

DMM Working Group
Internet-Draft
Intended status: Informational
Expires: 9 January 2021

J. Auge
G. Carofiglio
L. Muscariello
M. Papalini
Cisco
8 July 2020

Anchorless mobility through hICN
draft-auge-dmm-hicn-mobility-04

Abstract

This document first discusses the use of locators and identifiers in mobility management architectures, and their implication on various anchorless properties. A new architecture is then proposed that is purely based on identifiers, and more specifically names as defined in Hybrid-ICN (hICN). The document then focuses on two main cases: the end-point sends data (data producer) or the end-point receives data (data consumer). These two cases are taken into account entirely to provide anchorless mobility management in hICN.

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 9 January 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights

and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Locators, identifiers and anchorless mobility management . .	3
2.1. Terminology	4
2.2. Towards locator-independent network architectures	4
2.3. Information-Centric Networking (ICN)	6
2.4. Hybrid-ICN overview	6
3. Hybrid-ICN Anchorless Mobility Management (hICN-AMM)	6
3.1. Consumer mobility in hICN	7
4. Producer mobility architectures	7
4.1. Producer mobility in hICN	8
5. Protocol description	8
5.1. Signalization messages; acknowledgements and retransmission	9
5.2. Dynamic face creation and producer-triggered advertisements	9
5.3. Update protocol	10
5.3.1. Illustration	10
5.3.2. Message content	11
5.3.3. Processing at network routers	12
5.4. Notifications and scoped discovery	12
5.4.1. Notification processing	12
5.4.2. Illustration	13
6. Benefits	14
6.1. Overview	14
6.2. Simplicity, scalability, efficiency	15
6.3. Reduced latency through caching	15
6.4. Improved reliability through caching	17
6.5. Local mobility and recovery from common cache	17
6.6. Additional reliability through consumer multihoming . . .	18
6.7. Bandwidth aggregation with consumer multihoming	20
6.8. Traffic and signalization offload	21
7. Implementation considerations	22
7.1. Interaction with non-hICN enabled routers	23
7.2. Security considerations	23
7.3. Discussion	23
8. IANA Considerations	24
9. References	24
9.1. Normative References	24
9.2. Informative References	25
Authors' Addresses	27

1. Introduction

New usages of the network and the rapid growth of the Mobile Internet calls to reconsider the way we deploy and operate IP networks, where mobility is not built into the design, but rather added as an afterthought. Notable examples are IETF Mobile IP and its variants [RFC5944] and [RFC2275] 3GPP GTP-based architecture [TS29.274], both based on tunnelling and encapsulation.

One identified difficulty in proposing mobility models for IP lies in the semantic overloading of IP addresses which are both host identifiers, and locators used for routing. Starting with LISP, the identifier/locator split paradigm has shown promising results in virtue of its scalability properties with respect to routing tables entries, and the possibilities it offers in terms of mobility. Several solutions have been proposed around this concept, namely ILNP, ILSR, and ILA. One common facet of these proposals is to embrace the current trend of a clear separation between the control and data planes, which both allows for distribution of the control infrastructure, as well as anchorless operations in the data plane, including facilitated local breakout.

The counterpart of these architectures is an increased dependency on the control plane which is responsible for binding the identifiers used by the application at the edge to the locators used to forward the traffic in the network. Device mobility will typically induce a change of IP address, which makes performance of flows in progress dependent on interactions with those control elements which have to remain globally consistent.

In this document, we first propose to clarify protocol descriptions by adopting a new terminology of control-plane and data-plane anchors to characterize anchorless operations, and show their tight coupling with the use of locators and identifiers. This definition serves to position Loc/ID-split architectures with respect to the traditional use of tunnels, before advocating to push this step further and perform mobility management purely based on identifiers. We introduce a mobility approach based on Hybrid ICN (hICN) as described in [I-D.muscariello-intarea-hicn], for which we perform an in-depth analysis of mobility considerations. We show how this proposal can help addressing further challenges faced by networks today, such as multihoming and multipath, while preserving the simplicity and end-to-end design of the current Internet.

2. Locators, identifiers and anchorless mobility management

2.1. Terminology

The consideration of mobility in network design shares the challenges raised for routing for instance in [RFC6115], where the terminology for locators and identifiers is presented.

Because they are intrinsically bound to the topological location of a node, session established through locators require additional mechanisms to support mobility, including the presence of anchor points in the network. The notion of `_anchor_` and the resulting `_anchorless_` properties of mobility schemes are prone to different interpretation depending on the context. The following definitions attempt to reconcile those interpretations and propose a unifying terminology used throughout this document.

Anchor An anchor refers to a specialized node or service, possibly distributed, functionally required by a network architecture for forwarding or mobility. This is in contrast to decentralized architectures. Configuration of these anchors, including their location, will directly affect the overall scheme performance. We follow the classic distinction between control plane and user (or data, or forwarding) plane to distinguish control-plane anchors and user-plane anchors.

User-plane anchor A user-plane anchor is a node through which traffic is forced to pass. An example of such anchor is the indirection point in Mobile IP.

Control-plane anchor A control-plane anchor refers to a node that is not responsible for carrying traffic, but is needed for the operation of the forwarding and/or the mobility architecture. An example of such anchor is the resolution or mapping service of LISP. We remark that while not being on path, such anchors might affect the performance of the user-plane due to resolution delays or indirection (following a mapping cache miss for instance).

Anchorless This term qualifies approaches that do not involve any user-plane nor control-plane anchor. The challenge for such full anchorless approaches are exacerbated during mobility of both end points at the same time, as they cannot rely on any stable anchor point to preserve connectivity. Nor they can rely on routing mechanism that would be the source of overhead and instability.

2.2. Towards locator-independent network architectures

We distinguish network architectures based on their use of location independent identifiers (or names) and locators for forwarding.

Locator-based architectures

As mentioned earlier, IP architectures are typically operated based on locators corresponding to the IP addresses of the host interfaces in their respective network attachment, used also as session identifiers. This results in a complex mobility architecture built on top, involving traffic anchors and tunnels to preserve the identifier exposed to the transport layer: IP/IP or GRE tunnels in Mobile IP, and GTP tunnels in 3GPP architectures.

The limitations of locator-based schemes in terms of complexity, overhead and efficiency are well-recognized and led to other alternatives to be considered.

Locator-ID separation architectures

LISP [RFC6830] was the first proposal to distinguish between the usage of IP addresses as locators or identifiers by explicitly defining two namespaces, respectively used for endpoint identification and forwarding. A mapping service is further used to bind an identifiers to a given location, and updated after mobility. From there, several approaches have been defined, either host-based like SHIM6 [RFC5533] or HIP [RFC4423]), or network-based, like LISP [RFC6830], ILSR, ILNP [RFC6740] or ILA [I-D.herbert-intarea-ila] to cite a few.

An overview of these approaches and their use in mobility is presented in {I-D.bogineni-dmm-optimized-mobile-user-plane}}.

The main challenge consists in maintaining a (distributed) mapping service at scale, including the synchronization of local caches required for scalability and efficiency.

ID-based architectures

A third class of approaches exists that redefines IP communication principles (i.e. network and transport layers) around location-independent network identifiers of node/traffic. The interest of such architectures is highlighted in [I-D.vonhugo-5gangip-ip-issues] (referred to as ID-oriented Networking) for it removes the limitations introduced by locators and simplifies the management of mobility. The draft however question the possibility to realize such an architecture where node status and mobility would not affect routing table stability.

The work done around Information-Centric Networking (ICN) falls into such class of approaches that we refer to as purely ID-based, also known as name-based [I-D.irtf-icnrg-terminology], although as we will see, mobility management often departs from this principle.

2.3. Information-Centric Networking (ICN)

ICN is a new networking paradigm centering network communication around named data, rather than host location. Network operations are driven by location-independent data names, rather than location identifiers (IP addresses) to gracefully enable user-to-content communication.

Although there exist a few proposals, they share the same set of core principles, resulting in several advantages including a simplified mobility management [RFC7476]. For clarity, this section we focus on hICN [I-D.muscariello-intarea-hicn] an ICN implementation for IPv6.

2.4. Hybrid-ICN overview

Hybrid ICN (hICN) is an ICN architecture that defines integration of ICN semantics within IPv6. The goal of hICN is to ease ICN insertion in existing IP infrastructure by:

1. selective insertion of hICN capabilities in a few network nodes at the edge (no need for pervasive fully hICN network deployments);
2. guaranteed transparent interconnection with hICN-unaware IPv6 nodes, without using overlays;
3. minor modification to existing IP routers/endpoints;
4. re-use of existing IP control plane (e.g. for routing of IP prefixes carrying ID-semantics) along with performing mobility management and caching operations in forwarding plane;
5. fallback capability to traditional IP network/transport layer.

hICN architecture is described in [I-D.muscariello-intarea-hicn].

3. Hybrid-ICN Anchorless Mobility Management (hICN-AMM)

hICN, together with MAP-Me [I-D.irtf-icnrg-mapme], forms the basis for the mobility management architecture we describe in the rest of this document. Due to the pull based nature of hICN architecture, we distinguish consumer and producer nodes, for which mobility is handled differently

3.1. Consumer mobility in hICN

The consumer end-point is the logical communication termination that receives data. Due to the pull-based and connection-less properties of hICN communications, consumer mobility comes natively with ICN. It is indeed sufficient that the consumer reissues pending interests from the new point-of-attachment to continue the communication. Consumer mobility is anchorless by design, and managed without any impact on the transport session. It is however necessary to have an appropriate transport layer on top able to cope with eventual disruptions and path variations caused by the mobility event.

4. Producer mobility architectures

The producer end-point is the logical communication termination that sends data. Producer mobility is not natively supported by the architecture, rather handled in different ways according to the selected producer mobility management scheme, some of which diverge from the concept of pure ID-based architecture through their use of locators. Additional procedures have to be performed to maintain reachability as it moves in the network.

In fact, many schemes proposed for ICN are adaptations to the vast amount of work made in IP over the last two decades [RFC6301]. Surveys for the ICN family, resp. for CCN/NDN-specific solutions, are available in [SURVEYICN], respectively [SURVEY1] and [SURVEY2]. There has been however a recent trend towards anchorless mobility management, facilitated by ICN design principles, that has led to new proposals and an extension of previous classifications in [I-D.irtf-icnrg-mapme] and [MAPME] to the four following categories:

- * Resolution based solutions rely on dedicated rendez-vous nodes (similar to DNS) which map content names into routable location identifiers. To maintain this mapping updated, the producer signals every movement to the mapping system. Once the resolution is performed, packets can be correctly routed directly to the producer.
- * Anchor-based proposals are inspired by Mobile IP, and maintain a mapping at network-layer by using a stable home address advertised by a rendez-vous node, or anchor. This acts as a relay, forwarding through tunneling both interests to the producer, and data packets coming back.

- * _Tracing-based_ solutions allow the mobile node to create a hop-by-hop forwarding reverse path from its RV back to itself by propagating and keeping alive traces stored by all involved routers. Forwarding to the new location is enabled without tunneling.
- * _Anchorless_ approaches allow the mobile nodes to advertise their mobility to the network without requiring any specific node to act as a rendez-vous point.

4.1. Producer mobility in hICN

In an hICN network, regular routing protocols such as BGP, ISIS or OSPF can be used for propagating all prefix announcements and populate routers' FIBs. However, these protocols are not appropriate and should not be used to manage name prefix mobility, for scalability and consistency reasons.

The default mobility management for hICN is designed following the same principles and protocols as MAP-Me, an anchorless producer mobility management protocol initially proposed for ICN [I-D.irtf-icnrg-mapme] [MAPME]. It builds on an initial forwarding state bootstrapped by the routing protocol, and performs a lightweight path repair process as the producer moves. For simplicity, we refer to it as simply MAP-Me in the rest of the document.

In the rest of this section, we describe the specific realization of the protocol in an hICN context. Additional background and details are available in [I-D.irtf-icnrg-mapme] and [MAPME]. The solution is based on a path repair mechanism following mobility events, using dynamic FIB updates. Using data plane mechanisms for ensuring connectivity has been previously proposed in [DATAPLANE] to handle link failures, and has been proven more reactive than relying on typical control plane messaging.

5. Protocol description

5.1. Signalization messages; acknowledgements and retransmission

Signalization messages follow hICN design principles and use data plane packets for signalization. Signalization messages and acknowledgement are respectively Interest and Data packet (requests and replies) according to hICN terminology [I-D.muscariello-intarea-hicn]. Upon processing of those advertisements, the network will send an acknowledgement back to the producer using the name prefix as the source and the locator of the producer as the destination, plus the sequence number allowing the producer to match which update has been acknowledged.

Pending signalization interests that are not acknowledged are retransmitted after a given timeout.

5.2. Dynamic face creation and producer-triggered advertisements

The producer is responsible for mobility updates and should be hICN-enabled. It stores a sequence number incremented at each mobility event.

Faces in the producer are assumed synchronized with layer 2 adjacencies, upon joining a new point-of-attachment, a new face should be created. Face creation will trigger the increase of the sequence number, and per-prefix advertisement packets to be sent to the joined network.

Those advertisements should contain:

- * the name prefix as the destination address, plus a field indicating the associated prefix length;
- * the locator of the producer as the source address, that will serve for receiving acknowledgements;
- * a sequence number sequentially increased by the producer after each movement;
- * a security token (see Section 7.2).

Upon producer departures from a PoA, the corresponding face is destroyed. If this leads to the removal of the last next hop, then faces that were previously saved are restored in FIB to preserve the original forwarding tree and thus global connectivity.

5.3. Update protocol

Based on the information transmitted in the packet, and its local the network's local policy, the network might decide to update its forwarding state to reflect this change.

The update process consists in updating a few routers on the path between the new and a former point-of-attachment. More precisely, this is done in a purely anchorless fashion by sending a signalling message from the new location towards the name prefix itself. This packet will be forwarded based on the now-stale FIB entries, and will update forwarding entries to point to the ingress interfaces of each traversed router.

5.3.1. Illustration

{fig-mapme-update}} illustrates the situation of a P moving between access routers AR1 and AR2 while serving user requests:

1. A first interest towards the producer originates from remote. The producer answers with a Data packet.
2. Following this, the producer detaches from AR1 and moves to AR2.
3. As soon as it is attached to AR2, the producer sends an Interest Update towards its own prefix, which is forwarded from AR2 following the FIB towards AR1 (one of its former positions in the general case).
4. At each hop, the message fixes the FIB to point towards the ingress face, until the message cannot be further forwarded in AR1.
5. A subsequent message from remote will follow the updated FIB and correctly reach the producer's new location in AR2.

- * a sequence number sequentially increased by the producer after each movement;
- * an optional security token.

5.3.3. Processing at network routers

At the reception of advertisement/update packets, each router performs a name-based Longest Prefix Match lookup in FIB to compare sequence number from the received packet and from FIB}. According to that comparison:

- * if the packet carries a higher sequence number, the existing next hops associated to the lower sequence number in FIB are used to forward it further and temporarily stored to avoid loss of such information before completion of the acknowledgement process.
- * If the packet carries the same sequence number as in the FIB, its originating face is added to the existing ones in FIB without additional packet processing or propagation. This may occur in presence of multiple forwarding paths.
- * If the packet carries a lower sequence number than the one in the FIB, FIB entry is not updated as it already stores 'fresher information'. To advertise the latest update through the path followed by the packet, this one is re-sent through the originating face after having updated its sequence number with the value stored in FIB.

5.4. Notifications and scoped discovery

The update protocol is responsible for reestablishing global connectivity with minimal changes to the FIBs. In order to further improve the reactivity of the scheme and better support QoS constraints of latency-sensitive traffic, we propose an additional mechanism named *Notifications*. It assumes hICN-enabled routers at the edge, and the existence of links between access routers, which are typically used for handover, and proposes to exploit them during mobility events, or to delay updates when possible.

5.4.1. Notification processing

Upon receiving a valid advertisement, the point-of-attachment will remember the presence of the producer, update its corresponding FIB entry but send no update. Previous next hops should be saved and restored upon face deletion so as to preserve the forwarding tree/DAG structure.

6. Because the producer is attached to AR2, the producer can be directly reachable. Otherwise, AR2 would iterate the discovery to neighbours only if it had more recent information about the producer location than its predecessor (based on sequencing of regular Interests and Interest Updates).
7. The Data packet follows the reverse path to the consumer as usual.

6. Benefits

We now review the potential benefits of the general architecture we have presented using features from the hICN data plane for supporting consumer and producer mobility.

6.1. Overview

Native mobility : This mobility management process follows and exploits hICN design principles. It makes producer mobility native in the architecture by preserving all benefits of hICN even when consumers and/or producers are moving.

Anchorless mobility : Such approach belongs to the category of pure ID-based mobility management schemes whose objective is (i) to overcome the limitations of traditional locator-based solutions like Mobile IP (conf)using locators as identifiers and requiring tunnels, and (ii) to remove the need for a global mapping system as the one required by locator-identifier separation solutions. The result is a fully anchorless solution both in the data plane and in the control plane.

Local and decentralized mobility management : Mobility updates are handled locally and the routers that are affected are those on path between successive positions of the producer. In particular, remote endpoints are not affected by the event. Mobility is managed in a fully distributed manner and no third party is required. This does not prevent any centralized control (as discussed in [I-D.auge-dmm-hicn-mobility-deployment-options]), but makes the network robust to disconnectivity events.

The next section will discuss more in depth the following advantages

6.2. Simplicity, scalability, efficiency

As emerges from the points raised in the previous section, consumer mobility is transparently supported by an hICN network in virtue of the pull-based model and the way the forwarding path works. After moving, a consumer can just reissue pending interests once attached to the new access router at layer 2, without requiring any more information from L3 and above. This ensures a fast and simple handover, which can be further enhanced through additional mechanisms such as in-network caching. Consumer mobility is fully anchorless with hICN, and does not incur any signalization nor tunneling overhead.

This is particularly interesting considering that most mobile users are consumers only (e.g. linear video distribution, or large scale video conferencing where we typically have few presenters and most users are simply consumers).

Another aspect of using the unifying hICN architecture in replacement of the traditional tunnel-based mobile core is that it removes the need to maintain state for consumers and producers which are not mobile (eg. IoT sensors), or not currently moving.

6.3. Reduced latency through caching

While this is not strictly linked to mobility, we first illustrate the benefits of caching content close to the edge to reduce user latencies. This is particularly important for wireless networks such as WiFi or LTE which have non-negligible latencies especially during connection setup, or after an idle period. The characteristics of those radio networks are thus of interest for the performance of the mobility architecture as a whole.

The example in Figure 3 considers a mobile node that can move access two accesses linked to Access Routers (AR) AR1 and AR2, both connected to the Internet through a common gateway (GW). This same setup will be later used to illustrate the flow of packets during mobility events, eventually specializing AR into AP or eNB when it makes more sense.

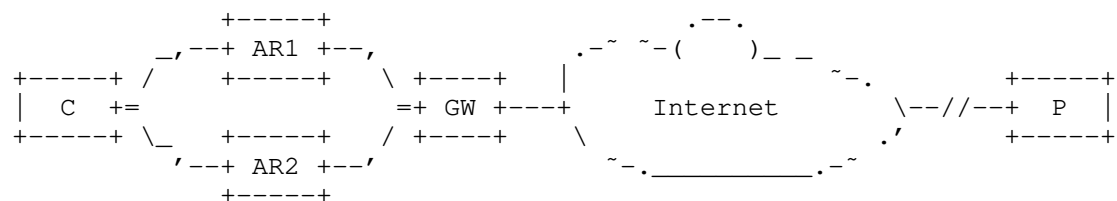
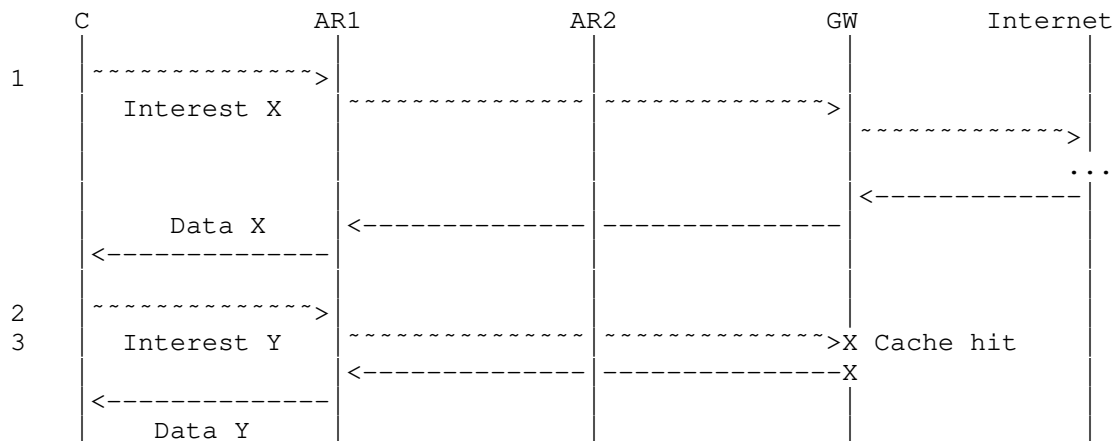


Figure 3: Simple topology with a multi-homed consumer

We represent in Figure 4 a simple data flow between the different entities involved in the communication between the mobile node M, and a remote producer available over the Internet.



