DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 29, 2020

S. Matsushima
SoftBank
C. Filsfils
M. Kohno
P. Camarillo
Cisco Systems, Inc.
D. Voyer
Bell Canada
C. Perkins
Futurewei

September 26, 2019

Segment Routing IPv6 for Mobile User Plane
draft-ietf-dmm-srv6-mobile-uplane-06

Abstract

This document shows the applicability of SRv6 (Segment Routing IPv6) to the user-plane of mobile networks. The network programming nature of SRv6 accomplish 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. This document describes the SRv6 mobile user plane behavior and defines the SID functions for that.

Status of This Memo

Table of Contents

1. Introduction 3
2. Conventions and Terminology 3
  2.1. Terminology 3
  2.2. Conventions 4
  2.3. Predefined SRv6 Functions 4
3. Motivation 5
4. A 3GPP Reference Architecture 6
5. User-plane behaviors 7
  5.1. Traditional mode 7
    5.1.1. Packet flow - Uplink 8
    5.1.2. Packet flow - Downlink 8
  5.2. Enhanced Mode 9
    5.2.1. Packet flow - Uplink 10
    5.2.2. Packet flow - Downlink 10
  5.3. Enhanced mode with unchanged gNB GTP behavior 11
    5.3.1. Interworking with IPv6 GTP 11
    5.3.2. Interworking with IPv4 GTP 14
    5.3.3. Extensions to the interworking mechanisms 17
6. SRv6 SID Mobility Functions 17
  6.1. Args.Mob.Session 17
  6.2. End.MAP 18
  6.3. End.M.GTP6.D 18
  6.4. End.M.GTP6.E 19
  6.5. End.M.GTP4.E 20
  6.6. T.M.GTP4.D 20
  6.7. End.Limit: Rate Limiting function 21
7. SRv6 supported 3GPP PDU session types 22
8. Network Slicing Considerations 22
9. Control Plane Considerations 22
10. Security Considerations 23
11. IANA Considerations 23

12. Acknowledgements 23
13. Contributors 24
14. References 24
14.1. Normative References 24
14.2. Informative References 24
Appendix A. Implementations 26
Authors' Addresses 26

1. Introduction

In mobile networks, mobility management systems provide connectivity while mobile nodes move. While the control-plane of the system signals movements of a mobile node, the user-plane establishes a tunnel between the mobile node and its anchor node over IP-based backhaul and core networks.

This document shows the applicability of SRv6 (Segment Routing IPv6) to those mobile networks. SRv6 provides source routing to networks so that operators can explicitly indicate a route for the packets to and from the mobile node. SRv6 endpoint nodes serve as the anchors of mobile user-plane.

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

- AMBR: Aggregate Maximum Bit Rate
- APN: Access Point Name (commonly used to identify a network or class of service)
- BSID: SR Binding SID [RFC8402]
- CNF: Cloud-native Network Function
- gNB: gNodeB
- NH: The IPv6 next-header field.
- NFV: Network Function Virtualization
- PDU: Packet Data Unit
- Session: TBD...
- SID: A Segment Identifier which represents a specific segment in a segment routing domain.
- SRH: The Segment Routing Header. [I-D.ietf-6man-segment-routing-header]
- TEID: Tunnel Endpoint Identifier
- UE: User Equipment
- UPF: User Plane Function
- VNF: Virtual Network Function

2.2. Conventions

- o NH=SRH means that NH is 43 with routing type 4.
- o Multiple SRHs may be present inside each packet, but they must follow each other. The next-header field of each SRH, except the last one, must be NH-SRH (43 type 4).
- o For simplicity, no other extension headers are shown except the SRH.
- o The SID type used in this document is IPv6 address (also called SRv6 Segment or SRv6 SID).
- o gNB::1 is an IPv6 address (SID) assigned to the gNB.
- o U1::1 is an IPv6 address (SID) assigned to UPF1.
- o U2::1 is an IPv6 address (SID) assigned to UPF2.
- o U2:: is some other IPv6 address (SID) assigned to UPF2.
- o 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.
- o (SA,DA) (S3, S2, S1; SL) represents an IPv6 packet with:

  * IPv6 header with source and destination addresses SA and DA respectively, and next-header SRH, with SID list <S1, S2, S3> with SegmentsLeft = SL
  * The payload of the packet is not represented.
- o 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. (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 behavior, the (S3, S2, S1; SL) notation is more convenient.
- o SRH[SL] represents the SID pointed by the SL field in the first SRH. In our example, SRH[2] represents S1, SRH[1] represents S2 and SRH[0] represents S3.
- o SRH[SL] can be different from the DA of the IPv6 header.

