DMM Working Group Internet-Draft Intended status: Standards Track Expires: September 6, 2018 S. Matsushima SoftBank C. Filsfils M. Kohno P. Camarillo Cisco Systems, Inc. D. Voyer Bell Canada C. Perkins Futurewei March 5, 2018

## Segment Routing IPv6 for Mobile User Plane draft-ietf-dmm-srv6-mobile-uplane-01

Abstract

This document discusses the applicability of SRv6 (Segment Routing IPv6) to user-plane of mobile networks. The source routing capability and the network programming nature of SRv6, accomplish mobile user-plane functions in a simple manner. The statelessness and the ability to control underlying layer will be even more beneficial to the mobile user-plane, in terms of providing flexibility and SLA control for various applications. It also simplifies the network architecture by eliminating the necessity of tunnels, such as GTP-U [TS.29281], PMIP [RFC5213], Mac-in-Mac, MPLS, and so on. In addition, Segment Routing provides an enhanced method for network slicing, which is briefly introduced by this document.

Status of This Memo

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

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 http://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 September 6, 2018.

# Copyright Notice

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

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

## Table of Contents

<u>1</u> .	Introduction	<u>3</u>
<u>2</u> .	Conventions and Terminology	<u>3</u>
<u>3</u> .	Motivation	<u>4</u>
<u>4</u> .	Reference Architecture	<u>5</u>
<u>5</u> .	User-plane behaviors	<u>6</u>
5	. <u>.</u> . Traditional mode (formerly Basic mode)	<u>6</u>
	5.1.1. Packet flow - Uplink	7
	5.1.2. Packet flow - Downlink	8
	5.1.3. IPv6 user-traffic	8
5		8
	5.2.1. Packet flow - Uplink	9
	5.2.2. Packet flow - Downlink	10
	5.2.3. IPv6 user-traffic	10
5		10
	5.3.1. Interworking with IPv6 GTP	11
	5.3.2. Interworking with IPv4 GTP	14
	5.3.3. Extensions to the interworking mechanisms	16
6.	SRv6 STD Mobility Functions	17
<u> </u>	1. End.MAP: Endpoint function with SID mapping	17
6	5.2 End M GTP6 D: Endpoint function with decansulation from	
Ŭ	TPv6/GTP tunnel	17
6	S 3 End M GTP6 E: Endnoint function with encansulation for	
0	TPv6/GTP tunnel	18
6	A End M GTP4 E: Endnoint function with encansulation for	
0	TDv//GTD tunnel	10
6	5 T M Tman: Transit behavior with TPv4/GTP decansulation	10
0	and manning into an SPV6 Policy	10
6	and mapping into an SKVO FOILCy	19
7	Notvork Sliging Considerations	<u>10</u>
<u>/</u> ·	Control Diana Considerations	20
ŏ.		<u>10</u>
<u>9</u> .	Security constaerations	<u>71</u>

<u>10</u> . IAN	A Considerations	<u>21</u>
<u>11</u> . Ack	nowledgements	<u>21</u>
<u>12</u> . Ref	erences	<u>21</u>
<u>12.1</u> .	Normative References	<u>21</u>
<u>12.2</u> .	Informative References	<u>22</u>
Authors	'Addresses	<u>23</u>

# **<u>1</u>**. Introduction

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

This document discusses the applicability of SRv6 (Segment Routing IPv6) to those mobile networks. SRv6 provides source routing to networks where operators can explicitly indicate a route for the packets from and to the mobile node. SRv6 endpoint nodes perform the roles of anchor 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 [<u>RFC2119</u>].

SRH is the abbreviation for the Segment Routing Header. We assume that the SRH may be present multiple times inside each packet.

NH is the abbreviation of the IPv6 next-header field.

NH=SRH means that the next-header field is 43 with routing type 4.

When there are multiple SRHs, they must follow each other: the nextheader field of all SRH, except the last one, must be SRH.

The effective next-header (ENH) is the next-header field of the IP header when no SRH is present, or is the next-header field of the last SRH.

In this version of the document, we assume that there is no other extension header than the SRH. This will be lifted in future versions of the document.