LEGEND:

~~~~~>: Interest      <---- Data      =====> Signalization  
 ....-: L2 detach      .....+ L2 attach      X failure      o event

Figure 4: Reduced latency through caching

Numbers on the left refer to the following comments relative to the data flow:

1. The consumer issues a first interest towards name prefix X, which is transported up to the producer. A Data packet comes back following the reverse path and populating intermediate caches. Latency of the exchange can be seen by the distance between lines on the vertical axis.
2. The consumer now requests content B, which has previously been requested by another consumer located on the same gateway. Interest B thus hits the cache, refreshes the entry, and allows a lower round-trip latency to content both for consumer C and any subsequent request of the same content.



#### 6.4. Improved reliability through caching

Mobile networks might consist in unreliable access technologies, such as WiFi, responsible for packet losses. Figure 5 considers a similar scenario with a lossy channel (eg. WiFi) between the mobile and the first access router.

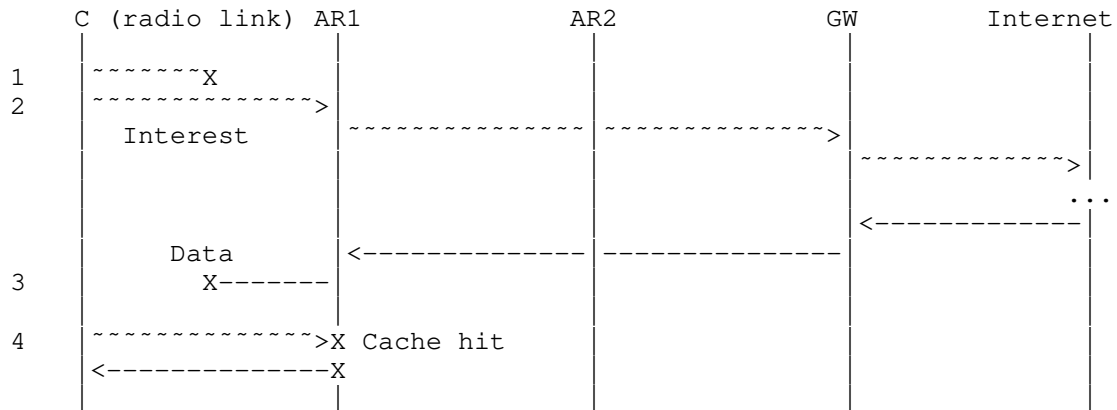


Figure 5: IU propagation example

1. The first issued interest is lost due to bad radio conditions.
2. Upon detection (either after a timeout, the consumer can just reissue the lost Interest).
3. This time the remote producer successfully replies but the Data packet is lost on the radio link.
4. Upon a similar detection, the consumer can reissue the lost Interest that will hit a locally stored copy at the router where the loss occurred (or again use a detection mechanism to retransmit the Data packet).

#### 6.5. Local mobility and recovery from common cache

The same mechanism can be used to recover from mobility losses after the consumer reconnects to the new network as the content will already be available in the cache at the junction router between the two accesses. This is particularly interesting in case of micro-mobility between access routers that are topologically close in the network.



Figure 7 illustrates a fallback to LTE which can be performed transparently for the application.

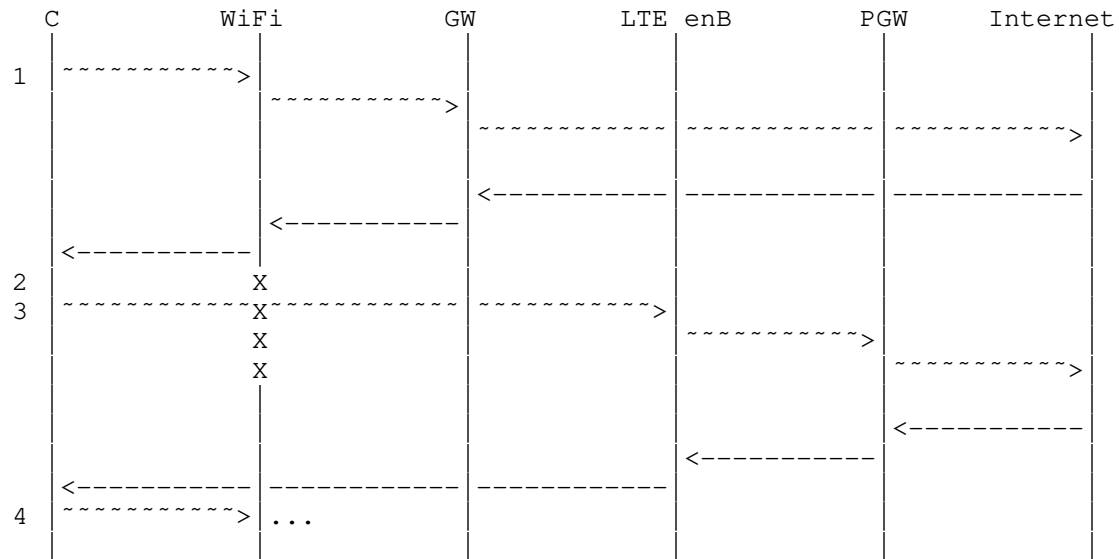


Figure 7

1. First interest is sent on the WiFi link.
2. WiFi unavailability due to failure for instance.
3. The application seamlessly switches to the LTE link.
4. Upon recovery, the traffic can be brought back on the WiFi interface.

hICN handles consumer mobility from one access to the other (e.g. WiFi to LTE or vice-versa) without any a-priori knowledge of the multiple networks to use as it is the case of MPTCP or QUIC approaches. Moving rate and congestion control at the receiver end results to be a significant advantage w.r.t. all existing alternatives in controlling dynamically multiple and new discovered network accesses in presence of mobility.

This use case highlights the importance of having a compatible transport, as the WiFi and LTE paths will have much different characteristics in terms of delay, jitter and capacity.

Evaluations of the scheme have shown this scheme to preserve the performance of flows like mobile video distribution up to very high switching rates in the order of a second, not even considering optimization occurring from network support.

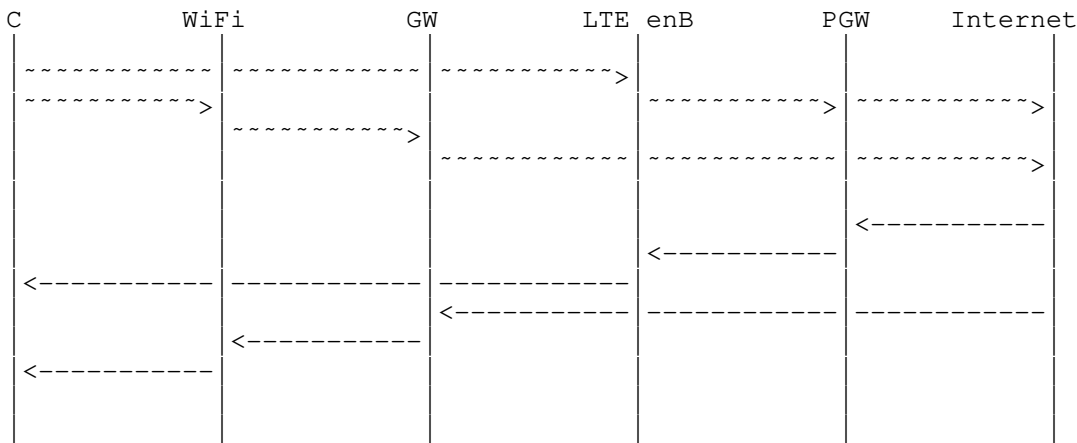
Mobility is handled transparently at the network layer with very fast handover times, due to the connection-less property of hICN. This makes it possible to offer reliable WiFi connectivity, besides its lossy nature as observed in previous use case and besides the frequency of mobility from one network access to the other.

#### 6.7. Bandwidth aggregation with consumer multihoming

Bandwidth aggregation can be realized dynamically through a congestion-aware load-balancing forwarding strategy at the client, with no a priori knowledge of paths. This is done similarly in the network by hICN forwarders, allowing a combination of multi-homing, multipath and multi-source data transfers. As all the paths are used at the same time, hICN offers the full network capacity to the users and tends to smooth fluctuations due to the radio channels.

Over an heterogeneous network access, hICN also offers a simple and cost-effective realization of heterogeneous channel bonding allowing an user to seamlessly roam across different radios or fixed lines (for increased reliability or reduced costs), or aggregate their bandwidth for high-throughput applications such as video streaming.

We illustrate a simplified data flows with one mobile consumer C alternating interests between a WiFi and LTE access points. The load balancing strategy would in that case optimize the split ratio between the two access to realize an optimal split. Note that in that case the relative distances on the vertical axis have not been respected here for readability.



Fine grained control from the application allows fully exploiting available bandwidth, resulting in an aggregated throughput equal to the sum of access throughputs, which is hard to achieve with existing solutions.

6.8. Traffic and signalization offload

As a natural consequence of its anchorless behavior, an hICN network can continue to operate in disconnected mode, for local mobility, even though no upstream entity is available.

In order to illustrate this, we slightly extend the previous topology in Figure 8 with an additional access network. To show that local mobility induces no traffic on upstream links, we further assume a failure on the link between the gateway (GW) and the Internet.

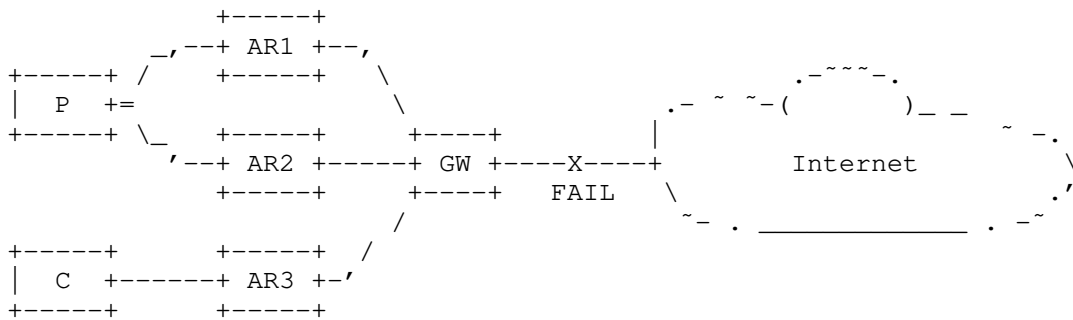


Figure 8: Access network disconnected from core

The data flow represented in Figure 9 illustrates the communication between a consumer connected to AR3, and a mobile producer moving from AR1 to AR2.

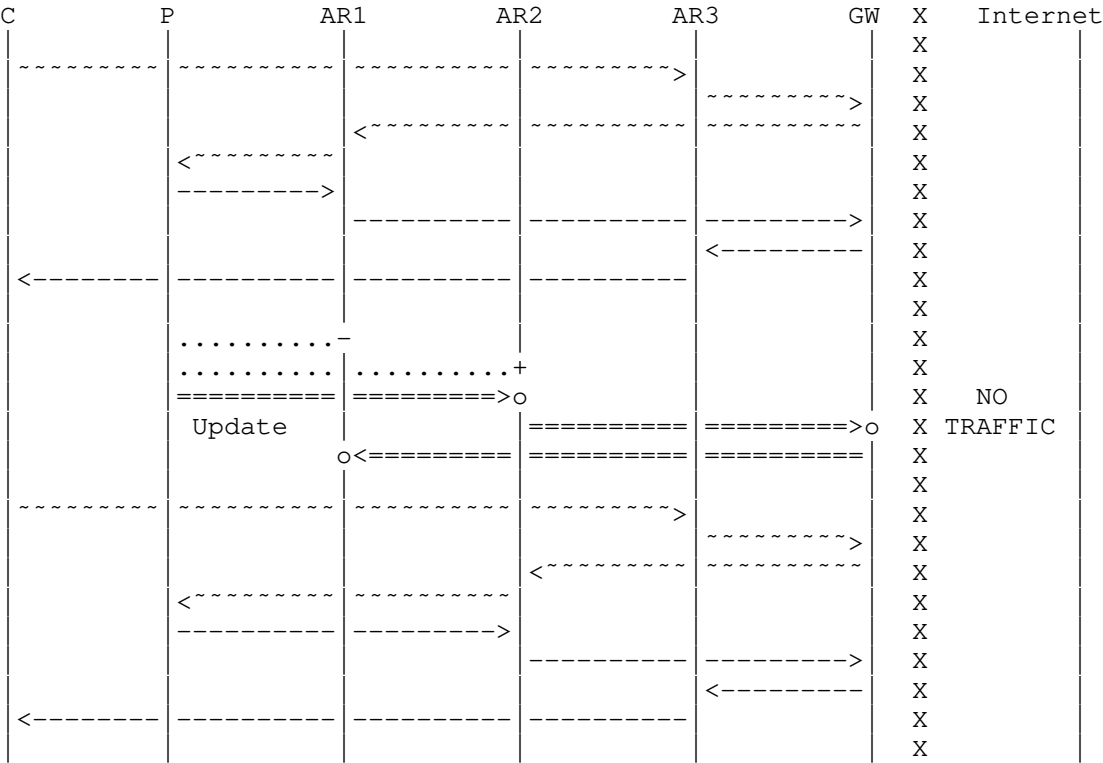


Figure 9: Anchorless mobility in network disconnected from core

We see that both the data and the signalization remain local to the zone where the mobility occurs, and that communications during mobility are not affected by the failure of the link upstream. This is a sign that the mobile core is not loaded with unnecessary traffic, and that communications remain local, thus improving user flow latencies. The offload of both data and signalization allows reducing the cost of the infrastructure by increasing the diversity of resources used at edge, mutualizing their capacity, and lower requiring network and compute capacity in the mobile core. A direct consequence is also a more robust and reliable network.

7. Implementation considerations

### 7.1. Interaction with non-hICN enabled routers

The realization of the architecture in a partial hICN deployment where some routers are not extended to support hICN mechanisms requires either to introduce additional functionalities or protocol support, or to reuse existing protocols achieving similar objectives (following hICN design).

One such example is the combination of ICMP redirect messages and Neighbour Discovery Proxies (NDProxy), that partially realizes the objectives of the update process:

- \* ICMP packets do not include sequence numbers however they can be transported as part of the payload; verification is deferred to the next hICN node which should send the packet backwards in case of verification failure to fix the incorrect path update.
- \* multipath is not supported in the pure-IP part of the network (which is the expected behaviour)

Identified concerns might be about the unexpected use of such protocols, the lack of available implementation for NDProxy, and security aspect related to redirect messages. The latter shares the fate of source routing, which has long been advocated against, and has recently gained popularity within the SPRING context. A proper security scheme is certainly the right way to address this problem, and we believe the set of benefits that we have listed are worth reconsidering such aspects.

### 7.2. Security considerations

As indicated in previous sections, signalization messages transmitted across trust boundaries must be secured. The choice of the solution will intimately depend on the selected protocols.

The use of ICMP packet might allow reusing existing security schemes such as AH headers [RFC4302], or SEND [RFC3971] (and its proxy extensions [RFC6496], [RFC5909]).

Alternatively, [SEC] has reviewed standard approaches from the literature and proposes a fast, lightweight and distributed approach that can be applied to MAP-Me and fits its design principles.

### 7.3. Discussion

Both consumer and producer mobility support multiple paths, however the support of mobility for a multihomed producer, is left for future updates of the present document.

Similarly, the proposed producer mobility solution is appropriate for the management of micro-mobility; its extension to multiple domains is out of scope.

## 8. IANA Considerations

This memo includes no request to IANA.

## 9. References

### 9.1. Normative References

- [RFC2275] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, DOI 10.17487/RFC2275, January 1998, <<https://www.rfc-editor.org/info/rfc2275>>.
- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<https://www.rfc-editor.org/info/rfc3971>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, DOI 10.17487/RFC4423, May 2006, <<https://www.rfc-editor.org/info/rfc4423>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<https://www.rfc-editor.org/info/rfc5533>>.
- [RFC5909] Combes, J-M., Krishnan, S., and G. Daley, "Securing Neighbor Discovery Proxy: Problem Statement", RFC 5909, DOI 10.17487/RFC5909, July 2010, <<https://www.rfc-editor.org/info/rfc5909>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised", RFC 5944, DOI 10.17487/RFC5944, November 2010, <<https://www.rfc-editor.org/info/rfc5944>>.
- [RFC6115] Li, T., Ed., "Recommendation for a Routing Architecture", RFC 6115, DOI 10.17487/RFC6115, February 2011, <<https://www.rfc-editor.org/info/rfc6115>>.



- [RFC6301] Zhu, Z., Wakikawa, R., and L. Zhang, "A Survey of Mobility Support in the Internet", RFC 6301, DOI 10.17487/RFC6301, July 2011, <<https://www.rfc-editor.org/info/rfc6301>>.
- [RFC6496] Krishnan, S., Laganier, J., Bonola, M., and A. Garcia-Martinez, "Secure Proxy ND Support for SEcure Neighbor Discovery (SEND)", RFC 6496, DOI 10.17487/RFC6496, February 2012, <<https://www.rfc-editor.org/info/rfc6496>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<https://www.rfc-editor.org/info/rfc6740>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<https://www.rfc-editor.org/info/rfc6830>>.
- [RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and S. Eum, "Information-Centric Networking: Baseline Scenarios", RFC 7476, DOI 10.17487/RFC7476, March 2015, <<https://www.rfc-editor.org/info/rfc7476>>.

## 9.2. Informative References

- [DATAPLANE]  
J, ., A, ., A, ., B, ., M, ., and . S, "Ensuring connectivity via data plane mechanisms.", 2013.
- [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", Work in Progress, Internet-Draft, draft-auge-dmm-hicn-mobility-deployment-options-03, 6 January 2020, <<http://www.ietf.org/internet-drafts/draft-auge-dmm-hicn-mobility-deployment-options-03.txt>>.
- [I-D.herbert-intarea-ila]  
Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", Work in Progress, Internet-Draft, draft-herbert-intarea-ila-01, 5 March 2018, <<http://www.ietf.org/internet-drafts/draft-herbert-intarea-ila-01.txt>>.

[I-D.irtf-icnrg-mapme]

Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "MAP-Me : Managing Anchorless Mobility in Content Centric Networking", Work in Progress, Internet-Draft, draft-irtf-icnrg-mapme-05, 9 June 2020, <<http://www.ietf.org/internet-drafts/draft-irtf-icnrg-mapme-05.txt>>.

[I-D.irtf-icnrg-terminology]

Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): CCNx and NDN Terminology", Work in Progress, Internet-Draft, draft-irtf-icnrg-terminology-08, 17 January 2020, <<http://www.ietf.org/internet-drafts/draft-irtf-icnrg-terminology-08.txt>>.

[I-D.muscariello-intarea-hicn]

Muscariello, L., Carofiglio, G., Auge, J., Papalini, M., and M. Sardara, "Hybrid Information-Centric Networking", Work in Progress, Internet-Draft, draft-muscariello-intarea-hicn-04, 20 May 2020, <<http://www.ietf.org/internet-drafts/draft-muscariello-intarea-hicn-04.txt>>.

[I-D.vonhugo-5gangip-ip-issues]

Hugo, D. and B. Sarikaya, "Review on issues in discussion of next generation converged networks (5G) from an IP point of view", Work in Progress, Internet-Draft, draft-vonhugo-5gangip-ip-issues-03, 13 March 2017, <<http://www.ietf.org/internet-drafts/draft-vonhugo-5gangip-ip-issues-03.txt>>.

[MAPME]

Auge, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., and X. Zeng, "MAP-Me: Managing Anchor-Less Producer Mobility in Content-Centric Networks", DOI 10.1109/tnsm.2018.2796720, IEEE Transactions on Network and Service Management Vol. 15, pp. 596-610, June 2018, <<https://doi.org/10.1109/tnsm.2018.2796720>>.

[SEC]

Compagno, A., Zeng, X., Muscariello, L., Carofiglio, G., and J. Auge, "Secure producer mobility in information-centric network", DOI 10.1145/3125719.3125725, Proceedings of the 4th ACM Conference on Information-Centric Networking, September 2017, <<https://doi.org/10.1145/3125719.3125725>>.

- [SURVEY1] Zhang, Y., Afanasyev, A., Burke, J., and L. Zhang, "A survey of mobility support in Named Data Networking", DOI 10.1109/infcomw.2016.7562050, 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April 2016, <<https://doi.org/10.1109/infcomw.2016.7562050>>.
- [SURVEY2] Feng, B., Zhou, H., and Q. Xu, "Mobility support in Named Data Networking: a survey", DOI 10.1186/s13638-016-0715-0, EURASIP Journal on Wireless Communications and Networking Vol. 2016, September 2016, <<https://doi.org/10.1186/s13638-016-0715-0>>.
- [SURVEYICN] Tyson, G., Sastry, N., Rimac, I., Cuevas, R., and A. Mauthe, "A survey of mobility in information-centric networks", DOI 10.1145/2248361.2248363, Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications - NoM '12, 2012, <<https://doi.org/10.1145/2248361.2248363>>.
- [TS29.274] "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", n.d..

#### Authors' Addresses

Jordan Auge  
Cisco Systems  
11, rue Camille Desmoulins  
92130 Issy-les-Moulineaux  
France

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

Giovanna Carofiglio  
Cisco Systems  
11, rue Camille Desmoulins  
92130 Issy-les-Moulineaux  
France

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

Luca Muscariello  
Cisco Systems  
11, rue Camille Desmoulins

92130 Issy-les-Moulineaux  
France

Email: lumuscar@cisco.com

Michele Papalini  
Cisco Systems  
11, rue Camille Desmoulins  
92130 Issy-les-Moulineaux  
France

Email: micpapal@cisco.com

dmm  
Internet-Draft  
Intended status: Informational  
Expires: December 31, 2018

K. Bogineni  
Verizon  
A. Akhavain  
Huawei Canada Research Centre  
T. Herbert  
Quantonium  
D. Farinacci  
lispers.net  
A. Rodriguez-Natal  
G. Carofiglio  
J. Auge  
L. Muscariello  
P. Camarillo  
Cisco Systems, Inc.  
S. Homma  
NTT  
June 29, 2018

Optimized Mobile User Plane Solutions for 5G  
draft-bogineni-dmm-optimized-mobile-user-plane-01

Abstract

3GPP CT4 has approved a study item to study different mobility management protocols for potential replacement of GTP tunnels between UPFs (N9 Interface) in the 3GPP 5G system architecture.

This document provides an overview of 5G system architecture in the context of N9 Interface which is the scope of the 3GPP CT4 study item [CP-173160-1], [TS.23.501-3GPP], [TS.23.502-3GPP], [TS.23.503-3GPP], [TS.29.244-3GPP], [TS.29.281-3GPP], [TS.38.300-3GPP], and [TS.38.401-3GPP].

Architecture requirements for evaluation of candidate protocols are provided. Optimization of the user plane can be in different ways - packet overhead, transport integration, etc.

Several IETF protocols are considered for comparison: SRv6, LISP, ILA and several combinations of control plane and user plane protocols.

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 December 31, 2018.

#### Copyright Notice

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

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

#### Table of Contents

|                                                            |    |
|------------------------------------------------------------|----|
| 1. Introduction . . . . .                                  | 4  |
| 1.1. Scope of 3GPP Study Items . . . . .                   | 5  |
| 1.2. Relevance to IETF . . . . .                           | 6  |
| 1.3. Rationale for GTP replacement . . . . .               | 6  |
| 1.4. Usage of GTP . . . . .                                | 7  |
| 1.5. Document Structure . . . . .                          | 7  |
| 2. Conventions and terminology . . . . .                   | 7  |
| 3. Overview of 3GPP Release 15 5G Architecture . . . . .   | 8  |
| 3.1. Non-Roaming Reference Architecture . . . . .          | 8  |
| 3.2. End-to-end Protocol Stack . . . . .                   | 10 |
| 3.3. Mobility Architecture with reference to N9 . . . . .  | 11 |
| 3.3.1. User Plane Function (UPF) Functionalities . . . . . | 12 |
| 3.3.2. N9 Interface . . . . .                              | 15 |
| 3.4. Roaming Architectures . . . . .                       | 16 |
| 3.4.1. Roaming and policy management . . . . .             | 17 |
| 3.4.2. Local Break Out Model . . . . .                     | 18 |
| 3.4.3. Home Routed Model . . . . .                         | 18 |
| 3.5. Support for Multiple PDU Sessions . . . . .           | 19 |
| 3.6. Service and Session Continuity Modes . . . . .        | 21 |
| 4. Architectural requirements . . . . .                    | 22 |
| 5. Data plane architecture models for N9 . . . . .         | 23 |

|        |                                                                        |    |
|--------|------------------------------------------------------------------------|----|
| 5.1.   | Overview . . . . .                                                     | 23 |
| 5.2.   | Forwarding and mobility paradigms . . . . .                            | 23 |
| 5.3.   | SRv6 . . . . .                                                         | 25 |
| 5.3.1. | Overview . . . . .                                                     | 25 |
| 5.3.2. | SRv6 with Traffic Engineering . . . . .                                | 26 |
| 5.3.3. | Service Programming with SRv6 . . . . .                                | 27 |
| 5.3.4. | SRv6 and Entropy . . . . .                                             | 28 |
| 5.3.5. | SRv6 and transport slicing . . . . .                                   | 28 |
| 5.3.6. | SRv6 and Alternative Approaches to Advanced Mobility Support . . . . . | 28 |
| 5.3.7. | Areas of Concerns . . . . .                                            | 29 |
| 5.4.   | LISP . . . . .                                                         | 30 |
| 5.4.1. | Overview . . . . .                                                     | 30 |
| 5.4.2. | LISP Encapsulation . . . . .                                           | 30 |
| 5.4.3. | LISP Mapping Systems . . . . .                                         | 30 |
| 5.4.4. | LISP Mobility Features . . . . .                                       | 31 |
| 5.4.5. | ILSR . . . . .                                                         | 31 |
| 5.5.   | ILA . . . . .                                                          | 31 |
| 5.5.1. | Overview . . . . .                                                     | 32 |
| 5.5.2. | Protocol Layering . . . . .                                            | 33 |
| 5.5.3. | Control plane . . . . .                                                | 33 |
| 5.5.4. | IP addressing . . . . .                                                | 34 |
| 5.5.5. | Traffic engineering . . . . .                                          | 36 |
| 5.5.6. | Locator Chaining with ILA . . . . .                                    | 36 |
| 5.5.7. | Security considerations . . . . .                                      | 36 |
| 5.6.   | Hybrid ICN (hICN) . . . . .                                            | 37 |
| 5.6.1. | Overview . . . . .                                                     | 37 |
| 5.6.2. | Consumer and Producer mobility . . . . .                               | 37 |
| 5.6.3. | Anchorless mobility support . . . . .                                  | 38 |
| 5.6.4. | Benefits . . . . .                                                     | 38 |
| 5.6.5. | Deployment considerations . . . . .                                    | 39 |
| 5.6.6. | hICN with SRv6 . . . . .                                               | 40 |
| 5.6.7. | Summary . . . . .                                                      | 41 |
| 6.     | Integration into the 5G framework . . . . .                            | 41 |
| 6.1.   | Locator based - SRv6 . . . . .                                         | 41 |
| 6.1.1. | Insertion in N9 interface . . . . .                                    | 41 |
| 6.1.2. | Control Plane considerations . . . . .                                 | 42 |
| 6.1.3. | Extensions to N3/F1-U/Xn-U interface . . . . .                         | 43 |
| 6.1.4. | Coexistence with GTP-based architecture . . . . .                      | 43 |
| 6.2.   | ID-LOC split . . . . .                                                 | 44 |
| 6.2.1. | Insertion in N9 interface . . . . .                                    | 44 |
| 6.2.2. | LISP control plane . . . . .                                           | 46 |
| 6.2.3. | ILA control plane . . . . .                                            | 47 |
| 6.2.4. | Extensions to N3/F1-U/Xn-U interface . . . . .                         | 47 |
| 6.2.5. | Coexistence with GTP-based architecture . . . . .                      | 48 |
| 6.3.   | ID-based - hICN . . . . .                                              | 50 |
| 6.3.1. | Insertion in N9 interface . . . . .                                    | 50 |
| 6.3.2. | Control plane considerations . . . . .                                 | 51 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 6.3.3. Extensions to N3/F1-U/Xn-U interface . . . . .              | 52 |
| 6.3.4. Coexistence with GTP-based architecture . . . . .           | 52 |
| 6.4. Coexistence of multiple protocols in network slices . . . . . | 53 |
| 6.5. Interoperability/Roaming considerations . . . . .             | 54 |
| 7. Summary . . . . .                                               | 55 |
| 8. Formal Syntax . . . . .                                         | 55 |
| 9. Security Consideration . . . . .                                | 56 |
| 10. IANA Considerations . . . . .                                  | 56 |
| 11. Acknowledgement . . . . .                                      | 56 |
| 12. References . . . . .                                           | 56 |
| 12.1. Normative References . . . . .                               | 56 |
| 12.2. Informative References . . . . .                             | 58 |
| Authors' Addresses . . . . .                                       | 63 |

## 1. Introduction

Release 15 of the 3GPP specifications provide the 5G System Architecture in [TS.23.501-3GPP], [TS.23.502-3GPP], and [TS.23.503-3GPP]. They come with significant changes to the radio and core architectures with respect to previous generations, with the objective of enabling new use case requirements expected from 5G networks. The user plane is however still based on GTP-U, and tunnelling user-traffic to anchor points in the core network. User, data and forwarding plane are used with the same meaning in this context.

3GPP CT4 is in charge of specifying the user plane interface named N9, and has approved a study item [CP-173160-1] to study possible candidates for user plane protocol for the 5GC in Release 16.

This document comprehensively describes the various user plane protocols and how they can be used in the 3GPP 5G architecture. Specifically Segment Routing v6 (SRv6), Locator Identifier Separation Protocol (LISP), Identifier Locator Addressing (ILA) and Hybrid Information-Centric Networking (hICN) are introduced and their use as replacement of GTP for N9 is further described.

Analysis work for clarifying the specifications of GTP-U as the current mobile user plane protocol and the architectural requirements of the 5G system is provided in [I-D.hmm-dmm-5g-uplane-analysis]. That provides observations of GTP-U, the architectural requirements for UP protocol, and some evaluation criteria based on the requirements.

Optimization of the user plane can be in one more more of the following:

- o reduction/elimination of encapsulation;



- o use of native routing mechanisms;
- o efficient forwarding during, and in between mobility events;
- o support of anchor-less mobility management and offloading of local traffic;
- o reduction of session state and signaling associated with mobility management;
- o convergence towards a flatter architecture, consistent with other mobility proposals.

#### 1.1. Scope of 3GPP Study Items

3GPP CT4 WG has approved a Release 16 study item [CP-173160-1] to study user-plane protocol for N9 in 5GC architecture as specified in [TS.23.501-3GPP] and [TS.23.502-3GPP]. This provides an opportunity to investigate potential limits of the existing user plane solution and potential benefits of alternative user plane solutions.

The following is extracted from the CT4 study item [CP-173160-1].

The expected work in CT4 will include:

- o Identify the possible candidate protocols for user-plane including existing protocol;
- o Define a list of evaluation criteria based on Rel-16 stage 2 requirements to evaluate the candidate protocols;
- o Evaluate the candidate solutions against the list of requirements and the potential benefits against the existing user plane solution in 5GS.

CT4 will coordinate with RAN3 for selecting the user plane protocols for N3 and F1-U interfaces in RAN. CT4 will also coordinate with CT3 Working Group for potential impacts to N6 interface and with SA2 for potential impacts on stage 2 specifications.

Coordination will also be required with CT3 for potential impacts on N6, and with SA2 if the work has possible impacts on the stage 2 specifications.

Extracted from [SP-180231-1], the work in SA2 Study item will study the feasibility of extending the service concept from 5GC control plane to the user plane function(s). Impact to User plane traffic processing is not expected in this study.

## 1.2. Relevance to IETF

IETF has some protocols for potential consideration as candidates. These protocols have the potential to simplify the architecture through reduction/elimination of encapsulation; use of native routing mechanisms; support of anchor-less mobility management; reduction of session state and reduction of signaling associated with mobility management.

This document provides an overview of the various protocols and how they can be used in the 3GPP 5G architecture. Details of the protocols will be provided as references in the respective sections, then described in the context of the 3GPP 5G architecture. ILNP is an end-to-end protocol and is not included in this document. The scenario of replacing GTP on N9 as the focus of CT4 study is discussed for each protocol. Additional scenarios are related to N3/F1-U; integration of mobility with transport; support for different mobility protocols on different slices of the 5G system, etc.

## 1.3. Rationale for GTP replacement

Although being different in terms of architecture or implementations, common objectives emerge from the different proposals and their positioning with respect to the GTP-U tunnel-based architecture. We succinctly discuss those aspects here, that will be detailed in the sections dedicated to each protocol, clarifying some terminology at the same occasion.

\_Simplification\_ : simplify the management of networks, flat and converge architecture with other mobility proposals.

\_Efficiency\_ : performance of the proposal for both packet forwarding, and handling of traffic during mobility events.

\_Overhead\_ : remove encapsulation overhead due to tunneling.

\_Data plane anchors\_ : remove anchoring of all communications in a central core location, and opt for distributed/decentralized/full removal of anchors.

\_Offloading of local communications\_ : a direct consequence on the distribution/removal of user plane anchors is the ability to offload local traffic from the core.

\_Control plane anchors\_ : remove dependency on additional control plane anchors, and interoperability with the SMF.

\_Transport\_ : Relieve transport and application layers from the impact of mobility and related management protocols.

#### 1.4. Usage of GTP

The main focus of the study is on the N9 interfaces that interconnect UPFs and could span over the mobile backhaul. However, GTP is used at multiple interfaces beyond N9.

N3 and N9 interfaces are tightly coupled and Section 6 discusses the possibility to extend the deployment of new user planes to N3. The impact on N3, F1-U, and Xn-U interfaces is still TBD.

#### 1.5. Document Structure

Section 3 provides a high level overview of the 5G system architecture and the relevant scenarios like roaming, support for multiple PDU sessions, etc. Section 4 provides a list of architectural requirements that candidate solutions should address are provided. Section 5 provides an overview of the various protocols. Section 6 discusses how various approaches can be integrated into the 5G framework. A summary is provided in Section 7.

## 2. Conventions and terminology

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

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 RFC 2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

In this document, the characters ">>" preceding an indented line(s) indicates a statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the portions of this RFC covered by these keywords.

#### Acronyms

\_AF\_: Application Function

\_AUSF\_: Authentication Server Function

\_AMF\_: Access and Mobility Management Function

\_DN\_: Data Network, e.g. operator services, Internet access or 3rd party services

\_NEF\_: Network Exposure Function

\_NRF\_: Network Repository Function

\_NSSF\_: Network Slice Selection Function

\_PCF\_: Policy Control Function

\_RAN\_: (Radio) Access Network

\_SMF\_: Session Management Function

\_UDM\_: Unified Data Management

\_UDR\_: Unified Data Repository

\_UE\_: User Equipment

\_UPF\_: User Plane Function

### 3. Overview of 3GPP Release 15 5G Architecture

This section briefly describes the 5G system architecture as specified in [TS.23.501-3GPP]. The key relevant features for session management and mobility management are:

- o Separate the User Plane (UP) functions from the Control Plane (CP) functions, allowing independent scalability, evolution and flexible deployments e.g. centralized location or distributed (remote) location.
- o Support concurrent access to local and centralized services. To support low latency services and access to local data networks, UP functions can be deployed close to the Access Network.
- o Support roaming with both Home routed traffic as well as Local breakout traffic in the visited PLMN.

#### 3.1. Non-Roaming Reference Architecture

This section briefly describes the 5G system architecture as specified in 3GPP TS 23.501, and represented in Figure 1 and Figure 2.

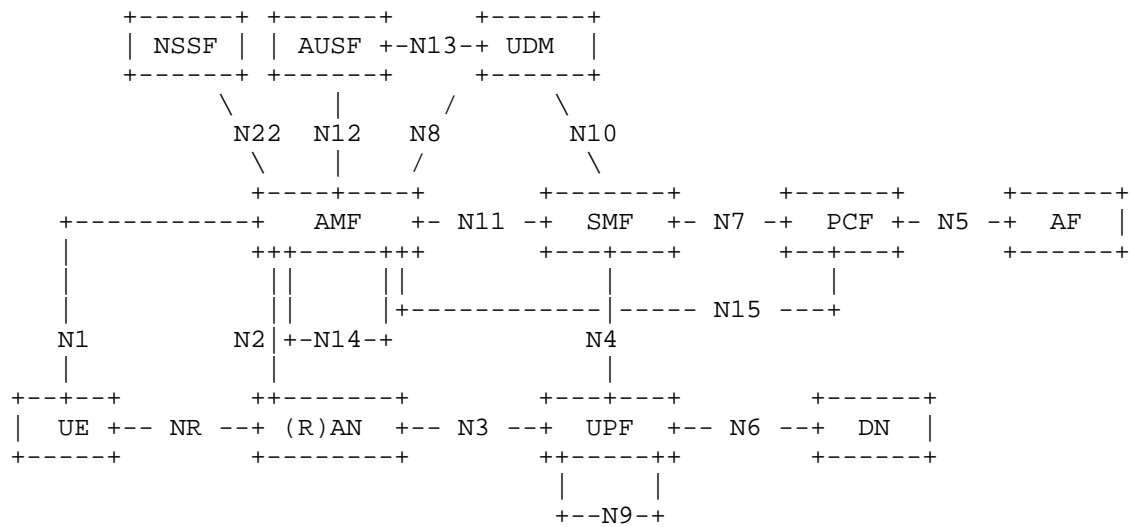


Figure 1: 5G System Architecture in Reference Point Representation

A short description of the network functions is provided below.  
Details are in [TS.23.501-3GPP].

Access and Mobility Management Function (AMF) interfaces with the Radio access network and provides management of registration/connection/reachability/mobility, access authentication and authorization, etc.

Session Management function (SMF) handles session management, UE IP address allocation & management, DHCP, ARP proxying, selection and control of UP function, traffic steering, interface to PCF, charging data collection, roaming, etc.

User Plane Function (UPF) is the anchor point mobility, packet routing/forwarding/marking, packet inspection, policy rule enforcement, lawful intercept, QoS handling, etc.

Policy Control Function (PCF) provides policy rules to Control Plane function(s) to enforce them.

Network Exposure Function (NEF) supports exposure of capabilities and events between network functions, to 3rd party, Application Functions, Edge Computing, etc.

Network Repository Function (NRF) supports service discovery function.

Unified Data Management (UDM) supports access authorization, subscription management, etc.

Authentication Server Function (AUSF) supports authentication for 3GPP access and untrusted non-3GPP access.

Network Slice Selection Function (NSSF) selects the set of Network Slice instances serving the UE, determines the allowed slices, etc.

Application Function (AF)

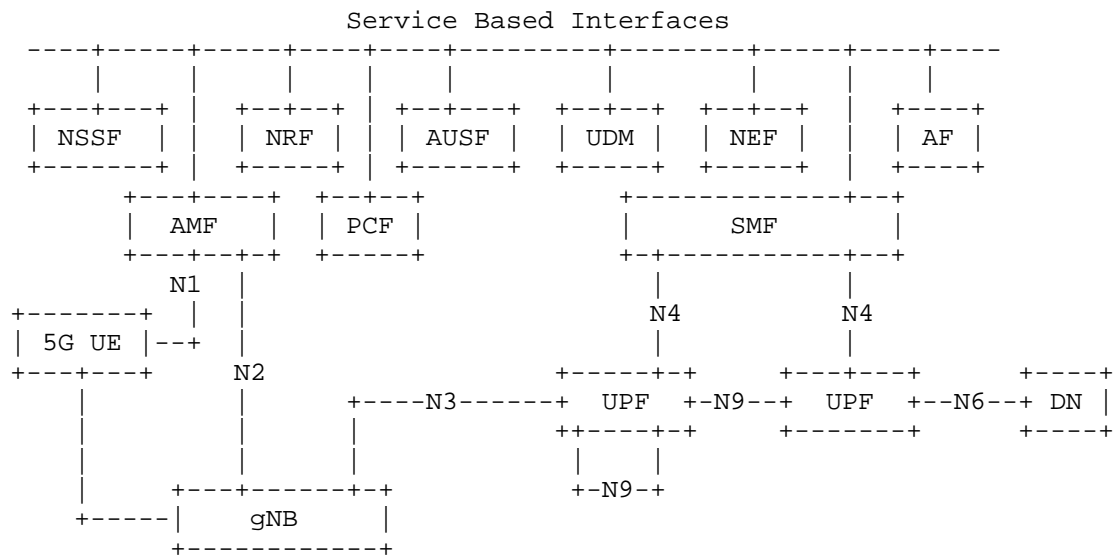
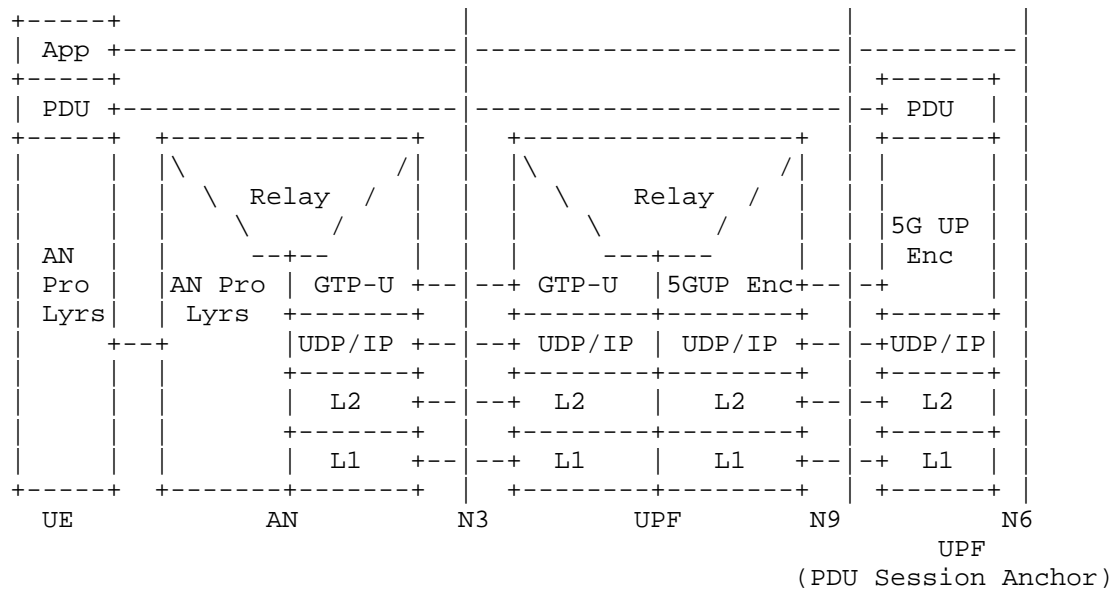


Figure 2: 5G Service Based Architecture

### 3.2. End-to-end Protocol Stack

The protocol stack for the User Plane transport for a PDU session is depicted below in Figure 3.



#### Legend:

- o PDU layer: This layer corresponds to the PDU carried between the UE and the DN over the PDU session. When the PDU session Type is IPV6, it corresponds to IPv6 packets; When the PDU session Type is Ethernet, it corresponds to Ethernet frames; etc.
- o GPRS Tunnelling Protocol for the user plane (GTP U): This protocol supports multiplexing traffic of different PDU sessions (possibly corresponding to different PDU session Types) by tunnelling user data over N3 (i.e. between the AN node and the UPF) in the backbone network. GTP shall encapsulate all end user PDUs. It provides encapsulation on a per PDU session level. This layer carries also the marking associated with a QoS Flow.
- o 5G Encapsulation: This layer supports multiplexing traffic of different PDU sessions (possibly corresponding to different PDU session Types) over N9 (i.e. between different UPF of the 5GC). It provides encapsulation on a per PDU session level. This layer carries also the marking associated with a QoS Flow.

Figure 3: Non-roaming 5G SA for multiple PDU Sessions

### 3.3. Mobility Architecture with reference to N9

This document focuses on the N9 interface which represents the user plane between UPFs in 5G architecture. Figure 4 shows the relevant functions and interfaces.

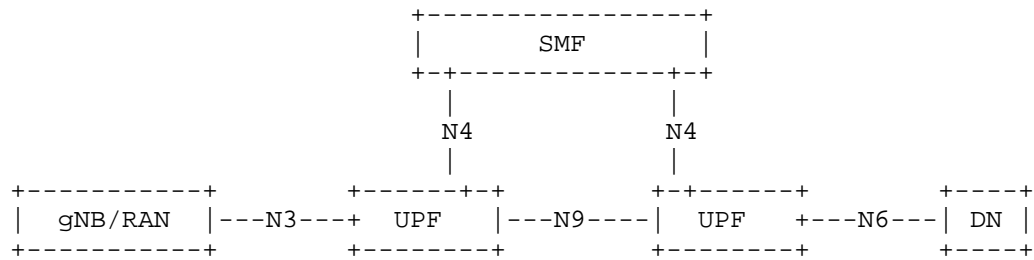


Figure 4: N3, N4, N9, and N6 interfaces in 5G Service Based Architecture

### 3.3.1. User Plane Function (UPF) Functionalities

The User plane function (UPF) is the function relevant to this evaluation and the N9 interface between two UPFs.

The User Plane Function (UPF) handles the user plane path of PDU sessions. The UPF transmits the PDUs of the PDU session in a single tunnel between 5GC and (R)AN. The UPF includes the following functionality. Some or all of the UPF functionalities may be supported in a single instance of a UPF. Not all of the UPF functionalities are required to be supported in an instance of user plane function of a Network Slice.

The following provides a brief list of main UPF functionalities. Please refer to section 6.2.3 of [TS.23.501-3GPP] for detailed description of UPF and its functionalities.

- o Anchor point for Intra-/Inter-RAT mobility (when applicable)"
- o Sending and forwarding of one or more end marker to the source NG-RAN node
- o External PDU Session point of interconnect to Data Network.
- o PDU session type: IPv4, IPv6, Ethernet, Unstructured (type of PDU totally transparent to 5GS)
- o Activation and release of the UP connection of an PDU session, upon UE transition between the CM-IDLE and CM-CONNECTED states(i.e. activation and release of N3 tunnelling towards the access network)
- o Data forwarding between the SMF and the UE or DN (e.g. IP address allocation or DN authorization during the establishment of a PDU session)



- o Packet routing and forwarding (e.g. support of Uplink classifier to route traffic flows to an instance of a data network, support of Branching point to support IPv6 multi-homed PDU session>
- o Branching Point to support routing of traffic flows of an IPv6 multi-homed PDU session to a data network, based on the source Prefix of the PDU
- o User Plane part of policy rule enforcement (e.g. Gating, Redirection, Traffic steering)
- o Uplink Classifier enforcement to support routing traffic flows to a data network, e.g. based on the destination IP address/Prefix of the UL PDU
- o Lawful intercept (UP collection)
- o Traffic usage reporting
- o QoS handling for user plane including:
  - \* packet filtering, gating, UL/DL rate enforcement, UL/DL Session-AMBR enforcement (with the Session-AMBR computed by the UPF over the Averaging window provisioned over N4, see subclause 5.7.3 of 3GPP [TS.23.501-3GPP]), UL/DL Guaranteed Flow Bit Rate (GFBR) enforcement, UL/DL Maximum Flow Bit Rate (MFBR) enforcement, etc
  - \* marking packets with the QoS Flow ID (QFI) in an encapsulation header on N3 (the QoS flow is the finest granularity of QoS differentiation in the PDU session)
  - \* enabling/disabling reflective QoS activation via the User Plane, i.e. marking DL packets with the Reflective QoS Indication (RQI) in the encapsulation header on N3, for DL packets matching a QoS Rule that contains an indication to activate reflective QoS
- o Uplink Traffic verification (SDF to QoS flow mapping, i.e. checking that QFIs in the UL PDUs are aligned with the QoS Rules provided to the UE or implicitly derived by the UE e.g. when using reflective QoS)
- o Transport level packet marking in the uplink and downlink, e.g. based on 5QI and ARP of the associated QoS flow.
- o Downlink packet buffering and downlink data notification triggering: This includes the support and handling of the ARP

priority of QoS Flows over the N4 interface, to support priority mechanism:

- \* "For a UE that is not configured for priority treatment, upon receiving the "N7 PDU-CAN Session Modification" message from the PCF with an ARP priority level that is entitled for priority use, the SMF sends an "N4 Session Modification Request" to update the ARP for the Signalling QoS Flows, and sends an "N11 SM Request with PDU Session Modification Command" message to the AMF, as specified in clause 4.3.3.2 of [TS.23.502-3GPP].
- \* "If an IP packet arrives at the UPF for a UE that is CM-IDLE over a QoS Flow which has an ARP priority level value that is entitled for priority use, delivery of priority indication during the Paging procedure is provided by inclusion of the ARP in the N4 interface "Downlink Data Notification" message, as specified in clause 4.2.3.4 of [TS.23.502-3GPP]."
- o ARP proxying as specified in [RFC1027] and / or IPv6 Neighbour Solicitation Proxying as specified in [RFC4861] functionality for the Ethernet PDUs. The UPF responds to the ARP and / or the IPv6 Neighbour Solicitation Request by providing the MAC address corresponding to the IP address sent in the request.
- o Packet inspection (e.g. Application detection based on service data flow template and the optional PFDs received from the SMF in addition)
- o Traffic detection capabilities.
  - \* For IP PDU session type, the UPF traffic detection capabilities may detect traffic using traffic pattern based on at least any combination of:
    - + PDU session
    - + QFI
    - + IP Packet Filter Set. Please refer to section 5.7.6.2 of 3GPP TS 23.501 for further details.
  - \* For Ethernet PDU session type, the SMF may control UPF traffic detection capabilities based on at least any combination of:
    - + PDU session
    - + QFI

- + Ethernet Packet Filter Set. Please refer to section 5.7.6.3 of 3GPP TS 23.501 for further details.

- o Network slicing Requirements for different MM mechanisms on different slice. The selection mechanism for SMF to select UPF based on the selected network slice instance, DNN and other information e.g. UE subscription and local operator policies.

### 3.3.2. N9 Interface

The details of N9 interface are extracted from [TR.29.891-3GPP].

The following information is sent in an encapsulation header over the N3 interface. N9 needs to support that.

- o QFI (QoS Flow Identifier), see subclause 5.7.1 of [TS.23.501-3GPP].
- o RQI (Reflective QoS Identifier), see subclause 5.7.5.4.2 of [TS.23.501-3GPP].
- o Support of RAN initiated QoS Flow mobility, when using Dual connectivity, also requires the QFI to be sent within End Marker packets. See subclause 5.11.1 of [TS.23.501-3GPP] and subclause 4.14.1 of [TS.23.502-3GPP] respectively.

GTPv1-U as defined in [TS.29.281-3GPP] is used over the N3 and N9 interfaces in Release 15. Release 15 is still work-in-progress and RAN3 will specify the contents of the 5GS Container. It is to be decided whether CT4 needs to specify new GTP-U extension header(s) in [TS.29.281-3GPP] for the 5GS Container.