2.3. Predefined SRv6 Functions

The following functions are defined in [I-D.ietf-spring-srv6-network-programming].

- o End.DT4 means to decapsulate and forward using a specific IPv4 table lookup.
- o End.DT6 means to decapsulate and forward using a specific IPv6 table lookup.

- End.DX4 means to decapsulate the packet and forward through a particular outgoing interface -or set of OIFs- configured with the SID.
- End.DX6 means to decapsulate and forward through a particular outgoing interface -or set of OIFs- configured with the SID.
- End.DX2 means to decapsulate the L2 frame and forward through a particular outgoing interface -or set of OIFs- configured with the SID.
- End.T means to forward using a specific IPv6 table lookup.
- End.X means to forward through a link configured with the SID.
- T.Encaps.Red means encapsulation without pushing SRH (resulting in "Reduced" packet size).
- PSP means Penultimate Segment Pop. The packet is subsequently forwarded without the popped SRH.

New SRv6 functions are defined in Section 6 to support the needs of the mobile user plane.

3. Motivation

Mobility networks are becoming more challenging to operate. On one hand, traffic is constantly growing, and latency requirements are more strict; on the other-hand, there are new use-cases like NFV that are also challenging network management.

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, VNF/CNF to VNF/CNF networking.

SRv6 specifies network-programming (see [I-D.ietf-spring-srv6-network-programming]). Applied to mobility, SRv6 can provide the user-plane functions needed for mobility management. SRv6 takes advantage of underlying transport awareness and flexibility to improve mobility user-plane functions.

The use-cases for SRv6 mobility are discussed in [I-D.camarilloelmalky-springdmm-srv6-mob-usecases].

4. A 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. Other architectures can be seen in [I-D.gundavelli-dmm-mfa] and [WHITEPAPER-5G-UP].

```
                                  +-----+
                                  | AMF |
                                  +-----+
                                 /    | [N11]
                          [N2]  /  +-----+
                        +------/   | SMF |
                       /           +-----+
                      /              / \
                     /              /   \  [N4]
                    /              /     \                    ________
                   /              /       \                  /        \
   +--+      +-----+ [N3] +-------+  [N9]  +-------+  [N6]    /          \
   |UE|------| gNB |------| UPF1 |--------| UPF2 |--------- \    DN    /
   +--+      +-----+      +-------+        +-------+          \________/
```

Figure 1: 3GPP 5G Reference Architecture

- o gNB: gNodeB
- o UPF1: UPF with Interfaces N3 and N9
- o UPF2: UPF with Interfaces N9 and N6
- o SMF: Session Management Function
- o AMF: Access and Mobility Management Function
- o 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 description in this document also work for such UPFs.

Each session from an 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 behaviors

This section describes some mobile user-plane behaviors using SRv6.

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 user-plane. In this mode there is no change to mobility networks architecture, except that GTP-U [TS.29281] is replaced by SRv6.

The second mode is the "Enhanced mode". In this mode the SR policy contains SIDs for Traffic Engineering and VNFs, which results in effective end-to-end network slices.

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).

We introduce two mechanisms for interworking with legacy access networks (N3 interface is unmodified). In these document we introduce them applied to the Enhanced mode, although they could be used in combination with the Traditional mode as well.

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

This document uses SRv6 functions defined in [I-D.ietf-spring-srv6-network-programming] as well as new SRv6 functions designed for the mobile user plane. The new SRv6 functions are detailed in Section 6.

5.1. Traditional mode

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

In existing 3GPP mobile networks, an UE 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 UE PDU Session.