SID: A Segment Identifier which represents a specific segment in segment routing domain. The SID type used in this document is IPv6 address (also referenced as SRv6 Segment or SRv6 SID). Matsushima, et al. Expires September 6, 2018 [Page 3]

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:

- o IPv6 header with source and destination addresses respectively SA and DA and next-header is SRH
- o SRH with SID list <S1, S2, S3> with SegmentsLeft = SL
- 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 The payload of the packet is omitted.

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.

FIB is the abbreviation for the forwarding table. A FIB lookup is a lookup in the forwarding table. When a packet is intercepted on a wire, it is possible that SRH[SL] is different from the DA.

## 3. Motivation

Every day mobility networks are getting 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.

Problem comes from the fact that the current architecture of mobile networks is agnostic to the underlying transport. Indeed, it rigidly fragments the user-plane into radio access, core and service networks and connects them by tunneling techniques through the user-plane roles such as access and anchor nodes. Such agnosticism and rigidness make it difficult for the operator to optimize and operate the data-path.

While the mobile network industry has been trying to solve those problems, applications have shifted to use IPv6, and network operators have started adopting IPv6 as their IP transport as well. SRv6, the IPv6 instantiation of Segment Routing [I-D.ietf-spring-segment-routing], integrates both the application Matsushima, et al. Expires September 6, 2018 [Page 4]

data-path and the underlying transport layer into one single protocol, allowing operators to optimize the network in a simplified manner and removing state from the network.

Further on, SRv6 introduces the notion of network-programming [I-D.filsfils-spring-srv6-network-programming], that applied to mobility fulfils the user-plane functions of mobility management. SRv6 takes advantage of underlying transport awareness and flexibility to deploy mobility user-plane functions in an optimized manner. Those are the motivations to adopt SRv6 for mobile user-plane.

#### 4. Reference Architecture

This section describes a reference architecture and possible deployment scenarios.

Figure 1 shows a reference architecture, based on 5G packet core architecture [TS.23501].

Please note that all the user-plane described in this document does not depend on any specific architecture. This architecture is just used as a reference based on the latest 3GPP standards at the time of writing this draft. Other type of architectures can be seen in [I-D.gundavelli-dmm-mfa] and [WHITEPAPER-5G-UP].



Figure 1: Reference Architecture

- o UE : User Equipment
- o gNB : gNodeB
- o UPF : User Plane Function
  - \* UPF1: Interfaces N3 and N9

Matsushima, et al. Expires September 6, 2018 [Page 5]

- \* UPF2: Interfaces N9 and N6
- \* Note: For simplicity we don't depict a UPF that is only connected to N9 interfaces, although the techniques described in this document are also valid in such case.
- o SMF : Session Management Function
- o AMF : Access and Mobility Management Function
- o DN : Data Network e.g. operator services, Internet access

A session from an UE gets assigned to an UPF. Sometimes more than one UPF may be used for providing a certain kind of richer service functions. UE gets its IP address from the DHCP block of its UPF. The UPF advertises the IP address block towards the Internet ensuring that return traffic is routed to the right UPF.

## 5. User-plane behaviors

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

In order to simplify the SRv6 adoption, we present two different "modes" that vary with respect the SRv6 SID allocation. The first one is the "Traditional mode", which inherits the traditional mobile user-plane. In this mode there is no change to mobility networks architecture, except for the pure replacement of GTP-U [TS.29281] for SRv6.

The second mode is the "Enhanced mode", which aggregates the mobile sessions and allocates SID on a per policy basis. The benefit of the latter is that the SR policy contains SIDs for Traffic Engineering and VNFs. Both of these modes assume both the gNB and UPFs are SRaware (N3 and N9 interfaces are SRv6).

Additionally, we introduce a new "Enhanced mode with unchanged gNB GTP behavior". This mode consists of two mechanisms for interworking with legacy access networks -interface N3 unmodified-. One of these mechanism is designed to interwork with legacy gNBs using GTP/IPv4. The second method is designed to interwork with legacy gNBs using GTP/IPv6.