A GTP-U tunnel is used per PDU session to encapsulate T-PDUs and GTP-U signaling messages (e.g. End Marker, Echo Request, Error Indication) between GTP-U peers.

A 5GS Container is defined as a new single GTP-U Extension Header over the N3 and N9 interfaces and the elements are added to this container as they appear with the forthcoming features and releases. This approach would allow to design the 5GS information elements independently from the tunneling protocol used within the 5GS, i.e. it would achieve the separation of the Transport Network Layer (TNL) and Radio Network Layer (RNL) as required in 3GPP TR 38.801 subclause 7.3.2. This would allow to not impact the RNL if in a future release a new transport network layer (TNL) other than GTP-U/UDP/IP (e.g. GRE/IP) was decided to be supported.

### 3.4. Roaming Architectures

3GPP specifies two roaming models in [TS.23.501-3GPP], namely the Local Break Out (LBO) and the Home Routed (HR) model.

- o Local Break Out Model: This model enables traffic to be offloaded locally in the visited network.
- o Home Routed Model: In this model, the traffic is always routed to the home network.

A given UE can have multiple simultaneous PDU sessions with different roaming models. In these scenarios, the HPLMN uses subscription data per Data Network Name(DNN) and per Single Network Slice Selection Assistance Information(S-NSSAI) to determine PDU sessions's roaming model.

They are represented in Figure 5 and Figure 6 to the extent relevant to N9.

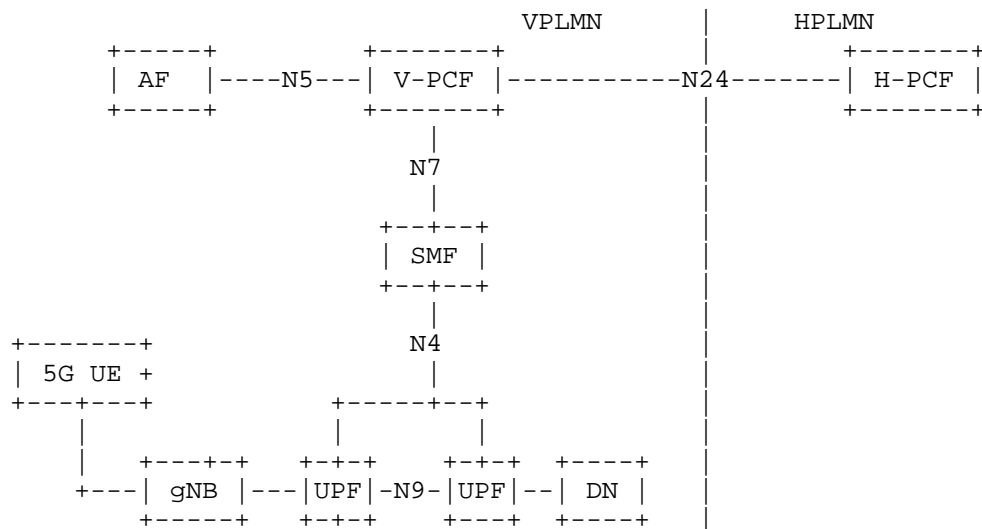


Figure 5: Roaming 5G System Architecture - Local Breakout Scenario

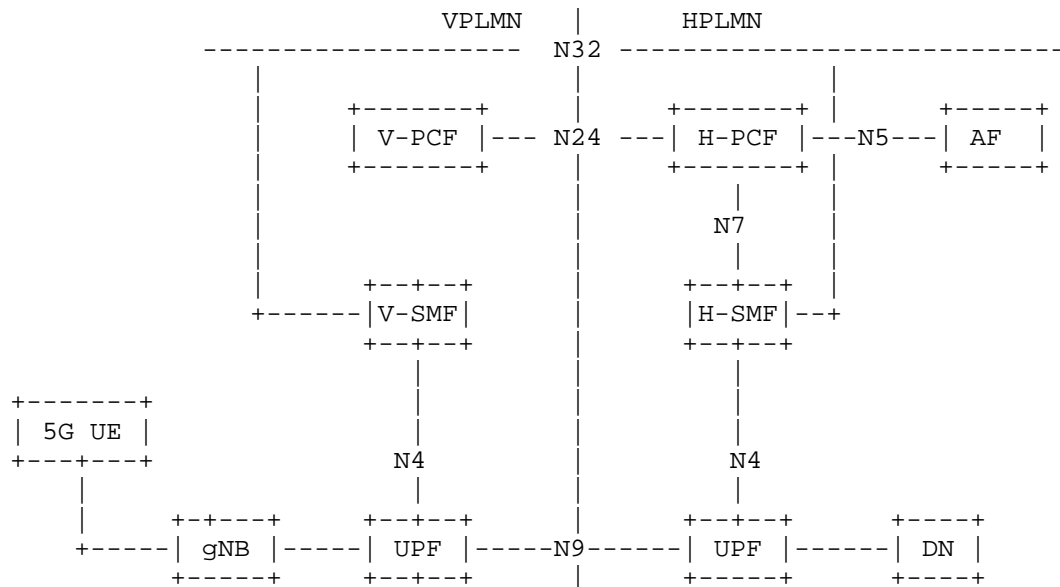


Figure 6: Roaming 5G System Architecture- Home Routed Scenario

#### 3.4.1.1. Roaming and policy management

In general, the Policy Control Functions (PCF)s in Home PLMN (HPLMN) and Visited PLMN (VPLMN) interact with their respective SMFs as well as one another to support roaming.

The interface between the PCF and SMF allows the PCF to have dynamic control over policy and charging decisions at SMF. More specifically, the interface

- o Enables the SMF to establish PDU session,
- o Allows policy and charging control decisions to be requested from the SMF to the PCF direction or to be provisioned from the opposite direction.
- o Provides a mean for SMF to deliver network events and PDU session parameters to PCF.
- o Provides for PDU session termination at either PCF or SMF end.

The N24 interface between V-PCF and H-PCF provides a communication path between these two entities. The interface enables H-PCF to provision access and mobility management related policies to V-PCF, and allows V-PCF to send Policy Association Establishment and

Termination requests to H-PCF during UE registration and deregistration procedures.

#### 3.4.2. Local Break Out Model

In the LBO model, visited operator routes user traffic locally through UPFs that are local to the visited operator. In this model, the SMF and all UPF(s) involved by the PDU Session are located and are under the control of the VPLMN.

In this model, the V-PCF generates Policy and Charging Control (PCC) rules from the local configuration data that are based on the roaming agreement with the HPLMN. The V-PCF might also use information from Application Function(AF) to generate PCC rules for VPLMN delivered services. Here, the H-PCF uses the N24 interface to deliver UE access selection, and PDU session selection policies to the V-PCF. The V-PCF can either provide access and mobility policy information on its own, or alternatively obtain the required information from the H-PCF via the N24 interface.

#### 3.4.3. Home Routed Model

In the HR model, user traffic is routed to the UPF in HPLMN via the UPF in the visited network. In this scenario, the SMF in HPLMN (H-SMF) selects the UPF(s) in the HPLMN and the SMF in VPLMN(V-SMF) selects the UPF(s) in the VPLMN. In this model, the UE obtains services from its home network. Here, the UPF acting as PGW resides in home network, and can directly communicate with policy and billing system.

In the HR roaming model:

- o The NAS SM terminates at the V-SMF.
- o The V-SMF forwards SM related information to the SMF in the HPLMN.
- o The V-SMF sends UE's Subscription Permanent Identifier(SUPI) to the H-SMF during the PDU session establishment procedure.
- o The V-SMF sends the PDU Session Establishment Request message to the H-SMF along with the S-NSSAI with the value from the HPLMN.
- o The H-SMF obtains subscription data directly from the Unified Data Management(UDM) and is responsible for checking the UE request with regard to the user subscription, and may reject the request in case of mismatch.

- o The H-SMF may send QoS requirements associated with a PDU Session to the V-SMF. This may happen at PDU Session establishment and after the PDU Session is established. The interface between H-SMF and V-SMF is also able to carry (N9) User Plane forwarding information exchanged between H-SMF and V-SMF. The V-SMF may check QoS requests from the H-SMF with respect to roaming agreements. At the user plane, the encapsulation header carries QoS flow ID (QFI) over N3, and N9 without any changes to the end to end packet header.
- o The AMF selects a V-SMF and a H-SMF, and provides the identifier of the selected H-SMF to the selected V-SMF.
- o The H-SMF performs IP address management procedure based on the selected PDU session type.

### 3.5. Support for Multiple PDU Sessions

Figure 7 depicts the non-roaming architecture for UEs concurrently accessing two (e.g. local and central) data networks using multiple PDU Sessions, using the reference point representation. This figure shows the architecture for multiple PDU Sessions where two SMFs are selected for the two different PDU Sessions. However, each SMF may also have the capability to control both a local and a central UPF within a PDU Session.

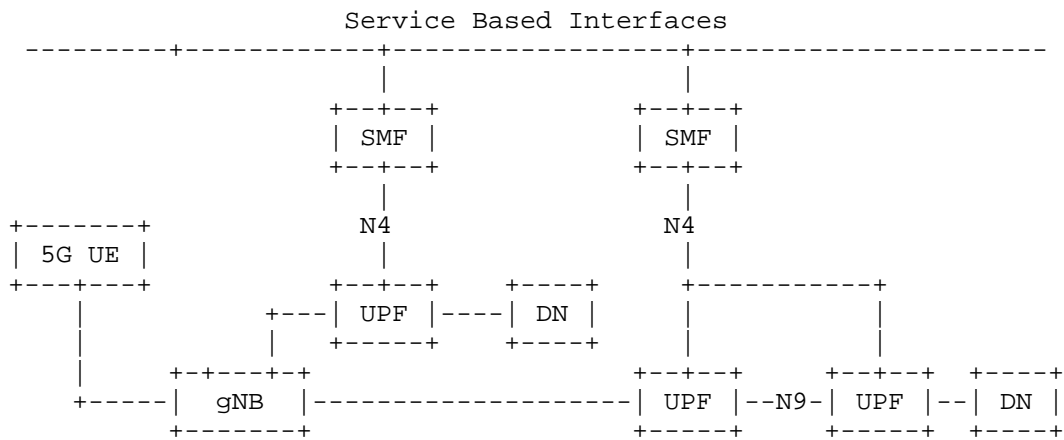


Figure 7: Non-roaming 5G System Architecture for multiple PDU Sessions Service Based Interface

Figure 8 depicts the non-roaming architecture in case concurrent access to two (e.g. local and central) data networks is provided within a single PDU Session.

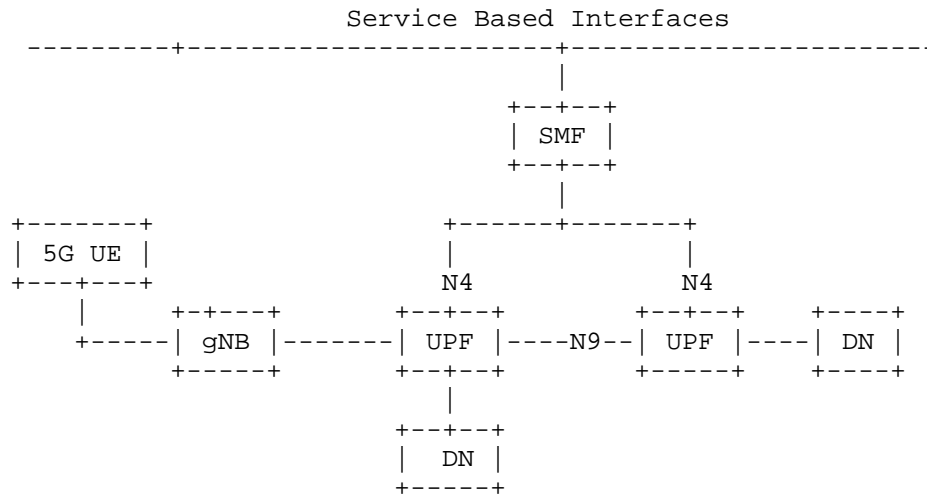


Figure 8: Non-roaming 5G System Architecture for Current Access to Two (e.g. local and central) Data Networks (single PDU Session option)

Figure 9 depicts overview of a network model where multiple UPFs are distributed geographically. Such networks have two types of UPFs: central UPF (cUPF) deployed for covering wide area, and local/distributed UPF (dUPF) deployed close to UEs' access points. UPFs are connected via N9 interfaces over transport network.



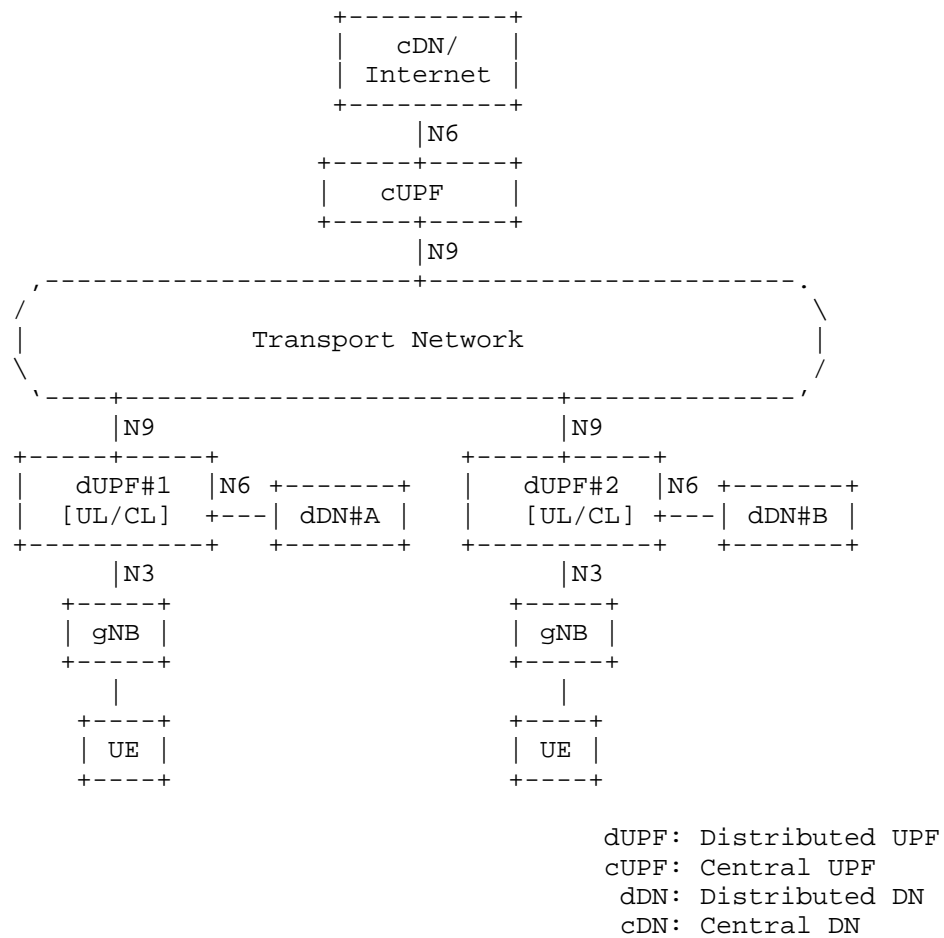


Figure 9: Overview of Network Model with Distributed UPFs

### 3.6. Service and Session Continuity Modes

The 5G System supports different session and service continuity (SSC) modes.

\_SSC mode 1\_: the network preserves the connectivity service provided to the UE.

\_SSC mode 2\_: the network may release the connectivity service delivered to the UE and release the corresponding PDU Session.

\_SSC mode 3\_: changes to the user plane can be visible to the UE, while the network ensures that the UE suffers no loss of connectivity. A connection through new PDU Session Anchor point is established before the previous connection is terminated in order to allow for better service continuity.

#### 4. Architectural requirements

[I-D.hmm-dmm-5g-uplane-analysis] provides a comprehensive summary of GTP architecture, and architectural requirements related to user plane collected from 3GPP specifications that we summarize here:

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

The 5G system defines four types of PDU session as IPv4, IPv6, Ethernet, and Unstructured, and UP protocol would be required to support to convey all of these PDUs session type.

ARCH-Req-2: Supporting IP Connectivity over N3, N6, and N9

The 5G system provides IP connectivity over N3, N6, and N9 interfaces.

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

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

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

The appropriate UPFs are selected for a PDU session based on parameters and information such as UPF's dynamic load or UE location information.

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

The number of UPF in the data path is not constrained by 3GPP specifications.

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

In the 5G system, a single tunnel/data-path includes multiple QFIs contrast to just one QoS Flow (a bearer) to one tunnel/data-path

User plane protocol needs to support fundamentally these requirements. In addition, [I-D.hmm-dmm-5g-uplane-analysis] provides evaluation aspects for user plane protocol that are mainly derived from the architectural requirements, such as Supporting PDU Session Type Variations, Nature of Data Path, Data Path Management, etc. The details are described in [I-D.hmm-dmm-5g-uplane-analysis].

For each protocol, we will attempt in the next section to discuss to what extent those architectural requirements are addressed. However, it is worth noticing that it is not mandatory that all those requirements are supported by the user plane protocol itself, as they might be realized through complementary mechanisms Section 6.6.

## 5. Data plane architecture models for N9

### 5.1. Overview

The user plane architectures considered for UPF connectivity in mobile packet core fall into two categories:

- o Interworking model:
  - \* This model uses GWs.
  - \* UPFs and 3GPP control remain unchanged.
  - \* 3GPP user plane becomes an overlay on top of new user planes
  - \* GWs convert GTP traffic to underlying user plane format.
- o Integrated model:
  - \* In this model UPFs transmit/receive packets in accordance with the new user plane format.
  - \* UPFs and 3GPP control will be modified.
  - \* 3GPP and transport user plane are collapsed into one user plane.

### 5.2. Forwarding and mobility paradigms

Based on their use of identifiers and locators, mobility approaches can be broadly categorized in the three following classes:

\*Locator-based\*

IP communication relies solely on locators (host interfaces' addresses) that are also used as node/service identifiers at network layer. Such semantic overloading of IP addresses as both identifiers and locators does not allow to disentangle locators from location-independent traffic identifiers, thus complexifying mobility management.

As a result, traffic anchors and tunnels have been introduced to handle mobility while preserving the identifier exposed to the transport layer.

#### \*Locator-ID separation\*

To overcome the limitations of purely locator-based architectures, "locator/identifier separation" (or Loc/ID split) schemes have been proposed, that use separate namespaces for so-called End-point Identifiers (EID) and Route Locators (RLOC), bound together through a mapping system. This service can be centralized, decentralized or distributed and offers control plane protocols for storage, update or retrieval of the correspondence between EIDs and RLOCs.

Loc/ID split has been originally proposed by LISP to solve the scalability challenges of Internet routing, and further adapted as a mobility management solution. This category includes most of the approaches reviewed in this document, namely ILA, ILSR and a SRv6-based solution, which all share the requirement for a mapping system.

#### \*ID-based\*

A third class of approaches exists that redefines IP communication principles (i.e. network and transport layers) around location-independent identifiers [I-D.vonhugo-5gangip-ip-issues].

Information-Centric Networking (ICN) approaches fall into such class of approaches that we refer to as purely ID-based, or in that specific case, as name-based [I-D.irtf-icnrg-terminology]. Previous work has highlighted the interest of ICN for mobility management [RFC7476].

The rest of this section details the set of reviewed approaches, namely SRv6, LISP, ILSR, ILA and hICN, as summarized in Figure 10. Each proposal consists in an overview with pointers to reference material for a more in depth description. The focus is then given to a discussion on its integration at N9 interface, as well as the benefits with respect to GTP-U. Extensions to N3 interface as well as alternative deployments preserving GTP tunnels as discussed later in this document in Section 6.

**\*Reviewed approaches\***

|                                  |            |
|----------------------------------|------------|
| _ Mobility Management            |            |
| _ Locator-based                  |            |
| _ Tunnelling                     |            |
| _ 3GPP / GTP-U                   | Sec. 4     |
| _ Packet steering                |            |
| _ SRv6 (backwards-compatible)    | Sec. 5.2.1 |
| _ Loc/ID split                   |            |
| _ Packet steering                |            |
| _ SRv6                           | Sec 5.2.2  |
| _ Encapsulation                  |            |
| _ LISP, LISP-MN, ILSR            | Sec. 5.3   |
| _ Address rewrite                |            |
| _ Network-based translation      |            |
| _ ILA                            | Sec. 5.5   |
| _ ID-based                       |            |
| _ Information-Centric Networking |            |
| _ ID-based mobility / IPv6       |            |
| _ Hybrid ICN                     | Sec. 5.6   |

Figure 10: Overview of reviewed approaches

**5.3. SRv6****5.3.1. Overview**

SRv6 [I-D.filsfils-spring-srv6-network-programming] is the IPv6 dataplane instantiation of Segment Routing [I-D.ietf-spring-segment-routing]. Segment Routing is a network architecture based on source-routing (the headend inserts the nodes that a packet must traverse for TE, NFV and VPN purposes). Thus confining flow states to the ingress nodes in the SR domain.

The SRv6 dataplane consists on leveraging the IPv6 extension headers, defined in RFC8200, to include in the IPv6 header a new "Segment Routing Header" [I-D.ietf-6man-segment-routing-header] (SRH).

SRv6 encodes segments (SIDs) as IPv6 addresses in the Segment List of its header. The IPv6 Destination Address (DA) specifies the active segment in the Segment List, while the Segments Left (SL) field of the SRH points to the next active segment in the Segment List. SRv6 routes over the shortest ECMP-aware path in the network up to the node instantiating the active segment. Once the packet has reached this node, the segment is executed. This implies running its associated function on the router, decrementing the SL value and updating the IPv6 DA to the next active segment. Notice that transit

routers neither inspect the SRH nor process it. Thus they only need to be IPv6 capable.

The main benefit of SRv6 overlays is the reduction of state in the network (there is no state in the forwarding fabric), with optimized MTU overhead, and its capability to integrate with the underlay (SLA; Traffic Engineering) and distributed NFVi. Hence there is no need NSH for NVF, RSVP for TE, or UDP for ECMP. SR also supports natively network slicing, which implies that SRv6 can offer end-to-end network slices that spans all those elements (overlay, underlay, NFV).

The versatility and adaptability of SR combined with IPv6's ample and flexible address space positions SRv6 as a viable user plane for the next generation of mobile user-plane, in particular the 3GPP N3 and N9 interfaces. Notice that SRv6 applicability does not require a new mobility control-plane. SRv6 can be combined with other control-planes such as LISP, hICN described later in this document or others such as DHT, proprietary CP, etc.

The applicability of SRv6 to mobility is described in [I-D.ietf-dmm-srv6-mobile-uplane].

SRv6 counts with three open-source implementations (Linux Kernel, FD.io VPP, P4) and several proprietary implementations (4xCisco, 1xBarefoot Networks, 1xUTStarcom) which have publicly participated in interops and all execute at linecard rate.

This section starts by summarising the use of SRv6 as a drop-in alternative for GTP-U over the N9 interface connecting different User Plane Functions (UPF). It then shows how SRv6 as a GTP-U replacement can then provide additional features such as TE, IP session aggregation, rate limiting, and distributed NFVi that are not natively available by GTP.

It must be noted that the SRv6 models discussed in this document can follow either of the interworking or the integration model mentioned earlier depending on operator's requirements.

SRv6 appears well placed as a mechanism to replace GTP-U with initially no control plane changes, but to then offer a progressive path towards many innovations in routing.

### 5.3.2. SRv6 with Traffic Engineering

SRv6 can be applied as a drop-in replacement for GTP without changes in the control-plane. This is a simple 1 to 1 replacement discussed in section 6.1. However, SRv6 offers much richer possibilities.

Traffic engineering is a native feature of SR. The SRv6 variant of SR of course supports both strict and loose models of source routing. Here, the SID list in SRH can represent a loose or strict path to UPFs. Therefore, traffic engineering can easily be supported regardless of any of the aforementioned approaches.

The main benefit of leveraging SRv6 for TE is the natural ability to create end-to-end network slices that spans both the UPFs and the underlaying transport network with TE optimization objectives (i.e. low-latency).

It must be noted that the SRH could contain multiple sets of SIDs each representing a TE path between a pair of UPFs. Alternatively, the SRH can contain a fully resolved end to end TE path that covers every intermediate node and UPF along the user plane.

SR considers segments to be instructions. Therefore each SID can represent a function that enforces a specific nodal or global packet treatment. Attributes such as jitter and delay requirement, rate limiting factors, etc. can be easily encoded in to SIDs in order to apply the desired treatment as packets traverse the network from UPF to UPF. [I-D.ietf-dmm-srv6-mobile-uplane] suggests a SID encoding mechanism for rate limiting purposes.

Please refer to the followings for further details about SR traffic engineering capabilities, the network programming concept, and some of the main SRv6 functions.

- o [I-D.ietf-spring-segment-routing]
- o [I-D.ietf-spring-segment-routing-policy]
- o [I-D.filsfils-spring-srv6-network-programming]
- o [I-D.ietf-6man-segment-routing-header]

### 5.3.3. Service Programming with SRv6

Service programming -or distributed NFVi- is another intrinsic feature of SR. Leveraging this capability, operators can steer user traffic through a set of UPFs where each UPF performs a specific service on the traffic.

Service programming is achieved through the use of SIDs in an identical manner to what was described in the previous section: the SRH is populated with a set of SIDs with each SID identifying a specific UPF in the network. Starting from the ingress SRv6 node,

packets are then forwarded through the network visiting the set of UPFs listed as SIDs in the SRH.

Please refer to [I-D.xuclad-spring-sr-service-chaining] for further detail.

#### 5.3.4. SRv6 and Entropy

Ability to provide a good level of entropy is an important aspect of user plane protocols. If included in network node's hashing, the TEID field in GTP tunnels algorithms can result in good load balancing. Therefore, any new user plane proposal should be able to deal with entropy in an efficient manner.

SRv6 natively supports entropy by using the IPv6 Flow Label. Additionally, SRv6 SIDs can easily accommodate entropy at a hop by hop level by reserving a set of bits in the SID construct itself. In this way, the hashing algorithm at different nodes distribute traffic flows based on the SID which has been copied to IPv6 DA field.

#### 5.3.5. SRv6 and transport slicing

Slicing is one of the main features in 5G [TS.23.501-3GPP]. Several Slices with different requirements can coexist on top of the common network infrastructure. Diverse flows belonging to different 5G slices can be completely disjoint or can share different parts of the network infrastructure. SRv6 has native support for network slicing spanning the UPFs, underlay -transport network- and NFVi. Also, SRv6 creates network slices without per-flow state in the fabric, hence simplifying the slicing paradigm.

Please refer to [I-D.ietf-spring-segment-routing-policy] for further detail.

#### 5.3.6. SRv6 and Alternative Approaches to Advanced Mobility Support

SRv6 flexibility enables it to support different methods of providing mobility in the network. ID-LOC for mobility support is one such option.

The previous sections discussed how SRv6 could be employed as a replacement for GTP tunnels while leaving the existing control plane intact. This section describes the use of SRv6 as a vehicle to implement Locator/ID Separation model for UPF user plane connectivity. It must be noted that SRv6 implementation of the ID-LOC architecture can employ a variety of different control planes including LISP, , different variety of DHT, proprietary, etc.



#### 5.3.6.1. UPF connectivity via SRv6 with Loc-ID separation (Interworking model)

SRv6 can easily implement ID-LOC Separation model for UPF connectivity. The SIDs are once again the main vehicle here. In this model, UPFs are considered to be the IDs while the nodes where the UPFs attach to take on the role of the Locators.

In this approach, UPFs connect to SRv6 capable Locators. UPFs use IPv4/IPv6 transport either in conjunction with GTP or without any GTP tunnel and send the packets to their associated Locator at the near end (Ingress SRv6 Locator).

It must be noted that use of GTP at UPFs allows us to leave the 3GPP control plane intact and hence provides a smooth migration path toward SRv6 with ID-Locator model.

#### 5.3.6.2. SRv6 Capable UPFs and RLOCs (Integration model)

In this model, the head-end UPF (Ingress UPF) is the ingress node and the entity that constructs the SRH in the SRv6 domain.

The 3GPP control plane is responsible for distributing UPF's endpoint information. But it requires some modifications to be able to convey endpoint information to interested parties.

The SMF can provide a fully resolved SID list by communicating with a centralised or distributed ID-LOC mapping system containing all the relevant data regarding the UPF-Locator relationship.

#### 5.3.6.3. Advanced Features in ID-Locator Architecture

SRv6's native features such as Traffic Engineering, QoS support, UPF Chaining, network slicing, etc. can be easily added to ID-Locator support. As it was noted earlier, these features are not readily available by GTP.

#### 5.3.7. Areas of Concerns

Support for IPv6 is a precondition for SRv6. Although SRv6 can support hybrid IPv4/IPv6 mobile user plane through an interworking node, support of UPFs with IPv4 address is rather complex.

Due to IPv6 128-bit address space, large SRH size can have a negative impact on MTU. Large SRH size can also exert undesirable header tax especially in the case of small payload size.

ID-LOC architecture relies on high performance mapping systems. The SRv6 support of ID-LOC as described earlier can employ different control planes. Distributed mapping systems using some form Distributed Hash Table(DHT) however, exhibit very promising results. But further investigation is needed to ensure conformance with performance metrics required by the mobile networks, specially for slice types supporting high speed mobility.

#### 5.4. LISP

##### 5.4.1. Overview

The Locator/Identifier Separation Protocol (LISP), which provides a set of functions for routers to exchange information used to map from Endpoint Identifiers (EIDs) that are not globally routable to routable Routing Locators (RLOCs). It also defines a mechanism for these LISP routers to encapsulate IP packets addressed with EIDs for transmission across a network infrastructure that uses RLOCs for routing and forwarding.

An introduction to LISP can be found in [I-D.ietf-lisp-introduction].

A complete RFC-set of specifications can be found in [RFC6830], [RFC6831], [RFC6832], [RFC6833], [RFC6836], [RFC7215], [RFC8061], [RFC8111]. They describe support and mechanisms for all combinations of inner and outer IPv4 and IPv6 packet headers for unicast and multicast packet flows that also interwork with non-LISP sites as well as two designs to realize a scalable mapping system.

A standards-track based set of drafts [I-D.ietf-lisp-rfc6830bis] [I-D.ietf-lisp-rfc6833bis] are products and work in progress of the LISP Working Group.

##### 5.4.2. LISP Encapsulation

LISP uses dynamic tunnel encapsulation as its fundamental mechanism for the data-plane. Fixed headers are used between the outer and inner IP headers which are 16 bytes in length. Details can be found in [RFC6830].

##### 5.4.3. LISP Mapping Systems

Many years of research dating back to 2007 have gone into LISP scalable mapping systems. They can be found at [LISP-WG] and [IRTF-RRG]. The two that show promise and have deployment experience are LISP-DDT [RFC8111] and LISP-ALT [RFC6836].

The control-plane API which LISP xTRs are the clients of is documented in [RFC6833]. Various mapping system and control-plane tools are available [RFC6835] [RFC8112] and are in operational use.

#### 5.4.4. LISP Mobility Features

LISP supports multi-homed shortest-path session survivable mobility. An EID can remain fixed for a node that roams while its dynamic binding changes to the RLOCs it uses when it reconnect to the new network location.

When the roaming node supports LISP, its EIDs and RLOCs are local to the node. This form of mobility is call LISP Mobile-Node. Details can be found in [I-D.ietf-lisp-mn].

When the roaming node does not support LISP, but LISP runs in the network the node roams to, the EIDs and RLOCs are not co-located in the same device. In this case, EIDs are assigned to the roaming node and RLOCs are assigned to LISP xTRs. So when the roaming node attaches to the network, its EIDs are mapped to the RLOCs of the LISP xTRs in the network. This form of mobility is called LISP EID-Mobility. Details can be found in [I-D.ietf-lisp-eid-mobility].

For a 3GPP mobile network, the LISP EID-Mobility form of mobility is recommended and is specified in the use-case document [I-D.farinacci-lisp-mobile-network].

#### 5.4.5. ILSR

ILSR is a specific recommendation for using LISP in the 3GPP 5G mobile network architecture. A detailed whitepaper can be found at [ILSR-WP]. The recommendation is to use the mechanisms in [I-D.farinacci-lisp-mobile-network].

#### 5.5. ILA

Identifier-Locator Addressing [I-D.herbert-intarea-ila] is a protocol to implement transparent network overlays without encapsulation. It addresses the need for network overlays in virtualization and mobility that are efficient, lightweight, performant, scalable, secure, provide seamless mobility, leverage and encourage use of IPv6, provide strong privacy, are interoperable with existing infrastructure, applicable to a variety of use cases, and have simplified control and management.

### 5.5.1. Overview

ILA is a form of identifier/locator split where IPv6 addresses are transformed from application-visible, non-topological "identifier" addresses to topological "locator" addresses. Locator addresses allow packets to be forwarded to the network location where a logical or mobile node currently resides or is attached. Before delivery to the ultimate destination, locator addresses are reverse transformed back to the original application visible addresses. ILA does address "transformation" as opposed to "translation" since address modifications are always undone. ILA is conceptually similar to ILNP and 8+8, however ILA is contained in the network layer. It is not limited to end node deployment, does not require any changes to transport layer protocols, and does not use extension headers.

ILA includes both a user plane and control plane. The user plane defines the address structure and mechanisms for transforming application visible identifier addresses to locator addresses. The control plane's primary focus is a mapping system that includes a database of identifier to locator mappings. This mapping database drives ILA transformations. Control plane protocols disseminate identifier to locator mappings amongst ILA nodes.

The use cases of ILA include mobile networks, datacenter virtualization, and network virtualization. A recent trend in the industry is to build converged networks containing all three of these to provide low latency and high availability. A single network overlay solution that works across multiple use cases is appealing.

Benefits of ILA include:

- o Facilitates node mobility and virtualization
- o Multiple use cases (mobile, datacenter, cloud)
- o Super efficient and performant user plane
- o Allows strong privacy in addressing
- o Promotes anchorless mobility
- o No typical tunneling issues (e.g. MTU) or management related to encapsulation
- o Flexible control plane that splits data and control
- o Modern "SDN" control protocols (e.g. RPC/TCP)

- o Scale number of nodes to billions for 5G, DC virtualization
- o Upstream Linux kernel data path and open source ctrl plane [ILACONTROL].

The ILA user plane protocol is described in [I-D.herbert-intarea-ila], motivation and problems areas are described in [ILAMOTIVE], ILA in the mobile user-plane is described in detail in [I-D.herbert-ila-mobile].

#### 5.5.2. Protocol Layering

Figure 11 illustrates the protocol layers of packets sent over various user plane interfaces in the downlink direction of data network to a mobile node. Note that this assumes the topology shown in Figure 2 where GTP-U is used over N3 and ILA is used on N9.

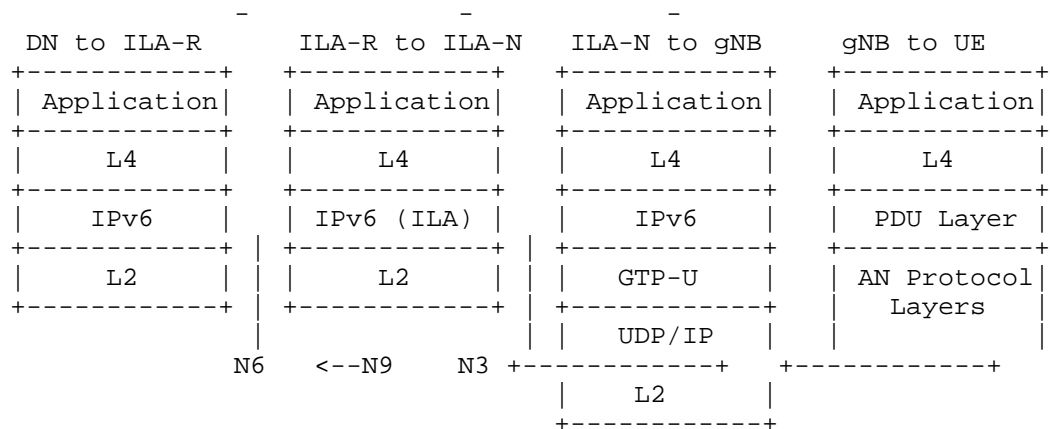


Figure 11: ILA and protocol layer in 5G

#### 5.5.3. Control plane

ILA-M provides the interface between the 5G services architecture and the common ILA control plane.

##### 5.5.3.1. ILA-M services interface

The control interface into ILA is via an ILA-M that interacts with 5G network services. ILA-M uses RESTful APIs to make requests to network services. An ILA-M receives notifications when devices enter the network, leave it, or move within the network. The ILA-M writes the ILA mapping entries accordingly.

ILA is a consumer of several 5G network services. The service operations of interest to ILA are:

- o Nudm (Unified Data Management): Provides subscriber information.
- o Nsmf (Service Management Function): Provides information about PDU sessions.
- o Namf (Core Access and Mobility Function): Provides notifications of mobility events.

#### 5.5.3.2. ILA control plane

The ILA control plane is composed of mapping protocols that manage and disseminate information about the mapping database. There are two levels of mapping protocols: one used by ILA routers that require the full set of ILA mappings for a domain, and one used by ILA nodes that maintain a caches of mappings.

The ILA mapping system is effectively a key/value datastore that maps identifiers to locators. The protocol for sharing mapping information amongst ILA routers can thus be implemented by a distributed database [I-D.herbert-ila-ilamp]. ILA separates the control plane from the user plane, so alternative control plane protocols may be used with a common user plane [I-D.lapukhov-bgp-ila-afi], [I-D.rodriqueznatal-ila-lisp].

The ILA Mapping Protocol [I-D.herbert-ila-ilamp] is used between ILA forwarding nodes and ILA mapping routers. The purpose of the protocol is to populate and maintain the ILA mapping cache in forwarding nodes. ILAMP defines redirects, a request/response protocol, and a push mechanism to populate the mapping table. Unlike traditional routing protocols that run over UDP, this protocol is intended to be run over TCP and may be RPC oriented. TCP provides reliability, statefulness implied by established connections, ordering, and security in the form of TLS. Secure redirects are facilitated by the use of TCP. RPC facilities such REST, Thrift, or GRPC leverage widely deployed models that are popular in SDN.

#### 5.5.4. IP addressing

ILA supports single address assignments as well as prefix assignments. ILA will also support strong privacy in addressing.

## 5.5.4.1. Singleton address assignment

Singleton addresses can use a canonical 64/64 locator/identifier split. Singleton addresses can be assigned by DHCPv6.

## 5.5.4.2. Network prefix assignment

Prefix assignment can be done via SLAAC or DHCPv6-PD.

To support /64 prefix assignment with ILA, the ILA identifier can be encoded in the upper sixty-four bits of an address. A level of indirection is used so that ILA transforms the upper sixty four bits to contain both a locator and an index into a locator (ILA-N) specific table. The entry in the table provides the original sixty-four bit prefix so that locator to identifier address transformation can be done.

As an example of this scheme, suppose network has a /24 prefix. The identifier address format for /64 assignment might be:

|                     |         |  |            |                     |         |  |
|---------------------|---------|--|------------|---------------------|---------|--|
| +-----+-----+-----+ |         |  |            | +-----+-----+-----+ |         |  |
|                     | 24 bits |  | 40 bits    |                     | 64 bits |  |
| +-----+-----+-----+ |         |  |            | +-----+-----+-----+ |         |  |
|                     | Network |  | Identifier |                     | IID     |  |
| +-----+-----+-----+ |         |  |            | +-----+-----+-----+ |         |  |

The IID part is arbitrarily assigned by the device, so that is ignored by ILA. All routing, lookups, and transformations (excepting checksum neutral mapping) are based on the upper sixty-four bits.

For identifier to locator address transformation, a lookup is done on the upper sixty-four bits. That returns a value that contains a locator and a locator table index. The resulting packet format may be something like:

|                     |         |  |                     |                     |         |  |
|---------------------|---------|--|---------------------|---------------------|---------|--|
| +-----+-----+-----+ |         |  |                     | +-----+-----+-----+ |         |  |
|                     | 24 bits |  | 20 bits   20 bits   |                     | 64 bits |  |
| +-----+-----+-----+ |         |  |                     | +-----+-----+-----+ |         |  |
|                     | Network |  | Locator   Loc index |                     | IID     |  |
| +-----+-----+-----+ |         |  |                     | +-----+-----+-----+ |         |  |

The packet is forwarded and routed to the ILA-N addressed by locator (/44 route in this case). At the ILA forwarding node, the locator index is used as a key to an ILA-N specific table that returns a 40 bit Identifier. This value is then written in the packet do ILA to identifier address transformation thereby restoring the original destination address.

The locator index is not globally unique, it is specific to each ILA-N. When a node attaches to an ILA-N, an index is chosen so that the table is populated at the ILA-N and the ILA mapping includes the locator and index. When a node detaches from an ILA, its entry in the table is removed and the index can be reused after a hold-down period to allow stale mappings to be purged.

#### 5.5.4.3. Strong privacy addresses

Note that when a /64 is assigned to UEs, the assigned prefix may become a persistent identifier for a device. This is a potential privacy issue.

#### 5.5.5. Traffic engineering

ILA is primarily a mechanism for mobility and network virtualization. Transport mechanisms for traffic engineering such as MPLS, network slices, encapsulation, routing based on flow hash(flow label) can be applied independently of ILA. This separation allows any discussion related to transport to be left to operator deployment.

#### 5.5.6. Locator Chaining with ILA

ILA transformations can be performed on a hop-by-hop bases. In this manner a packet can be source routed through a sequence of nodes. At each hop a determination is made as to the next hop the packet should visit. The locator for the target is then written into the destination. Eventually, the packet will be forwarded to an ILA forwarding node that will restore the original address before delivery to the final destination.

#### 5.5.7. Security considerations

A mobile public infrastructure has many considerations in security as well as privacy. Fundamentally, a system must protect against misdirection for the purposes of hijacking traffic, spoofing, revealing user identities, exposing accurate geo-location, and Denial of Service attacks on the infrastructure.

The ILA mapping system contains personally identifiable information (PII) including user identities and geo-location. The information must be safeguarded. An ILA domain is confined to one administrative domain, only trusted parties/entities in the domain participate in ILA. There is no concept of a global, public mapping system and UEs in public networks generally do not participate in ILA protocols since they are untrusted. ILA control protocols, include ILA redirects, use TCP. TLS or other protocols can be applied for strong security.



Privacy in addressing is a consideration. ILA endeavors to provide a mechanism of address assignment that prevents inference of user identity or location.

## 5.6. Hybrid ICN (hICN)

### 5.6.1. Overview

hICN Anchorless Mobility Management (hICN-AMM) refers to a novel mobility management approach, introduced in [I-D.auge-dmm-hicn-mobility], that leverages routable location-independent identifiers (IDs) and an Information-Centric Networking (ICN) communication model integrated in IPv6, (also referred to as Hybrid ICN, or hICN) [I-D.muscariello-intarea-hicn].

Such approach belongs to the category of pure ID-based mobility management schemes whose objective is (i) to overcome the limitations of traditional locator-based solutions like Mobile IP (conf)using locators as identifiers, (ii) to remove the need for a global mapping system as the one required by locator-identifier separation solutions.

### 5.6.2. Consumer and Producer mobility

In ICN and hICN endpoints can act as consumers and/or producers. Consumers when they emit requests for named data packets (so called Interests), producers when they send data packets in response to consumers request (pull-based transport model). Clearly a node can be a consumer and a producer at the same time (e.g. in a voice conversation).

Consumer and producer mobility are handled in a different way due to the pull-based request model. More specifically, consumer mobility is natively supported: consumers pull traffic by sending Interest packets towards named content (wherever produced/stored, the source is a priori unknown by the consumer). Interests are named-based forwarded using the information found in traversed routers' FIBs.

In case of consumer mobility, i.e. mobility of the endpoint issuing the requests, selection of a new available output interface and retransmission of not-yet-satisfied Interests is sufficient for data delivery to continue, independently from the underlying change of locators. Consumer mobility is fully anchorless with hICN, and does not incur any signalization nor tunneling overhead.

Producer mobility is not natively supported by ICN architecture, rather handled in different ways according to the selected producer mobility management scheme.

### 5.6.3. Anchorless mobility support

The selected mobility management scheme for hICN is MAP-Me, an anchorless producer mobility management solution originally proposed for ICN [I-D.irtf-icnrg-mapme] [MAPME] and further extended to hICN in [I-D.auge-dmm-hicn-mobility].

MAP-Me belongs to the class of anchorless approaches that relies on scope-limited forwarding updates triggered by producer mobility events to keep locally up-to-date FIB information for a low-latency guaranteed reroute of consumer Interests towards changing location of the producer. Forwarding and mobility management operations in hICN are based only on location-independent identifiers, preserving coexistence with IP locators whose existence may be required by non-hICN services and by control/management plane operations specific to the considered network architecture.