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

Our example topology is shown in Figure 2. In traditional mode 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 function End.MAP, defined in Section 6.2, is used.

```
                                                              ________
                  SRv6           SRv6                        /        \
+--+      +-----+ [N3] +-------+  [N9]  +-------+  [N6]     /          \
|UE|------| gNB |------| UPF1 |--------| UPF2 |--------- \    DN    /
+--+      +-----+      +-------+        +-------+            \________/
         SRv6 node     SRv6 node       SRv6 node
```

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)     -> T.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 T.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. U1::1 represents an anchoring SID specific for that session at UPF1. gNB obtains the SID U1::1 from the existing control plane (N2 interface).

When the packet arrives at UPF1, the SID U1::1 identifies a local End.MAP function. 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.DT function. 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::1) (Z,A)    -> T.Encaps.Red <U1::1>
      UPF1_out: (U2::, gNB::1) (Z,A)   -> End.MAP
      gNB_out : (Z,A)                  -> End.DX4 or End.DX6
```

When the packet arrives at the UPF2, the UPF2 maps that flow into a UE PDU Session. This UE PDU Session is associated with the segment endpoint <U1::1>. UPF2 performs a T.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::1 is a local End.MAP function. This function maps the SID to the next anchoring point and replaces U1::1 by gNB::1, that belongs to the next hop.

Upon packet arrival on gNB, the SID gNB::1 corresponds to an End.DX4 or End.DX6 function. 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, traffic steering and service programming [I-D.xuclad-spring-sr-service-programming], thanks to the use of multiple SIDs, instead of a single SID as done in the Traditional mode.

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

The gNB control-plane (N2 interface) is unchanged, specifically a single IPv6 address is given to the gNB.

- o The gNB MAY resolve the IP address into a SID list using a mechanism like PCEP, DNS-lookup, small augment for LISP control-plane, etc.

Note that the SIDs MAY use the arguments Args.Mob.Session if required by the UPFs.

Figure 3 shows an Enhanced mode topology. In the Enhanced mode, 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 a constraint path on a router requiring Traffic Engineering. S1 and C1 belong to the underlay and don't have an N4 interface, so they are not considered UPFs.

```
                                    +----+  SRv6               ________
                    SRv6          --| C1 |--[N3]              /        \
   +--+    +-----+  [N3]         /  +----+  \  +-------+ [N6] /         \
   |UE|----| gNB |--       SRv6 /    SRv6    --| UPF2 |------\   DN    /
   +--+    +-----+  \      [N3]/      TE       +-------+       \_______/
          SRv6 node  \ +----+ /               SRv6 node
                      -| S1 |-
                       +----+
                      SRv6 node
                        CNF
```

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)(U2::1, C1; SL=2)(A,Z)-> T.Encaps.Red<S1,C1,U2::1>
S1_out  : (gNB, C1)(U2::1, C1; SL=1 (A,Z)
C1_out  : (gNB, U2::1)(A,Z)              -> PSP
UPF2_out: (A,Z)                          -> End.DT4 or End.DT6
```

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 and GTP TEID T. gNB's control plane does a lookup on B to find the related SID list <S1, C1, U2::1>.

When gNB transmits the packet, it contains all the segments of the SR policy. The SR policy can include 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 UPF2, the active segment (U2::1) is an End.DT4/6 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:

```
UPF2_in : (Z,A)                                -> UPF2 maps the flow w/
                                                  SID list <C1,S1, gNB>
UPF2_out: (U2::1, C1)(gNB, S1; SL=2)(Z,A)     -> T.Encaps.Red
C1_out  : (U2::1, S1)(gNB, S1; SL=1)(Z,A)
S1_out  : (U2::1, gNB)(Z,A)                   -> PSP
gNB_out : (Z,A)                                -> End.DX4 or End.DX6
```

When the packet arrives at the UPF2, the UPF2 maps that particular flow into a UE PDU Session. This UE PDU Session is associated with the policy <C1, S1, gNB>. The UPF2 performs a T.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 or End.DX6 (depending on the underlying traffic). The gNB decapsulates the packet, removing the IPv6 header and all its extensions headers and forwards the traffic toward the 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, the other for IPv6.