This section makes reference to already existing SRv6 functions defined in [<u>I-D.filsfils-spring-srv6-network-programming</u>] as well as new SRv6 functions designed for the mobile userplane. The new SRv6 functions are detailed in the <u>Section 6</u>.

## **<u>5.1</u>**. Traditional mode (formerly Basic mode)

In the traditional mode, we assume that mobile user-plane functions are the same as existing ones except the use of SRv6 as the data

Matsushima, et al. Expires September 6, 2018 [Page 6]

plane instead of GTP-U. No impact to the rest of mobile system should be expected.

In the traditional mobile network, an UE session is mapped 1-for-1 with a specific GTP tunnel (TEID). This 1-for-1 mapping is replicated here to replace the GTP encaps with the SRv6 encaps, while not changing anything else.

This mode minimizes the changes required to the entire system and it is a good starting point for forming the common basis. Note that in this mode the TEID is embedded in each SID.

Our reference topology is shown in Figure 2. In this mode we assume that the gNB and the UPFs are SR-aware.

 SRv6
 SRv6
 /
 \

 +--+
 +--+
 [N3] +---+
 [N9] +---+
 [N6]
 \

 |UE|-----|
 gNB |-----|
 UPF1 |------|
 UPF2 |----- DN
 /

 +--+
 +--++
 +---++
 +---+
 \\_\_\_\_\_/

 SRv6 node
 SRv6 node
 SRv6 node
 SRv6 node

Figure 2: Traditional mode - Reference topology

## 5.1.1. Packet flow - Uplink

The uplink packet flow is the following:

UE_out :	(A,Z)	
gNB_out :	(gNB, U1::1) (A,Z)	-> T.Encaps.Reduced <u1::1></u1::1>
UPF1_out:	(gNB, U2::1) (A,Z)	-> End.MAP
UPF2_out:	(A,Z)	-> End.DT4 or End.DT6

The UE packet arrives to the gNB. The gNB performs a T.Encaps.Reduced operations. 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. The SID U1::1 is retrieved through the existing control plane (N2 interface).

Upon packet arrival on UPF1, the SID U1::1 is a local End.MAP function. This function maps the SID with the next anchoring point and replaces U1::1 by U2::1, that belongs to the next anchoring point.

Upon packet arrival on UPF2, the SID U2::1 corresponds to an End.DT function. UPF2 decapsulates the packet, performs a lookup in a specific table and forwards the packet towards the data network.

Matsushima, et al. Expires September 6, 2018 [Page 7]

## 5.1.2. Packet flow - Downlink

The downlink packet flow is the following:

```
UPF2_in : (Z,A)
UPF2_out: (U2::, U1::1) (Z,A) -> T.Encaps.Reduced <U1::1>
UPF1_out: (U2::, gNB::1) (Z,A) -> End.MAP
gNB_out : (Z,A) -> End.DX4 or End.DX6
```

When the packet arrives to the UPF2, the UPF2 will map that particular flow into a UE session. This UE session is associated with the policy <U1::1>. The UPF2 performs a T.Encaps.Reduced 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 with the next anchoring point and replaces U1::1 by gNB::1, that belongs to the next anchoring point.

Upon packet arrival on gNB, the SID gNB::1 corresponds to an End.DX4/ End.DX6 function. The gNB will decapsulates the packet, removing the IPv6 header and all it's extensions headers and will forward the traffic towards the UE.

#### 5.1.3. 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 a lower overhead. In such case, the functions used are T.Insert.Red at gNB, End.MAP at UPF1 and End.T at UPF2 on Uplink, T.Insert.Red at UPF2, End.MAP at UPF1 and End.X at gNB on Downlink.

## **<u>5.2</u>**. Enhanced Mode (formerly Aggregate mode)

This mode improves the scalability. In addition, it provides key improvements in terms of traffic steering and service chaining, thanks to the use of an SR policy of multiple SIDs, instead of single one in the Traditional mode.

Key points:

- o Several UE share the same SR Policy (and it's composing SID)
- o The SR policy MAY include SIDs for traffic engineering and service chaining on top of the UPF anchor.

Matsushima, et al. Expires September 6, 2018 [Page 8]

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 through a mechanism like PCEP, DNS-lookup, small augment for LISP controlplane, etc.