Signaling of mobility is only required upon producer movements and limited in scope to current-to-previous network hops. Unlike routing updates, it is not necessary to update all routers' FIBs after a node has moved, but only those located on the path between the new and a former position of the producer. Scalability of producer mobility is guaranteed by an efficient and secure FIB update process with minimal and bounded path stretch.

The difference w.r.t. to other classes of approaches is that it does not require an anchor neither in forwarding plane (no tunnel, traffic does not need to pass through a specific network node), nor in the control plane (no rendez-vous point, no mapping system).

### 5.6.4. Benefits

The appeal of purely ID-based architectures is that they move Loc/ID split one step further by embedding ID-awareness in the network and transport layer by default and as such completely decoupling data delivery from underlying network connectivity. The resulting mobility management solution is fully anchorless for both consumer and producer mobility. Forwarding is performed directly based on identifiers stored in routers' FIBs and no mapping of ID into locators is required. In this way, purely ID-based architectures remove the need to maintain a global mapping system at scale, and its intrinsic management complexity.

Additional benefits brought specifically by ICN principles motivate the consideration of ICN solutions for next generation mobility architectures, like for instance:

- o the flexibility of multi-source/multi-path connectionless pull-based transport. An example is the native support for consumer mobility, i.e. the transparent emission of data requests over multiple and varying available network interfaces during node mobility;
- o the opportunity to define fine-grained per-application forwarding and security policies (in the network, and in-between UPFs);
- o low-latency and multicast capabilities by means of in-path edge caching;
- o network-assisted transport.

An in depth analysis of benefits originating from the coupling between a purely identifier-based approach and from specific hICN properties can be found in [I-D.auge-dmm-hicn-mobility-deployment-options] along with some illustrative examples.

#### 5.6.5. Deployment considerations

##### \*Partial insertion\*

The benefits previously described can be obtained by an upgrade of only a few selected routers at the network edge. The design of hICN allows the rest of the infrastructure to remain unmodified, and to leverage existing management and monitoring tools. There exists thus a tradeoff between incremental deployment and benefits which are proportionally related to the degree of hICN penetration.

##### \*End-to-end deployment\*

The deployment of an hICN stack in endpoints is the preferred option and offers the full range of benefits. Both the hICN forwarder and the transport stack are available as reference implementations based on the CICON project [CICON]. They are both designed to facilitate insertion on routers and end-user devices thanks to implementation in user space, one targetting high-performance, the other aiming at wide support from major vendors including iOS, Android, Linux, MacOSX and Windows.

##### \*Network-contained deployment\*

It is not always possible nor desirable to affect endpoints, and a deployment fully contained in the network is possible through the deployment of proxies. An example would be the deployment of HTTP proxies at the ingress and egress (resp. first and last UPFs), in

order to benefit from content awareness in the network. Such configuration however reduces the flexibility and dynamic forwarding capabilities in endpoints. In particular, existing transport protocols have limited support for dynamically changing paths or network conditions.

Traffic that is not handled through hICN mechanisms can still benefit from the lower overhead and anchorless mobility capabilities coming from the removal of GTP tunnels, as well as dynamic forwarding capabilities that are inherent to the forwarding pipeline. This results from the ability to assign location-independent identifiers to endpoints. It preserves the advantage of removing the mapping system, and of a lightweight FIB update process. No encapsulation is required and packet headers are not modified, which allows the network to have visibility in the source and/or destination identifiers.

#### \*hICN in a slice\*

The use of hICN does not impose any specific slicing of the network. Rather, it can assist a transition of services towards hICN, and/or the coexistence of different hICN deployment options.

As an example of use of hICN in a slice, a service provider might for instance decide to use an hICN-enabled slice dedicated to video delivery, with appropriate mobility management, and dedicated hICN nodes with appropriate caching/forwarding strategies at places aggregating considerable number of user requests.

#### 5.6.6. hICN with SRv6

The association of hICN with other user plane technologies, such as SRv6, is investigated as a possibility to overcome the above-mentioned tradeoff yielding to a selective, yet fully beneficial insertion of hICN in IP networks. This would inherit all SRv6 advantages for underlay (TE, FRR) and service programming (NFV), but also extend the reach of hICN on regular IP routers with SRv6 functionality.

One realization consists in creating SRv6 domains in between hICN nodes. The hICN router (through forwarding strategies) would then act as a control plane for SRv6 by specifying the list of SIDs to insert in the packet.

#### 5.6.7. Summary

hICN proposes a general purpose network architecture that combines the benefits of a pure-ID architecture with those of ICN. While a full deployment is recommended to make efficient use of available network resources, it is still possible to opt for a partial or phased deployment, with the associated tradeoffs that we have reviewed here.

An hICN enabled network offers native offloading capabilities thanks to the anchorless properties resulting from the pure-ID communication scheme. It does so without the need for a third party mapping system, and further requires no change in the 5G architecture nor in its control plane. The architecture will further leverage the incremental insertion of information centric functionalities through proxies or direct insertion in user devices as the technology gets adopted and deployed.

### 6. Integration into the 5G framework

#### 6.1. Locator based - SRv6

##### 6.1.1. Insertion in N9 interface

Existing mobile backhaul employs GTP tunnels to carry user traffic flows in the network. These tunnels are unidirectional, are established via the control plane for a particular QoS level, and run on links between access and the different anchor nodes all the way to DN gateways.

The Tunnel Endpoint Id (TEID) field in the GTP tunnel plays a crucial role in stitching the data path between the above mentioned network nodes for a particular user flow. In other words, TEIDs are used to coordinate traffic hand off between different UPFs.

In its most basic form, SRv6 can be used as a simple drop-in alternative for GTP tunnels. The control plane in this approach remains the same, and still attempts to establish GTP-U tunnels and communicate TEIDs between the tunnel end points. However, at the next level, SRv6 capable nodes use SIDs to direct user traffic between the UPFs.

The simplest option here is to encapsulate the entire GTP frame as a payload within SRv6. This scheme still carries the GTP header as the payload and as such doesn't offer any significant advantage.

A much more promising and efficient option however is to use SIDs to carry tunnel related information. This is commonly known as the

Traditional Mode for SRv6 support for mobility. Here, TEIDs and other relevant data can be encoded into SRv6 SIDs which can be mapped back to TEID's at the intermediate UPFs thus requiring no changes except at the encapsulation and de-encapsulation points in the UPF chains.

Note that this is a direct replacement of GTP by SRv6. It's also worth noting that in this case the MTU overhead in the N9 interface is reduced.

[I-D.ietf-dmm-srv6-mobile-uplane] discusses the details of leveraging the existing control plane for distributing GTP tunnel information between the end nodes and employing SRv6 in user plane for UPF connectivity. The document defines a SID structure for conveying TEID, DA, and SA of GTP tunnels, shows how hybrid IPV4/IPV6 networks are supported by this model and in doing so, it paves a migration path toward a full SRv6 user plane.

Another alternative that can provide for a smooth migration toward SRv6 data plane between UPFs is via the use of "Tag", and optional TLV fields in SRH. Similar to the previously described method, this approach takes advantage of the existing control plane to deliver GTP tunnel information to the UPF endpoints. "Tag" and optional TLV fields in SRH are then used to encode tunnel information in the SRv6 user plane where the UPFs can determine the TEID etc. by inverting the mapping.

In yet another option, GTP tunnel information can be encoded as a separate SID either within the same SRH after the SID that identifies the UPF itself (SRH-UPF) or inside a separate SRH (SRH-GTP). This option resembles the MPLS label stacking mechanism which is widely used in different VPN scenarios. Here, we use one SID to carry traffic to the target UPF and use the other to encode and decode GTP related information.

It must be noted that in any of the above mentioned approaches, the ingress UPF in SRv6 domain can insert a SRH containing the list of SIDs that corresponds to all UPFs along the path. Alternatively, UPFs can stack a new SRH on top of the one inserted by the previous one as packets traverse network paths between different pairs of UPFs in the network.

#### 6.1.2. Control Plane considerations

SRv6, when applied in Traditional Mode follows the interworking model and as such does not require control-plane changes. It still attempts to establish GTP-U tunnels and communicate TEIDs between the tunnel

endpoints. AT the next level of user plane however, SRv6 capable nodes use SIDs to direct user traffic between the UPFs.

#### 6.1.3. Extensions to N3/F1-U/Xn-U interface

Although not strictly the object of study by 3GPP, previous solutions can (and would gain to) be extended beyond N9 to cover N3 interface too.

The immediate benefit is the complete removal of all GTP tunnels, along with associated mangement complexity and traffic overhead. In particular, this removes the need for internetworking between N3 and N9 technologies, and offers a uniform user plane as recommended in the specification.

Potential gains can result for an early handling of traffic right from the RAN and thus possibly closer to the UE. The result is a simpler and lighter architecture, allowing convergence with other non-3GPP accesses.

The mobile network would benefit of the application of SRv6 to both, N3 and N9 interfaces. The intrinsic ability of SRv6 to integrate, in a single protocol, the control of the overlay, underlay and NFV implies that if applied to the N3 interface the end-to-end SRv6-based network slice can start on the NodeB itself.

In addition, SRv6 could be applied to the F1-U interface for cloud-RAN and TE purposes.

#### 6.1.4. Coexistence with GTP-based architecture

An alternative vision, although not recommended, would be to preserve the current architecture as is, and deploy alternative user planes on top.

As explained in section 5.3.1, SRv6 can co-exist with the current GTP-based control plane. Additionally, the current control plane can be extended to suport TE as defined in 5.3.2.

From a dataplane perspective, SRv6 can coexist on the N9 interface together with GTP-U traffic.

This is important towards a slow migration from a GTP-based architecture into different architectures.

## 6.2. ID-LOC split

### 6.2.1. Insertion in N9 interface

An ID-LOC network architecture is able to decouple the identity of endpoints (ID) from their location in the network (LOC). Common ID-LOC architectures are based on two main components, ID-LOC data-plane nodes and an ID-LOC mapping system.

ID-LOC data-plane nodes act upon received data traffic and perform ID-LOC data-plane operation. The specific operation that these ID-LOC data-plane nodes perform is based on the particular ID-LOC data-plane protocol that they implement. ID-LOC data-plane protocols are usually divided in two categories, (1) those that encapsulate ID-based data-plane packets into LOC-based data-plane packets and (2) those that transform the addresses on the data-plane packets from ID-based addresses to LOC-based addresses. SRv6 and LISP-DP protocols are examples of the former while the ILA protocol is an example of the latter.

The ID-LOC mapping system is a database that provides mappings of Identity to Location for ID-LOC data-plane nodes to use. Usually, ID-LOC architectures use an ID-LOC control plane protocol to make available at the data-plane nodes the ID-LOC mappings that they need to operate. Examples of such ID-LOC control plane protocols are LISP-CP and ILAMP, which are discussed later in this section.

When integrating ID-LOC architecture into the 5G framework there are several aspects to take into account. One is that the ID-LOC data-plane function needs to be performed in the data-plane path as the packets enter and leave the ID-LOC domain. One option for this is to deploy ID-LOC data-plane nodes adjacent to UPFs to perform the ID-LOC operation on the traffic as it leaves or enters the UPFs (as shown in Fig. Figure 12). In this case the ID-LOC data-plane protocol will be part of the N9 interface along with current GTP.



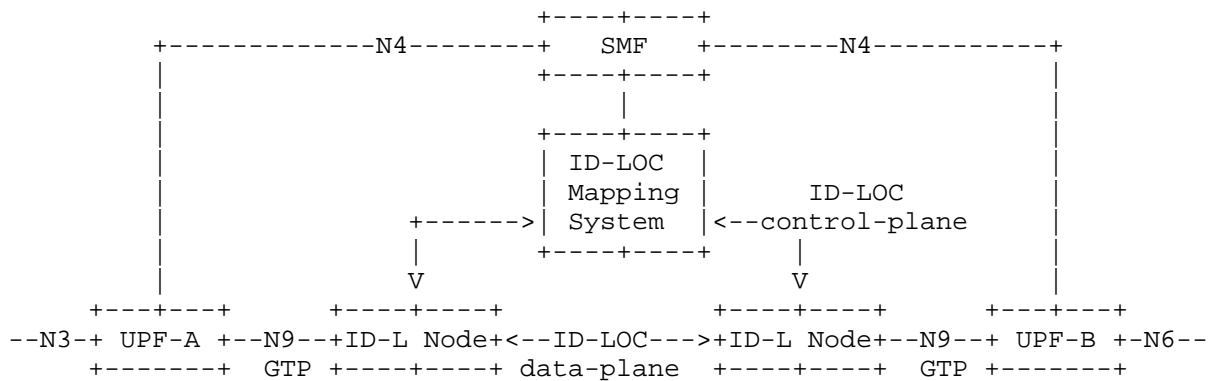


Figure 12: 5G Integration with ID-LOC (Interworking model)

Another option is to implement the ID-LOC data-plane function directly in the UPFs (as shown in Fig. Figure 13). In this case, these ID-LOC enabled UPFs will directly generate packets encapsulated or transformed and will be able to directly process packets encapsulated or transformed. In this case the ID-LOC protocol will completely replace GTP in the N9 interface.

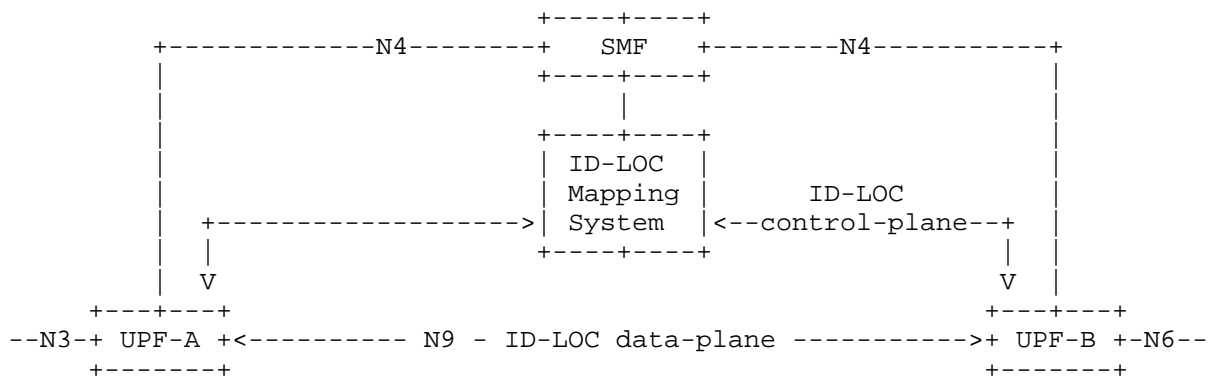


Figure 13: 5G Integration with ID-LOC (Integrated model)

Finally, another aspect to consider when integrating the ID-LOC architecture into the 5G framework is that the Mapping System needs to contain the appropriate ID-LOC mappings in coordination with the SMF. In order to do so, the mappings in the Mapping System are populated either by the SMF directly or by the LOC-nodes that should be in synch with the SMF. In the former case, an interface from the SMF to the Mapping System is needed (as shown in Figs. Figure 12 and Figure 13).

### 6.2.2. LISP control plane

The current LISP control-plane (LISP-CP) specification [I-D.ietf-lisp-rfc6833bis] is data-plane agnostic and can serve as control plane for different data-plane protocols (beyond the LISP data-plane). LISP-CP offers different mechanisms to register, request, notify and update ID-Loc mappings between ID-LOC data-plane elements and the ID-LOC Mapping System. In the sections below we describe how LISP-CP can serve to enable the operation of the ILA data-plane and the SRv6 data-plane.

It should be noted that the LISP-CP can run over TCP or UDP. The same signaling and logic applies independently of the transport. Additionally, when running over TCP, the optimizations specified in [I-D.kouvelas-lisp-map-server-reliable-transport] can be applied.

#### 6.2.2.1. LISP-CP for ILA

The LISP-CP can serve to resolve the Identifier-to-Locator mappings required for the operation of an ILA data-plane. The required ILA control plane operations of "request/response" and "push" are implemented via the LISP mechanisms defined in [I-D.ietf-lisp-rfc6833bis] and [I-D.ietf-lisp-pubsub] respectively. In addition, the ILA "redirect" operation is implemented via the mapping notifications described in [I-D.ietf-lisp-pubsub] triggered as response to data-plane events.

Furthermore, the LISP-CP can also be used to obtain the ILA Identifier when it is not possible to locally derivate it from the endpoint address. These two mapping operations, Endpoint-to-Identifier and Identifier-to-Locator, can be combined into one mapping operation to obtain the ILA Identifier and associated Locators in a single round of signaling.

The complete specification of how to use the LISP-CP in conjunction with an ILA data-plane can be found in [I-D.rodriqueznatal-ila-lisp].

#### 6.2.2.2. LISP-CP for SRv6

The LISP-CP can be used by an ingress SRv6 node to obtain the egress node SRv6 VPN SID and its corresponding SLA associated with such endpoint. Alternatively, an ingress SRv6 node can use the LISP-CP to obtain not only the egress SRv6 VPN segment for a particular endpoint but also the SRv6 SID list to steer the traffic to that egress SRv6 node.

The complete specification of how to use the LISP-CP in conjunction with an SRv6 data-plane can be found in [I-D.rodriqueznatal-lisp-srv6].

### 6.2.3. ILA control plane

The ILA control plane is composed of mapping protocols that manage and disseminate information about the mapping database. There are two levels of mapping protocols: one used by ILA routers that require the full set of ILA mappings for a domain, and one used by ILA nodes that maintain a caches of mappings.

The ILA mapping system is effectively a key/value datastore that maps identifiers to locators. The protocol for sharing mapping information amongst ILA routers can thus be implemented by a distributed database [I-D.herbert-ila-ilamp]. ILA separates the control plane from the user plane, so alternative control plane protocols may be used with a common user plane [I-D.lapukhov-bgp-ila-afi], [I-D.rodriqueznatal-ila-lisp].

The ILA Mapping Protocol [I-D.herbert-ila-ilamp] is used between ILA forwarding nodes and ILA mapping routers. The purpose of the protocol is to populate and maintain the ILA mapping cache in forwarding nodes. ILAMP defines redirects, a request/response protocol, and a push mechanism to populate the mapping table. Unlike traditional routing protocols that run over UDP, this protocol is intended to be run over TCP and may be RPC oriented. TCP provides reliability, statefulness implied by established connections, ordering, and security in the form of TLS. Secure redirects are facilitated by the use of TCP. RPC facilities such REST, Thrift, or GRPC leverage widely deployed models that are popular in SDN.

### 6.2.4. Extensions to N3/F1-U/Xn-U interface

While not the main focus of this document, it is worth noting that it is also possible to enable an ID-LOC data-plane over the N3 interface and to instantiate the ID-LOC overlay directly at the NodeB. In this case, the NodeB will implement the functionality of an ID-LOC node, i.e. it will retrieve ID-LOC mappings using an ID-LOC control protocol and will encapsulate/transform ID packets into LOC packets. Bringing the ID-LOC data-plane to the NodeB (closer to the UE) has several advantages: (1) complete removal of GTP tunnels, (2) unified management of the ID-LOC data-plane across the network, (3) improved data-plane latency due to traffic being forwarded to the destination ID-LOC node directly from the NodeB, and (4) lower handover time since the ID-LOC mobility event can start at the NodeB itself.

#### 6.2.5. Coexistence with GTP-based architecture

ID-Locator separation architecture can be implemented by control plane of a dedicated protocol such as LISP, ILA, etc., however, it may cause major impact to the specifications of 3GPP 5GS. The approach, described in [I-D.homma-dmm-5gs-id-loc-coexistence], enables to introduce such ID-Locator separation protocols into 5GS with no or low impacts. It would also support a migration path toward a network which an ID-Locator separation protocol is completely incorporated.

This approach establishes an individual domain/slice in which an ID-Locator

separation protocol works as packet forwarding mechanism, and divert the appropriate packets (e.g., packets for UE-to-UE communication) to the domain at local/distributed UPFs by using Up-Link Classifier (ULCL). ULCL is a fundamental function of UPF, and it diverts uplink traffic based on filter rules indicated by SMF. The other packets to a central UPF (e.g., packets for Internet access) are forwarded with GTP-U via N9 interface.

The architecture is shown in Figure 14.

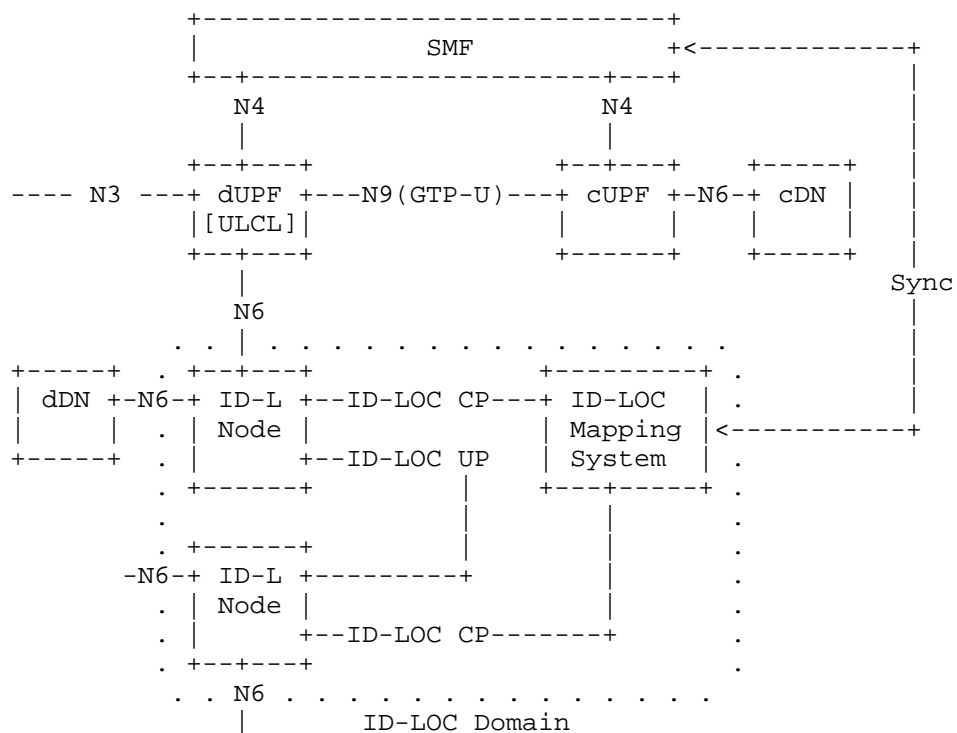


Figure 14: Architecture of 5GS and ID-LOC Coexistence

Coexistence approach allows to use GTP-U or any other forwarding protocol described in this document as user plane mechanism. However, each LOC-Node must be connected to the all other LOC-Nodes, and thus it may cause complexity of path management if you use a protocol which needs session establishment.

Regarding to control plane of this approach, every dedicated ID-Locator separation protocol described in this document can be used. For management of mobility of UEs in ID-Locator separation domain, some cooperation between SMF and mapping system is needed. In this approach, a UE is attached to a LOC-Node only when it communicates to another UE or an NF in a dDN. In 5GS, SMF manages sessions, and thus SMF may be required to update mapping database when an UE moves to under another UPF or an NF is moved to another dDN. The impact caused by such cooperation can be reduced by using Naf interface which is defined in 5GS specifications.

This approach provides a mechanism for introducing ID-Locator separation architecture into 5GS with no or nominal impact, and achieves optimization of forwarding path and session continuity.

Moreover, this can keep scalability on forwarding on down link from cDN/Internet because it can use the current GTP-based mechanism.

Meanwhile, this approach causes an extra hop when diverting packets to ID-Locator separation domain, and it may leads to increase of latency.

### 6.3. ID-based - hICN

By operating directly on routers' FIBs for mobility updates, dynamic hop-by-hop forwarding strategies etc., hICN inherits the simplicity of IP forwarding and reuses IP routing protocols for ID prefixes advertisement and routing. In this way it removes the challenges of managing a distributed mapping service at scale (cache update/refresh, etc.). In addition it remains compatible with the exiting control plane architecture as proposed in the 3GPP standard, with no change required to N1, N2 or N4.

MAP-Me anchorless producer mobility management does not imply SMF interaction, but does not exclude neither to use SMF signaling to trigger MAP-Me updates or to handle FIB updates, at the condition to follow the same procedure described for MAP-Me. However, the absence of SMF interaction might be beneficial in case of dense deployments or failure of the central control entities (infrastructure-less communication scenarios) to empower distributed control of local mobility within an area.

#### 6.3.1. Insertion in N9 interface

Insertion of hICN in 5G IP infrastructure is facilitated by its design allowing a selective insertion of hICN capabilities in a few network nodes at the edge (no need for pervasive fully hICN network enablement), and to guarantee a transparent interconnection with hICN-unaware IP nodes, without using overlays.

The deployment of hICN routers allow to avoid the reliance on GTP tunnels, and to provide an agile transport and native anchorless mobility natively. The resulting protocol stack is shown in Figure 15. We remark that in the protocol layer, hICN is associated to IPv6 PDU layer, transported in N9 directly over L2.

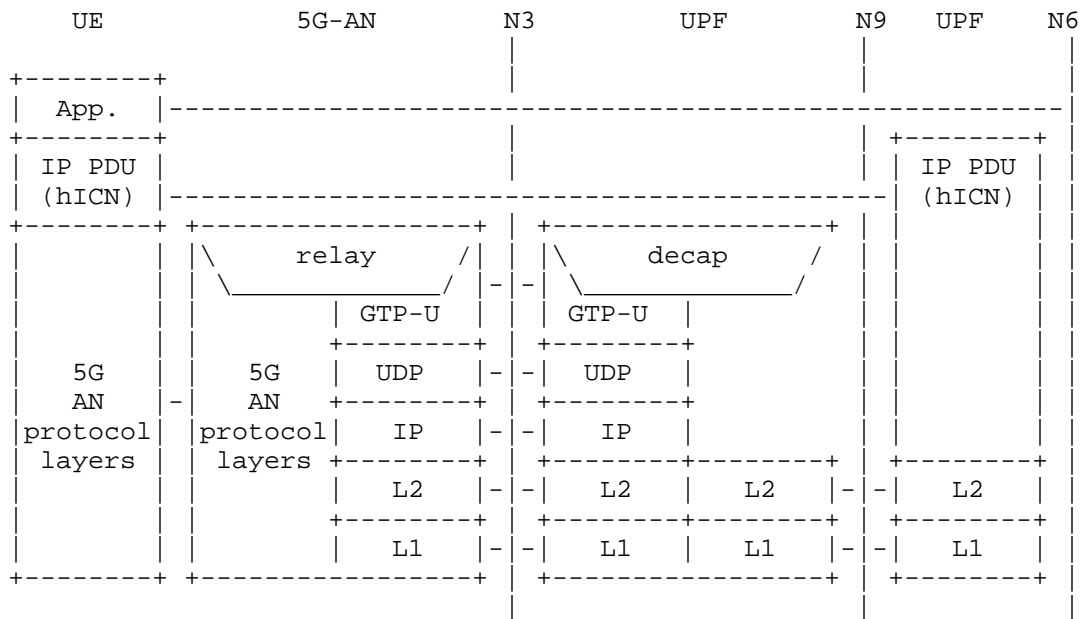


Figure 15: Replacement of N9 interface - Protocol layers

### 6.3.2. Control plane considerations

By operating directly on routers' FIBs for mobility updates, dynamic hop-by-hop forwarding strategies etc., hICN inherits the simplicity of IP forwarding and reuses IP routing protocols for ID prefixes advertisement and routing. In this way it removes the challenges of managing a distributed mapping service at scale (cache update/refresh, etc.). In addition it remains compatible with the existing control plane architecture as proposed in the 3GPP standard, with no change required to N1, N2 or N4.

MAP-Me anchorless producer mobility management does not imply SMF interaction, but does not exclude neither to use SMF signaling to trigger MAP-Me updates or to handle FIB updates, at the condition to follow the same procedure described for MAP-Me. However, the absence of SMF interaction might be beneficial in case of dense deployments or failure of the central control entities (infrastructure-less communication scenarios) to empower distributed control of local mobility within an area.

### 6.3.3. Extensions to N3/F1-U/Xn-U interface

This option ensures that forwarding beyond the radio access is directly managed through hICN. As a consequence, no additional state nor signaling is required for static and mobile consumers, nor for static producers. The impact of producer mobility is low because of the small number of impacted routers.

Dynamic forwarding capabilities are extended in this configuration to the selection of the first UPF, with the potential of additional performance improvement and higher traffic offload because of the deployment of hICN functionalities closer to the UE. A significant advantage arises in dense deployments scenarios where it becomes possible to isolate the core network from the locally-management mobility (a design objective of the mobile architecture), while allowing distributed selection of ingress UPFs, and dynamic per-packet load balancing of traffic across them.

### 6.3.4. Coexistence with GTP-based architecture

This section discusses the insertion of hICN-AMM in an unmodified 3GPP 5G reference architecture, where GTP tunnels are preserved. As previously stated, maintaining GTP tunnels does not allow to overcome limitations of anchor-based approaches. However, a transparent integration of hICN-AMM limits to the minimum deployment costs and already brings advantages over the baseline architecture presented earlier.

The first option shares some similarities with the previous situation and proposes to deploy hICN-AMM within Mobile Edge Computing (MEC) platforms. It relies on the local breakout capability introduced in 5G through the UL/CL. This function is used to realize the hICN punting function described in [I-D.muscariello-intarea-hicn], i.e. to identify hICN traffic (Interest and Data packets) and forward it to the local MEC hICN instance. Although it preserves tunnels and anchor points, this option permits an early termination of tunnels and the distribution of hICN capabilities close to the edge like in path caching and rate/loss/congestion control which may be leveraged for efficient low-latency content distribution especially in presence of consumer mobility.

The second option consists in the deployment of hICN-AMM as User Plane Function (UPF) inside mobile user plane. It has the advantage of preserving hICN benefits in terms of consumer mobility and flexible transport.

A more in depth presentation of those alternative deployments can be found in [I-D.auge-dmm-hicn-mobility-deployment-options].



#### 6.4. Coexistence of multiple protocols in network slices

Slicing is one of the main features in 5G. Several Slices with different requirements can coexist on top of the common network infrastructure. Diverse flows belonging to different 5G slices can be completely disjoint or can share different parts of the network infrastructure.

All proposals reviewed in this draft lend themselves well to 5G slicing paradigm, that can assist a transition of services towards these new user plane protocols, or allow the coexistence of different deployment options.

Figure 16 illustrates the use of network slices with the different proposals. All categories of approach can coexist in separate slices, so as different deployments of the same approach. We refer to previous sections for more details about the possible configurations for ID-LOC, and limit our discussion here to the possibility for different slices to deploy their own mapping system, or share it as illustrated here.

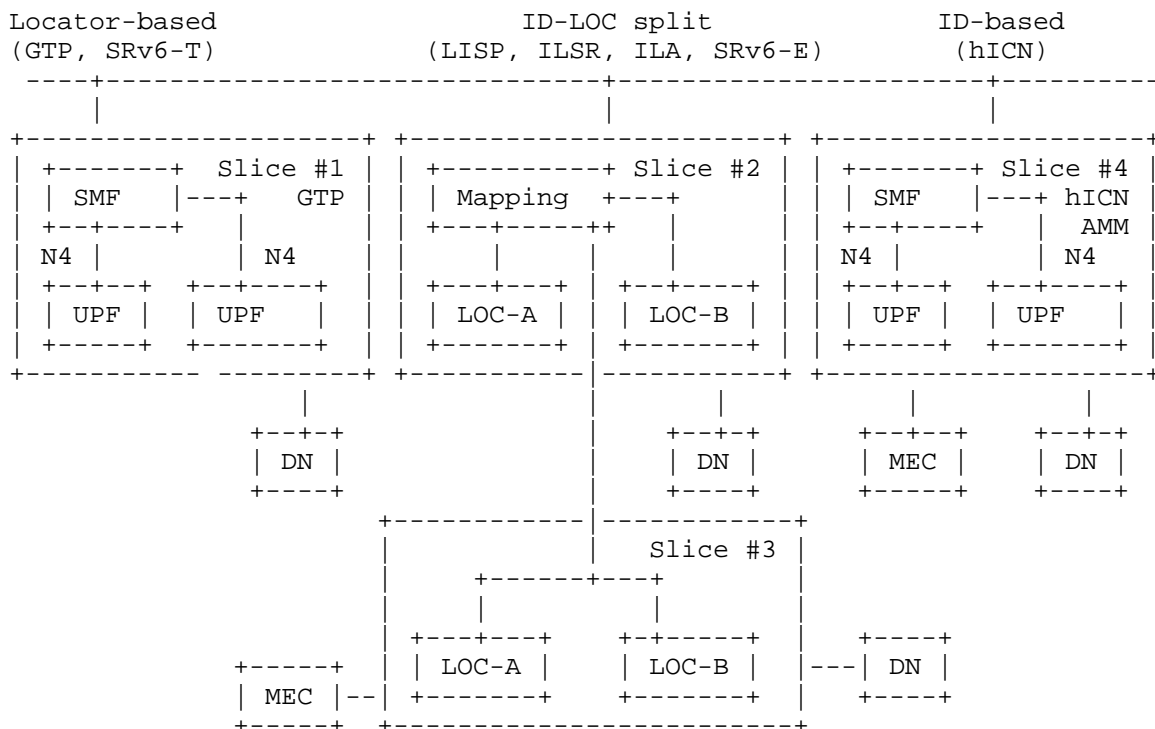


Figure 16: Network slices in 5G

#### \*Locator-based\*

Slice #1 illustrates legacy use of UPFs with GTP in a slice. New approaches can be deployed incrementally or in parts of the network. As demonstrated, the use of network slices can provide domain isolation for this.

#### \*ID-LOC split\*

Slice #2 and #3 support ID-LOC. We illustrate in slice #2 a typical deployment with ILA. Mapping then corresponds to ILA-M, LOC-A to ILA-N and LOC-B to ILA-R.

Some number of ILA-Ns and ILA-Rs are deployed. ILA transformations are performed over the N9 interface. ILA-Rs would be deployed at the N6 interface to perform transformations on packets received from a data network. ILA-Ns will be deployed deeper in the network at one side of the N3 interface. ILA-Ns may be supplemented by ILA-Rs that are deployed in the network. ILA-M manages the ILA nodes and mapping database within the slice.

Slice #3 shows another slice that supports ILA. In this scenario, the slice is for Mobile Edge Computing. The slice contains ILA-Rs and ILA-Ns, and as illustrated, it may also contain ILA\_Hs that run directly on edge computing servers. Note in this example, one ILA-M, and hence one ILA domain, is shared between slice #2 and slice #3. Alternatively, the two slices could each have their own ILA-M and define separate ILA domains.

#### \*ID-based\*

Finally, in slice #4, a slice using hICN-AMM is shown, that does not require any mapping system nor changes in N4.

### 6.5. Interoperability/Roaming considerations

Different situations including roaming scenarios might require the coexistence of different mobility protocols for the same user plane. In Figure 17 and Figure 18, we illustrate two possible deployments for the Home-Routed Roaming Scenario, respectively using a UPF supporting several protocols, and relying on an exchange service point for interconnection.

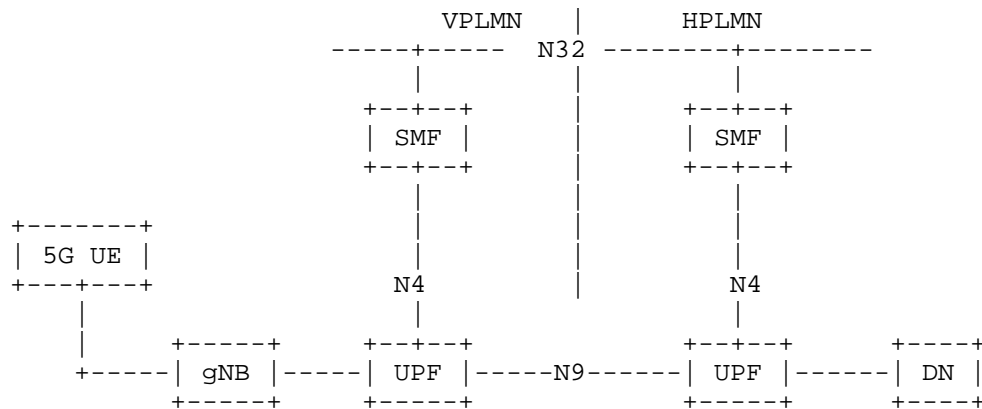


Figure 17: Direct Connectivity between operators with UPFs supporting more than one mobility protocols

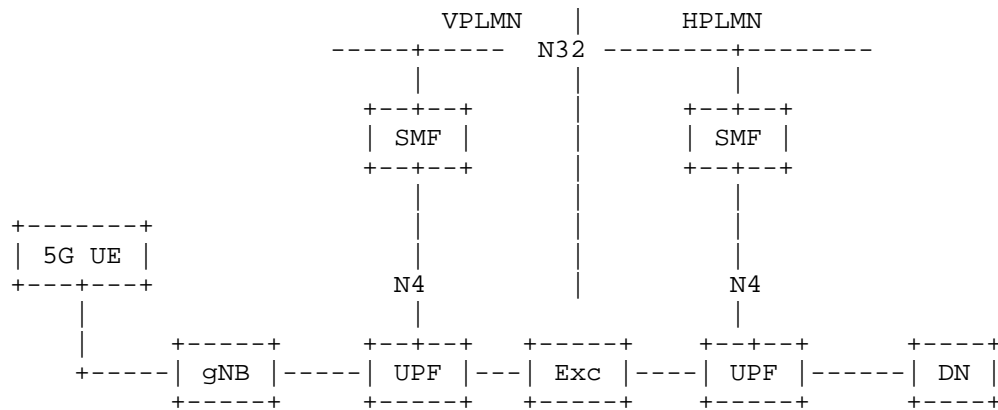


Figure 18: Connectivity between operators using an Exchange that supports multiple mobility protocols

## 7. Summary

This document summarizes the various IETF protocol options for GTP replacement on N9 interface of 3GPP 5G architecture. The document also proposes optional replacements of GTP in N3 interface.

## 8. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [RFC2234].

## 9. Security Consideration

All 3GPP security aspects apply to all the protocols discussed in this document.

## 10. IANA Considerations

There are no IANA considerations in this specification.

## 11. Acknowledgement

The authors would like to thank Farooq Bari, Devaki Chandramouli, Ravi Guntupalli, Sri Gundavelli, Peter Ashwood Smith, Satoru Matsushima, Michael Mayer, Vina Ermagan, Fabio Maino, Albert Cabellos, Cameron Byrne for reviewing various iterations of the document and for providing content into various sections.

## 12. References

### 12.1. Normative References

- [RFC1027] Carl-Mitchell, S. and J. Quarterman, "Using ARP to implement transparent subnet gateways", RFC 1027, DOI 10.17487/RFC1027, October 1987, <<https://www.rfc-editor.org/info/rfc1027>>.
- [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>>.
- [RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, DOI 10.17487/RFC2234, November 1997, <<https://www.rfc-editor.org/info/rfc2234>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<https://www.rfc-editor.org/info/rfc6830>>.

- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <<https://www.rfc-editor.org/info/rfc6831>>.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <<https://www.rfc-editor.org/info/rfc6832>>.
- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <<https://www.rfc-editor.org/info/rfc6833>>.
- [RFC6835] Farinacci, D. and D. Meyer, "The Locator/ID Separation Protocol Internet Groper (LIG)", RFC 6835, DOI 10.17487/RFC6835, January 2013, <<https://www.rfc-editor.org/info/rfc6835>>.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <<https://www.rfc-editor.org/info/rfc6836>>.
- [RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <<https://www.rfc-editor.org/info/rfc7215>>.
- [RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and S. Eum, "Information-Centric Networking: Baseline Scenarios", RFC 7476, DOI 10.17487/RFC7476, March 2015, <<https://www.rfc-editor.org/info/rfc7476>>.
- [RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <<https://www.rfc-editor.org/info/rfc8061>>.
- [RFC8111] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)", RFC 8111, DOI 10.17487/RFC8111, May 2017, <<https://www.rfc-editor.org/info/rfc8111>>.

- [RFC8112] Farinacci, D., Jain, A., Kouvelas, I., and D. Lewis, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT) Referral Internet Groper (RIG)", RFC 8112, DOI 10.17487/RFC8112, May 2017, <<https://www.rfc-editor.org/info/rfc8112>>.

## 12.2. Informative References

- [CICN]        Linux Foundation fd.io, "CICN project", 2018.
- [CP-173160-1]        3rd Generation Partnership Project (3GPP), "New Study Item on User Plane Protocol in 5GC", December 2017.
- [I-D.auge-dmm-hicn-mobility]        Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "Anchorless mobility through hICN", draft-auge-dmm-hicn-mobility-00 (work in progress), June 2018.
- [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-00 (work in progress), June 2018.
- [I-D.farinacci-lisp-mobile-network]        Farinacci, D., Pillay-Esnault, P., and U. Chunduri, "LISP for the Mobile Network", draft-farinacci-lisp-mobile-network-03 (work in progress), March 2018.
- [I-D.filsfils-spring-srv6-network-programming]        Filsfils, C., Li, Z., 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., and M. Sharif, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-04 (work in progress), March 2018.
- [I-D.herbert-ila-ilamp]        Herbert, T., "Identifier Locator Addressing Mapping Protocol", draft-herbert-ila-ilamp-00 (work in progress), December 2017.

## [I-D.herbert-ila-mobile]

Herbert, T. and K. Bogineni, "Identifier Locator Addressing for Mobile User-Plane", draft-herbert-ila-mobile-01 (work in progress), March 2018.

## [I-D.herbert-intarea-ila]

Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", draft-herbert-intarea-ila-01 (work in progress), March 2018.

## [I-D.homma-dmm-5g-uplane-analysis]

Homma, S., Miyasaka, T., and S. Matsushima, "User Plane Protocol and Architectural Analysis on 3GPP 5G System", 2018.

## [I-D.homma-dmm-5gs-id-loc-coexistence]

Homma, S., Kawakami, K., and A. Akhavain, "Co-existence of 3GPP 5GS and Identifier Locator Separation Architecture", draft-homma-dmm-5gs-id-loc-coexistence-01 (work in progress), May 2018.

## [I-D.ietf-6man-segment-routing-header]

Previdi, S., Filsfils, C., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-13 (work in progress), May 2018.

## [I-D.ietf-dmm-srv6-mobile-uplane]

Matsushima, S., Filsfils, C., Kohno, M., Camarillo, P., daniel.voyer@bell.ca, d., and C. Perkins, "Segment Routing IPv6 for Mobile User Plane", draft-ietf-dmm-srv6-mobile-uplane-01 (work in progress), March 2018.

## [I-D.ietf-lisp-eid-mobility]

Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-02 (work in progress), May 2018.

## [I-D.ietf-lisp-introduction]

Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.

- [I-D.ietf-lisp-mn]  
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", draft-ietf-lisp-mn-02 (work in progress), April 2018.
- [I-D.ietf-lisp-pubsub]  
Rodriguez-Natal, A., Ermagan, V., Leong, J., Maino, F., Cabellos-Aparicio, A., Barkai, S., Farinacci, D., Boucadair, M., Jacquenet, C., and S. Secci, "Publish/Subscribe Functionality for LISP", draft-ietf-lisp-pubsub-00 (work in progress), April 2018.
- [I-D.ietf-lisp-rfc6830bis]  
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos-Aparicio, "The Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-rfc6830bis-12 (work in progress), March 2018.
- [I-D.ietf-lisp-rfc6833bis]  
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio, "Locator/ID Separation Protocol (LISP) Control-Plane", draft-ietf-lisp-rfc6833bis-10 (work in progress), March 2018.
- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.
- [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-01 (work in progress), June 2018.
- [I-D.irtf-icnrg-mapme]  
Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "MAP-Me : Managing Anchorless Mobility in Content Centric Networking", draft-irtf-icnrg-mapme-00 (work in progress), March 2018.
- [I-D.irtf-icnrg-terminology]  
Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): CCN and NDN Terminology", draft-irtf-icnrg-terminology-00 (work in progress), December 2017.



- [I-D.kouvelas-lisp-map-server-reliable-transport]  
Cassar, C., Leong, J., Lewis, D., Kouvelas, I., and J. Arango, "LISP Map Server Reliable Transport", draft-kouvelas-lisp-map-server-reliable-transport-04 (work in progress), September 2017.
- [I-D.lapukhov-bgp-ila-afi]  
Lapukhov, P., "Use of BGP for dissemination of ILA mapping information", draft-lapukhov-bgp-ila-afi-02 (work in progress), October 2016.
- [I-D.muscariello-intarea-hicn]  
Muscariello, L., Carofiglio, G., Auge, J., and M. Papalini, "Hybrid Information-Centric Networking", draft-muscariello-intarea-hicn-00 (work in progress), June 2018.
- [I-D.rodriqueznatal-ila-lisp]  
Rodriguez-Natal, A., Ermagan, V., Maino, F., and A. Cabellos-Aparicio, "LISP control-plane for Identifier Locator Addressing (ILA)", draft-rodriqueznatal-ila-lisp-01 (work in progress), April 2018.
- [I-D.rodriqueznatal-lisp-srv6]  
Rodriguez-Natal, A., et al., "LISP Control Plane for SRv6 Endpoint Mobility", draft-rodriqueznatal-lisp-srv6-00 (work in progress), June 2018.
- [I-D.vonhugo-5gangip-ip-issues]  
Hugo, D. and B. Sarikaya, "Review on issues in discussion of next generation converged networks (5G) from an IP point of view", draft-vonhugo-5gangip-ip-issues-03 (work in progress), March 2017.
- [I-D.xuclad-spring-sr-service-chaining]  
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Segment Routing for Service Chaining", draft-xuclad-spring-sr-service-chaining-01 (work in progress), March 2018.
- [ILACONTROL]  
Herbert, T., "Identifier Locator Addressing Mapping Protocol draft-herbert-ila-ilamp-00", December 2017.
- [ILAGRPS]  
Herbert, T., "Identifier Groups draft-herbert-idgroups-00", February 2018.

- [ILAMOTIVE] Herbert, T., "Identifier Locator Addressing: Problem Areas, Motivation, and Use Cases draft-herbert-ila-motivation-00", January 2018.
- [ILSR-WP] Kurebayashi, R., Ashwood-Smith, P., and D. Farinacci, "Evolving 5G Routing", December 2017.
- [IRTF-RRG] Li, T., "IRTF Routing Research Group (rrg)", November 2012.
- [LISP-WG] Halrpen, J. and L. Iannone, "IETF Locator/ID Separation Protocol (lisp) Working Group", June 2018.
- [MAPME] Auge, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., and X. Zeng, "MAP-Me: Managing Anchor-Less Producer Mobility in Content-Centric Networks", IEEE Transactions on Network and Service Management Vol. 15, pp. 596-610, DOI 10.1109/tnsm.2018.2796720, June 2018.
- [SP-180231-1] 3rd Generation Partnership Project (3GPP), "New Study on Enhancements to the Service-Based 5G System Architecture", March 2018.
- [TR.29.891-3GPP] 3rd Generation Partnership Project (3GPP), "5G System ? Phase 1, CT WG4 Aspects, 3GPP TR 29.891 v15.0.0", December 2017.
- [TS.23.203-3GPP] 3rd Generation Partnership Project (3GPP), "Policy and Charging Control Architecture, 3GPP TS 23.203 v2.0.1", December 2017.
- [TS.23.501-3GPP] 3rd Generation Partnership Project (3GPP), "System ARchitecture for the 5G System; Stage 2, 3GPP TS 23.501, v15.2.0", June 2018.
- [TS.23.502-3GPP] 3rd Generation Partnership Project (3GPP), "Procedures for 5G System; Stage 2, 3GPP TS 23.502, v15.2.0", June 2018.

- [TS.23.503-3GPP]  
3rd Generation Partnership Project (3GPP), "Policy and Charging Control System for 5G Framework; Stage 2, 3GPP TS 23.503 v15.2.0", June 2018.
- [TS.29.244-3GPP]  
3rd Generation Partnership Project (3GPP), "Interface between the Control Plane and the User Plane Nodes; Stage 3, 3GPP TS 29.244 v15.2.0", June 2018.
- [TS.29.281-3GPP]  
3rd Generation Partnership Project (3GPP), "GPRS Tunneling Protocol User Plane (GTPv1-U), 3GPP TS 29.281 v15.3.0", June 2018.
- [TS.38.300-3GPP]  
3rd Generation Partnership Project (3GPP), "NR and NG-RAN Overall Description: Stage 2, 3GPP TS 38.300 v15.2.0", June 2018.
- [TS.38.401-3GPP]  
3rd Generation Partnership Project (3GPP), "NG-RAN: Architecture Description, 3GPP TS 38.401 v15.2.0", June 2018.
- [TS.38.801-3GPP]  
3rd Generation Partnership Project (3GPP), "Study on new radio access technology: Radio access architecture and interfaces", March 2017.