In the interworking scenarios as illustrated in Figure 4, gNB does not support SRv6. gNB supports GTP encapsulation over IPv4 or IPv6. To achieve interworking, a SR Gateway (SRGW-UPF1) 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.

```
                                                         _______
                  IP GTP          SRv6                  /       \
 +--+      +-----+ [N3] +-------+  [N9]  +-------+  [N6]   /         \
 |UE|------| gNB |------| UPF1 |--------| UPF2 |---------\   DN    /
 +--+      +-----+      +-------+        +-------+          \_______/
                       SR Gateway       SRv6 node
```

Figure 4: Example topology for interworking

5.3.1. Interworking with IPv6 GTP

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

Key points:

o The gNB is unchanged (control-plane or user-plane) and encapsulates into GTP (N3 interface is not modified).
o The 5G Control-Plane (N2 interface) is unmodified; one IPv6 address is needed (i.e. a BSID at the SRGW).
o The SRGW removes GTP, finds the SID list related to DA, and adds SRH with the SID list.
o There is no state for the downlink at the SRGW.
o There is simple state in the uplink at the SRGW; using Enhanced mode results in fewer SR policies on this node. A SR policy can be shared across UEs.
o When a packet from the UE leaves the gNB, it is SR-routed. This simplifies network slicing [I-D.hegdeppsenak-isis-sr-flex-algo].
o In the uplink, the IPv6 DA BSID steers traffic into an SR policy when it arrives at the SRGW-UPF1.

An example topology is shown in Figure 5. In this mode the gNB is an unmodified gNB using IPv6/GTP. The UPFs are SR-aware. As before, the SRGW maps IPv6/GTP traffic to SRv6.

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

```
                               +----+
             IPv6/GTP         -| S1 |-                            ___
+--+  +-----+ [N3]           / +----+ \                          /
|UE|--| gNB |-         SRv6 /   SRv6   \ +----+   +-------+ [N6] /
+--+  +-----+ \        [N9]/     VNF    -| C1 |---| UPF2  |------\  DN
        GTP    \ +-------+ /             +----+   +-------+       \___
                -| UPF1  |-               SRv6      SRv6
                 +-------+                 TE
                SR Gateway
```

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::1, C1; SL=2)(A,Z) -> B is an End.M.GTP6.D
                                              SID at the SRGW
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 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 signalled 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 is 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::1) 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) -> T.Encaps.Red
C1_out  : (U2::1, S1)(gNB, 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 a T.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 inserted 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 UE PDU Session; the state state can be shared among UEs. This enables much more scalable SRGW deployments compared to a solution holding millions of states, one or more per UE.

5.3.1.4. IPv6 user-traffic

For IPv6 user-traffic it is RECOMMENDED to perform encapsulation. However based on local policy, a service provider MAY choose to do SRH insertion. The main benefit is lower overhead.

5.3.2. Interworking with IPv4 GTP

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

Key points:

- o The gNB is unchanged and encapsulates packets into GTP (the N3 interface is not modified).
- o In the uplink, traffic is classified by SRGW's Uplink Classifier and steered into an SR policy. The SRGW is a UPF1 functionality and can coexist with UPF1's Uplink Classifier functionality.
- o SRGW removes GTP, finds the SID list related to DA, and adds a 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.

```
                                +----+
             IPv4/GTP         -| S1 |-                            ___
+--+  +-----+ [N3]           / +----+ \                          /
|UE|--| gNB |-          SRv6 /   SRv6   \ +----+   +------+ [N6] /
+--+  +-----+ \         [N9]/     VNF    -| C1 |---| UPF2 |------\  DN
        GTP    \ +------+ /              +----+   +------+       \___
                -| UPF1 |-                SRv6      SRv6
                 +------+                  TE
                SR Gateway
```

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)    -> T.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 Uplink Classifier rule for incoming traffic from the gNB, that steers the traffic into an SR policy by using the function T.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.