Our reference topology is shown in Figure 3. In this mode we assume that the gNB and the UPF are SR-aware. We also assume that we have two services segments, S1 and C1. S1 represents a VNF in the network, and C1 represents a constraint path on a router over which we are going to perform Traffic Engineering. Note that S1 and C1 belong to the underlay and don't have an N4 interface. For this reason we don't consider them UPFs.



Figure 3: Enhanced mode - Reference topology

## 5.2.1. Packet flow - Uplink

The uplink packet flow is the following:

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 session to its gNB. gNB's CP associates that session from the UE(A) with the IPv6 address B and GTP TEID T. gNB's CP does a lookup on B (by reverseDNS, LISP, etc.) to find the related SID list <S1, C1, U2::1>.

Once the packet leaves the gNB, it already contains all the segments of the SR policy. This SR policy contains segments for traffic engineering (C1) and for service chaining (S1). Matsushima, et al. Expires September 6, 2018 [Page 9]

The nodes S1 and C1 perform their related Endpoint functionality and forward.

When the packet arrives to UPF2, the active segment (U2::1) is an End.DT4/6 which performs the decapsulation (removing the IPv6 header with all it's extension headers) and forward towards the data network.

Note that in case several APNs are using duplicated IPv4 private address spaces, then the aggregated SR policies are unique per APNs.

## 5.2.2. Packet flow - Downlink

The downlink packet flow is the following:

UPF2_in :	(Z,A)				->	UPF2 maps the flow w/
						SID list <c1,s1, gnb=""></c1,s1,>
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 to the UPF2, the UPF2 will map that particular flow into a UE session. This UE session is associated with the policy <C1, S1, gNB>. The UPF2 performs a T.Encaps.Reduced 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 to the gNB, the IPv6 DA corresponds to an End.DX4 or End.DX6 (depending on the underlying traffic). The gNB will decapsulate the packet, removing the IPv6 header and all it's extensions headers and will forward the traffic towards the UE.

## 5.2.3. 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 a lower overhead. In such case, the functions used are T.Insert.Red at gNB and End.T at UPF2 on Uplink, T.Insert.Red at UPF2 and End.X at gNB on Downlink.

### 5.3. Enhanced mode with unchanged gNB GTP behavior

In this section we introduce two mechanisms for interworking with legacy gNBs that still use GTP. One of the mechanisms is valid for IPv4 while the other for IPv6.

In this scenario, it is assumed that gNB does not support SRv6. It just supports GTP encapsulation over IPv4 or IPv6. Hence in order to achieve interworking we are going to add a new SR Gateway (SRGW-UPF1) entity. This SRGW is going to map the GTP traffic into SRv6. Note that the SR GW is not an anchor point.

The SRGW maintains very little state on it. For this reason, both of these methods (IPv4 and IPv6) scale to millions of UEs.



Figure 4: Reference topology for interworking

#### 5.3.1. Interworking with IPv6 GTP

In this interworking mode we assume that the gNB is using GTP over IPv6 in the N3 interface  $% \left[ 1 + 1 \right] = \left[ 1 + 1 \right] \left[ 1$ 

Key points:

- o gNB is unchanged (control-plane or user-plane) and encaps into GTP (N3 interface is not modified).
- o 5G Control-Plane (N2 interface) is unmodified: 1 IPv6 address (i.e. a BSID at the SRGW)
- o SRGW removes GTP, finds SID list related to DA, add 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 (leveraging the enhanced mode results in few SR policies on this node. A SR policy can be shared across UEs).
- o As soon as the packet leaves the gNB (uplink), the traffic is SR-routed. This simplifies considerably network slicing [<u>I-D.hegdeppsenak-isis-sr-flex-algo</u>].
- o In the uplink, we use the IPv6 DA BSID to steer the traffic into an SR policy when it arrives at the SRGW-UPF1-.

Our reference topology is shown in Figure 5. In this mode we assume that the gNB is an unmodified gNB using IPv6/GTP. The UPFs are SRaware. Also, as explained before, we introduce a new SRGW entity that is going to map the IPv6/GTP traffic to SRv6.