- [WLDR] Carofiglio, G., Muscariello, L., Papalini, M., Rozhnova, N., and X. Zeng, "Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks", Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking - ACM-ICN '16, DOI 10.1145/2984356.2984361, 2016.

## Authors' Addresses

Kalyani Bogineni  
Verizon

Email: kalyani.bogineni@verizon.com

Arashmid Akhavain  
Huawei Canada Research Centre  
Email: arashmid.akhavain@huawei.com

Tom Herbert  
Quantonium  
Email: tom@quantonium.net

Dino Farinacci  
lispers.net  
Email: farinacci@gmail.com

Alberto Rodriguez-Natal  
Cisco Systems, Inc.  
Email: natal@cisco.com

Giovanna Carofiglio  
Cisco Systems, Inc.  
Email: gcarofig@cisco.com

Jordan Auge  
Cisco Systems, Inc.  
Email: jordan.auge@cisco.com

Luca Muscariello  
Cisco Systems, Inc.  
Email: lumuscar@cisco.com

Pablo Camarillo Garvia  
Cisco Systems, Inc.  
Email: pcamaril@cisco.com

Shunsuke Homma  
NTT

Email: [homma.shunsuke@lab.ntt.co.jp](mailto:homma.shunsuke@lab.ntt.co.jp)

DMM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 26, 2019

P. Camarillo  
C. Filsfils  
Cisco Systems, Inc.  
L. Bertz  
Sprint  
A. Akhavan  
Huawei Canada Research Centre  
S. Matsushima  
SoftBank  
D. Voyer  
Bell Canada  
April 24, 2019

Segment Routing IPv6 for mobile user-plane PoCs  
draft-camarillo-dmm-srv6-mobile-pocs-02

Abstract

This document describes the ongoing proof of concepts of [I-D.ietf-dmm-srv6-mobile-uplane] and their progress.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 26, 2019.

Copyright Notice

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

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

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

## Table of Contents

|                                                                       |   |
|-----------------------------------------------------------------------|---|
| 1. Introduction . . . . .                                             | 2 |
| 2. Terminology . . . . .                                              | 2 |
| 3. M-CORD C3PO . . . . .                                              | 3 |
| 3.1. PoC phases . . . . .                                             | 3 |
| 3.2. Activity report . . . . .                                        | 3 |
| 3.2.1. Phase 1 . . . . .                                              | 3 |
| 4. Open Air Interface . . . . .                                       | 4 |
| 4.1. PoC phases . . . . .                                             | 4 |
| 4.1.1. Phase 1: Mobile Core Migration from IPv4-GTP to SRv6 . . . . . | 5 |
| 4.2. Activity report . . . . .                                        | 6 |
| 5. Contributors . . . . .                                             | 6 |
| 6. Informative References . . . . .                                   | 7 |
| Authors' Addresses . . . . .                                          | 7 |

## 1. Introduction

The [I-D.ietf-dmm-srv6-mobile-uplane] proposes SRv6 as userplane protocol for mobile networks. As part of this work we have decided to create a series of PoCs with the objective to prove the viability and feasibility of such proposal.

For this reason we have two ongoing PoCs using M-CORD C3PO and OAI, that are progressing towards a full implementation of the mechanisms described in such I-D.

This I-D contains a formal definition of the PoCs and will summarize it's findings. Anyone interested in participating in the ongoing PoCs or propose new ones is welcome to join us.

## 2. Terminology

This document adopts the terminology of [I-D.ietf-dmm-srv6-mobile-uplane].

This document uses the terms N3, N6 and N9 interfaces, as well as UPF and gNB as referred to in [TS.23501].

### 3. M-CORD C3PO

M-CORD <<https://www.opennetworking.org/m-cord/>> is an open-source project from ONF focused on building a cloud-native virtualized and disaggregated RAN and EPC.

As part of the M-CORD project, the C3PO component is part of the NGIC (Next Generation Infrastructure Core) <<https://gerrit.opencord.org/#/admin/projects/ngic>>.

The scope of this PoC is to extend the C3PO component to support natively SRv6 on the N6 and N9 interfaces and have SRv6-supported UPFs.

#### 3.1. PoC phases

This PoC is divided in several phases:

1. SRv6 in transport network with no impact to EPC
2. SRv6 native in N6 interface (GiLAN) with SRv6 transport network
3. SRv6 native in N6 and N9 interfaces with N3 interworking mechanisms

#### 3.2. Activity report

Phase 1 has been completed. Ongoing development of phase 2.

##### 3.2.1. Phase 1

We used FD.io VPP <<https://fd.io/technology/>> to simulate an SRv6 transport network with three SRv6 routers in the N9 interface simulating a transport network.

As part of this transport network, we run two simulations:

In the first simulation we steered the IPv4/GTP traffic into an SR policy that encapsulated the packet with an SRv6 header containing two SIDs.

In the second simulation we steered the IPv4/GTP traffic into an SR policy that removed the IPv4/GTP headers and placed the GTP header information (i.e. TEID) into an SRv6 SID. The last SID of the SR policy corresponds to an End.M.GTP4.E function, that decapsulates SRv6 traffic restoring the IPv4/GTP header. The objective of the second simulation is to show the IPv4/GTP interworking mechanism via an uplink classifier behaving as SR-GW, as defined in Section 6.4 of [I-D.ietf-dmm-srv6-mobile-uplane] .



After Phase 1, we concluded that SRv6 as mobility transport network works fine, with an expected MTU overhead due to the original PDU encapsulation. The IPv4/GTP interworking mechanism in the scope of phase 1 is also fully functional. This mechanism will be further tested as the POC progresses and a native SRv6-based UPF is developed.

#### 4. Open Air Interface

Open Air Interface (OAI) is an open-source software <[http://www.openairinterface.org/?page\\_id=2762](http://www.openairinterface.org/?page_id=2762)> that implements the 3GPP stack. OAI is composed of two major projects: OAI-RAN and OAI-CN.

- o OAI-RAN implements the 4G LTE and 5G Radio Access Network. Both the gNB as well as the UE are implemented.
- o OAI-Core Network implements the 4G LTE Evolved Packet Core (EPC) and 5G Core Network.

The scope of this PoC is to extend the OAI-RAN and OAI-CN components to support natively SRv6 on the N3 and N9 interfaces, and have SRv6-supported gNBs and UPFs.

##### 4.1. PoC phases

The primary goal of this POC is to show SRv6 as a data plane replacement for GTP on both N3 and N9 interfaces. The POC also aims to demonstrate a smooth migration path during deployment and transition period from IPv4-GTP and IPv6-GTP to an end to end SRv6 data plane.

The PoC functions within the existing OAI model. OAI currently doesn't provide support for S5/S8 interface. The implementation instead provides an integrated SGW and PGW S/PGW module and therefore there is no GTP tunnel between these two entities. This limitation has an impact on the POC strategy and its implementation phases.

This PoC is divided into several phases:

- 1.- N3 via SRv6 GW VNFs and no impact on 3GPP control plane.
  - 1.1.- Mobile Core Migration from IPv4-GTP to SRv6
  - 1.2.- Mixed IPv4-GTP/IPv6-GTP Mobile Core Over SRv6
- 2.- N3 via SRv6 eNB and S/PGW integrated modules and no impact on 3GPP control plane.
  - 2.1.- Mobile Core Migration from IPv4-GTP to SRv6

## 2.2.- Mixed IPv4-GTP/IPv6-GTP Mobile Core Over SRv6

### 3.- N3 via SRv6 support of ID-LOC architecture

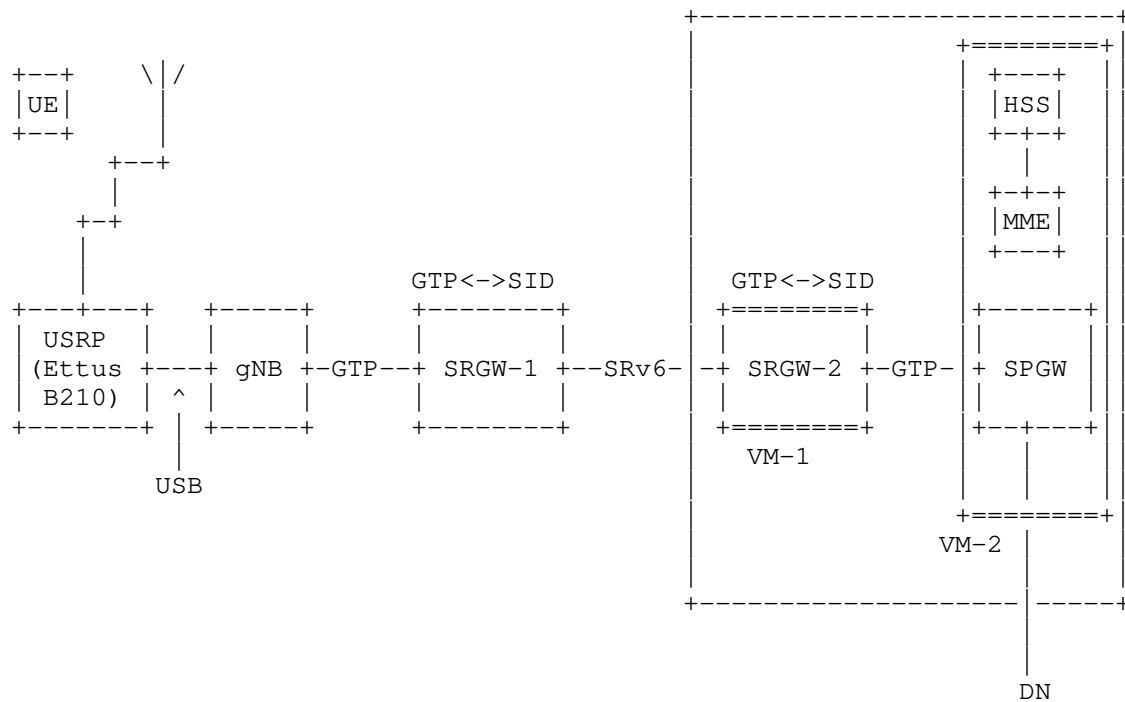
#### Important notes:

- The above phases and solution strategy can easily be extended to the N9 interface. However, although the N9 interface is well within the scope of this PoC, the effort required to change the OAI code base to support S5/S8 and separate SGW and PGW modules will push the project well beyond the timeline of this PoC and as such are not currently part of the PoC.
- Support for service programming, TE, QoS, entropy, and other enhanced features are also within the scope of this PoC, but will also fall beyond the time line of this project and are not currently considered in this PoC.
- The above items can be pulled back into the project based on demand and assistance from others.

#### 4.1.1. Phase 1: Mobile Core Migration from IPv4-GTP to SRv6

Phase one of this POC focuses on demonstrating a smooth migration path from the existing mobile core networks with IPv4 GTP based user plane to SRv6 user plane with absolutely no impact on 3GPP control plane. The idea is to employ SRv6 gateways between mobile core equipment such as eNB, SGW, and PGW, intercept GTP traffic, and carry UE's payload through SRv6 network by encoding GTP information into the SIDs.

In this POC as it was mentioned earlier we use OAI open source software. OAI implements gNB as a stand alone entity, but bundles MME, SGW and PGW into a single package. We employ three Linux PCs in our setup. Two of these machines run the gNB and one of the SRv6 GWs. The third machine employs virtualisation and instantiates two virtual machines. The second SRv6 gateway runs in one of the virtual machines while the other virtual machines executes the code for the combined MME, SGW, PGW. The code in SRv6 gateways is based on VPP implementation in Linux Foundation. We modified this code to intercept GTP packets, extract GTP information, and encode GTP information into the SIDs. Given that today's mobile core don't deal with multiple UPFs, the resulting SRv6 header doesn't require any SRH to carry GTP information across the network. Therefore, in this phase, the resulting SRv6 packets are simply IPv6 packets with their DA set to SIDs. The following diagram shows the POC configuration.



#### POC Configuration

In this implementation, the SRGW at one end extracts relevant GTP information (SA, DA, TEID) from GTP and encodes them into the lower 96 bits of SID. The SID is then copied into the DA of IPv6 header and the packet is forwarded toward the SRGW at the far end. Receiving the SRv6 packet, the far end SRGW recognizes the SID as local and executes a set of functions that extracts GTP information from the SID, forms the GTP packet by adding relevant UDP and GTP headers and forwards this reconstructed GTP packet to its associated mobile core node.

#### 4.2. Activity report

Development started. Phase 1 has been completed.

#### 5. Contributors

Chenchen Liu  
Huawei Technologies Co., Ltd.  
Shenzhen, China

Email: liuchencheng1@huawei.com

Arun Rajagopal  
Sprint  
United States of America

Email: Arun.Rajagopal@sprint.com

Mark Bales  
Sprint  
United States of America

Email: Mark.Bales@sprint.com

Robert Butler  
Sprint  
United States of America

Email: Robert.Butler@sprint.com

## 6. Informative References

[I-D.filsfils-spring-srv6-network-programming]  
Filsfils, C., Camarillo, P., Leddy, J.,  
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6  
Network Programming", draft-filsfils-spring-srv6-network-  
programming-07 (work in progress), February 2019.

[I-D.ietf-dmm-srv6-mobile-uplane]  
Matsushima, S., Filsfils, C., Kohno, M., Camarillo, P.,  
daniel.voyer@bell.ca, d., and C. Perkins, "Segment Routing  
IPv6 for Mobile User Plane", draft-ietf-dmm-srv6-mobile-  
uplane-04 (work in progress), March 2019.

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

## Authors' Addresses

Pablo Camarillo Garvia  
Cisco Systems, Inc.  
Spain

Email: pcamaril@cisco.com

Clarence Filsfils  
Cisco Systems, Inc.  
Belgium

Email: cf@cisco.com

Lyle T Bertz  
Sprint  
United States of America

Email: Lyle.T.Bertz@sprint.com

Arashmid Akhavain  
Huawei Canada Research Centre  
Canada

Email: arashmid.akhavain@huawei.com

Satoru Matsushima  
SoftBank  
Tokyo  
Japan

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

Daniel Voyer  
Bell Canada  
Canada

Email: daniel.voyer@bell.ca

DMM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 17, 2019

U. Chunduri, Ed.  
R. Li  
Huawei USA  
J. Tantsura  
Nuage Networks  
L. Contreras  
Telefonica  
X. De Foy  
InterDigital Communications, LLC  
July 16, 2018

Transport Network aware Mobility for 5G  
draft-clt-dmm-tn-aware-mobility-01

Abstract

This document specifies a framework and a mapping function for 5G mobile user plane with transport network slicing, integrated with Mobile Radio Access and a Virtualized Core Network. The integrated approach specified in a way to address all the mobility scenarios defined in [TS23.501] and to be backward compatible with LTE [TS.23.401-3GPP] network deployments.

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

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 17, 2019.

#### Copyright Notice

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

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

#### Table of Contents

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 1. Introduction and Problem Statement . . . . .                 | 3  |
| 1.1. Acronyms . . . . .                                         | 3  |
| 1.2. Solution Approach . . . . .                                | 5  |
| 2. Transport Network (TN) and Slice aware Mobility on N3/N9 . . | 5  |
| 2.1. Discrete Approach . . . . .                                | 6  |
| 2.2. Integrated Approach . . . . .                              | 7  |
| 3. Using PPR as TN Underlay . . . . .                           | 9  |
| 3.1. PPR with Transport Slicing aware Mobility on N3/N9 . . .   | 9  |
| 3.2. Path Steering Support to native IP user planes . . . . .   | 11 |
| 3.3. Service Level Guarantee in Underlay . . . . .              | 11 |
| 3.4. PPR with various 5G Mobility procedures . . . . .          | 11 |
| 3.4.1. SSC Model . . . . .                                      | 12 |
| 3.4.2. SSC Mode2 . . . . .                                      | 13 |
| 3.4.3. SSC Mode3 . . . . .                                      | 13 |
| 4. Other TE Technologies Applicability . . . . .                | 14 |
| 5. New Control Plane and User Planes . . . . .                  | 15 |
| 5.1. LISP and PPR . . . . .                                     | 15 |
| 5.2. ILA and PPR . . . . .                                      | 15 |
| 6. Summary and Benefits with PPR . . . . .                      | 15 |
| 7. Acknowledgements . . . . .                                   | 16 |
| 8. IANA Considerations . . . . .                                | 16 |
| 9. Security Considerations . . . . .                            | 16 |
| 10. References . . . . .                                        | 16 |
| 10.1. Normative References . . . . .                            | 16 |

|                                        |    |
|----------------------------------------|----|
| 10.2. Informative References . . . . . | 16 |
| Authors' Addresses . . . . .           | 18 |

## 1. Introduction and Problem Statement

3GPP Release 15 for 5GC is defined in [TS.23.501-3GPP], [TS.23.502-3GPP], [TS.23.503-3GPP]. A new user plane interface N9 [TS.23.501-3GPP] has been created between 2 User Plane Functionalities (UPFs). While user plane for N9 interface being finalized for REL16, both GTP-U based encapsulation or any other compatible approach is being considered [CT4SID]. Concerning to this document another relevant interface is N3, which is between gNB and the UPF. N3 interface is similar to the user plane interface S1U in LTE [TS.23.401-3GPP]. This document:

- o does not propose any change to existing N3 user plane encapsulations to realize the benefits with the approach specified here
- o and can work with any encapsulation (including GTP-U) for the N9 interface.

[TS.23.501-3GPP] defines various Session and Service Continuity (SSC) modes and mobility scenarios for 5G with slice awareness from Radio and 5G Core (5GC) network. 5G System (5GS) as defined, allows transport network between N3 and N9 interfaces work independently with various IETF Traffic Engineering (TE) technologies.

However, lack of underlying Transport Network (TN) awareness can be problematic for some of the 5GS procedures, for real-time, mission-critical or for any deterministic latency services. These 5GS procedures including but not limited to Service Request, PDU Session, or User Equipment (UE) mobility need same service level characteristics from the TN for the Protocols Data Unit (PDU) session, similar to as provided in Radio and 5GC for various 5QI's defined in [TS.23.501-3GPP] .

### 1.1. Acronyms

|     |                                                |
|-----|------------------------------------------------|
| 5QI | - 5G QoS Indicator                             |
| AMF | - Access and Mobility Management Function (5G) |
| BP  | - Branch Point (5G)                            |
| CSR | - Cell Site Router                             |
| DN  | - Data Network (5G)                            |



|       |   |                                                         |
|-------|---|---------------------------------------------------------|
| eMBB  | - | enhanced Mobile Broadband (5G)                          |
| FRR   | - | Fast ReRoute                                            |
| gNB   | - | 5G NodeB                                                |
| GBR   | - | Guaranteed Bit Rate (5G)                                |
| IGP   | - | Interior Gateway Protocols (e.g. IS-IS, OSPFv2, OSPFv3) |
| LFA   | - | Loop Free Alternatives (IP FRR)                         |
| mIOT  | - | Massive IOT (5G)                                        |
| MPLS  | - | Multi Protocol Label Switching                          |
| QFI   | - | QoS Flow ID (5G)                                        |
| PPR   | - | Preferred Path Routing                                  |
| PDU   | - | Protocol Data Unit (5G)                                 |
| PW    | - | Pseudo Wire                                             |
| RQI   | - | Reflective QoS Indicator (5G)                           |
| SBI   | - | Service Based Interface (5G)                            |
| SID   | - | Segment Identifier                                      |
| SMF   | - | Session Management Function (5G)                        |
| SSC   | - | Session and Service Continuity (5G)                     |
| SST   | - | Slice and Service Types (5G)                            |
| SR    | - | Segment Routing                                         |
| TE    | - | Traffic Engineering                                     |
| ULCL  | - | Uplink Classifier (5G)                                  |
| UPF   | - | User Plane Function (5G)                                |
| URLLC | - | Ultra reliable and low latency communications (5G)      |



component Transport Network Function (TNF). A Cell Site Router (CSR) is shown connecting to gNB. Though it is shown as a separate block from gNB, in some cases both of these can be co-located. This document concerns with backhaul TN, from CSR to UPF on N3 interface or from Staging UPF to Anchor UPF on N9 interface.

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

N3 interface is similar to S1U in 4G/LTE [TS.23.401-3GPP] network and uses GTP-U [TS.29.281-3GPP] encapsulation to transport any UE PDUs (IPv4, IPv6, IPv4v6, Ethernet or Unstructured). N9 interface is a new interface to connect UPFs in SSC Mode3 Section 3.4.3 and right user plane protocol/encapsulation is being studied through 3GPP CT4 WG approved study item [CT4SID] for REL-16.

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

## 2.1. Discrete Approach

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

In this case, the encapsulation provided by GTP-U helps carry different PDU session types (IPv4, IPv6, IPv4IPv6, Ethernet and Unstructured) independent of the underlying transport or user plane (IPv4, IPv6 or MPLS) network. Mapping of the PDU sessions to TE paths can be done based on the source UDP port ranges (if these are assigned based on the PDU session QCI, as done in some deployments with 4G/LT) of the GTP-U encapsulated packet or based on the 5QI or RQI values in the GTP-U header. Here, TNF as shown in Figure 1 need

not be part of the 5G Service Based Interface (SBI). Only management plane functionality is needed to create, monitor, manage and delete (life cycle management) the transport TE paths/transport slices from gNB to UPF (on N3/N9 interfaces). This approach provide partial integration of the transport network into 5GS with some benefits.

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

## 2.2. Integrated Approach

Network Slice Selection Function (NSSF) as defined in [TS.23.501-3GPP] concerns with multiple aspects related to creation, selection, mobility, roaming and co-ordination among other CP functions in 5GS. However, the scope is only in 5GC (both control and user plane) and NG Radio Access network including N3IWF for non-3GPP access. Slice functionality is per PDU session granularity. While this fully covers needed functionality and resources from UE registration, Tracking Area (TA) updates, mobility and roaming, resources and functionalities needed from transport network is not specified. This is seen as independent functionality though part of 5GS. If transport network is not factored in an integrated fashion w.r.t available resources (with network characteristics from desired bandwidth, latency, burst size handling and optionally jitter) some of the gains made with optimizations through 5G NR and 5GC can be degraded.

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

A south bound interface Ns is shown which interacts with the gNB/CSR. 'Ns' can use one or more mechanism available today (PCEP [RFC5440], NETCONF [RFC6241], RESTCONF [RFC8040] or gNMI) to provision the L2/L3 VPNs along with TE underlay paths from gNB to UPF.

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

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

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

- o In the Network Slice Selection Assistance Information (NSSAI) PDU session's assigned transport VPN and the TE path information is needed.
- o For transport slice assignment for various SSTs (eMBB, URLLC, MIoT) corresponding underlay paths need to be created and monitored from each transport end point (gNB/CSR and UPF).
- o During PDU session creation, apart from radio and 5GC resources, transport network resources needed to be verified matching the characteristics of the PDU session traffic type.
- o Mapping of PDU session parameters to underlay SST paths need to be done. One way to do this is through 5QI/QFI information in the GTP-U header and map the same to the underlying transport path (including VPN or PW). This works for uplink (UL) direction.
- o For downlink direction RQI need to be considered to map the DL packet to one of the underlay paths at the UPF.
- o If any other form of encapsulation (other than GTP-U) either on N3 or N9 corresponding 5QI/QFI or RQI information MUST be there in the encapsulation header.
- o If SSC Mode3 Section 3.4.3 is used, segmented path (gNB to staging/ULCL/BP-UPF to anchor-point-UPF) with corresponding path characteristics MUST be used. This includes a path from gNB/CSR to UL-CL/BP UPF [TS.23.501-3GPP] and UL-CL/BP UPF to eventual UPF access to DN.
- o Continuous monitoring of transport path characteristics and reassignment at the endpoints MUST be performed. For all the effected PDU sessions, degraded transport paths need to be updated dynamically with similar alternate paths.
- o During UE mobility event similar to 4G/LTE i.e., gNB mobility (Xn based or N2 based), for target gNB selection, apart from radio resources, transport resources MUST be factored. This enables

handling of all PDU sessions from the UE to target gNB and this require co-ordination of AMF, SMF with the TNF module.

Changes to detailed signaling to integrate the above for various 5GS procedures as defined in [TS.23.502-3GPP] is beyond the scope of this document.

### 3. Using PPR as TN Underlay

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

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

Details of the IGP extensions for PPR are provided here:

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

#### 3.1. PPR with Transport Slicing aware Mobility on N3/N9

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

- o For 3 different SSTs, 3 PPR-IDs can signaled from any node in the transport network. For Uplink traffic, gNB will choose the right PPR-ID of the UPF based on the 5QI value in the encapsulation header of the PDU session. Similarly in the Downlink direction matching PPR-ID of the gNB is chosen for the RQI value in the encapsulated SL payload. The table below shows a typical mapping:

| 5QI (Ranges)/<br>RQI (Ranges)               | SST                             | Transport Path<br>Info      | Transport Path<br>Characteristics                                        |
|---------------------------------------------|---------------------------------|-----------------------------|--------------------------------------------------------------------------|
| Range Xx - Xy<br>X1, X2(discrete<br>values) | MIOT<br>(massive<br>IOT)        | PW ID/VPN info,<br>PPR-ID-A | GBR (Guaranteed<br>Bit Rate)<br>Bandwidth: Bx<br>Delay: Dx<br>Jitter: Jx |
| Range Yx - Yy<br>Y1, Y2(discrete<br>values) | URLLC<br>(ultra-low<br>latency) | PW ID/VPN info,<br>PPR-ID-B | GBR with Delay<br>Req.<br>Bandwidth: By<br>Delay: Dy<br>Jitter: Jy       |
| Range Zx - Zy<br>Z1, Z2(discrete<br>values) | EMBB<br>(broadband)             | PW ID/VPN info,<br>PPR-ID-C | Non-GBR<br>Bandwidth: Bx                                                 |

Figure 2: 5QI/RQI Mapping with PPR-IDs on N3/N9

- o It is possible to have a single PPR-ID for multiple input points through a PPR tree structure separate in UL and DL direction.
- o Same set of PPRs are created uniformly across all needed gNBs and UPFs to allow various mobility scenarios.
- o Any modification of TE parameters of the path, replacement path and deleted path needed to be updated from TNF to the relevant ingress points. Same information can be pushed to the NSSF, AMF and SMF as needed.
- o PPR can be supported with any native IPv4 and IPv6 data/user planes (Section 3.2 with optional TE features Section 3.3 . As

this is an underlay mechanism it can work with any overlay encapsulation approach including GTP-U as defined currently for N3 interface.

### 3.2. Path Steering Support to native IP user planes

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

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

### 3.3. Service Level Guarantee in Underlay

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

### 3.4. PPR with various 5G Mobility procedures

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

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



## 3.4.1.1. SSC Model

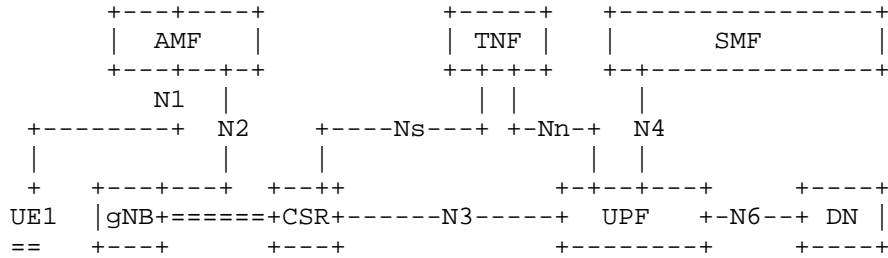


Figure 3: SSC Model with integrated Transport Slice Function

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

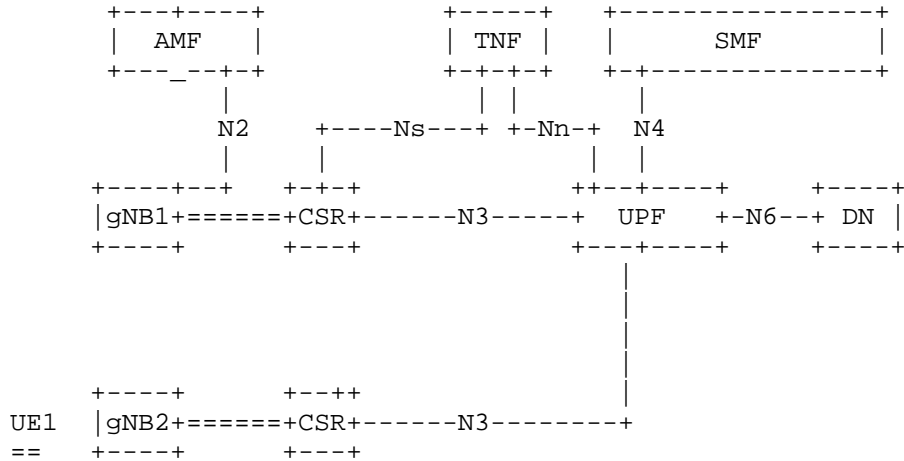


Figure 4: SSC Model with integrated Transport Slice Function

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

## 3.4.2.    SSC Mode2

In this case, if IP Address is changed during mobility (different UPF area), then corresponding PDU session is released. No session continuity from the network is provided and this is designed as an application offload and application manages the session continuity, if needed. For PDU Session, Service Request and Mobility cases mechanism to select the transport resource and the PPR-ID (TE Path) is similar to SSC Model.

## 3.4.3.    SSC Mode3

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

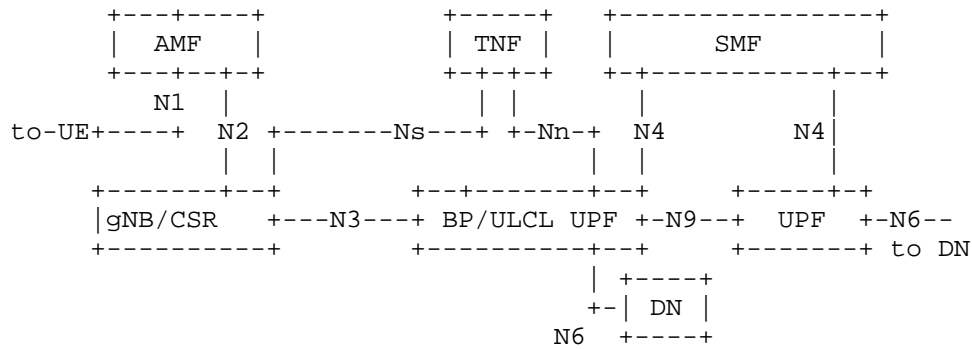


Figure 5: SSC Mode3 and Service Continuity

In the uplink direction for the traffic offloading from UL/CL case, packet has to reach to the right exit UPF. In this case packet gets re-encapsulated with ULCL marker (with either GTP-U or the chosen encapsulation) after bit rate enforcement and LI to the anchor UPF. At this point packet has to be on the appropriate VPN/PW to the anchor UPF. This mapping is done based on the 5QI to the PPR-ID of the exit node by selecting the respective TE PPR-ID (PPR path) of the UPF. If it's a non-MPLS underlay, destination IP address of the encapsulation header would be the mapped PPR-ID (TE path).

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

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

#### 4. Other TE Technologies Applicability

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

SR-TE [I-D.ietf-spring-segment-routing] does not explicitly signal neither bandwidth reservation nor mechanism to guarantee latency on the nodes/links on SR path. But, SR allows path steering for any flow at the ingress and particular path for a flow can be chosen. Some of the issues around path overhead/tax, MTU issues are documented at Section 5.3 of [I-D.bogineni-dmm-optimized-mobile-user-plane]. Also SR allows reduction of the control protocols to one IGP (with out needing for LDP and RSVP).

However, as specified above with PPR (Section 3), in the integrated transport network function (TNF) a particular RSVP-TE path for MPLS or SR path for MPLS and IPv6 with SRH user plane, can be supplied to NSSF/AMF/SMF for mapping a particular PDU session to the transport path.

## 5. New Control Plane and User Planes

### 5.1. LISP and PPR

PPR can also be used with LISP control plane for 3GPP as described in [I-D.farinacci-lisp-mobile-network]. This can be achieved by mapping the UE IP address (EID) to PPR-ID, which acts as Routing Locator (RLOC). Any encapsulation supported by LISP can work well with PPR. If the RLOC refers to an IPv4 or IPv6 destination address in the LISP encapsulated header, packets are transported on the preferred path in the network as opposed to traditional shortest path routing with no additional user plane overhead related to TE path in the network layer.

Some of the distinct advantages of the LISP approach is, its scalability, support for service continuity in SSC Mode3 as well as native support for session continuity (session survivable mobility). Various other advantages are documented at [I-D.farinacci-lisp-mobile-network].

### 5.2. ILA and PPR

If an ILA-prefix is allowed to refer to a PPR-ID, ILA can be leveraged with all the benefits (including mobility) that it provides. This works fine in the DL direction as packet is destined to UE IP address at UPF. However, in the UL direction, packet is destined to an external internet address (SIR Prefix to ILA Prefix transformation happens on the Source address of the original UE packet). One way to route the packet with out bringing the complete DFZ BGP routing table is by doing a default route to the UPF (ILA-R). In this case, how TE can be achieved is TBD (to be expanded further with details).

## 6. Summary and Benefits with PPR

This document specifies an approach to transport and slice aware mobility with a simple mapping function from PDU Session to transport path applicable to any TE underlay.

This also describes PPR [I-D.chunduri-lsr-isis-preferred-path-routing], a transport underlay routing mechanism, which helps with goal of optimized user plane for N9 interface. PPR provides a method for N3 and N9 interfaces to support transport slicing in a way which does not erase the gains made with 5G NR and virtualized cellular core network features for various types of 5G traffic (e.g. needing low and deterministic latency, real-time, mission-critical or AR/VR traffic). PPR provides path steering, optionally guaranteed services with TE, unique Fast-

ReRoute (FRR) mechanism with preferred backups (beyond shortest path backups through existing LFA schemes) in the mobile backhaul network with any underlay being used in the operator's network (IPv4, IPv6 or MPLS) in an optimized fashion.

## 7. Acknowledgements

TBD.

## 8. IANA Considerations

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

## 9. Security Considerations

This document does not introduce any new security issues.

## 10. References

### 10.1. 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>>.

### 10.2. Informative References

[I-D.bashandy-rtgwg-segment-routing-ti-lfa]  
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., and d. daniel.voyer@bell.ca, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-04 (work in progress), April 2018.

[I-D.bogineni-dmm-optimized-mobile-user-plane]  
Bogineni, K., Akhavain, A., Herbert, T., Farinacci, D., Rodriguez-Natal, A., Carofiglio, G., Auge, J., Muscariello, L., Camarillo, P., and S. Homma, "Optimized Mobile User Plane Solutions for 5G", draft-bogineni-dmm-optimized-mobile-user-plane-01 (work in progress), June 2018.

[I-D.chunduri-lsr-isis-preferred-path-routing]  
Chunduri, U., Li, R., White, R., Tantsura, J., Contreras, L., and Y. Qu, "Preferred Path Routing (PPR) in IS-IS", draft-chunduri-lsr-isis-preferred-path-routing-01 (work in progress), July 2018.

- [I-D.chunduri-lsr-ospf-preferred-path-routing]  
Chunduri, U., Qu, Y., White, R., Tantsura, J., and L. Contreras, "Preferred Path Routing (PPR) in OSPF", draft-chunduri-lsr-ospf-preferred-path-routing-01 (work in progress), July 2018.
- [I-D.farinacci-lisp-mobile-network]  
Farinacci, D., Pillay-Esnault, P., and U. Chunduri, "LISP for the Mobile Network", draft-farinacci-lisp-mobile-network-03 (work in progress), March 2018.
- [I-D.ietf-dmm-srv6-mobile-uplane]  
Matsushima, S., Filsfils, C., Kohno, M., Camarillo, P., daniel.voyer@bell.ca, d., and C. Perkins, "Segment Routing IPv6 for Mobile User Plane", draft-ietf-dmm-srv6-mobile-uplane-02 (work in progress), July 2018.
- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <<https://www.rfc-editor.org/info/rfc6830>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8370] Beeram, V., Ed., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP-TE Deployments", RFC 8370, DOI 10.17487/RFC8370, May 2018, <<https://www.rfc-editor.org/info/rfc8370>>.
- [TS.23.401-3GPP]  
3rd Generation Partnership Project (3GPP), "Procedures for 4G/LTE System; 3GPP TS 23.401, v15.4.0", June 2018.
- [TS.23.501-3GPP]  
3rd Generation Partnership Project (3GPP), "System Architecture for 5G System; Stage 2, 3GPP TS 23.501 v2.0.1", December 2017.
- [TS.23.502-3GPP]  
3rd Generation Partnership Project (3GPP), "Procedures for 5G System; Stage 2, 3GPP TS 23.502, v2.0.0", December 2017.
- [TS.23.503-3GPP]  
3rd Generation Partnership Project (3GPP), "Policy and Charging Control System for 5G Framework; Stage 2, 3GPP TS 23.503 v1.0.0", December 2017.
- [TS.29.281-3GPP]  
3rd Generation Partnership Project (3GPP), "GPRS Tunneling Protocol User Plane (GTPv1-U), 3GPP TS 29.281 v15.1.0", December 2017.

## Authors' Addresses

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

Email: [uma.chunduri@huawei.com](mailto:uma.chunduri@huawei.com)

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

Email: renwei.li@huawei.com

Jeff Tantsura  
Nuage Networks  
755 Ravendale Drive  
Mountain View, CA 94043  
USA

Email: jefftant.ietf@gmail.com

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

Email: luismiguel.contrerasmurillo@telefonica.com

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

Email: Xavier.Defoy@InterDigital.com



DMM WG  
Internet-Draft  
Intended status: Standards Track  
Expires: March 23, 2019

S. Gundavelli  
Cisco  
M. Liebsch  
NEC  
S. Matsushima  
SoftBank  
September 19, 2018

Mobility-aware Floating Anchor (MFA)  
draft-gundavelli-dmm-mfa-01.txt

Abstract

IP mobility protocols are designed to allow a mobile node to remain reachable while moving around in the network. The currently deployed mobility management protocols are anchor-based approaches, where a mobile node's IP sessions are anchored on a central node. The mobile node's IP traffic enters and exits from this anchor node and it remains as the control point for all subscriber services. This architecture based on fixed IP anchors comes with some complexity and there is some interest from the mobile operators to eliminate the use of fixed anchors, and other residual elements such as the overlay tunneling that come with it.

This document describes a new approach for realizing a mobile user-plane that does not require fixed IP anchors. The architectural-basis for this approach is the separation of control and user plane, and the use of programmability constructs of the user-plane for traffic steering. This approach is referred to as, Mobility-aware Floating Anchor (MFA).

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 <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 March 23, 2019.

## Copyright Notice

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

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

## Table of Contents

|                                                               |    |
|---------------------------------------------------------------|----|
| 1. Introduction . . . . .                                     | 3  |
| 2. Conventions and Terminology . . . . .                      | 4  |
| 2.1. Conventions . . . . .                                    | 4  |
| 2.2. Terminology . . . . .                                    | 4  |
| 3. Overview . . . . .                                         | 6  |
| 3.1. The Network Topology Database . . . . .                  | 9  |
| 3.2. The Node Location Database . . . . .                     | 9  |
| 3.3. Determination of the Correspondent Node Anchor . . . . . | 10 |
| 3.4. Traffic Steering Approaches . . . . .                    | 10 |
| 3.5. Mobile Node Attachment Triggers . . . . .                | 12 |
| 3.6. Programming the User-plane . . . . .                     | 12 |
| 4. Life of a Mobile Node in a MFA Domain . . . . .            | 14 |
| 4.1. MN's Initial Attachment to a MFA Domain . . . . .        | 15 |
| 4.2. MN's Roaming within the MFA Domain . . . . .             | 17 |
| 4.3. Traffic Steering State Removal . . . . .                 | 20 |
| 4.4. Mobile Node's new IP flows . . . . .                     | 21 |
| 5. MFA in 5G System Architecture . . . . .                    | 21 |
| 6. IANA Considerations . . . . .                              | 23 |
| 7. Security Considerations . . . . .                          | 23 |
| 8. Acknowledgements . . . . .                                 | 23 |
| 9. References . . . . .                                       | 23 |
| 9.1. Normative References . . . . .                           | 23 |
| 9.2. Informative References . . . . .                         | 23 |
| Authors' Addresses . . . . .                                  | 24 |

## 1. Introduction

IP mobility protocols are designed to allow a mobile node to remain reachable while moving around in the network. The currently deployed mobility management protocols are anchor-based approaches, where a mobile node's IP sessions are anchored on a central node. The mobile node's IP traffic enters and exits from this anchor node and it remains as the control point for all subscriber services. This architecture based on fixed IP anchors comes with some complexity and there is some interest from the mobile operators to eliminate the use of fixed anchors, and other residual elements such as the overlay tunneling that come with it. Some of the key objectives for this effort are listed below.

- o Access-agnostic, shared user-plane that can be used for multiple access technologies
- o Optimized Routing for the mobile node's IP flows with topology awareness and leveraging the transport QoS
- o Elimination of overlay tunnels from the user-plane network for avoiding packet fragmentation, and reducing encapsulation related packet-size overhead
- o Elimination of centralized mobility anchors and shift towards a distributed mobility architecture, leveraging the edge compute at radio-access network for offloading some of the subscriber management services
- o Co-existence with control-plane and user-plane separated architecture; a stateless user-plane with no tunnels, and a control plane with the business/service logic
- o Support for services including accounting, charging, lawful-interception and other user plane services

Currently, there is a study item in 3GPP to explore options for simplifying the mobile user-plane. There are few proposals in IETF, which are presented as candidate solutions for user-plane simplification. However, each of these proposals come with certain complexity and do not leverage the 3GPP control plane, or the programmability aspects of the user-plane. For example, ILA defines a translation scheme without the need for overlay tunnels, but it also introduces significant amount of translation related state in the user-plane, and additionally introduces a new control-plane protocol for managing the mapping tables and the cache states. Therefore, we believe that none of the currently known approaches can adequately meet the stated goals for user-plane simplification.