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,SRGW::SA:DA:TEID>
UPF2_out: (U2::1, C1)(SRGW::SA:DA:TEID, S1; SL=2)(Z,A) ->T.Encaps.Red
C1_out  : (U2::1, S1)(SRGW::SA:DA:TEID, S1; SL=1)(Z,A)
S1_out  : (U2::1, SRGW::SA:DA:TEID)(Z,A)
SRGW_out: (SA, DA)(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 T.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 forward 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 inserted by the UPF. 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 an Uplink Classifier. There is state for steering the different sessions on a SR policies. However, SR policies are shared among several UE/sessions.

5.3.2.4. IPv6 user-traffic

For IPv6 user-traffic it is RECOMMENDED to perform encapsulation. Based on local policy, a service provider MAY choose to do SRH insertion. The main benefit is a lower overhead.

5.3.3. Extensions to the interworking mechanisms

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

Furthermore, although these mechanisms are designed for interworking with legacy RAN at the N3 interface, these methods could also be applied for interworking with a non-SRv6 capable UPF at the N9 interface (e.g. L3-anchor is SRv6 capable but L2-anchor is not).

6. SRv6 SID Mobility Functions

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.DT and End.DX functions. Note that proposed format is applicable for 5G networks, while similar formats could be proposed for legacy networks.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   QFI     |R|U|                PDU Session ID                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PDU Sess(cont')|
+-+-+-+-+-+-+-+-+
```

Args.Mob.Session format

- o QFI: QoS Flow Identifier [TS.38415]
- o R: Reflective QoS Indication [TS.23501]. This parameter indicates the activaton of reflective QoS towards the UE for the transfered packet. Reflective QoS enables the UE to map UL User Plane traffic to QoS Flows without SMF provided QoS rules.

o U: Unused and for future use. MUST be 0 on transmission and ignored on receipt.

o PDU Session ID: Identifier of PDU Session. The GTP-U equivalent is TEID.

Since the SRv6 function is likely NOT to be instantiated per PDU session, Args.Mob.Session helps the UPF to perform the functions which require per QFI and/or per PDU Session granularity.

6.2. End.MAP

The "Endpoint function with SID mapping" function (End.MAP for short) is used in several scenarios. Particularly in mobility, End.MAP is used in the UPFs for the PDU Session anchor functionality.

When a SR node N receives a packet destined to S and S is a local End.MAP SID, N does the following:

```
1.    Lookup the IPv6 DA in the mapping table
2.    update the IPv6 DA with the new mapped SID            ;; Ref1
3.    IF segment_list > 1
4.       insert a new SRH
5.    forward according to the new mapped SID
```

Ref1: The SIDs in the SRH are NOT modified.

6.3. End.M.GTP6.D

The "Endpoint function with IPv6/GTP decapsulation into SR policy" function (End.M.GTP6.D for short) is used in interworking scenario for the uplink toward from the legacy gNB using IPv6/GTP. Suppose, for example, this SID is associated with an SR policy <S1, S2, S3> and an IPv6 Source Address A.

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

```
 1. IF NH=UDP & UDP_DST_PORT = GTP THEN
 2.    copy TEID to form SID S3
 3.    pop the IPv6, UDP and GTP headers
 4.    push a new IPv6 header with a SR policy in SRH <S1, S2, S3>
 5.    set the outer IPv6 SA to A
 6.    set the outer IPv6 DA to S1
 7.    set the outer IPv6 NH                                  ;; Ref1
 8.    forward according to the S1 segment of the SRv6 Policy
 9. ELSE
10.    Drop the packet
```

Ref1: 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 prefix of last segment(S3 in above example) SHOULD be followed by an Arg.Mob.Session argument space which is used to provide the session identifiers.

The prefix of A SHOULD be an End.M.GTP6.E SID instantiated at an SR gateway.

6.4. End.M.GTP6.E

The "Endpoint function with encapsulation for IPv6/GTP tunnel" function (End.M.GTP6.E for short) is used in 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 S, and S is a local End.M.GTP6.E SID, N does the following:

```
 1. IF NH=SRH & SL = 1  THEN                                ;; Ref1
 2.    store SRH[0] in variable new_DA
 3.    store TEID in variable new_TEID from IPv6 DA         ;; Ref2
 4.    pop IP header and all its extension headers
 5.    push new IPv6 header and GTP-U header
 6.    set IPv6 DA to new_DA
 7.    set IPv6 SA to A
 8.    set GTP_TEID to new_TEID
 9.    lookup the new_DA and forward the packet accordingly
10. ELSE
11.    Drop the packet
```

Ref1: An End.M.GTP6.E SID MUST always be the penultimate SID.

Ref2: TEID is extracted from the argument space of the current SID.

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

6.5. End.M.GTP4.E

The "Endpoint function with encapsulation for IPv4/GTP tunnel" function (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:

```
1. IF (NH=SRH and SL = 0) or ENH=4 THEN
2.    store IPv6 DA in buffer S
3.    store IPv6 SA in buffer S'
4.    pop the IPv6 header and its extension headers
5.    push UDP/GTP headers with GTP TEID from S
6.    push outer IPv4 header with SA, DA from S' and S
7. ELSE
8.    Drop the packet
```

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

```
0                                                          127
+-----------------------+-------+----------------+---------+
|  SRGW-IPv6-LOC-FUNC   |IPv4DA |Args.Mob.Session|0 Padded |
+-----------------------+-------+----------------+---------+
       128-a-b-c            a             b            c
```

End.M.GTP4.E SID Encoding

S' has the following format:

```
0                                                          127
+----------------------+--------+--------------------------+
|  Source UPF Prefix   |IPv4 SA | any bit pattern(ignored) |
+----------------------+--------+--------------------------+
         128-a-b            a                  b
```

IPv6 SA Encoding for End.M.GTP4.E

6.6. T.M.GTP4.D

The "Transit with tunnel decapsulation and map to an SRv6 policy" function (T.M.GTP4.D for short) is used in the direction from legacy user-plane to SRv6 user-plane network.

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

```
1. IF Payload == UDP/GTP THEN
2.    pop the outer IPv4 header and UDP/GTP headers
3.    copy IPv4 DA, TEID to form SID B
4.    copy IPv4 SA to form IPv6 SA B'
5.    encapsulate the packet into a new IPv6 header   ;;Ref1
6.    set the IPv6 DA = B
7.    forward along the shortest path to B
8. ELSE
9.    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:

```
0                                                          127
+-----------------------+-------+----------------+---------+
|Destination UPF Prefix |IPv4DA |Args.Mob.Session|0 Padded |
+-----------------------+-------+----------------+---------+
       128-a-b-c            a            b           c
```

T.M.GTP4.D SID Encoding

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

The prefix of B' SHOULD be an End.M.GTP4.E SID with its format instantiated at an SR gateway with the IPv4 SA of the receiving packet.

6.7. End.Limit: Rate Limiting function

The mobile user-plane requires a rate-limit feature. For this purpose, we define a new function "End.Limit". The "End.Limit" function 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 an same AMBR group. The encoding format of the rate limit segment SID is as follows:

```
+----------------------+----------+-----------+
| LOC+FUNC rate-limit  | group-id | limit-rate|
+----------------------+----------+-----------+
      128-i-j                i          j
```

End.Limit: Rate limiting function 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 all the 3GPP PDU session types without any protocol overhead by using the corresponding SRv6 functions (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 PDU sessions; End.DX2 for Unstructured PDU sessions).

8. Network Slicing Considerations

A mobile network may be required to implement "network slices", which logically separate network resources. User-plane functions 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.hegdeppsenak-isis-sr-flex-algo]
- Inter-Domain policies [I-D.ietf-spring-segment-routing-central-epe]

Furthermore, these can be combined with ODN/AS [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.

The control plane could be the current 3GPP-defined control plane with slight modifications to the N4 interface [TS.29244].

Alternatively, SRv6 could be used in conjunction with a new mobility control plane as described in LISP [I-D.rodrigueznatal-lisp-srv6], hICN [I-D.auge-dmm-hicn-mobility-deployment-options], MFA [I-D.gundavelli-dmm-mfa] or in conjunction with FPC [I-D.ietf-dmm-fpc-cpdp]. The analysis of new mobility control-planes and its applicability to SRv6 is out of the scope of this document.

Section 11 allocates SRv6 endpoint function types for the new functions defined in this document. Control-plane protocols are expected to use these function type codes to signal each function.