We also assume that we have two service segment, S1 and C1. S1 represents a VNF in the network, and C1 represents a router over which we are going to perform Traffic Engineering.

+---+ IPv6/GTP -| S1 |-+--+ +----+ [N3] / +---+ \ |UE|--| gNB |-SRv6 / SRv6 \ +----+ +----+ [N6] / +--+ +----+ \ [N9]/ NFV -| C1 |---| UPF2 |-----\ DN GTP \ +----+ / +---+ +----+ -| UPF1 |-SRv6 SRv6 +---+ ΤE SR Gateway

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

## 5.3.1.1. Packet flow - Uplink

The uplink packet flow is the following:

The UE sends a packet destined to Z towards the gNB on a specific bearer for that session. The gNB, which is unmodified, encapsulates the packet into a new 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 session, B, is now the IPv6 DA. B is an SRv6 Binding SID instantiated at the SRGW. Hence the packet, will be routed up to the SRGW.

When the packet arrives at the SRGW, the SRGW realises that B is an End.M.GTP6.D BindingSID. Hence, the SRGW will remove the IPv6, UDP and GTP headers, and will push a new IPv6 header with its own SRH containing the SIDs bound to the SR policy associated with this BindingSID.

The nodes S1 and C1 perform their related Endpoint functionality and forward.

Matsushima, et al. Expires September 6, 2018 [Page 12]

When the packet arrives to UPF2, the active segment is (U2::1) which bound to End.DT4/6 which is going to perform the decapsulation (removing the outer IPv6 header with all it's extension headers) and forward towards the data network.

## 5.3.1.2. Packet flow - Downlink

The downlink packet flow is the following:

When a packet destined to A arrives at the UPF2, the UPF2 performs a lookup in the associated table to A and finds the SID list <C1, S1, SRGW::TEID, gNB>. The UPF2 performs a T.Encaps.Reduced 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 to the SRGW, the SRGW realizes the active SID is an End.M.GTP6.E function. The SRGW removes the IPv6 header and all it's extensions headers. The SRGW generates an 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 the arguments of the received End.M.GTP6.E SID. The SRGW pushes the headers to the packet and forwards the packet towards the gNB.

Once the packet arrives to 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 imposed by the UPF2. The UPF2 must have the UE states as the session anchor point.

For the uplink traffic, the state at the SRGW does not necessarily need to be per UE session basis. A state of SR policy of which state can be shared among UE's. Hence it is possible to deploy SRGW in

Matsushima, et al. Expires September 6, 2018 [Page 13]

very scalable way compared to hold millions of states per UE session basis.

#### 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 a lower overhead.

## 5.3.2. Interworking with IPv4 GTP

In this interworking mode we assume that the gNB is using GTP over  $\ensuremath{\mathsf{IPv4}}$  in the N3 interface

Key points:

- o gNB is unchanged and encaps into GTP (N3 interface is not modified).
- o In the uplink, traffic is classified at SRGW by UL CL(Uplink Classifier) and steered into an SR policy. The SRGW is a UPF1 functionality, hence it can coexist with UPF UL CL functionality.
- o SRGW removes GTP, finds SID list related to DA, add SRH with SID list.

Our reference topology is shown in Figure 6. In this mode we assume that the gNB is an unmodified gNB using IPv4/GTP. The UPFs are SR-aware. Also, as explained before, we introduce a new SRGW entity that is going to map the IPv4/GTP traffic to SRv6.

We also assume that we have two service segment, S1 and C1. S1 represents a VNF in the network, and C1 represents a router over which we are going to perform Traffic Engineering.



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

## 5.3.2.1. Packet flow - Uplink

The uplink packet flow is the following:

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.Tmap 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 towards 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 to the SRGW -UPF1-, the SRGW has an UL CL (uplink classifier) rule for incoming traffic from the gNB that steers the traffic into an SR policy by using the function T.M.TMap. 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.

The nodes S1 and C1 perform their related Endpoint functionality and forward.

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 it's extension headers) and forwards towards the data network.