This document describes a new approach for realizing a mobile user-plane that does not require any fixed IP anchors. The first-hop router on the link where the mobile node is attached remains as the IP anchor and thereby eliminating the need for IP tunneling to some central anchor node. Even when the mobile node moves in the network and changes its point of attachment, the IP anchor is always the first-hop router on that new link. The MFA entities will track the mobile node's movements in the network and will ensure the mobile node's IP flows always take the most optimal routing path. This is achieved by MFA entities programming the needed traffic steering rules for moving mobile node's IP packets directly between the correspondent node and the mobile node's edge anchor, which can be relocated to a new edge, e.g. in case of mobility. Furthermore, this approach does not require a new control-plane protocol, but instead leverages the SDN interfaces of the user-plane, and the mobility events in the control-plane for managing IP mobility. The architectural basis for this approach is the separation of control and user plane, and the use of programmability constructs of the user-plane for traffic steering. This approach is referred to as, Mobility-aware Floating Anchor (MFA). The rest of the document explains the operational details of the MFA approach.

## 2. Conventions and Terminology

### 2.1. Conventions

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 RFC 2119 [RFC2119].

### 2.2. Terminology

All mobility related terms used in this document are to be interpreted as defined in the IETF mobility specifications, including [RFC5213] and [RFC6275]. Additionally, this document uses the following terms:

#### MFA Domain

MFA domain refers to the network where the mobility management of a mobile node is handled by the MFA entities. The MFA domain includes MFA mobile node anchors, MFA corresponding node anchors, and MFA node controller, between which security associations can be set up for authorizing the configuration of traffic steering policies and other mobility management functions.

#### MFA Mobile Node Anchor (MFA-MNA)

Its an MFA function located in the user-plane network very close to the layer-2 access-point to where the mobile node is attached. It is typically on the first-hop router for the mobile node's IP traffic. The node hosting this function is required to support the standard IPv6 packet forwarding function, FPC or a similar interface for policy configuration, and packet steering functions such as based on SRv6 or alternative means that can support per-flow or per-flow-aggregate traffic steering. Typically, the MFA-MNA function will be collocated with the User Plane Function (UPF) in the 3GPP 5G system architecture.

#### MFA Corresponding Node Anchor (MFA-CNA)

Its an MFA function located in the user-plane node in the path between the mobile node and the correspondent node. If the correspondent node is another mobile node in the MFA domain, then the MFA-CNA is on the first hop router on the link shared with the correspondent node. The node hosting this function is required to support the standard IPv6 packet forwarding function, FPC or a similar interface for policy configuration, and packet steering functions such as based on SRv6 or alternative means that can support per-flow or per-flow-aggregate traffic steering. Typically, the MFA-CNA function will be collocated with the IP forwarding nodes on the N6 interface of the 3GPP 5G system architecture.

#### MFA Node

A generic term used for referring to MFA-MNA, or the MFA-CNA.

#### MFA Node-Controller (MFA-NC)

The is the function that controls the forwarding policies on the MFA-MNA and MFA-CNA nodes. This entity interfaces with the MFA node using the FPC interface [I-D.ietf-dmm-fpc-cdp], or a similar interface that support user-plane policy configuration. This is typically co-located with the SMF, or the AMF functions in the 3GPP 5G system architecture, and on WLAN controller in the case of Wi-Fi access architectures.

#### Node Location Database (NLDB)

A database that contains the location information of every mobile node that is part of the MFA domain and is currently attached to the network.

#### Network Topology Database (NTDB)

A database that contains the MFA node information along with the link state and directly connected neighbor information.

#### Home Network Prefix (HNP)

An IPv6 prefix assigned to the mobile node. This prefix is hosted by the MFA-MNA on the access link shared with the mobile node. The network will provide mobility support for the HNP prefixes. A meta-data tag indicating the mobility property [I-D.ietf-dmm-ondemand-mobility] is included in router advertisements and in address assignment related protocol messages.

#### Local Network Prefix (LNP)

An IPv6 prefix assigned to the mobile node. This prefix is hosted by the MFA-MNA on the access link shared with the mobile node. The network will not provide mobility support for the LNP prefixes. A meta-data tag indicating that there is no mobility support [I-D.ietf-dmm-ondemand-mobility] is included in router advertisements and in address assignment related protocol messages.

### 3. Overview

This specification describes the MFA protocol. The MFA protocol is designed for providing mobility management support to a mobile node without the need for a fixed IP anchor. In this approach the mobile node's IP session is always anchored on the first-hop router sharing the link with the mobile node. The entities in the MFA domain track the mobile node's movements in the MFA domain and will provision the forwarding states in the user-plane nodes for optimal routing and for ensuring the anchor is always the first-hop router. Any time the mobile node moves within the MFA domain and resulting in the mobile node's IP flows going through the previous anchor, the mobility entities detect this event and a corrective action is taken by provisioning the forwarding nodes with the path stitching rules. The result of this approach is an user-plane with no fixed anchors, and dynamically programmed user-plane for mobility and optimal packet routing.

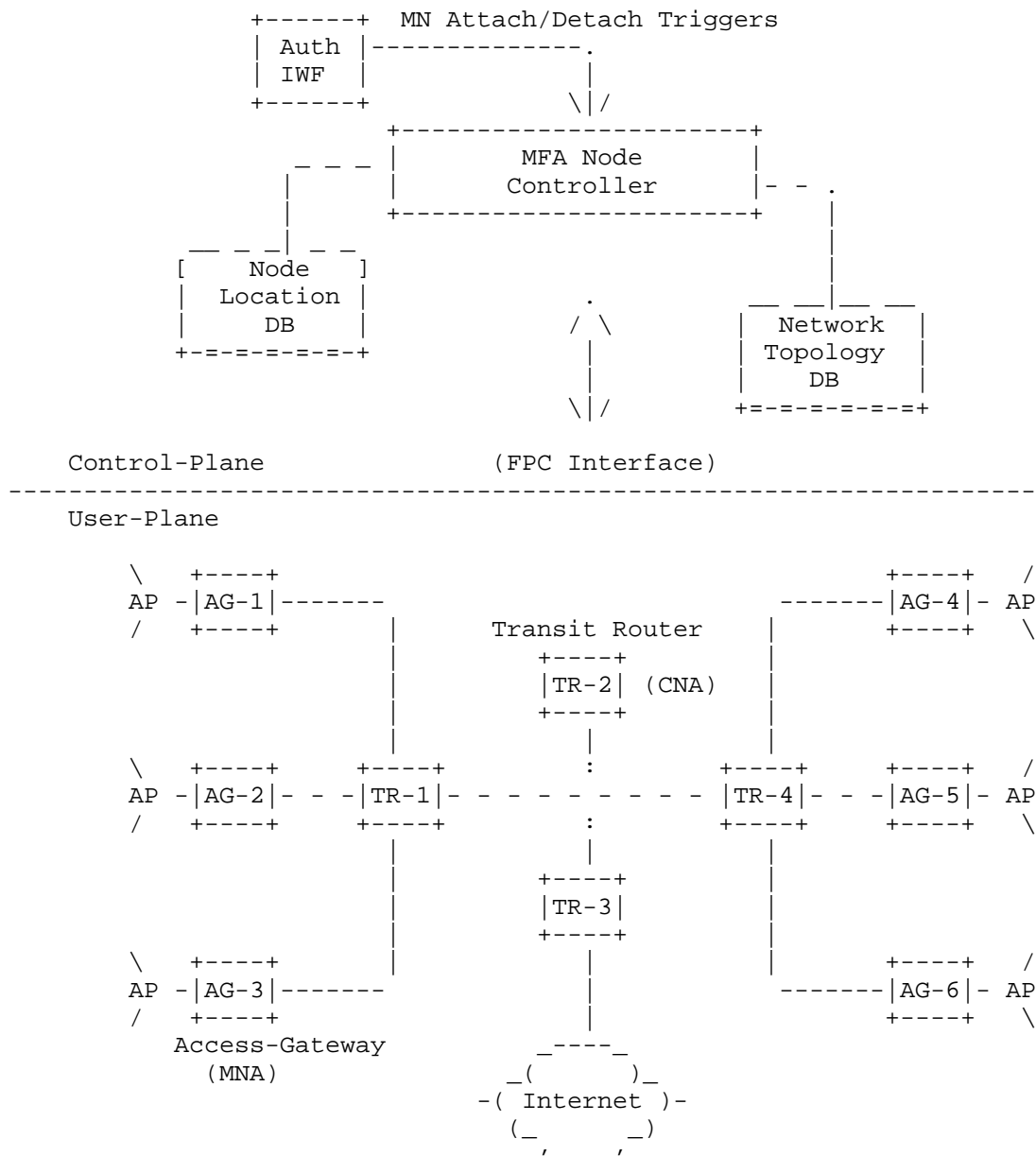
The following are the key functional entities in the MFA domain:

- o MFA Node Controller (MFA-NC)
- o MFA Mobile Node Anchor (MFA-MNA)

- o MFA Correspondent Node anchor (MFA-CNA)

The MFA-NC is typically collocated with the access network specific control-plane functions. It interfaces with the radio network/ authentication functions for detecting the mobile node's movements in the MFA domain for managing the forwarding states in the user-plane entities, MFA-MNA and MFA-CNA. The MFA node controller requires access to node location database and network topology database. These are the conceptual entities that can be realized using existing elements that are already present in different access architectures.

The MFA-MNA and the MFA-CNA are the functions in the user-plane network and they are collocated with the elements in the network that perform IP packet forwarding functions. The MFA-MNA is typically located on the first-hop router and whereas the MFA-CNA can be collocated with the access-gateways and transit routers. These entities interface with the MNA-NC using FPC ([I-D.ietf-dmm-fpc-cpdp]), or an alternative interface), for managing the IP forwarding policies.



\* MFA-MNA is collocated with the access gateways

\*\* MFA-CNA is collocated with the access gateways and transit routers

Figure 1: Example of a MFA Domain



### 3.1. The Network Topology Database

The network topology database contains the complete and the current information about all the MFA nodes in the network. The information includes the capabilities of each node, supported functions, supported interfaces with the interface-type, connected neighbors, hosted prefixes on each link, security configuration and other related configuration elements. The topology database can be used to determine the route between two nodes within the MFA domain, or the best exit gateway for reaching a correspondent node outside the MFA domain.

### 3.2. The Node Location Database

The node location database consists of location information of each mobile node that is currently attached to the MFA domain. It also includes the type of attachment, previous anchor, and other information elements, such as the mobile node's connection status and detailed or approximate location (e.g. tracking area) in case of device dormancy. Typically, the MFA entities obtain this information from the control-plane functions in the access network. For example, a WLAN controller and the authentication functions will be able to provide this information in IEEE 802.11 based networks. In 5G system architecture this information can be obtained from AMF/SMF functions.

Below diagram is an example NLDB database.

| MN Identifier | Current Anchor | Previous Anchor | Handover Type |
|---------------|----------------|-----------------|---------------|
| MN1@ietf.org  | AG1            | -               | NEW_ATTACH    |
| MN2@ietf.org  | AG6            | AG2             | HANDOVER      |
| MN3@ietf.org  | -              | AG4             | UNKNOWN       |

Figure 2: Example NLDB Table

### 3.3. Determination of the Correspondent Node Anchor

The anchor for a correspondent node is a MFA node that is closest to the correspondent node and is in path for all the MN-CN IP traffic flows. The MFA node controller leverages the topology database for the CN-anchor determination.

If the correspondent node is another mobile node in the MFA domain, then the CN-Anchor for that correspondent node is the access gateway to which it is currently attached.

If the correspondent node is outside the MFA domain, then the CN-anchor is typically the exit gateway, or any MFA node that is always in path for reaching the CN's network. This is typically the PE router of the data center that hosts the correspondent node service, or a programmable data plane node inside the data center.

The below illustration is an example topology of a MFA domain. The domain consists of MFA nodes, mobile and correspondent nodes. A query for CN2's anchor should result in finding AG4, as that is the MFA node in the traffic path and closest to CN2. Similarly, the query for CN3's anchor which is outside the MFA domain should result in finding TR3 as that is the last exit gateway in the MFA domain and closest to the CN3.

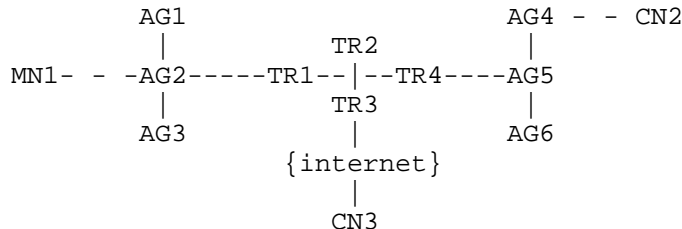


Figure 3: CN Anchor Determination - Example Topology

### 3.4. Traffic Steering Approaches

The MFA nodes support traffic steering approaches for moving the mobile node's IP traffic between the MFA nodes over the most optimal routing path. Segment Routing for IPv6 (SRv6) is one approach that this specification focuses on for steering the traffic between two points in the network, whereas the MFA-NC can utilize the available information from Network Topology- and Node Location Database to enforce policies in the MFA nodes in support of alternative data

plane protocols to enable traffic steering. Future versions of the document may include information about additional mechanisms.

When using SRv6 for traffic steering, the approaches specified in [I-D.ietf-dmm-srv6-mobile-uplane] and [I-D.filsfils-spring-srv6-network-programming] will be leveraged for moving the mobile node's IP traffic between the MFA-MNA and the MFA-CNA nodes. The SRv6 policy including the SID information and the associated functions are pushed from the MFA Node controller to the MFA nodes. This document mostly leverages the functions specified in those documents, but may require some changes to the SRv6 functions for reporting the flow meta-data of the non-optimal traffic flows to the MFA node controller. The definitions of those SRv6 functions will be specified in either in the future revisions of this document, or in other IETF documents.

The following table captures the possible SRv6 function activation when IP traffic steering approach is in use. This is only an example.

| FLOW DIRECTION | MN-Anchor                                                                                                                          | CN-Anchor                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| MN to CN       | Variant of T.Insert<br>(Transit with insertion of SRv6 policy and may require trigger to MFA-NC such as activation of Flow.Report) | Variant of End.X<br>(Or, End.B6, instantiation of a binding SID);<br>Or, End.T for internet traffic                                |
| CN to MN       | Variant of End.X<br>(Layer-3 cross connect<br>(Or, End.B6, instantiation of a binding SID                                          | Variant of T.Insert<br>(Transit with insertion of SRv6 policy and may require trigger to MFA-NC such as activation of Flow.Report. |

Figure 4: Using SRv6 for Traffic Steering - Example

### 3.5. Mobile Node Attachment Triggers

The MFA domain relies on the access network for certain key events related to the mobile node's movements in the network. These events include:

- o INITIAL\_ATTACH - Initial Attachment of the mobile node to the MFA domain
- o HANDOVER - Layer-2/Layer-3 Handover of the mobile node within the MFA Domain
- o DETACH - Detachment of the mobile node from the MFA domain
- o UNKNOWN - State of the mobile node is Unknown; TBD

The MFA node controller interfaces with the radio network and the authentication infrastructure for these events. These events drive the policy configuration on the MFA nodes.

### 3.6. Programming the User-plane

The MFA-NC leverages suitable southbound semantics and operation to enforce traffic steering rules in the selected access gateways (AG) and/or transient routers (TR). One suitable data model and operation is being specified in [I-D.ietf-dmm-fpc-cpdp] for Forwarding Policy Configuration (FPC). The model and operation applies in between a FPC Client function and an FPC Agent function.

A deployment of FPC with the specification per this document about MFA, the FPC Client is co-located with the MFA-NC, whereas the FPC Agent function is co-located with functions that enforce user plane configuration per the rules received from the FPC Client. The FPC Agent can either reside on an transport network- or SDN controller and be in charge of the configuration of multiple user plane nodes (MFA-TR, MFA-MA, MFA-CA), or an FPC Agent resides on each MFA node.

The following figure schematically draws an example how FPC can integrate with the functional MFA architecture per this specification. The example assumes that MFA nodes can be programmatically configured by an SDN Controller. Details about whether a single or multiple distributed SDN Controllers are deployed are left out.

The FPC data model includes the following components:

**Data Plane Nodes (DPN) Model:**

Representation of nodes in the data plane which can be selected and enforce rules per the control plane's directives. DPNs take a particular role, which is identified in the model. In the context of this document, the role of a DPN can be, for example, an anchor node or a transit router.

**Topology Model:**

Representation of DPNs in the network and associate in between DPNs. The FPC Client and Agent use the Topology to select most appropriate data plane node resources for a communication. In the context of this document, Topology has can be leveraged to implement the NTDB for the selection of steering paths and associated DPNs which function as MFA-MNA, MFA-CNA, or MFA-TR.

**Policy Model:**

Defines and identifies rules for enforcement at DPNs.

**Mobility-Context:**

Holds information associated with a mobile node and its mobility sessions. In the context of this document, Mobility-Context can be enriched with traffic steering related rules.

**Monitor:**

Provides mechanisms to register monitors (traffic, events) in the data plane and define status reporting schedules, which can be periodic or event-based. In the context of this document, Monitors may be used to detect traffic from a CN to an MN on an MFA node, which could result in a notification to the MFA-NC for path optimization and associated steering of traffic to the MN's current MFA-MNA.

Please refer to [I-D.ietf-dmm-fpc-cpdp] for model and operational details.

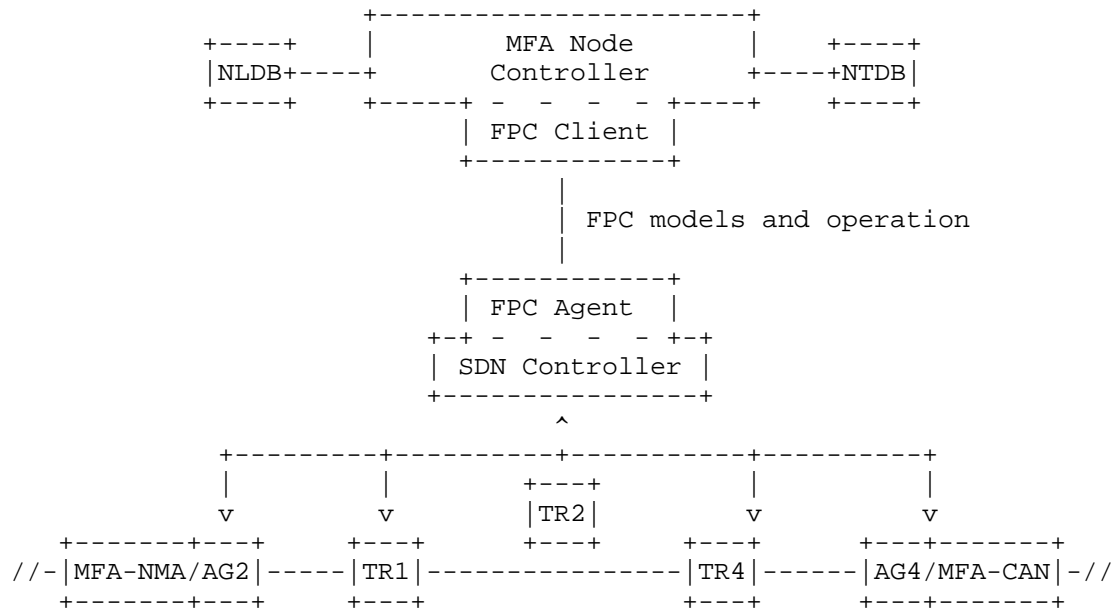


Figure 5: Deployment of the FPC models and operation in between the MFA-NC and MFA nodes on the user plane

#### 4. Life of a Mobile Node in a MFA Domain

##### Reference Topology

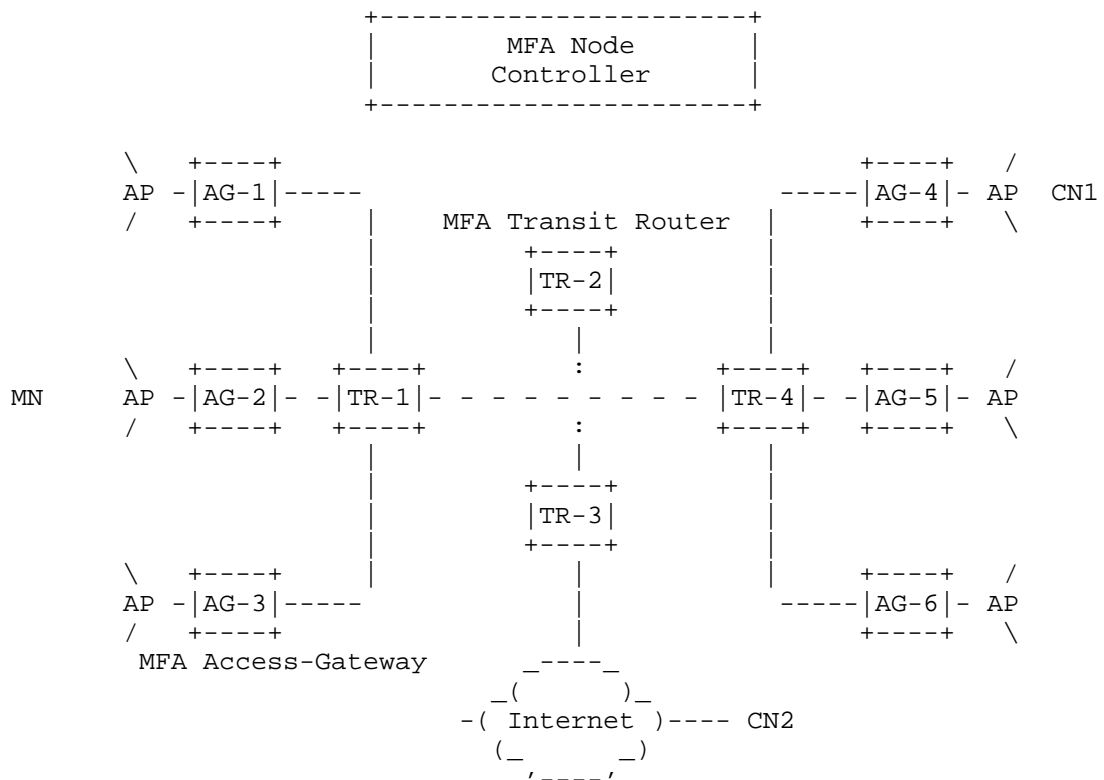


Figure 6: Reference Topology

#### 4.1. MN's Initial Attachment to a MFA Domain

A mobile node, MN enters the MFA domain and attaches to the access point on the gateway AG-2.

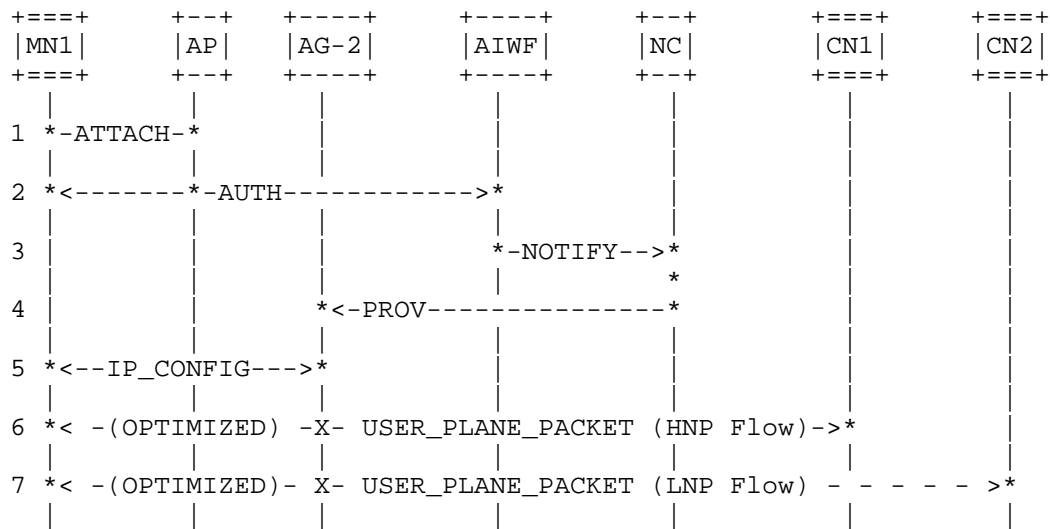


Figure 7: Mobile Node's Initial Attachment to a MFA Domain

- o 1-ATTACH: The mobile node with NAI (MN1@ietf.org) performs a layer-2 attach to the access point. This access point is connected to the access-gateway, AG-2, over a layer-2 link. The mobile node anchor function is supported on AG-2 and is active.
- o 2-AUTH: The mobile node completes the access authentication access technology specific access mechanisms. The mobile node's identity is established and is authorized for MFA domain access. The Authentication interworking (AUTH-IWK) function records the mobile node's identity, type of attach as `INITIAL_ATTACH`, and the current location of the mobile node in the access-network, to the node location database.
- o 3-NOTIFY: The Auth-IWK function delivers the attach event to the MFA node controller. The information elements that are delivered include the mobile node identifier (MN-1@ietf.org), type of attach as `INITIAL_ATTACH`, and the identity of the access gateway, which is AG-2.
- o 4-PROV: The NC provisions AG-2 for hosting the MN's home-network prefix(es). The assigned prefixes are HNP, H1::/64 and LNP, L1::/64. These prefixes are from a larger aggregate block (Ex: H1::/48; L1::/48) which are topologically anchored on AG-2. The policies for hosting the HNP prefixes on the link are provisioned using FPC interface. The AG-2 will include meta-data in the IPv6 RA messages for indicating the properties of the prefixes; H1::/64



as the prefix with mobility support and L1 as the prefix with no mobility support.

- o 5-IP\_CONFIG: The mobile node generates one or more IPv6 addresses using the prefixes H1 and L1. The generated addresses are tagged with the property meta-data in the host's source address policy table. This allows the applications on the mobile node to pick the addresses based on the application's mobility requirements.
- o 6-USER\_PLANE\_PACKET: The mobile node establishes IP flow with CN1. The source address is based on the prefix H1. This IP address will have mobility support. The packets associated with this flow will take the optimized routing path. There are no tunnels, or special traffic steering rules in the network.
- o 7-USER\_PLANE\_PACKET: The mobile node establishes IP flow with CN2. The source address is based on the prefix L1. This IP address will not have mobility support. There are no tunnels, or special traffic steering rules in the network.

#### 4.2. MN's Roaming within the MFA Domain

The mobile node roams and changes its point of attachment. It was initially attached to the access network on AG-2 and now it attaches to access network on AG-6. At the time of roaming, the mobile node had two active IPv6 prefixes HNP, H1::/64 and LNP, L1::/64 and there were two active IP flows, one to CN1 using an IPv6 address from the prefix H1::/64 and another flow to CN2 using an IPv6 address from the prefix L1::/64. The MFA network will ensure the prefix H1::/64 will be routable on the new network and the active flow to CN1 will survive, however the prefix L1::/64 will not be routable in the new access network and therefore the flow to CN2 will not survive.

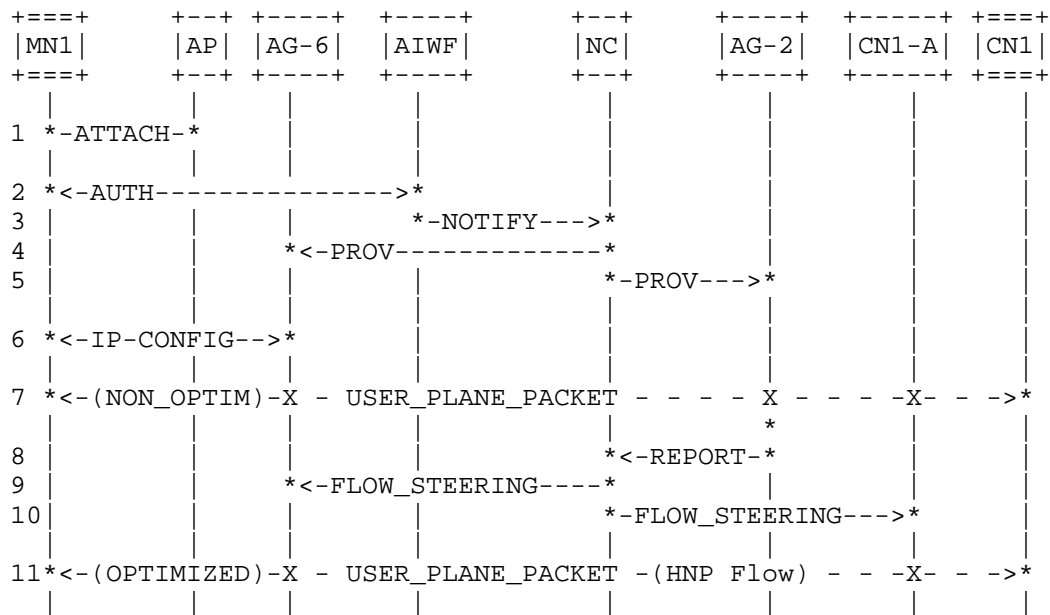


Figure 8: Mobile Node's Roaming within the MFA Domain

- o 1-ATTACH: The mobile node with NAI (MN1@ietf.org) roams in the network from AG-2 to AG-6.
- o 2-AUTH: The mobile node completes the handover to the new access network using access network specific security mechanisms. The Auth-IWK function updates the mobile node's location in the node-location database. The updated entry in the node location database will include the mobile node's NAI, attach type as HANDOVER, and the current access-network location as AG-6.
- o 3-NOTIFY: The Auth-IWK function delivers the handover event to the MFA node controller. The information elements that are delivered include the mobile node identifier (MN-1@ietf.org), type of attach as HANDOVER, and the identity of the access gateway as AG-6.
- o 4-IP\_PROV: The NC provisions AG-6 for hosting the MN's home-network prefix and local network prefix. The home network prefix, H1::/64 is from the previous anchor, AG-2 and is not topologically anchored on AG-6. However, for supporting mobility the prefix is hosted on the access link while the mobile node is attached to that access network and till there are active flows. The NC also

provisions AG-6 for hosting a new local network prefix, L2::/64. This prefix, L2::/64 is from a larger aggregate block that is topologically anchored on AG-6. The AG-6 will include meta-data in the IPv6 RA messages for indicating the properties of the prefixes; H1::/64 as the prefix with mobility support and L2::/64 as the prefix with no mobility support. The NC also provisions a traffic steering rule to steer all uplink IP traffic with source address H1::/64 through the previous anchor AG-2.

- o 5-IP\_PROV: The NC provisions AG-2 to steer all IP traffic to destination addresses matching the prefix, H1::/64 to AG-6, and it also provisions a rule to report flow meta-data of those flows taking the non-optimal traffic path through AG-2. This essentially allows the NC to learn about any mobile node's IP flows still going through AG-2, so it can stitch the optimized path for those flows and remove AG-2 from the path for those flows.
- o 6-IP\_CONFIG: The prefix H1::/64, obtained at the new location, will continue to be available on the new access link. The new local network prefix L2::/64 will also be available on the new access link and will be marked as a prefix with no mobility property. The mobile node may generate one, or more IPv6 addresses using the prefix L2::/64. The prefix L1::/64 is no longer hosted on the new link and the mobile node will remove it from interface configuration.
- o 7-USER\_PLANE\_PACKET: Any uplink IP link from CN1 will come to AG-2, as its the topological anchor for that address/prefix and AG-2 will steer the traffic directly to AG-6. On detecting an IP flow with the IP address belonging to prefix H1::/64, AG-2 will report the CN1-MN1 flow meta-data to NC.
- o 8-Report: The NC on receiving this event will lookup the CN anchor for the flow in its node location database. If the CN is another MN within the MFA domain, its current anchor information is retrieved from the node location database. However, if the CN is a node outside the MFA domain, the anchor for this node can be any transit router in the MFA domain which is always in path for that destination. The CN-anchor determination for nodes outside the MFA domain will be based on the network topology database.
- o 9-FLOW\_STEERING: The NC inserts a IP traffic steering rule on AG-6 to steer the MN1-CN1's IP flows using H1::/64 directly to CN1's anchor which is CN1-A, and bypassing AG-2.
- o 10-FLOW\_STEERING: The NC inserts a IP traffic steering rule on CN1-A to steer the MN1-CN1 IP flows using H1::/64 directly to

MN1's current anchor which is AG-6, and bypassing AG-2.

- o 11-USER\_PLANE\_PACKET: The MN1-CN1's IP flows using H1::/64 will be steered directly from CN1-A to AG-6; AG-2 will not be in the path.

#### 4.3. Traffic Steering State Removal

The mobile node's IP flows that were established at the previous location are no longer active. The steering state that was introduced at AG-6 and CN1-A will be removed on detecting the inactive flows. The network may also optionally choose to withdraw the prefix H1::/64 and may assign a new HNP prefix which are topologically anchored in the new location.

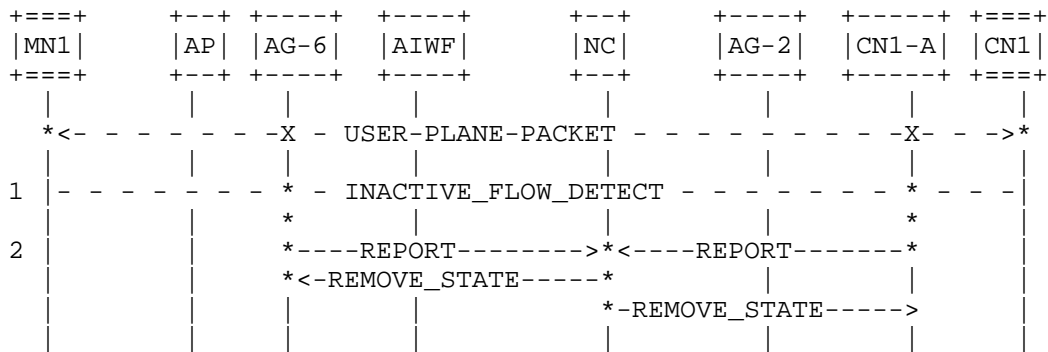


Figure 9: State Removal

- o 1-IN\_ACTIVE\_FLOW\_DETECT: At some point the MN1-CN1 flow using the prefix H1::/64 is no longer active.
- o 2-REPORT: Both AG-6 and AG-2 will detect the inactive flows and may report this event to the NC. The steering state associated with MN1-CN1 flow using the prefix H1::/64 may be removed prior to reporting to the NC. Optionally, the NC on receiving the INACTIVE\_FLOW\_DETECT event may provision AG-6 and CN1-A to remove the steering state.
- o 4-REMOVE\_STATE:

#### 4.4. Mobile Node's new IP flows

The mobile node's IP flows that were established at the previous location are no longer active and any created steering state was removed. The network may optionally choose to withdraw the prefix H1::/64 and may assign a new HNP prefix which is topologically anchored in the new location. All new IP flows will use the new prefix and the flows will take optimal routing path.

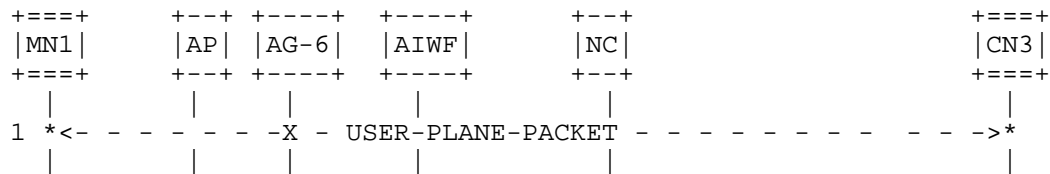


Figure 10: New Flows

- o 1-USER\_PLANE\_PACKET: The mobile node's has established some IP flows using the IP address from the new HNP and LNP assigned at the new location. These IP flows will take optimal routing path and there is no need for any steering state, or the use of tunnels in the network for the mobile node's traffic.

## 5. MFA in 5G System Architecture

3GPP is specifying the 5G System Architecture, which follows a split between control- and data plane. Key control plane functions, which have interfaces to the data plane, are the Access Network and Mobility Management Function (AMF), and the Session Management Function (SMF). AMF and SMF cooperate to set up data plane nodes in the (radio) access network ((R)AN) and the core network, which comprises one or multiple User Plane Functions (UPF). As soon as a mobile node (UE) attaches to the network, as Packet Data Unit (PDU) Session is established and the SMF in the control plane selects one UPF as PDU Session Anchor, which serves also as IP address anchor. The SMF may select one more UPF on the path in between the PDU Session Anchor and the (R)AN, which enables routing traffic in between the UE and a local packet data network (PDN) with a correspondent node or service without the need to traverse the PDU Session Anchor.

In the view of MFA, each UPF can represent a locator for the UE's downlink traffic on the N9 as well as on the N6 reference point in

the 5G System Architecture. Since the SMF is in charge of UPF selection and configuration, the MFA-NC can leverage the SMF to retrieve node location information per this specification's procedure to access the NLDB from the MFA-NC. For MFA node selection and traffic steering, the MFA-NC may need more information about the data plane in terms of the transport network nodes and topology. Details about the NTDB are left out of this version of the document, but a realization may exploit available Topology information per [I-D.ietf-dmm-fpc-cpdp].

In the figure below, a UE's UPFs can function as MFA nodes, either as MFA-MNA or as MFA-CNA in case of mobile to mobile communication. Other transport network nodes, which may function as MFA-CNA for the UE's communication with a (non-mobile) correspondent node or service, are not explicitly depicted in the below figure. The MFA function can be tightly coupled with a UFP (co-located) or loosely coupled (separated). The MFA-NC utilized the FPC models and operation to enforce traffic steering policies in the MFA nodes. In case of loose coupling, the SMF utilizes the N4 protocol per the 3GPP standard to configure the selected UPF, whereas the MFA-NC uses FPC to enforce policies in the associated (loosely coupled) MFA node. In case of tight coupling, the MFA-NC may be co-located with the SMF and a single reference point and associated protocol may be used in between the SMF/MFA-NC and a UPF/MFA node.

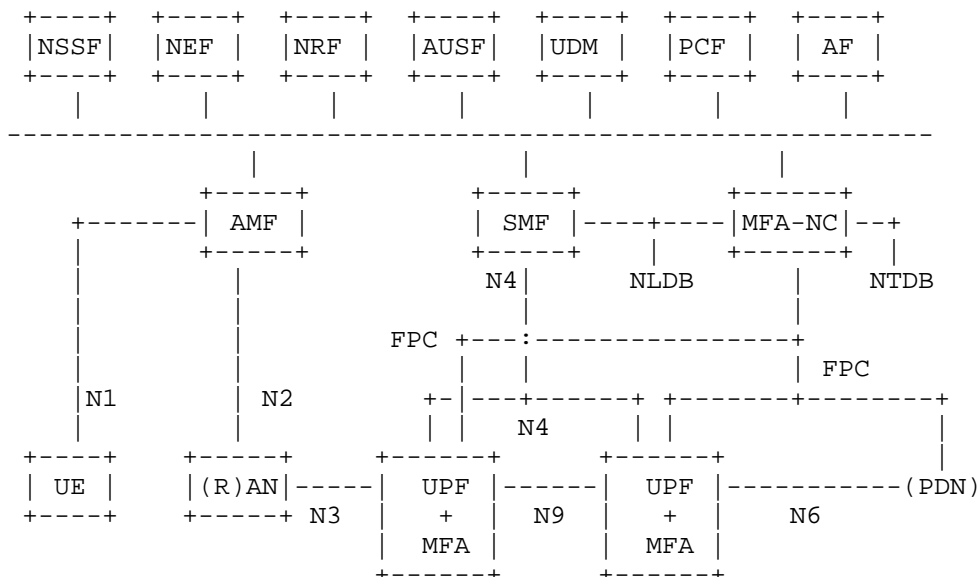


Figure 11: New Flows

## 6. IANA Considerations

TBD

## 7. Security Considerations

This specification allows a mobility node controller to provision IP traffic steering policies on the user plane nodes. It essentially leverages the FPC interface [I-D.ietf-dmm-fpc-cdpd] for interfacing with the user-plane anchor nodes. The security considerations specified in the FPC specification are sufficient for securing the messages carried on this interface.

The traffic steering rules that are provisioned on the MFA nodes by the MFA node controller are the standard policy rules that the FPC interface defines and does not require any new security considerations.

## 8. Acknowledgements

TBD

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

[I-D.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", draft-filsfils-spring-srv6-network-programming-03 (work in

progress), December 2017.

- [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-09  
(work in progress), October 2017.
- [I-D.ietf-dmm-ondemand-mobility]  
Yegin, A., Moses, D., Kweon, K., Lee, J., Park, J., and S.  
Jeon, "On Demand Mobility Management",  
draft-ietf-dmm-ondemand-mobility-13 (work in progress),  
January 2018.
- [I-D.ietf-dmm-srv6-mobile-uplane]  
Matsushima, S., Filsfils, C., Kohno, M.,  
daniel.voyer@bell.ca, d., and C. Perkins, "Segment Routing  
IPv6 for Mobile User-Plane",  
draft-ietf-dmm-srv6-mobile-uplane-00 (work in progress),  
November 2017.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V.,  
Chowdhury, K., and B. Patil, "Proxy Mobile IPv6",  
RFC 5213, DOI 10.17487/RFC5213, August 2008,  
<<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility  
Support in IPv6", RFC 6275, DOI 10.17487/RFC6275,  
July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.

#### Authors' Addresses

Sri Gundavelli  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA  
  
Email: [sgundave@cisco.com](mailto:sgundave@cisco.com)



Marco Liebsch  
NEC  
Kurfuersten-Anlage 36  
D-69115 Heidelberg,  
Germany

Email: [liebsch@neclab.eu](mailto:liebsch@neclab.eu)

Satoru Matsushima  
SoftBank  
Tokyo,  
Japan

Email: [satoru.matsushima@g.softbank.co.jp](mailto:satoru.matsushima@g.softbank.co.jp)



DMM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 25, 2019

S. Homma  
NTT  
T. Miyasaka  
KDDI Research  
S. Matsushima  
SoftBank  
D. Voyer  
Bell Canada  
October 22, 2018

User Plane Protocol and Architectural Analysis on 3GPP 5G System  
draft-hmm-dmm-5g-uplane-analysis-02

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 25, 2019.

Copyright Notice

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

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

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

## Table of Contents

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                            | 2  |
| 1.1. Current Status of Mobile User Plane for 5G . . . . .            | 3  |
| 1.2. Our Way of Analysis Work . . . . .                              | 3  |
| 2. Terms and Abbreviations . . . . .                                 | 4  |
| 3. GTP-U Specification and Observation . . . . .                     | 5  |
| 3.1. GTP-U Tunnel . . . . .                                          | 6  |
| 3.2. GTP-U Header Format . . . . .                                   | 10 |
| 3.3. Control Plane Protocol for GTP-U . . . . .                      | 12 |
| 3.4. GTP-U message . . . . .                                         | 13 |
| 3.5. Packet Format . . . . .                                         | 14 |
| 3.6. Observations Summary . . . . .                                  | 16 |
| 4. 5GS Architectural Requirements for User Plane Protocols . . . . . | 16 |
| 4.1. Overview of 5G System Architecture . . . . .                    | 16 |
| 4.1.1. UPF Functionalities . . . . .                                 | 18 |
| 4.2. Architectural Requirements for User Plane Protocols . . . . .   | 19 |
| 5. Evaluation Aspects . . . . .                                      | 22 |
| 5.1. Supporting PDU Session Type Variations . . . . .                | 23 |
| 5.2. Nature of Data Path . . . . .                                   | 23 |
| 5.3. Supporting Transport Variations . . . . .                       | 24 |
| 5.4. Data Path Management . . . . .                                  | 24 |
| 5.5. QoS Control . . . . .                                           | 25 |
| 5.6. Traffic Detection and Flow Handling . . . . .                   | 26 |
| 5.7. Supporting Network Slicing Diversity . . . . .                  | 26 |
| 6. Conclusion . . . . .                                              | 27 |
| 7. Security Consideration . . . . .                                  | 27 |
| 8. Acknowledgement . . . . .                                         | 28 |
| 9. Informative References . . . . .                                  | 28 |
| Authors' Addresses . . . . .                                         | 32 |

## 1. Introduction

This document analyzes the mobile user plane protocol and the architecture specified by 3GPP 5G documents. The background of the work is that 3GPP requests through a liaison statement that the IETF to provide any information for the User Plane Protocol Study work in 3GPP [CP-180116-3GPP]. Justification and the objectives of the study can be found from [CP-173160-3GPP].

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

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

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

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

#### 1.1. Current Status of Mobile User Plane for 5G

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

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

#### 1.2. Our Way of Analysis Work

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

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

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

[I-D.bogineni-dmm-optimized-mobile-user-plane] will provide the candidate protocols on IETF side to the 3GPP study.

## 2. Terms and Abbreviations

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

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

GTP: GPRS Tunneling Protocol

GTP-U: User Plane part of GTP

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

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

Noted that the PDU in 3GPP includes IP header in case that PDU session type is IPv4 or IPv6. In contrast, in IETF it is supposed that PDU is the payload of IP packet so that it doesn't include IP/TCP/UDP header in end-to-end.

T-PDU: Transport PDU.

G-PDU: GTP encapsulated user Plane Data Unit.

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

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

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

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

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

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

QFI: QoS Flow Identifier

UPF: User Plane Function

SMF: Session Management Function

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

RAN: Radio Access Network

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

### 3. GTP-U Specification and Observation

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

### 3.1. GTP-U Tunnel

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

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

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

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

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

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

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

N3: Between RAN and UPF

N9: Between different UPFs

GTP-U allows one tunnel endpoint node to send out a G-PDU to be received by multiple tunnel endpoints by utilizing IP multicast capability of underlay IP networks. That is described in section 4.2.6 of [TS.29.281-3GPP]. It looks GTP-U has Point-to-Multipoint (P2MP) tunneling capability. The P2MP tunneling is used for MBMS (Multimedia Broadcast Multicast Service) through GTP-U tunnel.

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

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



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

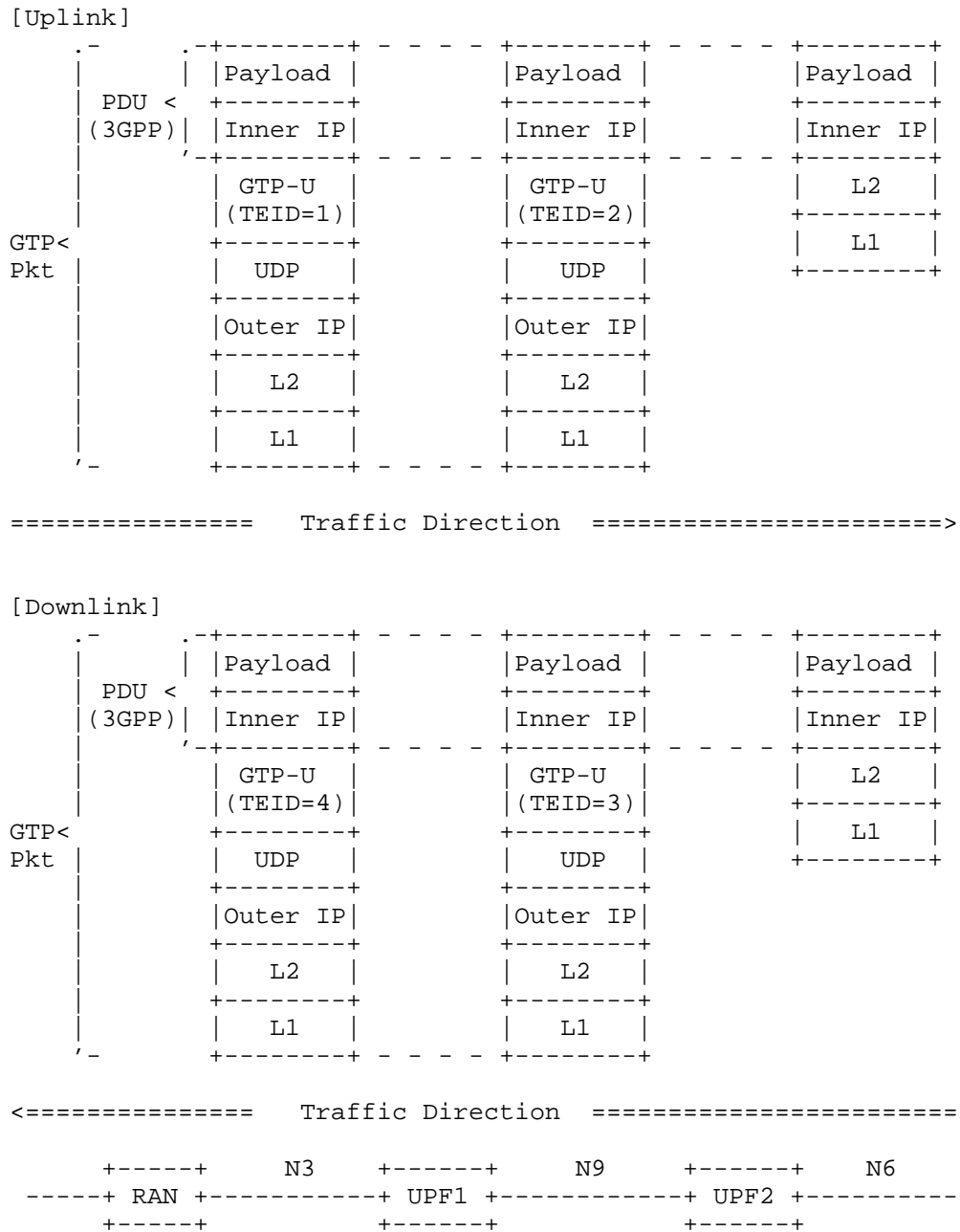


Figure 1: Protocol Stack by GTPv1-U for Uplink and Downlink Traffic Flow

IPv6 flow label [RFC6437] is also candidate method for load balancing especially for IP-in-IPv6 tunnel [RFC6438] like GTP-U. However, how to use IPv6 flow label of GTP-U is not described in [TS.29.281-3GPP]. Though this method is limited to a case of IPv6 encapsulated GTP-U tunnel, it is worth to study usage of IPv6 flow label in 3GPP.

[GTP-U-4]: GTP-U does not support IPv6 flow label for load balancing in case of IPv6 transport.

GTP-U supports both IPv4 and IPv6 as underlying transport layer protocol. As for IPv6, GTP-U specification refers [RFC2460], which is described in section 4.2.3 of [TS.29.281-3GPP]. As [RFC2460] does not allow the tunnel protocols on top of UDP to set the checksum value to zero, the GTP-U specification inherits it while the IPv4 transport for GTP-U case allows UDP zero checksum. It is noted that the IPv6 specification in IETF has been updated to [RFC8200] which allows UDP zero checksum for the tunnel. [RFC6935] describes benefits of zero checksum for tunnel protocol over UDP. If GTP-U support UFP zero checksum in future version, possible interoperability issue between previous generations which does not support zero checksum may raise.

[GTP-U-5]: UDP zero checksum is not available in case of IPv6 transport.

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

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

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

### 3.2. GTP-U Header Format

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

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

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

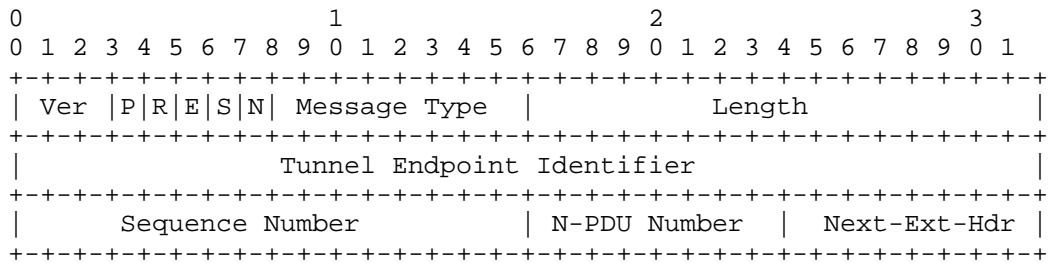


Figure 2: GTP-U Header

- o Ver: Version field (Set to '1')
- o P: Protocol Type (Set to '1')
- o R: Reserved bit (Set to '0')
- o E: Extension Header Flag (Set to '1' if extension header exists)
- o S: Sequence Number Flag (Set to '1' if sequence number exists)
- o N: N-PDU Number Flag (Set to '1' if N-PDU number exists)
- o Message Type: Indicates the type of GTP-U message
- o Length: Indicates the length in octets of the payload

- o Tunnel Endpoint Identifier (TEID)
- o Sequence Number: Indicates increasing sequence number for T-PDUs is transmitted via GTP-U tunnels
- o N-PDU Number: It is used only for inter SGSN, 2G-3G handover case, etc.
- o Next-Ext-Hdr: Indicates following extension header type

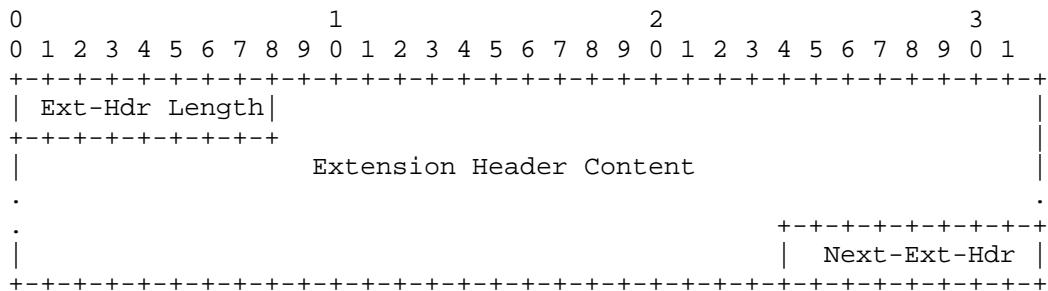


Figure 3: Extension GTP-U Header

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

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

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

- o NR RAN Container (Next-Ext-Hdr = 10000100)
- o PDU Session Container (Next-Ext-Hdr = 10000101)
- o PDCP PDU Number (Next-Ext-Hdr = 11000000)

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

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

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

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

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

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

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

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

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

### 3.3. Control Plane Protocol for GTP-U

Control plane protocol for GTP-U signals TEID between tunnel endpoint nodes. GTPv2-C [TS.29.274-3GPP] is the original control plane

protocol tied with GTP-U in previous generation architectures before CUPS (Control and User Plane Separation).

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

### 3.4. GTP-U message

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

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

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

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

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

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

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

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

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

### 3.5. Packet Format

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

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

