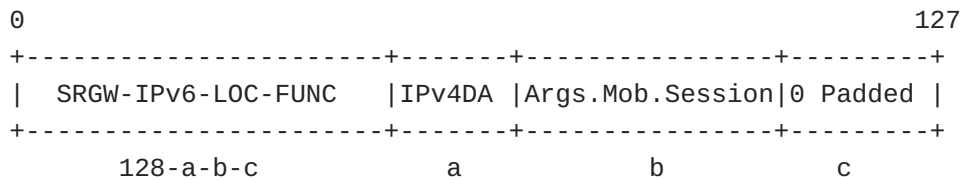


Figure 9: End.M.GTP4.E SID Encoding

The IPv6 Source Address has the following format:

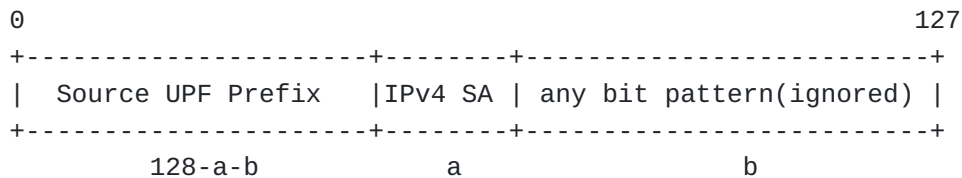


Figure 10: IPv6 SA Encoding for End.M.GTP4.E

## 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 SRGW-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
S06.   Set the IPv6 DA = B
S07.   Forward along the shortest path to B
S08. ELSE
S09.   Drop the packet

```

The SID B has the following format:

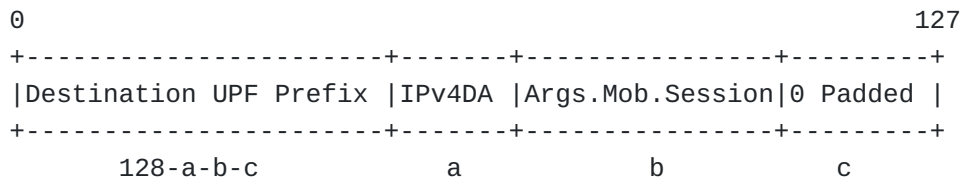


Figure 11: H.M.GTP4.D SID Encoding

The SID B MAY be an SRv6 Binding SID instantiated at the first UPF (U1) to bind an SR policy [[RFC9256](#)].

## 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:

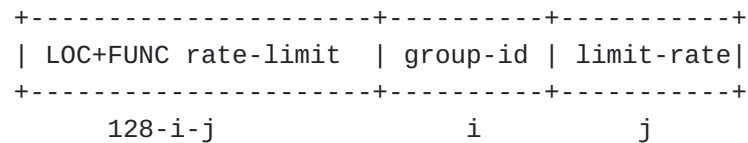


Figure 12: End.Limit: Rate limiting behavior argument format

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.

[[RFC9256](#)] 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) [[RFC9256](#)] 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 operate on control plane information, including information within the received SRH payload on which the behaviors operate. Altering the behaviors requires that an attacker alter the SR Domain as defined in [RFC8754]. 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

- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <<https://www.rfc-editor.org/info/rfc9256>>.
- [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]**

Camarillo, P., 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-26, 17 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lsr-flex-algo-26>>.

**[I-D.ietf-spring-segment-routing-central-epe]**

Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "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-06, 9 June 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-06>>.

**[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](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., Camarillo, P., Horn, J., Voyer, D., Zadok, S., Meilik, I., Agrawal, A., and K. Perumal, "Segment Routing IPv6 Mobile User Plane Architecture for Distributed Mobility Management", Work in Progress, Internet-Draft, draft-mhkk-dmm-srv6mup-architecture-04, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-mhkk-dmm-srv6mup-architecture-04>>.

**[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 open-source 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](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](mailto:satoru.matsushima@g.softbank.co.jp)

Clarence Filsfils  
Cisco Systems, Inc.  
Belgium

Email: [cf@cisco.com](mailto:cf@cisco.com)

Miya Kohno  
Cisco Systems, Inc.  
Japan

Email: [mkohno@cisco.com](mailto:mkohno@cisco.com)

Pablo Camarillo Garvia (editor)  
Cisco Systems, Inc.  
Spain

Email: [pcamaril@cisco.com](mailto:pcamaril@cisco.com)

Daniel Voyer  
Bell Canada  
Canada

Email: [daniel.voyer@bell.ca](mailto: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](mailto:charliep@computer.org)