## 5.3.2.2. Packet flow - Downlink

The downlink packet flow is the following:

When a packet destined to A arrives to the UPF2, the UPF2 performs a lookup in the associated table to A and finds the SID list <C1, S1, SRGW::SA:DA:TEID>. The UPF2 performs a T.Encaps.Reduced operation,

Matsushima, et al. Expires September 6, 2018 [Page 15]

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 to the SRGW, the SRGW realizes the active SID is an End.M.GTP4.E function. The SRGW removes the IPv6 header and all it's extensions headers. The SRGW generates an IPv4, UDP and GTP headers. The IPv4 SA and DA will the ones received as part of the 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 towards the gNB.

Once the packet arrives to 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 imposed by the UPF. The UPF must have this UE-base state anyway (it is its anchor point).

For the uplink traffic, the state at the SRGW is dedicated on a per UE/session basis. This is an UL CL (uplink classifier). There is state for steering the different sessions on a SR policies. Notice however that the 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. However 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. End.MAP: Endpoint function with SID mapping

The "Endpoint function with SID mapping" function (End.MAP for short) is used in several scenarios. Particularly in mobility, it is used in the UPFs for the anchor functionality in some of the use-cases.

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

- 1. look up the IPv6 DA in the mapping table
- update the IPv6 DA with the new mapped SID ;; Ref1
- 5. forward according to the new mapped SID
- 8. ELSE
- 9. Drop the packet

Ref1: Note that the SID in the SRH is NOT modified.

# <u>6.2</u>. End.M.GTP6.D: Endpoint function with decapsulation from IPv6/GTP tunnel

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 towards from the legacy gNB using IPv6/GTP. 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\_PORT = GTP THEN

- 2. pop the IP, UDP and GTP headers
- 3. push a new IPv6 header with its own SRH <S2, S3>
- 4. set the outer IPv6 SA to A
- 5. set the outer IPv6 DA to S1
- 6. forward according to the first segment of the SRv6 Policy
- 7. ELSE
- 8. Drop the packet

# <u>6.3</u>. End.M.GTP6.E: Endpoint function with encapsulation for IPv6/GTP tunnel

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

The End.M.GTP6.E function has a 32-bit argument space. This argument corresponds to the GTP TEID.