```

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
Outer IPv6 Header
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|      DSCP=EF      |      Flow Label      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Payload Length      | NxtHdr=17(UDP)|      Hop Limit  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+
|
+      Source IPv6 Address      +
|      2001:db8:1:1::1          |
+
+-----+-----+-----+-----+-----+-----+-----+-----+
|
+
|
+      Destination IPv6 Address  +
|      2001:db8:1:2::1          |
+
+-----+-----+-----+-----+-----+-----+-----+-----+

Outer UDP Header
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Source Port = xxxx      |      Dest Port = 2152      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      UDP Length      |      UDP Checksum (Non-zero)      |
+-----+-----+-----+-----+-----+-----+-----+-----+

GTP-U header

```



```

+-----+
| 0x1 | 1|0|1|0|0|      0xff      |          Length          |
+-----+
|                                     TEID = 1654                 |
+-----+
|      Sequence Number = 0          |N-PDU Number=0|NextExtHdr=0x85|
+-----+

```

#### GTP-U Extension Header (PDU Session Container)

```

+-----+
| ExtHdrLen=2 |Type=0| Spare |0|0|   QFI   | PPI | Spare |
+-----+
|                                     Padding                    |NextExtHdr=0x0|
+-----+

```

#### Inner IPv6 Header

```

+-----+
|Version|      DSCP=0      |          Flow Label          |
+-----+
|      Payload Length      |      NextttHdr      |      Hop Limit      |
+-----+
|
+
|
+
|      Source IPv6 Address
|      2001:db8:2:1::1
+
|
+-----+
|
+
|
+
|      Destination IPv6 Address
|      2001:db8:3:1::1
+
|
+-----+

```

#### Payload