SRv6's network programming nature allows a flexible and dynamic UPF placement.

10. Security Considerations

TBD

11. IANA Considerations

IANA is requested to allocate, within the "SRv6 Endpoint Types" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry [I-D.ietf-spring-srv6-network-programming], the following values:

Value/Range	Hex	Endpoint function	Reference
TBA	TBA	End.MAP	[This.ID]
TBA	TBA	End.M.GTP6.D	[This.ID]
TBA	TBA	End.M.GTP6.E	[This.ID]
TBA	TBA	End.M.GTP4.E	[This.ID]
TBA	TBA	End.Limit	[This.ID]

Table 1: SRv6 Mobile User-plane Endpoint 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 and Aeneas Dodd-Noble for their useful comments of this work.

13. Contributors

Kentaro Ebisawa
Toyota Motor Corporation
Japan

Email: ebisawa@toyota-tokyo.tech

14. References

14.1. Normative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-23 (work in progress), September 2019.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-03 (work in progress), May 2019.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-02 (work in progress), September 2019.

[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>.

14.2. Informative References

[I-D.ali-spring-network-slicing-building-blocks]
Ali, Z., Filsfils, C., Camarillo, P., and d. daniel.voyer@bell.ca, "Building blocks for Slicing in Segment Routing Network", draft-ali-spring-network-slicing-building-blocks-01 (work in progress), March 2019.

[I-D.auge-dmm-hicn-mobility-deployment-options]
Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "Anchorless mobility management through hICN (hICN-AMM): Deployment options", draft-auge-dmm-hicn-mobility-deployment-options-02 (work in progress), July 2019.

[I-D.camarillo-dmm-srv6-mobile-pocs]
Camarillo, P., Filsfils, C., Bertz, L., Akhavain, A., Matsushima, S., and d. daniel.voyer@bell.ca, "Segment Routing IPv6 for mobile user-plane PoCs", draft-camarillo-dmm-srv6-mobile-pocs-02 (work in progress), April 2019.

[I-D.camarilloelmalky-springdmm-srv6-mob-usecases]
Camarillo, P., Filsfils, C., Elmalky, H., Matsushima, S., daniel.voyer@bell.ca, d., Cui, A., and B. Peirens, "SRv6 Mobility Use-Cases", draft-camarilloelmalky-springdmm-srv6-mob-usecases-02 (work in progress), August 2019.

[I-D.gundavelli-dmm-mfa]
Gundavelli, S., Liebsch, M., and S. Matsushima, "Mobility-aware Floating Anchor (MFA)", draft-gundavelli-dmm-mfa-01 (work in progress), September 2018.

[I-D.hegdeppsenak-isis-sr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., and A. Gulko, "ISIS Segment Routing Flexible Algorithm", draft-hegdeppsenak-isis-sr-flex-algo-02 (work in progress), February 2018.

[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.

[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", draft-ietf-spring-segment-routing-central-epe-10 (work in progress), December 2017.

[I-D.rodrigueznatal-lisp-srv6]
Rodriguez-Natal, A., Ermagan, V., Maino, F., Dukes, D., Camarillo, P., and C. Filsfils, "LISP Control Plane for SRv6 Endpoint Mobility", draft-rodrigueznatal-lisp-srv6-02 (work in progress), July 2019.

[I-D.xuclad-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-xuclad-spring-sr-service-programming-02 (work in progress), April 2019.

[TS.23501]
3GPP, "System Architecture for the 5G System", 3GPP TS 23.501 15.0.0, November 2017.

[TS.29244]
3GPP, "Interface between the Control Plane and the User Plane Nodes", 3GPP TS 29.244 15.0.0, December 2017.

[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 functions. These functions have an open-source P4 implementation available in <https://github.com/ebiken/p4srv6>.

There are also implementations in M-CORD NGIC and Open Air Interface (OAI). Further details can be found in [I-D.camarillo-dmm-srv6-mobile-pocs].

Authors' Addresses

Satoru Matsushima
SoftBank
Tokyo
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
Cisco Systems, Inc.
Spain

Email: pcamaril@cisco.com

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

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

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