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

```
1. IF NH=SRH & SL = 1 THEN
                                                             ;; Ref1
2.
      decrement SL
      store SRH[SL] in variable new_DA
3.
4.
      store TEID in variable new_TEID
                                                             ;; Ref2
 5.
      pop IP header and all it's extension headers
      push new IPv6 header and GTP-U header
 6.
      set IPv6 DA to new_DA
7.
      set GTP_TEID to new_TEID
8.
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.

# <u>6.4</u>. End.M.GTP4.E: Endpoint function with encapsulation for IPv4/GTP tunnel

The "Endpoint function with encapsulation for IPv4/GTP tunnel" function (End.M.GTP4.UP 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:

IF NH=SRH & SL > 0 THEN
 decrement SL
 update the IPv6 DA with SRH[SL]
 pop the SRH
 push header of TUN-PROTO with tunnel ID from S ;; Ref1
 push outer IPv4 header with SA, DA from S
 ELSE
 Drop the packet

Matsushima, et al. Expires September 6, 2018 [Page 18]

Ref1: TUN-PROTO indicates target tunnel type.

Note that S has the following format:

+----+ | SRGW-IPv6-LOC-FUNC |IPv4DA |IPv4SA |TUN-ID | +----+ 128-a-b-c a b c

End.M.GTP4.E SID Encoding

# 6.5. T.M.Tmap: Transit behavior with IPv4/GTP decapsulation and mapping into an SRv6 Policy

The "Transit with tunnel decapsulation and map to an SRv6 policy" function (T.Tmap 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:

IF P.PLOAD == TUN-PROTO THEN ;; Ref1
 pop the outer IPv4 header and tunnel headers
 copy IPv4 DA, SA, TUN-ID to form SID B with SRGW-IPv6-Prefix
 encapsulate the packet into a new IPv6 header ;; Ref2, Ref2bis
 set the IPv6 DA = B
 forward along the shortest path to B
 ELSE
 Drop the packet

Ref1: TUN-PROTO indicates target tunnel type.

Note that B has the following format:

+----+ | SRGW-IPv6-LOC-FUNC |IPv4DA |IPv4SA |TUN-ID | +----+ 128-a-b-c a b c

End.M.GTP4.E SID Encoding

Note that the B SID, is going to be an SRv6 BindingSID instantiated at the first UPF (anchor point). A static format is leveraged to instantiate this Binding SIDs in order to remove state from the SRGW.

Matsushima, et al. Expires September 6, 2018 [Page 19]

#### 6.6. End.Limit: Rate Limiting function

Mobile user-plane requires a rate-limit feature. SID is able to encode limiting rate as an argument in SID. Multiple flows of packets should have same group identifier in SID when those flows are in an same AMBR group. This helps to keep user-plane stateless. That enables SRv6 endpoint nodes which are unaware from the mobile control-plane information. Encoding format of rate limit segment SID is following:

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

End.Limit: Rate limiting function argument format

In case of j bit length is zero in SID, the node should not do rate limiting unless static configuration or control-plane sets the limit rate associated to the SID.

### 7. 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.

A simple way to represent slice would be to apply L2/L3 VPN described in [<u>I-D.filsfils-spring-srv6-network-programming</u>]. Segment Routing with [<u>I-D.hegdeppsenak-isis-sr-flex-algo</u>] provides even more advanced separation based on metrics like link-delay. Thus, a service provider would be able to have network slices per required SLA.

The SRv6 SID and quite a few SR extended capability would be a powerful tool for providing logical separation/integration within a network. Details are for further study.

#### 8. Control Plane Considerations

This documents focuses on the dataplane behavior. The control planes could be based on the existing 3GPP based signalling for N4 interface [TS.29244], [I-D.ietf-dmm-fpc-cpdp], control-plane protocols described in [WHITEPAPER-5G-UP], etc. and to be discussed further.

Note that the IANA section of this document allocates the SRv6 endpoint function types for the new functions defined in this document. All control-plane protocols are expected to leverage these function type-codes to signal each function.

It's notable that SRv6's network programming nature allows a flexible and dynamic anchor placement.

### 9. Security Considerations

TBD

#### **10**. IANA Considerations

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

+	++		++
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

## **<u>11</u>**. Acknowledgements

The authors would like to thank Daisuke Yokota, Bart Peirens, Ryokichi Onishi, Kentaro Ebisawa, Peter Bosch and Darren Dukes for their useful comments of this work.

## **<u>12</u>**. References

#### <u>12.1</u>. Normative References

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Leddy, J., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Steinberg, D., Raszuk, R., Matsushima, S., Lebrun, D., Decraene, B., Peirens, B., Salsano, S., Naik, G., Elmalky, H., Jonnalagadda, P., Sharif, M., Ayyangar, A., Mynam, S., Henderickx, W., Bashandy, A., Raza, K., Dukes, D., Clad, F., and P. Camarillo, "SRv6 Network Programming", <u>draft-filsfils-</u> <u>spring-srv6-network-programming-03</u> (work in progress), December 2017.

[I-D.ietf-spring-segment-routing]

Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", <u>draft-ietf-spring-segment-routing-15</u> (work in progress), January 2018.

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

## **<u>12.2</u>**. Informative References

[I-D.gundavelli-dmm-mfa]

Gundavelli, S., Liebsch, M., and S. Matsushima, "Mobilityaware Floating Anchor (MFA)", <u>draft-gundavelli-dmm-mfa-00</u> (work in progress), February 2018.

[I-D.hegdeppsenak-isis-sr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., and A. Gulko, "ISIS Segment Routing Flexible Algorithm", <u>draft-hegdeppsenak-</u> <u>isis-sr-flex-algo-02</u> (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", <u>draft-ietf-dmm-fpc-cpdp-09</u> (work in progress), October 2017.

[RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", <u>RFC 5213</u>, DOI 10.17487/RFC5213, August 2008, <<u>https://www.rfc-editor.org/info/rfc5213</u>>.

## [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.

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

Matsushima, et al. Expires September 6, 2018 [Page 23]