```

+-----+
|
|
|      TCP/UDP/etc., Data
|
|
+-----+

```

Figure 4: GTP-U Protocol Stack Example

### 3.6. Observations Summary

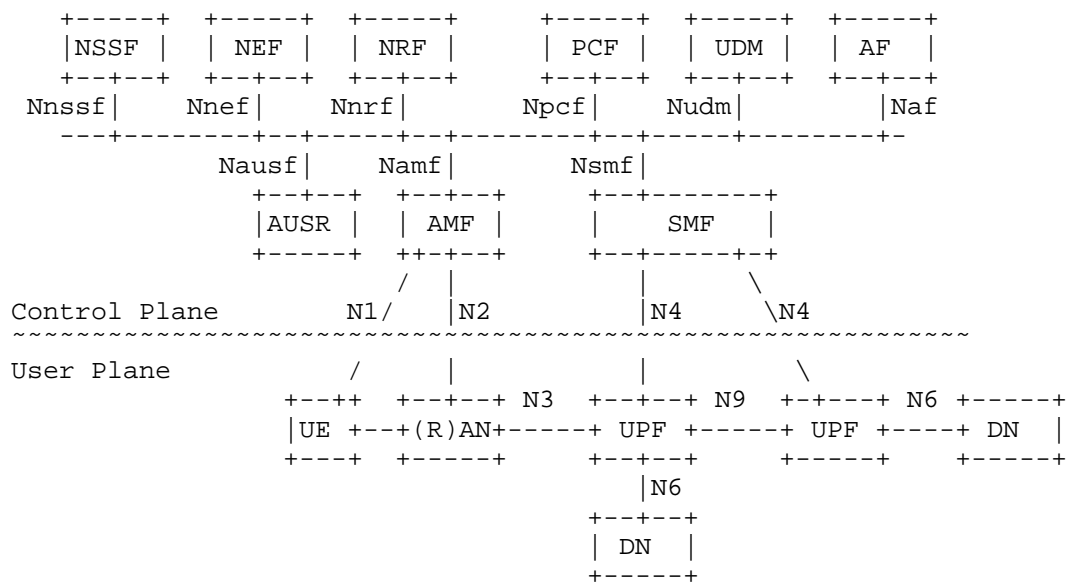
- [GTP-U-1]: An unidirectional Point-to-Point tunneling protocol.
- [GTP-U-2]: Supports Point-to-Multipoint tunneling.
- [GTP-U-3]: Supports load balancing by using dynamic UDP port allocation.
- [GTP-U-4]: Does not support IPv6 flow label for load balancing in case of IPv6 transport.
- [GTP-U-5]: UDP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

## 4. 5GS Architectural Requirements for User Plane Protocols

### 4.1. Overview of 5G System Architecture

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

The architecture overview is shown in Figure 5. The details of functions are described in [TS.23.501-3GPP]. User plane path and applied functions for a tunnel will be manipulated based on application requirements that the PDU session corresponding to the tunnel. These tunnels are available to be handled by other authorized functions through the control plane.



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

#### 4.1.1.1. UPF Functionalities

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

The followings are defined as UPF functionalities for traffic handling:

- o User Plane part of policy rule enforcement, e.g. Gating, Redirection, Traffic steering)
- o QoS handling for user plane, e.g. UL/DL rate enforcement, Reflective QoS marking in DL
- o Transport level packet marking in the uplink and downlink

Other functionalities are described in the section 6.2.3 of [TS.23.501-3GPP]

UPF shall detect user plane traffic flow depending on information indicated by SMF. User data traffic is detected with combination of the following information:

- o For IPv4 or IPv6 PDU Session type
  - \* PDU Session
  - \* QFI
  - \* Application Identifier: The Application ID is an index to a set of application detection rules configured in UPF
- o For Ethernet PDU Session type
  - \* PDU Session
  - \* QFI
  - \* Ethernet Packet Filter Set:
    - + Source/destination MAC address
    - + Ethertype as defined in IEEE 802.3
    - + Customer-VLAN tag(C-TAG) and/or Service-VLAN tag(S-TAG) VID fields as defined in IEEE 802.1Q

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

Such information for traffic detection (Traffic Detection Information) is described in the section 5.8.2.4 of [TS.23.501-3GPP].

#### 4.2. Architectural Requirements for User Plane Protocols

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

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

The requirements for UP protocols are described below:

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

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

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

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

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

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

DNS running on IPv4 transport connect to UPF running on IPv6 transport.

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

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

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

Multihoming is provided with Branching Point (BP) or Uplink Classifier (UL CL) which are functionalities of UPF. BP provides forwarding of UL traffic towards the different PDU Session Anchors based on the source IPv6 prefixes and merge of DL traffic to the UE. UL CL provides destination based multihoming for load balancing.

On UL side, multihoming of a single PDU session is achieved by a point-to-point (P2P) tunnel per anchor UPF. It means that multiple P2P paths are established from one source gNB or UPF to the multiple destination anchor UPFs for the PDU session.

On DL side, one single multipoint-to-point (MP2P) path exists from the anchor UPFs to the source gNB or UPF for the PDU session in this multihoming case. It means that the paths from the anchor UPFs are merged into just one tunnel state at the source gNB or UPF for the PDU session.

Multiple P2P paths on DL could also be used for multihoming. However it should be the multiple PDU sessions multihoming case where the destination gNB or UPF needs to maintain multiple tunnel states under the one PDU session to one UP tunnel architectural principle.

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

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

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

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

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

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

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

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

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

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

In addition, deployment of multiple UPFs as anchors closed to UEs' site and connecting them without extra anchor points enable to make data path more efficient. This usage is described in the section 6.5 of [TS.22.261-3GPP].

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

It is expected that multiple UPFs with per session tunnel handling for a PDU session becomes complicated task more and more for a SMF by increasing number of UPFs, and UP protocol shall support to aggregate several PDU sessions into a tunnel or shall be a session-less tunnel.

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

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

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

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

ARCH-Req-7: Supporting network slicing

The 5GS fundamentally supports network slicing for provision the appropriate end-to-end communication to various services. In the relevant documents (e.g., [TS.23.501-3GPP], [TS.28.531-3GPP]), a network slice is defined as virtual network and it is structured with SMF, RANs, UPFs and DN. Each network slice is independent and its user plane (including network functions and links) should be noninteractive against the others.

Note that 3GPP focuses on only mobility management and specifications to synchronize with other networks including transport networks is not clearly defined.

## 5. Evaluation Aspects

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



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

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

#### 5.1. Supporting PDU Session Type Variations

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

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

#### 5.2. Nature of Data Path

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

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

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

### 5.3. Supporting Transport Variations

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

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

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

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

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

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

### 5.4. Data Path Management

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

That powerful user plane capability could affect data path management complicated and difficult to be managed through the control plane, or operation support systems (OSS). Perhaps it could be the case where the UP protocol nature is P2P and it only supports per session base data path handling.

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

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

#### 5.5. QoS Control

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

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

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

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

## 5.6. Traffic Detection and Flow Handling

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

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

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

## 5.7. Supporting Network Slicing Diversity

To embody network slicing, it is expected that various means should be available in case by case, or operator by operator, for their 5G systems while it follows the fundamental slicing concept, as described in Section 4.1.

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

The assumed slice operation in 5G architecture is that UPFs connect to each other through direct (virtual) link as Section 4.1 describes that UPFs compose a network slice on an UP. So IP routing and transport system underneath the UP are not visible from the 5G system. However some means need to indicate a slice on the shared underlying networks of the UP over the wire.

That's just one way for network slicing, but it would help to reduce the operational burden. Even it depends on each operator's policy, sharing network instances, UPFs, and the interfaces among slices makes operators relax and not to be much hustled on slice lifecycle management., such as create, update, and delete operations for slices.

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

It could also make sense in case that each network slice requires different forwarding policies in the middle of the path. Some identifier in the packets for a slice could be a glue between UP path and its underlying transport networks. For example, if a slice requires certain level of latency with dedicated route, traffic engineering (TE) embodies appropriate forwarding policy through the underlay transport network.

The expected evaluation points from this aspect should be whether the candidate protocols can support to indicate a network slice in the UP packets that could be enough bits space on stable ID space to put slice identifier for the slice, or the forwarding policy within the slice. Since 5G control plane is not designed to handle transport resources, such as VLAN, MPLS Label, Tunnel ID except GTP-U, less impact to the control plane protocol and the APIs should be much preferable.

## 6. Conclusion

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

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

As we identified some potential gaps between the current UP protocol and the architectural requirements even for Release 15, it is worth to study possible candidate UP protocols for the 5G system including current one. Our conclusion here is that we suggest the UP protocol study work in 3GPP takes into account the evaluation aspects described in Section 5.

## 7. Security Consideration

TBD

## 8. Acknowledgement

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

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

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

## 9. Informative References

### [C4-185491-3GPP]

3rd Generation Partnership Project (3GPP), "LS OUT on User Plane Analysis", July 2018,  
<[http://www.3gpp.org/ftp/tsg\\_ct/WG4\\_protocollars\\_ex-CN4/TSGCT4\\_85bis\\_Sophia\\_Antipolis/Docs/C4-185491.zip](http://www.3gpp.org/ftp/tsg_ct/WG4_protocollars_ex-CN4/TSGCT4_85bis_Sophia_Antipolis/Docs/C4-185491.zip)>.

### [CP-173160-3GPP]

3rd Generation Partnership Project (3GPP), "New Study Item on User Plane Protocol in 5GC", December 2017,  
<[http://www.3gpp.org/ftp/tsg\\_ct/TSG\\_CT/TSGC\\_78\\_Lisbon/Docs/CP-173160.zip](http://www.3gpp.org/ftp/tsg_ct/TSG_CT/TSGC_78_Lisbon/Docs/CP-173160.zip)>.

### [CP-180116-3GPP]

3rd Generation Partnership Project (3GPP), "LS on user plane protocol study", March 2018,  
<[http://www.3gpp.org/ftp/tsg\\_ct/TSG\\_CT/TSGC\\_79\\_Chennai/Docs/CP-180116.zip](http://www.3gpp.org/ftp/tsg_ct/TSG_CT/TSGC_79_Chennai/Docs/CP-180116.zip)>.

### [I-D.bogineni-dmm-optimized-mobile-user-plane]

Bogineni, K., Akhavain, A., Herbert, T., Farinacci, D., Rodriguez-Natal, A., Carofiglio, G., Auge, J., Muscariello, L., Camarillo, P., and S. Homma, "Optimized Mobile User Plane Solutions for 5G", draft-bogineni-dmm-optimized-mobile-user-plane-01 (work in progress), June 2018.

### [IAB-Statement]

Internet Architecture Board (IAB), "IAB Statement on IPv6", November 2016,  
<<https://www.iab.org/2016/11/07/iab-statement-on-ipv6/>>.

### [RFC2460]

Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.

- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <<https://www.rfc-editor.org/info/rfc6935>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [TR.29.891-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TR 29.891 (V15.0.0): 5G System Phase 1, CT WG4 Aspects", December 2017, <[http://www.3gpp.org/FTP/Specs/2017-12/Rel-15/29\\_series/29891-f00.zip](http://www.3gpp.org/FTP/Specs/2017-12/Rel-15/29_series/29891-f00.zip)>.
- [TS.22.261-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TS 22.261 (V15.4.0): Service requirements for 5G system stage 1", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/22\\_series/22261-f40.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/22_series/22261-f40.zip)>.
- [TS.23.060-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TS 23.060 (V15.3.0): General Packet Radio Service (GPRS); Service description; Stage 2", June 2018, <[http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.060/23060-f30.zip](http://www.3gpp.org/ftp//Specs/archive/23_series/23.060/23060-f30.zip)>.

## [TS.23.501-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V15.3.0): System Architecture for 5G System; Stage 2", September 2018, <[http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.501/23501-f30.zip](http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-f30.zip)>.

## [TS.23.502-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.502 (V15.1.0): Procedures for 5G System; Stage 2", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23\\_series/23502-f10.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23_series/23502-f10.zip)>.

## [TS.23.503-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 23.503 (V15.1.0): Policy and Charging Control System for 5G Framework; Stage 2", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23\\_series/23503-f10.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/23_series/23503-f10.zip)>.

## [TS.28.530-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.530 (V1.0.0): Management and orchestration of networks and network slicing; Concepts, use cases and requirements (work in progress)", June 2018, <[http://ftp.3gpp.org//Specs/archive/28\\_series/28.530/28530-100.zip](http://ftp.3gpp.org//Specs/archive/28_series/28.530/28530-100.zip)>.

## [TS.28.531-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.531 (V1.0.0): Management and orchestration of networks and network slicing; Provisioning; Stage 1 (Release 15)", June 2018, <[http://ftp.3gpp.org//Specs/archive/28\\_series/28.531/28531-100.zip](http://ftp.3gpp.org//Specs/archive/28_series/28.531/28531-100.zip)>.

## [TS.28.532-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.532 (V0.3.0): Management and orchestration of networks and network slicing; Provisioning; Stage 2 and stage 3 (Release 15)", June 2018, <[http://www.3gpp.org/ftp//Specs/archive/28\\_series/28.532/28532-030.zip](http://www.3gpp.org/ftp//Specs/archive/28_series/28.532/28532-030.zip)>.

## [TS.28.533-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 28.533 (V0.3.0): Management and orchestration of networks and network slicing; Management and orchestration architecture (Release 15)", June 2018, <[http://www.3gpp.org/ftp//Specs/archive/28\\_series/28.533/28533-030.zip](http://www.3gpp.org/ftp//Specs/archive/28_series/28.533/28533-030.zip)>.



## [TS.29.244-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.244 (V15.1.0): Interface between the Control Plane and the User Plane Nodes; Stage 3", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/29\\_series/29244-f10.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/29_series/29244-f10.zip)>.

## [TS.29.274-3GPP]

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

## [TS.29.281-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.281 (V15.4.0): GPRS Tunneling Protocol User Plane (GTPv1-U)", September 2018, <[http://www.3gpp.org/ftp//Specs/archive/29\\_series/29.281/29281-f40.zip](http://www.3gpp.org/ftp//Specs/archive/29_series/29.281/29281-f40.zip)>.

## [TS.29.510-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.510 (V15.0.0): 5G System; Network Function Repository Services; Stage 3", June 2018, <[http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29\\_series/29510-f00.zip](http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29_series/29510-f00.zip)>.

## [TS.29.561-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 29.561 (V15.0.0): 5G System; Interworking between 5G Network and external Data Networks; Stage 3", June 2018, <[http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29\\_series/29561-f00.zip](http://www.3gpp.org/FTP/Specs/2018-06/Rel-15/29_series/29561-f00.zip)>.

## [TS.38.300-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.300 (v15.1.0): NR and NG-RAN Overall Description; Stage 2", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38\\_series/38300-f10.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38_series/38300-f10.zip)>.

## [TS.38.401-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.401 (v15.1.0): NG-RAN; Architecture Description", March 2018, <[http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38\\_series/38401-f10.zip](http://www.3gpp.org/FTP/Specs/2018-03/Rel-15/38_series/38401-f10.zip)>.

[TS.38.415-3GPP]

3rd Generation Partnership Project (3GPP), "3GPP TS 38.415  
(v15.1.0): NG-RAN; PDU Session User Plane protocol",  
September 2018, <[http://www.3gpp.org/ftp//Specs/  
archive/38\\_series/38.415/38415-f10.zip](http://www.3gpp.org/ftp//Specs/archive/38_series/38.415/38415-f10.zip)>.

#### Authors' Addresses

Shunsuke Homma  
NTT

Email: [homma.shunsuke@lab.ntt.co.jp](mailto:homma.shunsuke@lab.ntt.co.jp)

Takuya Miyasaka  
KDDI Research

Email: [ta-miyasaka@kddi-research.jp](mailto:ta-miyasaka@kddi-research.jp)

Satoru Matsushima  
SoftBank

Email: [satoru.matsushima@g.softbank.co.jp](mailto:satoru.matsushima@g.softbank.co.jp)

Daniel Voyer  
Bell Canada

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

dmm  
Internet-Draft  
Intended status: Standards Track  
Expires: October 25, 2019

S. Homma  
K. Kawakami  
NTT  
A. Akhavain  
Huawei Canada Research Centre  
A. Rodriguez-Natal  
R. Shekhar  
Cisco Systems Inc.  
April 23, 2019

Co-existence of 3GPP 5GS and Identifier-Locator Separation Architecture  
draft-homma-dmm-5gs-id-loc-coexistence-03

## Abstract

This document describes an approach to introduce Identifier Locator Separation architecture into 3GPP 5GS with low-impact on its specifications, and shows the features and considerations of this approach.

## Status of This Memo

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

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

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

This Internet-Draft will expire on October 25, 2019.

## Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                                    | 3  |
| 2. Definition of Terms . . . . .                                             | 3  |
| 2.1. Terms of ID-LOC Protocols . . . . .                                     | 4  |
| 2.2. Terms of 5GS . . . . .                                                  | 5  |
| 3. Mechanism on Data Plane . . . . .                                         | 5  |
| 4. Mechanisms on Control Plane . . . . .                                     | 10 |
| 4.1. Model 1: Independent Control Planes . . . . .                           | 11 |
| 4.2. Model 2: Interworking Control Planes . . . . .                          | 11 |
| 4.3. Model 3: Integrated Control Planes . . . . .                            | 12 |
| 5. Features Analysis . . . . .                                               | 12 |
| 5.1. Benefits . . . . .                                                      | 12 |
| 5.2. Issues . . . . .                                                        | 12 |
| 6. Security Considerations . . . . .                                         | 12 |
| 7. IANA Considerations . . . . .                                             | 12 |
| 8. Acknowledgement . . . . .                                                 | 13 |
| 9. Informative References . . . . .                                          | 13 |
| Appendix A. Case Studies on Use of LISP . . . . .                            | 15 |
| A.1. UE-to-UE Communication . . . . .                                        | 16 |
| A.1.1. Case A-1: UEs allocated different dUPF . . . . .                      | 16 |
| A.1.2. Case A-2: UEs allocated the same xTR . . . . .                        | 18 |
| A.1.3. Consideration of Case that UE Moves to under Another<br>xTR . . . . . | 19 |
| A.2. UE-to-dDN Communication . . . . .                                       | 19 |
| A.2.1. Case A-3: UE communicates with neighbor dDN . . . . .                 | 19 |
| A.2.2. Case A-4: UE communicates with non-neighbor dDN . . . . .             | 21 |
| A.3. UE-to-cDN/Internet Communication . . . . .                              | 22 |
| A.3.1. Case A-5: UE communicates with cDN . . . . .                          | 23 |
| Appendix B. Case Studies on Use of ILA . . . . .                             | 24 |
| B.1. UE-to-UE Communications . . . . .                                       | 25 |
| B.1.1. Case B-1: UEs allocated different dUPF . . . . .                      | 25 |
| B.1.2. Case B-2: UEs allocated the same ILA node . . . . .                   | 26 |
| B.2. UE-to-dDN Communication . . . . .                                       | 28 |
| B.2.1. Case B-3: UE communicates with neighbor dDN . . . . .                 | 28 |
| B.2.2. Case B-4: UE communicates with non-neighbor dDN . . . . .             | 30 |
| B.3. UE-to-cDN/Internet Communication . . . . .                              | 32 |
| B.3.1. Case B-5: Internet Communication . . . . .                            | 33 |
| B.3.2. Case B-6: Internet Communication . . . . .                            | 34 |
| Authors' Addresses . . . . .                                                 | 36 |

## 1. Introduction

Identifier-Locator Separation (ID-LOC) architectures aim to simplify management of network, devices, and sessions by employing two namespaces: Identifier for device's identity, and Locator for its location in the network.

An ID-LOC architecture can be implemented by a dedicated protocol such as LISP [I-D.ietf-lisp-rfc6833bis], ILA [I-D.herbert-intarea-ila], ILNP [RFC6740], etc. The control plane of such ID-LOC protocols can be combined with one of different encapsulation techniques such as GTP-U [TS.29281], SRv6 [I-D.filsfils-spring-srv6-network-programming], MPLS [RFC3031], VXLAN [RFC7348], etc. at data plane to provide a customized solution. Furthermore, regarding control plane of ID-LOC, it can optionally even take advantage of enhanced PUB/SUB capable distributed databases to store ID-LOC mappings.

ID-LOC protocols are also expected to be used for optimizing user-plane of mobile network [I-D.bogineni-dmm-optimized-mobile-user-plane]. Different alternatives to introduce ID-LOC architecture into 3GPP 5GS (5th Generation System), are under consideration in related IETF WG such as DMM WG.

Introducing ID-LOC architecture into mobile network can involve modifications to 5GS architecture and specifications that might span over several 5GS releases.

Therefore, an approach that enables the introduction of ID-LOC architecture into 5GS without change of its specifications and supports migration path toward a native ID-LOC network can be useful to operators. Here, ID-LOC native network refers to a network that employs the ID-LOC architecture as only mechanism for packet forwarding.

The document aims to describe one such approach and clarify different features, and benefits.

## 2. Definition of Terms

This section describes general terms of ID-LOC architecture. This document also refers definitions of 3GPP 5GS [TS.23.501-3GPP], and some of such terms which are used in this document are listed in this section.

The LISP terms are described in [I-D.ietf-lisp-rfc6833bis].

The ILA terms are described in [I-D.herbert-intarea-ila].

The SRv6 terms are described in [I-D.ietf-6man-segment-routing-header].

## 2.1. Terms of ID-LOC Protocols

**Device Identifier (ID):** An ID is an identifier of host or end point such as UE or network function including VM instance, container, etc. In ID-LOC architectures, an IP or MAC address is generally assigned to an end device as identifier. In this case, IDs are used as values for the source and destination IP/MAC address fields of packets sent from end points. Alternatively, other attributes of the end point, such as its Fully Qualified Domain Name (FQDN), can also be used as IDs.

**Locator (LOC):** A LOC is generally an address (e.g. IPv4, IPv6, MAC, etc) of the ID-LOC node. In the case of SRv6 it can be the ID-LOC node's local SID representing the segment for which the ID-LOC node is the segment termination node.

**ID-LOC node;** An ID-LOC node is a node that has at least one unique locator within a network domain, and has functionalities to obtain destination locator and to forward packets to the ID-LOC node which has the destination locator. This node has an ID-LOC mapping cache and obtains destination locator by looking up destination ID (destination address of a data packet) from the mapping cache. If ID of the received packet is not present in its own mapping cache, an ID-LOC node requests mapping information of the ID and the assigned locator to ID-to-LOC mapping system. Also an ID-LOC node forwards packet to a peered LOC node by encapsulation or conversion of the IP header field such as IP address field, and decapsulates or reconverts packets received from another ID-LOC node. Different implementations of ID-LOC architecture use different forwarding mechanisms. LISP data-plane, for example, uses IPv4/v6 header and LISP header for encapsulation, whereas ILA tightly couples itself with IPv6, and SRv6 uses IPv6 and SIDs (Segment Identifiers).

**ID-to-LOC Mapping System:** An ID-to-LOC mapping system is a database which contains all known ID-to-LOC mappings within an ID-LOC domain. The mapping information is updated when an end point moves to under another ID-LOC node. This database can be logically centralized, distributed across the ID-LOC nodes, or a combination of both. If the database is logically centralized, each ID-LOC node has an interface to the system to send a request and receive mapping information.

**ID-to-LOC Mapping Cache:** An ID-LOC mapping cache is a table in an ID-LOC node that stores ID-to-LOC mapping information and it is used for obtaining destination LOC from ID of received packet. ID-to-LOC mapping cache typically contains a small piece of database. The cache is updated when the ID-LOC node receives a new ID-to-LOC mapping information from ID-to-LOC mapping system.

## 2.2. Terms of 5GS

**User Plane Function (UPF):** An UPF handles the user plane paths. An UPF is connected to SMF with N4 interface. More detailed information is described in [TS.23.501-3GPP]. This document defines two types of UPF, Central UPF (cUPF) and Distributed UPF (dUPF). Their features are described in Section 3

**Uplink Classifier (ULCL):** An ULCL is an UPF functionality that aims at diverting Uplink traffic, based on filter rules provided by SMF, towards Data Network (DN).

**Data Network (DN):** A DN is a network where network functions and entities, including operator or 3rd party services, are deployed. This document defines two types of DN, Central DN (cDN) and Distributed DN (dDN). Their features are described in Section 3.

**Radio Access Network (RAN):** A RAN is an access network where radio bearer sent by UEs traverse. A RAN encapsulate users' packets with GTP-U.

**Session Management Function (SMF):** An SMF is a function which provides control plane functionalities for handling user traffic.

**Application Function (AF):** An AF is a control plane functionality and connected to SMF with Naf interfaces.

## 3. Mechanism on Data Plane

This approach achieves traffic forwarding with optimized path and session continuity by using ID-LOC and ULCL for particular communication including UE-to-UE or MEC (Mobile Edge Computing) communication. ULCL is one of fundamental functions of 5GC Rel.15 and it provides functionalities of packet filtering and divert for uplink packets sent by UEs.

The overview of the assumed 5GC architecture of data plane where the proposal approach works is shown in Figure 1. The details of numbered interfaces in the figure are described in [TS.23.501-3GPP].





addresses (IDs) for each UE. The traffic transmitted from UEs are basically sent to the cUPF.

- o Distributed UPFs (dUPFs) and Distributed DN (dDN) are deployed and geographically distributed at user edge side. A unique address space (it's not necessarily globally unique) is assigned to dDN. When a dUPF forwards a UE's uplink packet, and if the subnet of the destination address is the same as the one assigned to dDN at proximity, then dUPF, with the help of ULCL, may divert the packet to that dDN. Here, the ULCL identifies each encapsulated uplink packet to be diverted, by checking if the destination of the inner packet is one of IP addresses assigned to the dDN. A dUPF removes GTP-U header from the packets, and sends them to dDN via N6. When dUPF receives packets from dDN, dUPF encapsulates them with GTP-U header, and merges them into downlink packets from cUPF. An overview of behaviors of dUPF and ULCL is shown in Figure 2.
- o Network topology between RAN and dUPF/cUPF adopts tree structure and the section between RAN and dUPF and the section between dUPF and cUPF are connected with GTP-U.

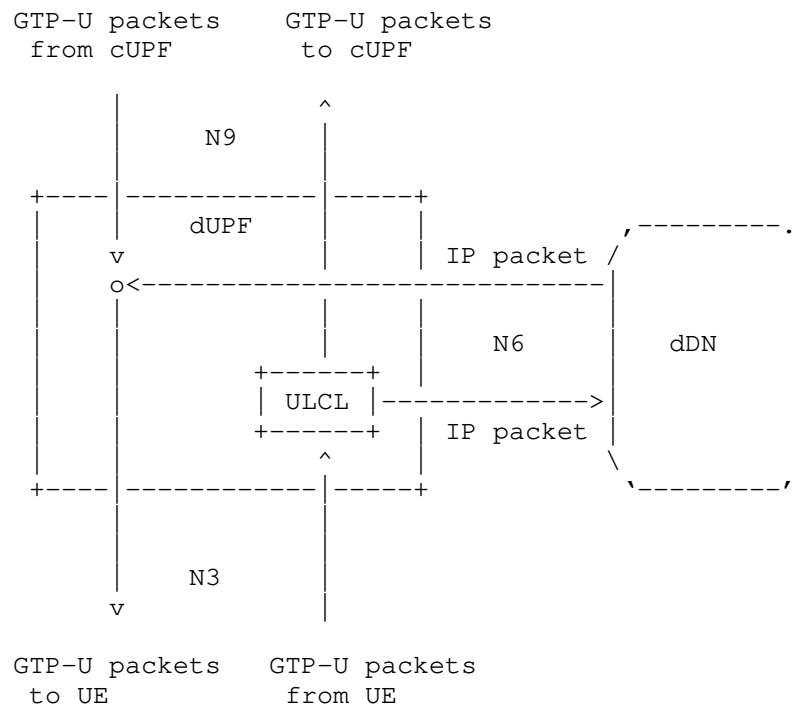


Figure 2: Behaviors of dUPF and ULCL

In the proposed approach, IDs are assumed to be IP addresses and an ID-LOC node is installed between dUPF and dDN. ID-LOC nodes are connected with a IP mechanism such as IP tunnels or translation of destination IP field. As examples of such data plane protocols for providing connectivity between ID-LOC nodes, IPv4/v6 header with LISP header or SRv6 ([I-D.ietf-6man-segment-routing-header]) can be used. In addition, each ID-LOC node has connectivity with one or more Mapping Systems. The overview is shown in Figure 3.

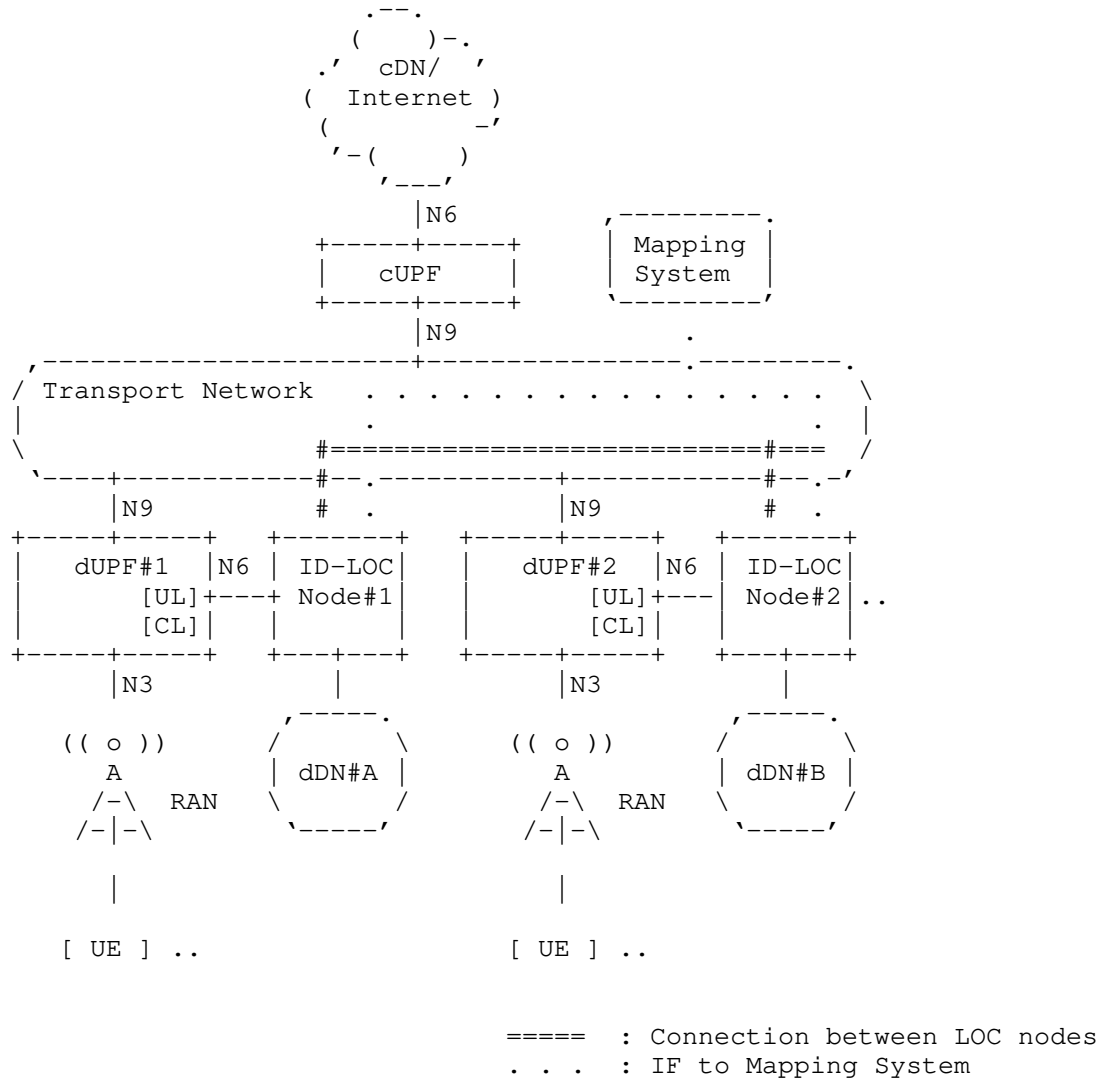


Figure 3: Proposal Network Architecture

Each dUPF has a filter table of ULCL. Each filter table is configured to match the addresses of UEs within the network domain (i.e., addresses for UEs assigned by the cUPF). Filter tables can also be configured to match the address corresponding to the address space (or part of) corresponding with the dDNs in the network domain. UPFs monitor each uplink GTP-U packet with its ULCL and divert it to the connected ID-LOC node with decapsulation of GTP-U if the

destination address of the inner packet (payload) matches the filtering table. When ID-LOC node receives a packet from the dUPF, it obtains LOC which the destination of the packet (ID) belongs to by looking up its own ID-to-LOC mapping table or querying it from the Mapping System according ID-LOC mechanism. Then it sends the packet to peered ID-LOC node indicated by the LOC. The peered ID-LOC node converts the received packet to appropriate form and forwards them the destination by following its own forwarding table.

From such processes, forwarding paths of user traffic diverted by ULCL from 5GC to ID-LOC node are optimized.

A cUPF is connected with dUPFs via N9 interface and packets are forwarded with GTP-U encapsulation between cUPF and dUPF.

Some case studies of ID-LOC protocols are described in Appendix A and Appendix B.

#### 4. Mechanisms on Control Plane

For ID-LOC mechanism in mobile networks, a control plane mechanism is required to manage location information of UEs and NFs in each dDN. There are mainly three models to realize control plane mechanism for ID-LOC as follows:

Model 1: Independent Control Planes

Model 2: Interworking Control Planes

Model 3: Integrated Control Planes

Some of models may require to use 5GS interfaces or add some functionalities to functions of 5GC. 5GS architecture and the service-based interfaces are shown in Figure 4. The details of functions and interfaces are described in [TS.23.501-3GPP].

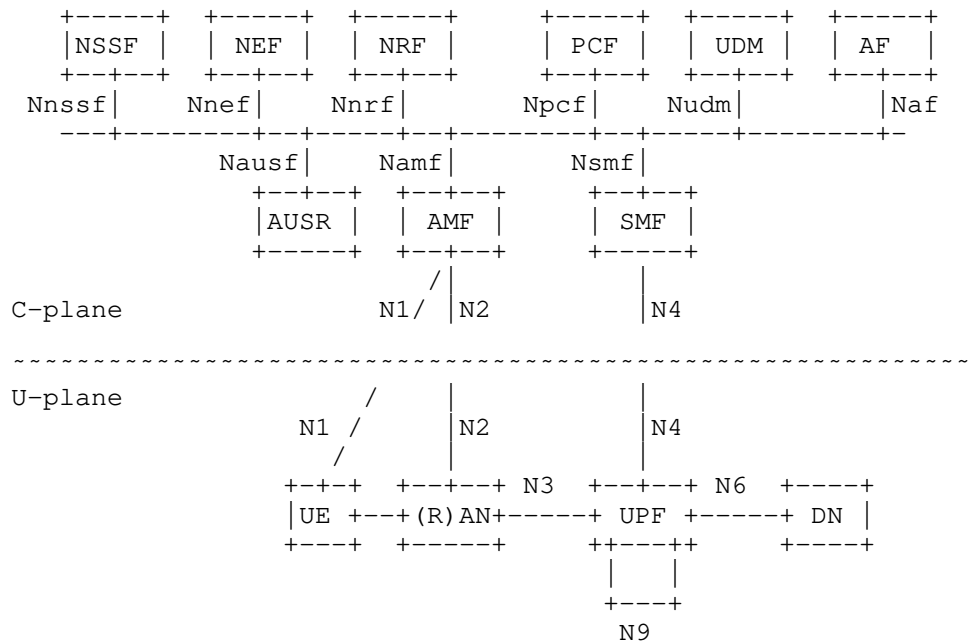


Figure 4: 5GS Architecture and Service-based Interfaces

#### 4.1. Model 1: Independent Control Planes

In this model, control plane of 5GC and ID-to-LOC mapping mechanism are completely separated. Information of a UE and an ID-LOC node which the UE is attached is sent to a mapping system and registered in the mapping database only when the ID-LOC node receives a packet from the UE and the UE is not registered yet.

This model does not cause any impacts on 5GC architecture. However, in this model, a UE cannot be accessed from other UEs within the same network domain until a packet from the UE is diverted to the ID-LOC node by the UPF which the UE is located and the ID and LOC are registered to the Mapping System.

#### 4.2. Model 2: Interworking Control Planes

In this model, a mapping system interworks with an SMF which manages sessions of each UE. A scheme to inform, that a UE moves and is relocated to another UPF, from SMF to AF via Naf interface is defined in 5GS ([TS.23.502-3GPP])\* . A Mapping System is installed as an AF and obtains mobility information of UEs with the above scheme.

\* The stage 3 of discussion of 5GS has not been fixed yet and the specification may be changed.

This model would not cause any impacts on 5GS architecture, and a mapping system can always keep the current mobility information of each UE.

#### 4.3. Model 3: Integrated Control Planes

In this model, SMF functionalities are integrated into a mapping system. In other words, the mapping system becomes a part of 5GS. In 5GS architecture, an SMF has a role of session management of UEs, and it updates its own mapping database depending on movement of a UE.

This approach enables to always keep mapping databases the latest status, however, it obviously requires extension or replacement of SMF actually deployed in 5GS network.

### 5. Features Analysis

#### 5.1. Benefits

- o This approach provides a mechanism for introducing ID-LOC architecture into 5GS with no or nominal impact, and achieves optimized forwarding with session continuity in the assumed use cases such as UE-to-UE or UE-to-dDN communications.
- o Regarding communication to the cDN, this approach can keep scalability because it does not change the current mechanism of 5GS.

#### 5.2. Issues

- o dUPF and ID-LOC node are separated, and thus an extra hop may occur against the optimized forwarding. However, it can be resolved by implementing dUPF and ID-LOC node within a same box or application.

### 6. Security Considerations

TBD

### 7. IANA Considerations

This memo includes no request to IANA.

## 8. Acknowledgement

The authors would like to thank Ryosuke Kurebayashi, Koji Tsubouchi, Toru Okugawa, and Dino Farinacci for their kind reviews and technical feedback.

## 9. Informative References

- [I-D.bogineni-dmm-optimized-mobile-user-plane]  
Bogineni, K., Akhavain, A., Herbert, T., Farinacci, D., Rodriguez-Natal, A., Carofiglio, G., Auge, J., Muscariello, L., Camarillo, P., and S. Homma, "Optimized Mobile User Plane Solutions for 5G", draft-bogineni-dmm-optimized-mobile-user-plane-01 (work in progress), June 2018.
- [I-D.farinacci-lisp-mobile-network]  
Farinacci, D., Pillay-Esnault, P., and U. Chunduri, "LISP for the Mobile Network", draft-farinacci-lisp-mobile-network-05 (work in progress), March 2019.
- [I-D.filsfils-spring-srv6-network-programming]  
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-07 (work in progress), February 2019.
- [I-D.herbert-intarea-ila]  
Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", draft-herbert-intarea-ila-01 (work in progress), March 2018.
- [I-D.ietf-6man-segment-routing-header]  
Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-18 (work in progress), April 2019.
- [I-D.ietf-lisp-eid-mobility]  
Portoles-Comeras, M., Ashtaputre, V., Moreno, V., Maino, F., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", draft-ietf-lisp-eid-mobility-03 (work in progress), November 2018.
- [I-D.ietf-lisp-predictive-rlocs]  
Farinacci, D. and P. Pillay-Esnault, "LISP Predictive RLOCs", draft-ietf-lisp-predictive-rlocs-03 (work in progress), November 2018.

- [I-D.ietf-lisp-pubsub]  
Rodriguez-Natal, A., Ermagan, V., Leong, J., Maino, F.,  
Cabellos-Aparicio, A., Barkai, S., Farinacci, D.,  
Boucadair, M., Jacquenet, C., and S. Secci, "Publish/  
Subscribe Functionality for LISP", draft-ietf-lisp-  
pubsub-03 (work in progress), March 2019.
- [I-D.ietf-lisp-rfc6830bis]  
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A.  
Cabellos-Aparicio, "The Locator/ID Separation Protocol  
(LISP)", draft-ietf-lisp-rfc6830bis-26 (work in progress),  
November 2018.
- [I-D.ietf-lisp-rfc6833bis]  
Fuller, V., Farinacci, D., and A. Cabellos-Aparicio,  
"Locator/ID Separation Protocol (LISP) Control-Plane",  
draft-ietf-lisp-rfc6833bis-24 (work in progress), February  
2019.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol  
Label Switching Architecture", RFC 3031,  
DOI 10.17487/RFC3031, January 2001,  
<<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC6740] Atkinson, R.J. and SN. Bhatti, "Identifier-Locator Network  
Protocol (ILNP) Architectural Description", RFC 6740,  
DOI 10.17487/RFC6740, November 2012,  
<<https://www.rfc-editor.org/info/rfc6740>>.
- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The  
Locator/ID Separation Protocol (LISP) for Multicast  
Environments", RFC 6831, DOI 10.17487/RFC6831, January  
2013, <<https://www.rfc-editor.org/info/rfc6831>>.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller,  
"Interworking between Locator/ID Separation Protocol  
(LISP) and Non-LISP Sites", RFC 6832,  
DOI 10.17487/RFC6832, January 2013,  
<<https://www.rfc-editor.org/info/rfc6832>>.
- [RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-  
Pascual, J., and D. Lewis, "Locator/Identifier Separation  
Protocol (LISP) Network Element Deployment  
Considerations", RFC 7215, DOI 10.17487/RFC7215, April  
2014, <<https://www.rfc-editor.org/info/rfc7215>>.



- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <<https://www.rfc-editor.org/info/rfc8061>>.
- [RFC8111] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and A. Smirnov, "Locator/ID Separation Protocol Delegated Database Tree (LISP-DDT)", RFC 8111, DOI 10.17487/RFC8111, May 2017, <<https://www.rfc-editor.org/info/rfc8111>>.
- [TS.23.501-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TS 23.501", December 2017, <[http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.501](http://www.3gpp.org/ftp//Specs/archive/23_series/23.501)>.
- [TS.23.502-3GPP]  
3rd Generation Partnership Project (3GPP), "3GPP TS 23.502", December 2017, <[http://www.3gpp.org/ftp//Specs/archive/23\\_series/23.502](http://www.3gpp.org/ftp//Specs/archive/23_series/23.502)>.
- [TS.29281]  
3GPP, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)", 3GPP TS 29.281 15.1.0, December 2017.

#### Appendix A. Case Studies on Use of LISP

This Appendix describes detailed processes of the proposal approach with LISP mechanism in the following types of communications.

1. UE-to-UE Communication
2. UE-to-dDN Communication
3. UE-to-cDN/Internet Communication

In the following description of case studies, ID and Locator are called EID (End-point Identifier) and RLOC (Routing Locator) in LISP terms. Mapping Server has the master of EID-to-RLOC mapping database, and each xTR (Ingress/Egress Tunnel Router) has EID-to-RLOC mapping cache. An xTR obtains the destination RLOC from its own

cache by looking up the destination EID of received packet. They obtain mappings from the mapping system if an EID looked up is not registered in the cache. Packets are passed between xTRs with some tunnel protocols.

#### A.1. UE-to-UE Communication

In the current architecture, a cUPF becomes an anchor point for UEs, and all packets between UEs even those which are located to the same dUPF are transferred through the anchor point. This may cause communication delay and inefficient resource usage. In the proposed procedure, packets can be transferred without going through an anchor point, and low latency and efficient resource usage can be achieved.

The UE-to-UE communications include communications between UEs located to different dUPFs (Case 1), and communication between UEs located to the same dUPF (Case 2). In this section, the detailed procedures of the cases are described.

Moreover, in a mobile network, a UE may move during communications. This section describes considerations about UE's handover in such case.

##### A.1.1. Case A-1: UEs allocated different dUPF

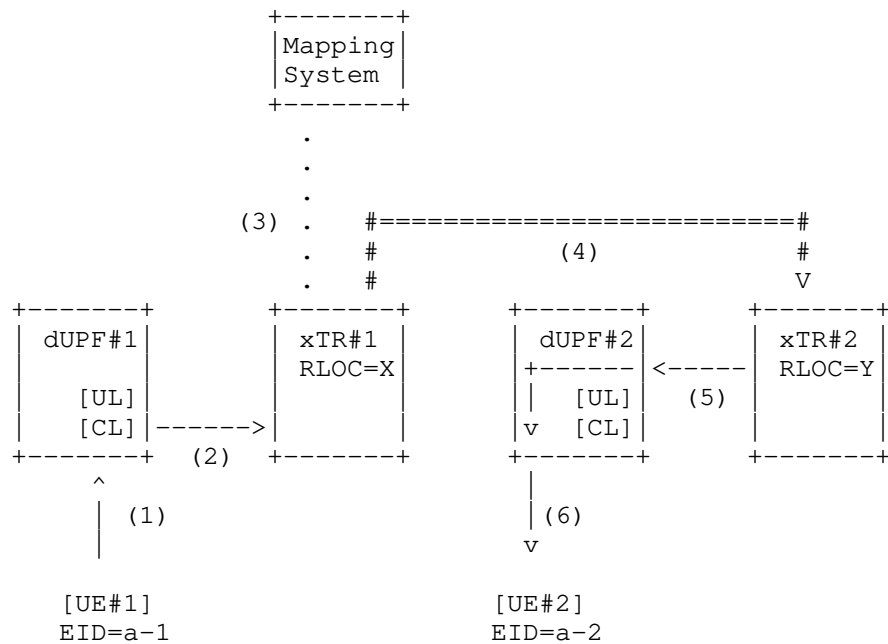


Figure 5: Procedure in Case A-1

- (0) Within this network, addresses are assigned to UEs from a address space [A]. These addresses are described as a-n (n=1,2,..). EID=a-1 and a-2 are assigned to UE#1 and UE#2.
- (1) UE#1 sends packets to UE#2 with setting EID=a-2 as the destination IP address.
- (2) dUPF#1 monitors inner packet of received GTP-U packet and divert it to xTR#1 with decapsulation if the destination address is one of address space [A].
- (3) xTR#1 updates own EID-to-RLOC mapping cache by interaction with Mapping System (if needed).
- (4) xTR#1 obtains the RLOC(=Y) of EID=a-2 from the EID-to-RLOC mapping cache, and sends the packets to the xTR#2 with a tunnel with RLOC=Y as the destination address.
- (5) xTR#2 decapsulate the packets, and sends them to dUPF#2.
- (6) dUPF#2 encapsulate packets with GTP-U header, and sends them to UE#2.

## A.1.2. Case A-2: UEs allocated the same xTR

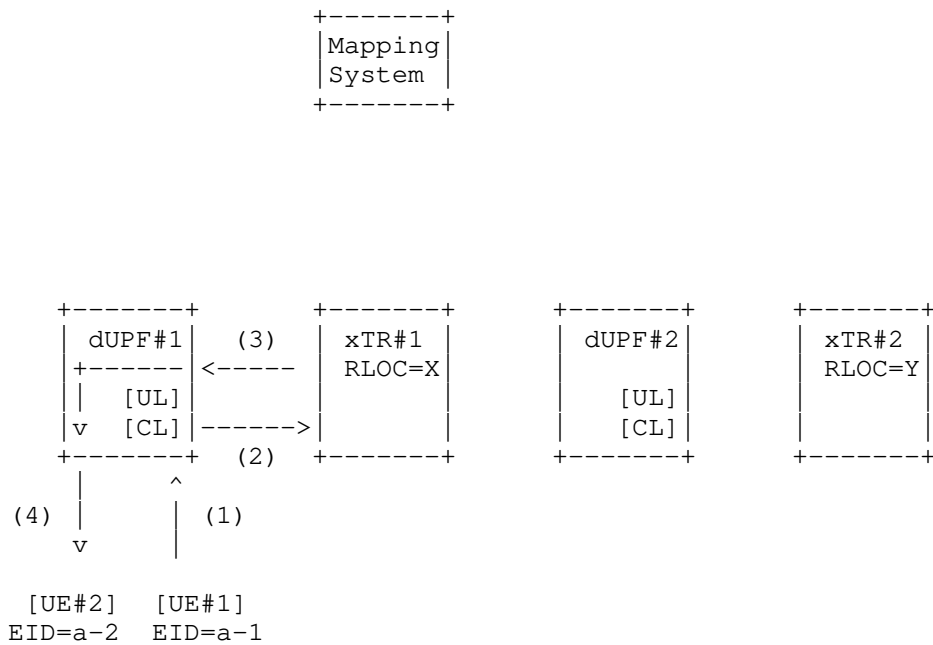


Figure 6: Procedure in Case A-2

- (0) Within this network, addresses are assigned to UEs from a address space [A]. These addresses are described as a-n (n=1,2,..). EID=a-1 and a-2 are assigned to UE#1 and UE#2.
- (1) UE#1 sends packets to UE#2 with setting EID=a-2 as the destination IP address.
- (2) dUPF#1 monitors inner packets of received GTP-U traffic and divert it to xTR#1 with decapsulation if the destination address is one of address space [A].
- (3) Since xTR#1 serves UE#2, it locally routes the traffic for EID=a-2. xTR#1 sends the received packets back to dUPF#1.
- (4) dUPF#1 encapsulate packets with GTP-U, and sends them to UE#2.

### A.1.3. Consideration of Case that UE Moves to under Another xTR

When a UE moves to a serving area of another dUPF during communication with another UE, EID-to-RLOC mapping database of a Mapping System and the tables of the xTR and the peered xTR must be updated. Unless some of the mechanism described below are in place, the xTRs can't send packets to the appropriate xTR during the updating, and thus packet drop or stalling may occur.

For example, a mechanism that immediately advertise the update of location of UEs to Mapping System and the appropriate xTRs depending on movement of each UE might be required. Some documents (e.g., [I-D.ietf-lisp-eid-mobility], [I-D.ietf-lisp-pubsub]) discuss such mechanisms. Alternatively, a mechanism that replicates packets to both the old and new location while the UE is in transit could also be used. This approach is discussed in detail in [I-D.ietf-lisp-predictive-rlocs].

### A.2. UE-to-dDN Communication

The UE-to-dDN communications basically correspond the communication between a UE and neighbor dDN (Case3). On the other hand, if a UE moved under another dUPF during usage of a stateful application, or the application is not uniformly deployed in every dDN, the UE needs to continue to communicate with the previous dDN (Case4).

In such cases, in the current architecture, all packets are needed to go through the anchor point or dynamic GTP tunnel reconfiguration between dUPF is required. The former solution causes additional communication delay and inefficient resource usage. The latter solution increase the cost of 5GS control plane to dynamically update the GTP tunnel with multiple UPFs and their ULCL filter tables along with the movement of the UE. The proposed approach achieves appropriate packet transfer in such cases.

In this section, the detailed procedures of communications between a UE and neighbor dDN and communications between a UE and non-neighbor dDN

#### A.2.1. Case A-3: UE communicates with neighbor dDN

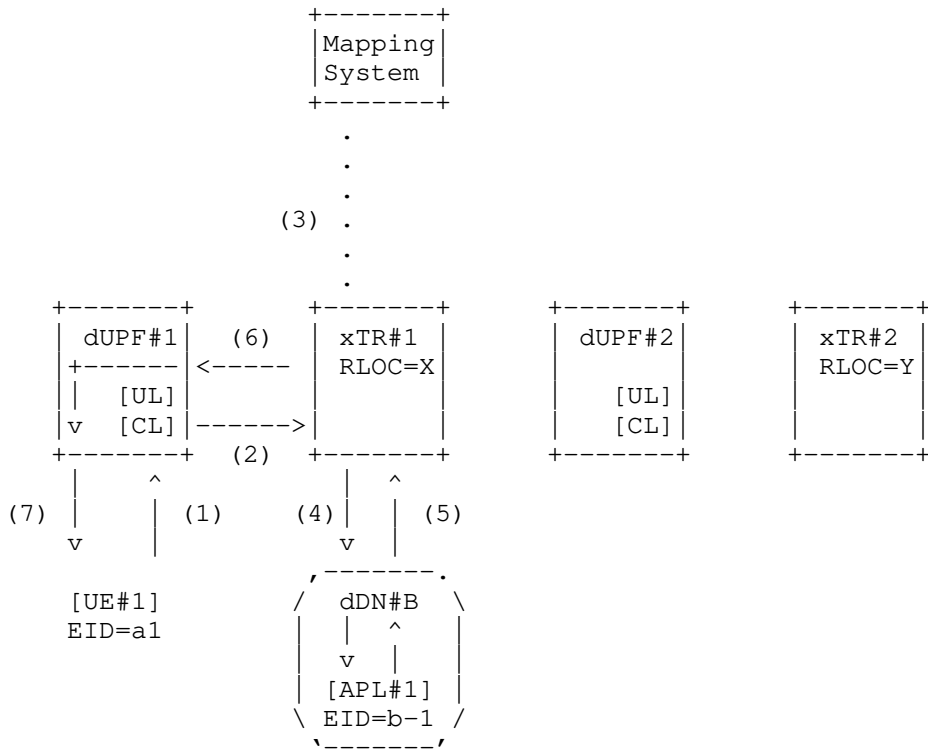


Figure 7: Procedure in Case A-3

- (0) Within this network, UEs are assigned their addresses from an address space [A]. These addresses are described as a-n (n=1,2,...). Also, applications in dDN#B are assigned their addresses from a address space [B]. These addresses are described as b-n (n=1,2,...). EID=a-1 and b-1 assigned to UE#1 and APL#1 which is located in dDN#B.

[Uplink Processes]

- (1) UE#1 sends packets to dDN#B with setting EID=b-1 as the destination IP address.
- (2) dUPF#1 monitors inner of received GTP-U packets and divert it to xTR#1 with decapsulation if the destination IP address is one of address space [B].

- (3) xTR#1 updates own EID-to-RLOC mapping cache by interaction with Mapping System (if needed). Or xTR#1 may update its own cache by a Map-Notify message when an APL is deployed or deleted in dDB#B.
- (4) Since xTR#1 serves dDN#B, it locally routes the traffic for EID=b-1. xTR#1 sends the packets to the dDN#B.

[Downlink Processes]

- (5) APL#1 in dDN#B sends packets to UE#1 with setting EID=a-1 as the destination IP address.
- (6) Since xTR#1 serves UE#1, it locally routes the traffic for EID=a-1. xTR#1 sends packets to dUPF#1.
- (7) dUPF#2 encapsulates packets with GTP-U, and sends them to UE#1.

A.2.2. Case A-4: UE communicates with non-neighbor dDN

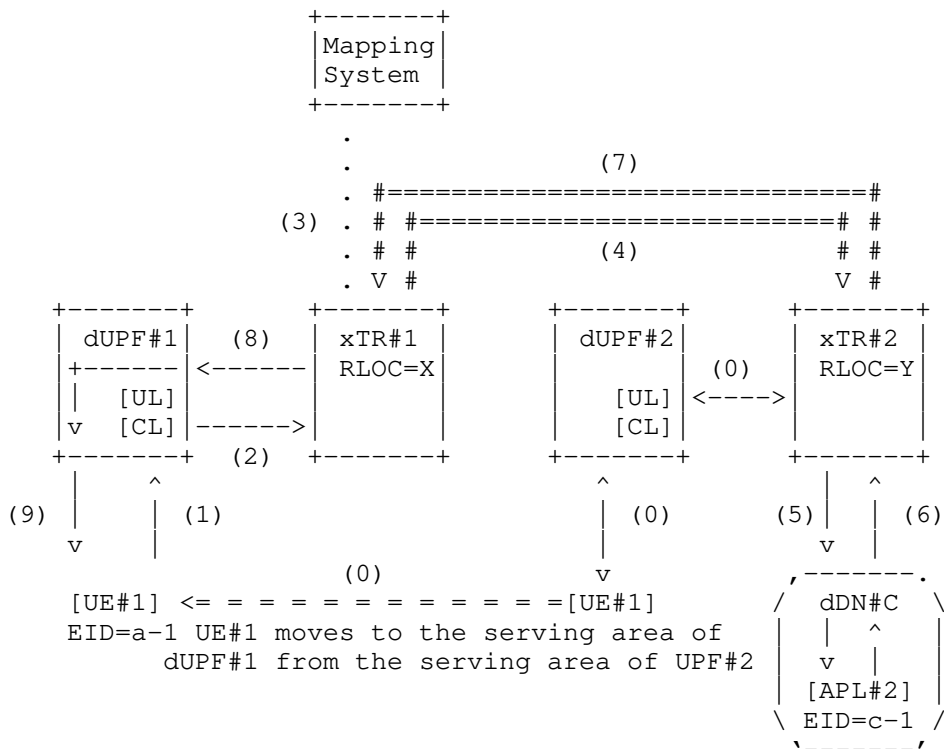


Figure 8: Procedure in Case A-4

- (0) Within this network, UEs are assigned their addresses from an address space [A]. These addresses are described as a-n (n=1,2,...). And applications in dDN#C are assigned their addresses from an address space [C]. These addresses are described as c-n (n=1,2,...). EID=a-1 and c-1 assigned to UE#1 and APL#2 which is located in dDN#C. UE#1 has moved to the serving area of dUPF#1 from the serving area of UPF#2 while communicating to APL#2.

[Uplink Processes]

- (1) UE#1 sends packets to APL#2 with setting EID=c-1 as the destination IP address.
- (2) dUPF#1 monitors each inner packet of received GTP-U traffic and divert it to xTR#1 with decapsulation if the destination address is one of address space [C].
- (3) xTR#1 updates own EID-to-RLOC mapping cache by interaction with Mapping System (if needed).
- (4) xTR#1 obtains RLOC(=Y) of EID=c-1 from the EID-to-RLOC mapping cache, and sends the packet to the xTR#2 with a tunnel with RLOC=Y as the destination address.
- (5) xTR#2 decapsulates the packets received from xTR#1, and sends them to dDN#C depending on its forwarding table.

[Downlink Processes]

- (6) APL#2 sends packets to UE#1 with setting EID=a-1 as the destination IP address.
- (7) xTR#2 obtains RLOC(=X) of EID=a-1 from the EID-to-RLOC mapping cache, and sends the packets to the xTR#1 with a tunnel with RLOC=X as the destination address.
- (8) xTR#1 decapsulates the packets received from xTR#2m and sends them to the dUPF#1 depending on its forwarding table.
- (9) dUPF#1 encapsulates the packets with GTP-U and sends packets to UE#1.

### A.3. UE-to-cDN/Internet Communication

UE-to-cDN/Internet communication is achieved by GTP-U mechanism originally equipped in 3GPP 5GS architecture. In this section, we



describe processes of UE-to-cDN communication in the proposal architecture as an example.

#### A.3.1. Case A-5: UE communicates with cDN

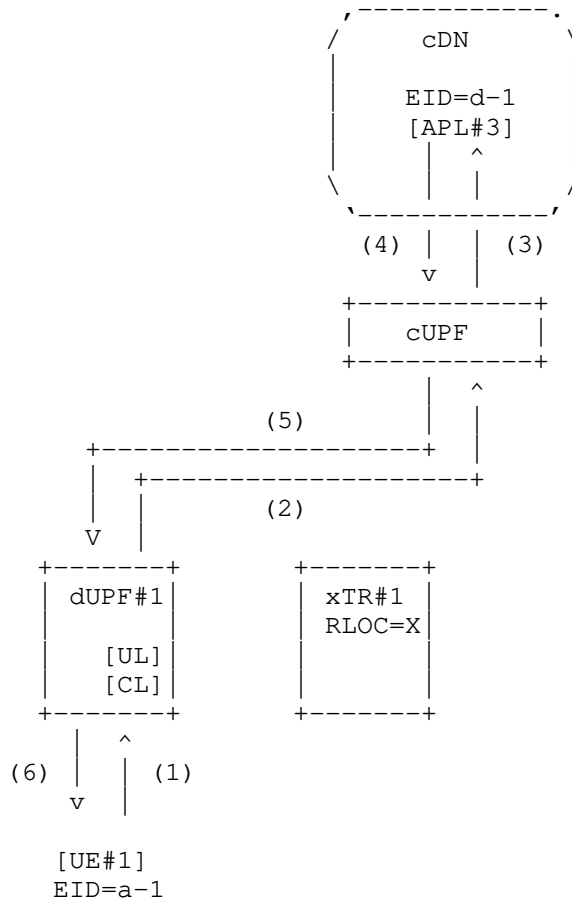


Figure 9: Procedure in Case A-5

- (0) Within this network, UEs are assigned their addresses from an address space [A]. These addresses are described as a-n (n=1,2,..). And applications in cDN are assigned their addresses from an address space [D]. These addresses are described as d-n (n=1,2,..). EID=a-1 and d-1 assigned to UE#1 and APL#3 which is located in cDN.

#### [Uplink Processes]

- (1) UE#1 sends packets to cDN with setting EID=d-1 as the destination IP address.
- (2) dUPF#1 monitors inner of received GTP-U packets. Since the destination IP address (EID=d-1) does not hit the filter of ULCL, dUPF#1 re-encapsulates the packet to another GTP-U connecting to cUPF and forwards to cUPF.
- (3) cUPF decapsulates GTP-U packets and forwards them to APL#3 in cDN depending on its own forwarding table.

#### [Downlink Processes]

- (4) APL#3 in cDN sends packets to UE#1 with setting EID=a-1 as the destination IP address.
- (5) cUPF encapsulates the packets received from APL#3 and forwards them to dUPF#1 depending on its own forwarding table.
- (6) dUPF re-encapsulates the packets to another GTP-U and forwards to UE#1.

### Appendix B. Case Studies on Use of ILA

This Appendix describes detailed processes of the proposal approach with ILA mechanism in the following types of communications.

1. UE-to-UE Communication
2. UE-to-dDN Communication
3. UE-to-cDN/Internet Communication

Each ILA node has ID-to-LOC mapping table. Mappings are propagated amongst ILA routers or hosts in a network using mapping propagation protocols.

In the following description of case studies, a mapping system, called ILA resolver in ILA terms, has the master of ID-to-LOC mapping database, and each ILA node obtains mappings from the mapping system. In some cases, each ILA node has an ID-to-LOC mapping database.

In ILA, an SIR address expressed by composition of SIR prefix and identifier is assigned to each UE or VM instance. An SIR prefix and an identifier are described  $SIR\_prefix\_n$  and  $id\_m$  ( $n=1,2,\dots$ ,  $m=1,2,\dots$ ), and an SIR address is expressed as  $SIR\_addr\_x = [n,m]$

(x=1,2,...) in the following description. Also, each ILA-Nodes are assigned unique Locators, which is a network prefix that routes to a host. Locators are described as loc\_n (n=1,2,...).

#### B.1. UE-to-UE Communications

The overview of this communication type is described in A.1.

##### B.1.1. Case B-1: UEs allocated different dUPF

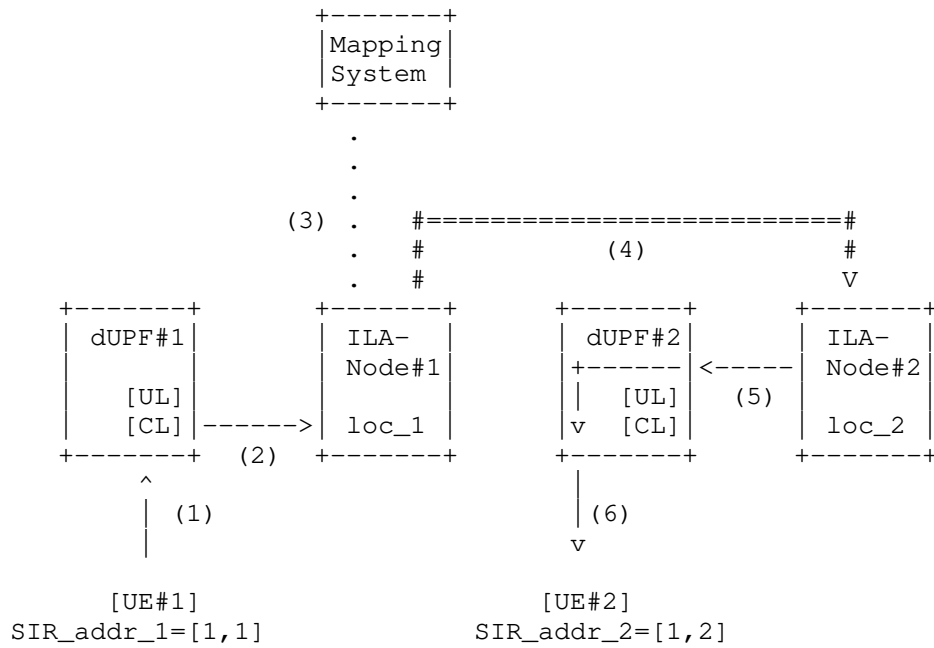


Figure 10: Procedure in Case B-1

- (0) Within this network, UEs are belonged to the same ILA domain, and the same SIR prefix is assigned to UEs. SIR\_addr\_1=[1,1] and SIR\_addr\_2=[1,2] are assigned to UE#1 and UE#2.
- (1) UE#1 sends packets to UE#2 with setting SIR\_addr\_2 as the destination IP address.

- (2) dUPF#1 monitors inner packet of received GTP-U packet and diverts it to ILA-Node#1 with decapsulation if the prefix of the destination address is SIR\_prefix\_1.
- (3) ILA-Node#1 updates own ID-to-LOC mapping table by interaction with the mapping system (if needed).
- (4) ILA-Node#1 obtains loc\_2 as Locator of the ILA node#2 from the ID-to-LOC mapping table. ILA-Node#1 converts the prefixes of the source and destination addresses to loc\_1 (Locator of id\_1) and loc\_2 (Locator of id\_2). ILA-Node#1 sends the packet to the ILA-Node#2.
- (5) ILA-Node#2 receives the packet and converts the prefixes of the source and destination addresses to SIR\_prefix\_1, and then sends packets to dUPF#2.
- (6) dUPF#2 encapsulate packets with GTP-U header, and sends them to UE#2.

B.1.2. Case B-2: UEs allocated the same ILA node

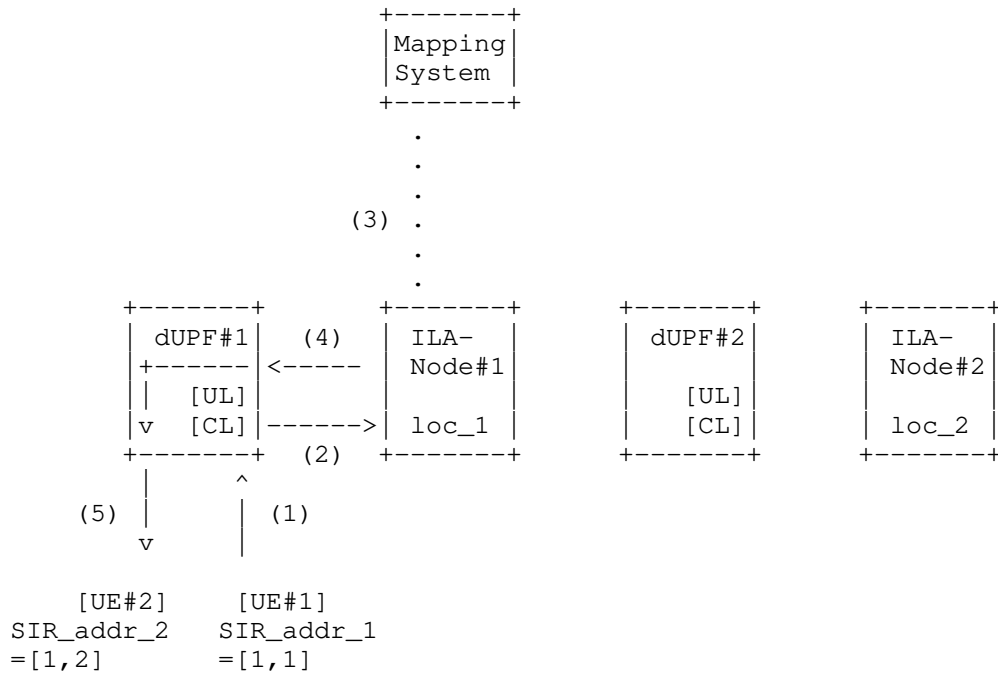


Figure 11: Procedure in Case B-2

- (0) Within this network, UEs are belonged to the same ILA domain, and the same SIR prefix is assigned to UEs.  $SIR\_addr\_1=[1,1]$  and  $SIR\_addr\_2=[1,2]$  are assigned to UE#1 and UE#2.
- (1) UE#1 sends packets to UE#2 with setting  $SIR\_addr\_2$  as the destination IP address.
- (2) dUPF#1 monitors inner packet of received GTP-U packet and diverts it to ILA-Node#1 with decapsulation if the prefix of the destination address is  $SIR\_prefix\_1$ .
- (3) ILA-node#1 updates own ID-to-LOC mapping table by interaction with Mapping System (if needed).

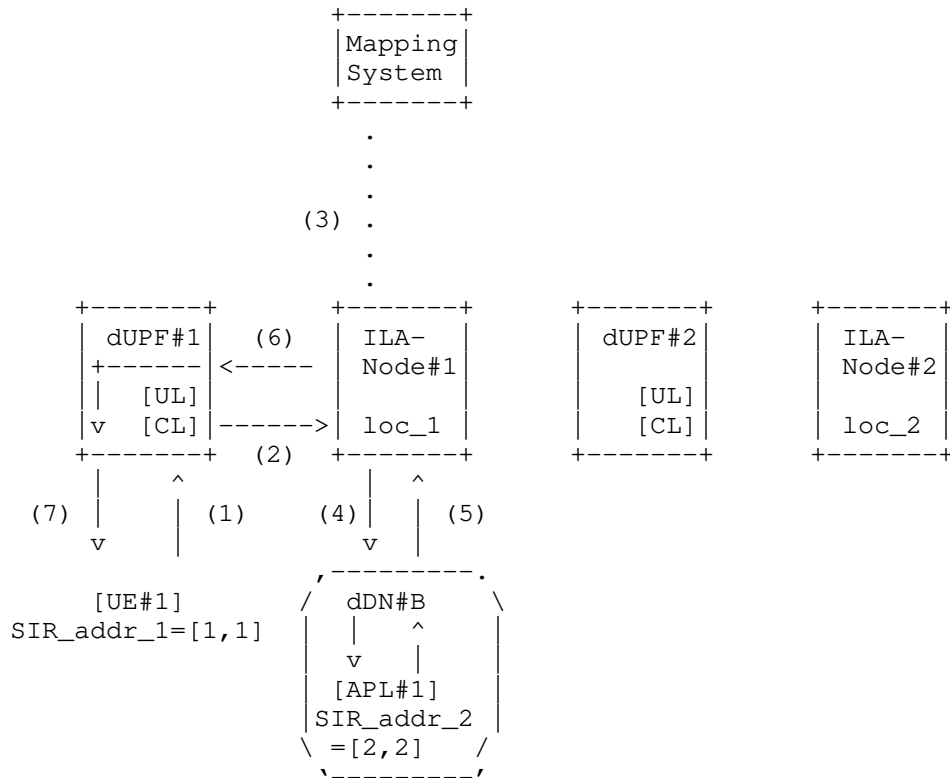
(4) ILA-Node#1 obtains loc\_1 as Locator of ILA node#2 from the ID-to-LOC mapping table. Since loc\_1 indicates itself, ILA-Node#1 sends the packets back to dUPF#1.

(5) dUPF#1 encapsulate packets with GTP-U, and sends them to UE#2.

#### B.2. UE-to-dDN Communication

The overview of this communication type is described in A.2.

##### B.2.1. Case B-3: UE communicates with neighbor dDN



Legend: SIR\_addr\_x=[(SIR\_Prefix), (Identifier)]

Figure 12: Procedure in Case B-3

- (0) Within this network, UEs are belonged to the same ILA domain, and the same SIR prefix (SIR\_prefix\_1) are assigned to UEs. Applications in dDN#B are belonged to different ILA domain. and different SIR prefix (SIR\_prefix\_2) is assigned to these applications. SIR\_addr\_1=[1,1] and SIR\_addr\_2=[2,2] are assigned to UE#1 and APL#1. APL#1 is located in dDN#B.

#### Uplink Processes

- (1) UE#1 sends packets to APL#1 with setting SIR\_addr\_2 as the destination IP address.

- (2) dUPF#1 monitors inner packet of received GTP-U packet and diverts it to ILA-Node#1 with decapsulation if the prefix of the destination address is SIR\_prefix\_2.
- (3) ILA-Node#1 updates own ID-to-LOC mapping table by interaction with a mapping system (if needed). Or ILA-Node#1 may update its own table by a Map-Notify message when an APL is deployed or deleted in dDB#B.
- (4) ILA-Node#1 obtains loc\_1 as Locator of id\_2 from the ID-to-LOC mapping table. Since loc\_1 indicates itself, ILA-Node#1 sends the packets to the dDN#B.

#### Downlink Processes

- (5) APL#1 in dDN#B sends packets to UE#1 with setting SIR\_address\_1 as the destination IP address.
- (6) ILA-Node#1 obtains loc\_1 as Locator of id\_1 from the ID-to-LOC mapping table. Since loc=1 indicates itself, ILA-Node#1 sends packets to dUPF#1.
- (7) dUPF#2 encapsulates packets with GTP-U, and sends them to UE#1.

#### B.2.2. Case B-4: UE communicates with non-neighbor dDN





- (2) dUPF#1 monitors inner packet of received GTP-U packet and diverts it to ILA-Node#1 with decapsulation if the prefix of the destination address is SIR\_prefix\_3.
- (3) ILA-Node#1 updates own ID-to-LOC mapping table by interaction with Mapping System (if needed).
- (4) ILA-Node#1 obtains loc\_2 as Locator of id\_3 from the ID-to-LOC mapping table. ILA-Node#1 converts the prefix of the source address to loc\_1 (Locator of id\_1), and the prefix of the destination address to loc\_2 (Locator of id\_3). ILA-Node#1 sends the packet to the ILA-Node#2.
- (5) ILA-Node#2 converts the prefix of the source address to SIR\_prefix\_1, and the prefix of the destination address to SIR\_prefix\_3, and then sends packets to dDN#C depending on its forwarding table.

#### Downlink Processes

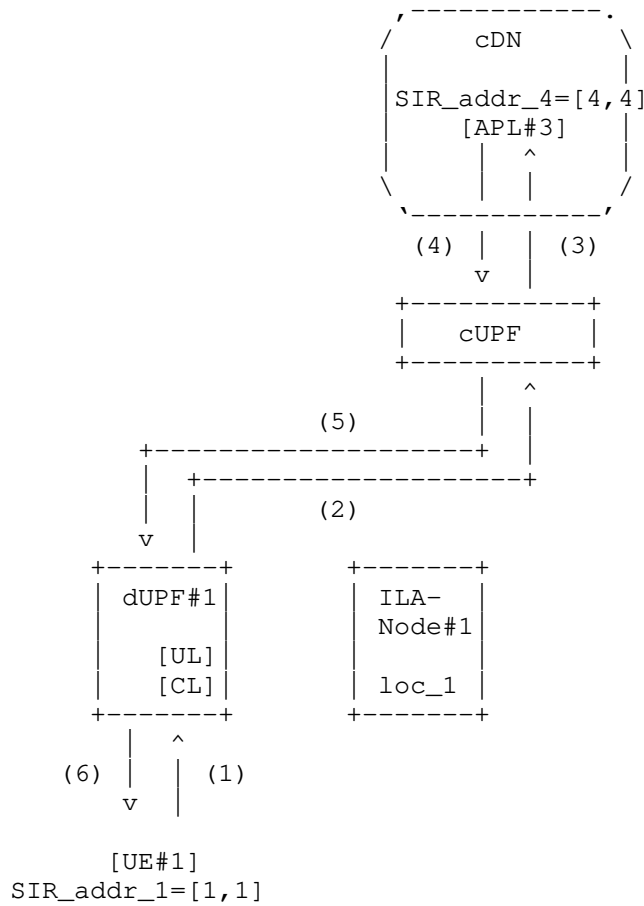
- (6) APL#2 sends packets to UE#1 with setting SIR\_address\_1 as the destination IP address.
- (7) ILA-Node#2 obtains loc\_1 as Locator of id\_1 from the ID-to-LOC mapping table. ILA-Node#2 converts the prefix of the source address to loc\_2 (Locator of id\_3), and the prefix of the destination address to loc\_1 (Locator of id\_1). ILA-Node#1 sends the packet to the ILA-Node#1.
- (8) ILA-Node#1 converts the prefix of the source address to SIR\_prefix\_3, and the prefix of the destination address to SIR\_prefix\_1, and then sends packets to d#UPF1 depending on its forwarding table.
- (9) dUPF#1 encapsulates the packets with GTP-U and sends packets to UE#1.

#### B.3. UE-to-cDN/Internet Communication

UE-to-cDN/Internet communication are basically achieved by GTP-U mechanism originally equipped in 3GPP 5GS architecture. ILA causes some limitation on IP addressing to UEs (e.g., all UEs in an ILA domain have the same SIR prefix), and thus some IP translation node such as NAT (Network Address Translation) may be required to enable UEs to access to external network. In this section, we describe processes of UE-to-cDN/Internet communication in the proposal architecture. In Internet communication, from aspect of privacy or routing with external network, SIR addresses assigned to UEs are

translated by NAT function deployed between dUPF and connection point.

### B.3.1. Case B-5: Internet Communication



Legend:  $SIR\_addr\_x=[(SIR\_Prefix), (Identifier)]$

Figure 14: Procedure in Case B-5

- (0) Within this network, UEs are belonged to the same ILA domain, and the same SIR prefix ( $SIR\_prefix\_1$ ) are assigned to UEs. Applications in cDN are belonged to different ILA domain. and

different SIR prefix (SIR\_prefix\_4) is assigned to these applications. SIR\_addr\_1=[1,1] and SIR\_addr\_4=[4,4] are assigned to UE#1 and APL#3. APL#3 is located in cDN.

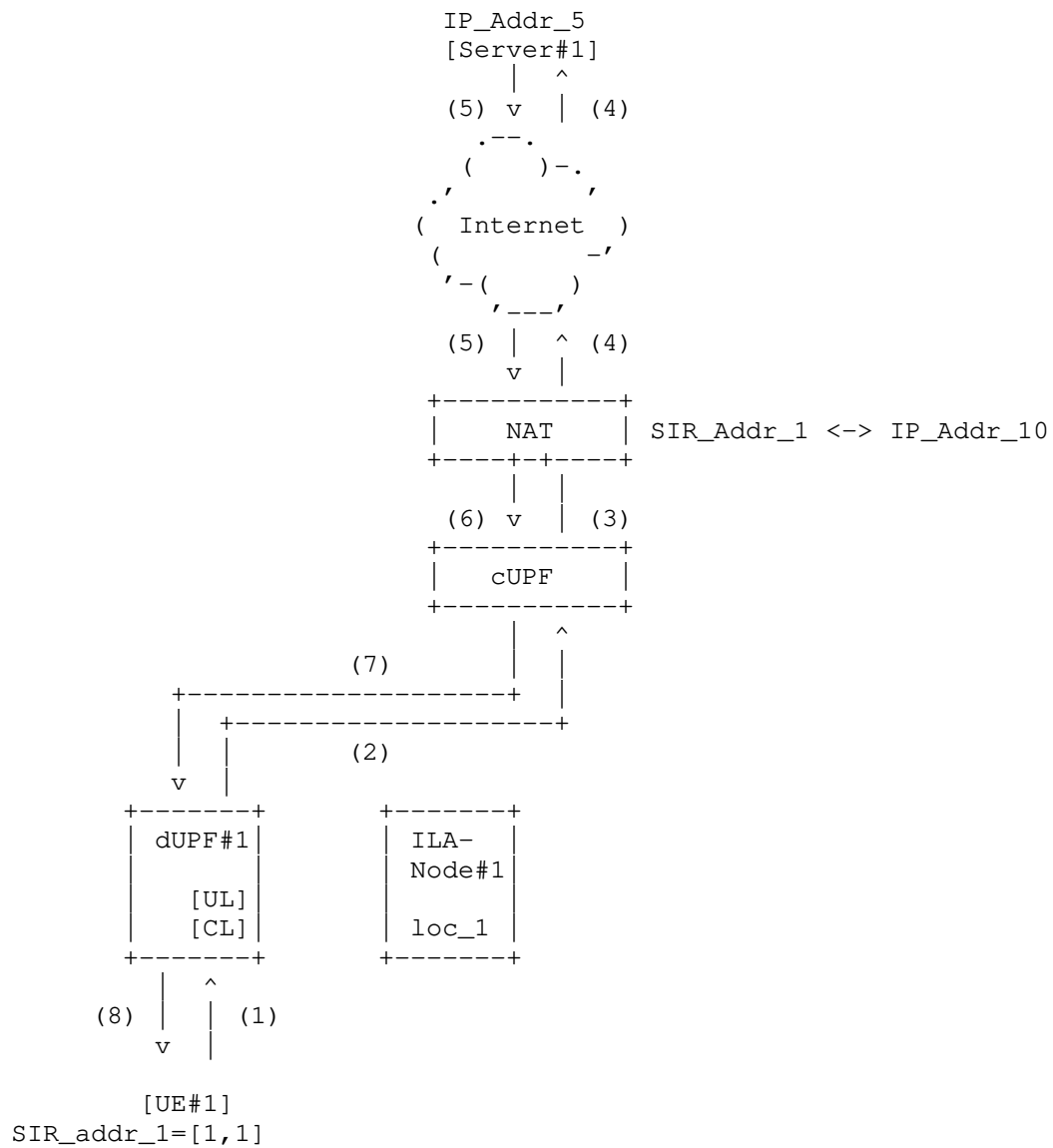
#### Uplink Processes

- (1) UE#1 sends packets to APL#3 with setting SIR\_adrr\_4 as the destination IP address.
- (2) dUPF#1 monitors inner of received GTP-U packets. Since the destination IP address (SIR\_adder\_4) does not hit the filter of ULCL, dUPF#1 re-encapsulates the packet to another GTP-U connecting to cUPF and forwards to cUPF.
- (3) cUPF decapusalates GTP-U packets and forwards them to APL#3 in cDN depending on its own forwarding table.

#### Downlink Processes

- (4) APL#3 in cDN sends packets to UE#1 with setting SIR\_addr\_1 as the destination IP address.
- (5) cUPF encapsulates the packets received from APL#3 and forwards them to dUPF#1 with GTP-U encapsulation depending on its own forwarding table.
- (6) dUPF re-encapsulates the packets to another GTP-U and forwards to UE#1.

#### B.3.2. Case B-6: Internet Communication



Legend: `SIR_addr_x=[ (SIR_Prefix), (Identifier) ]`

Figure 15: Procedure in Case B-6

- (0) Within this network, UEs are belonged to the same ILA domain, and the same SIR prefix (SIR\_prefix\_1) are assigned to UEs. SIR\_addr\_1=[1,1] assigned to UE#1 and server#1 has IP\_addr\_5. UE#1 communicate with server#1 over Internet.

#### Uplink Processes

- (1) UE#1 sends packets to server#1 with setting IP\_adrr\_5 as the destination IP address.
- (2) dUPF#1 monitors inner of received GTP-U packets. Since the destination IP address (SIR\_adddr\_4) does not hit the filter of ULCL, dUPF#1 re-encapsulates the packet to another GTP-U connecting to cUPF and forwards to cUPF.
- (3) cUPF decapusalates GTP-U packets and forwards them to Internet depending on its own forwarding table.
- (4) NAT translates SIR\_adddr\_1 of received packets to IP\_addr\_10 and the packets are forwarded to server#1 over Internet.

#### Downlink Processes

- (5) Server#1 sends packets to UE#1 with setting IP\_adddr\_1 as the destination IP address.
- (6) NAT translates IP\_addr\_10 of received packets to SIR\_adddr\_1, and packets are sent to cUPF.
- (7) cUPF encapsulates the packets with GTP-U and sends them to dUPF#1 depending on its own forwarding table.
- (8) dUPF re-encapsulates the packets to another GTP-U and forwards to UE#1.

#### Authors' Addresses

Shunsuke Homma  
NTT  
3-9-11, Midori-cho  
Musashino-shi, Tokyo 180-8585  
Japan

Email: shunsuke.homma.fp@hco.ntt.co.jp

Kenta Kawakami  
NTT  
3-9-11, Midori-cho  
Musashino-shi, Tokyo 180-8585  
Japan

Email: kawakami.kenta@lab.ntt.co.jp

Arashmid Akhavain  
Huawei Canada Research Centre  
Canada

Email: arashmid.akhavain@huawei.com

Alberto Rodriguez-Natal  
Cisco Systems Inc.  
USA

Email: natal@cisco.com

Ravi Shekhar  
Cisco Systems Inc.  
India

Email: ravishek@cisco.com

DMM  
Internet-Draft  
Intended status: Informational  
Expires: September 8, 2020

H. Chan, Ed.  
X. Wei  
Huawei Technologies  
J. Lee  
Sangmyung University  
S. Jeon  
Sungkyunkwan University  
CJ. Bernardos, Ed.  
UC3M  
March 7, 2020

Distributed Mobility Anchoring  
draft-ietf-dmm-distributed-mobility-anchoring-15

Abstract

This document defines distributed mobility anchoring in terms of the different configurations and functions to provide IP mobility support. A network may be configured with distributed mobility anchoring functions for both network-based or host-based mobility support according to the needs of mobility support. In a distributed mobility anchoring environment, multiple anchors are available for mid-session switching of an IP prefix anchor. To start a new flow or to handle a flow not requiring IP session continuity as a mobile node moves to a new network, the flow can be started or re-started using an IP address configured from the new IP prefix anchored to the new network. If the flow needs to survive the change of network, there are solutions that can be used to enable IP address mobility. This document describes different anchoring approaches, depending on the IP mobility needs, and how this IP address mobility is handled by the network.

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 September 8, 2020.

## Copyright Notice

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

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

## Table of Contents

|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                                                                      | 2  |
| 2. Conventions and Terminology . . . . .                                                                       | 3  |
| 3. Distributed Mobility Anchoring . . . . .                                                                    | 6  |
| 3.1. Configurations for Different Networks . . . . .                                                           | 6  |
| 3.1.1. Network-based DMM . . . . .                                                                             | 7  |
| 3.1.2. Client-based DMM . . . . .                                                                              | 8  |
| 4. IP Mobility Handling in Distributed Anchoring Environments -<br>Mobility Support Only When Needed . . . . . | 9  |
| 4.1. Nomadic case (no need of IP mobility): Changing to new IP<br>prefix/address . . . . .                     | 10 |
| 4.2. Mobility case, traffic redirection . . . . .                                                              | 12 |
| 4.3. Mobility case, anchor relocation . . . . .                                                                | 15 |
| 5. Security Considerations . . . . .                                                                           | 16 |
| 6. IANA Considerations . . . . .                                                                               | 17 |
| 7. Contributors . . . . .                                                                                      | 17 |
| 8. References . . . . .                                                                                        | 18 |
| 8.1. Normative References . . . . .                                                                            | 18 |
| 8.2. Informative References . . . . .                                                                          | 19 |
| Authors' Addresses . . . . .                                                                                   | 20 |

## 1. Introduction

A key requirement in distributed mobility management [RFC7333] is to enable traffic to avoid traversing a single mobility anchor far from an optimal route. This document defines different configurations, functional operations and parameters for distributed mobility anchoring and explains how to use them to avoid unnecessarily long routes when a mobile node moves.

Companion distributed mobility management documents are already addressing source address selection [RFC8653], and control-plane data-plane signaling [I-D.ietf-dmm-fpc-cpdp]. A number of distributed mobility solutions have also been proposed, for example, in [I-D.seite-dmm-dma], [I-D.ietf-dmm-pmipv6-dlif], [I-D.sarikaya-dmm-for-wifi], [I-D.yhkim-dmm-enhanced-anchoring], and [I-D.matsushima-stateless-uplane-vepc].

Distributed mobility anchoring employs multiple anchors in the data plane. In general, control plane functions may be separated from data plane functions and be centralized but may also be co-located with the data plane functions at the distributed anchors. Different configurations of distributed mobility anchoring are described in Section 3.1.

As a Mobile Node (MN) attaches to an access router and establishes a link between them, a /64 IPv6 prefix anchored to the router may be assigned to the link for exclusive use by the MN [RFC6459]. The MN may then configure a global IPv6 address from this prefix and use it as the source IP address in a flow to communicate with its Correspondent Node (CN). When there are multiple mobility anchors assigned to the same MN, an address selection for a given flow is first required before the flow is initiated. Using an anchor in a MN's network of attachment has the advantage that the packets can simply be forwarded according to the forwarding table. However, after the flow has been initiated, the MN may later move to another network which assigns a new mobility anchor to the MN. Since the new anchor is located in a different network, the MN's assigned prefix does not belong to the network where the MN is currently attached.

When the MN wants to continue using its assigned prefix to complete ongoing data sessions after it has moved to a new network, the network needs to provide support for the MN's IP address and session continuity, since routing packets to the MN through the new network deviates from applying default routes. The IP session continuity needs of a flow (application) determines how the IP address used by this flow has to be anchored. If the ongoing IP flow can cope with an IP prefix/address change, the flow can be reinitiated with a new IP address anchored in the new network. On the other hand, if the ongoing IP flow cannot cope with such change, mobility support is needed. A network supporting a mix of flows both requiring and not requiring IP mobility support will need to distinguish these flows.

## 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.

All general mobility-related terms and their acronyms used in this document are to be interpreted as defined in the Mobile IPv6 (MIPv6) base specification [RFC6275], the Proxy Mobile IPv6 (PMIPv6) specification [RFC5213], the "Mobility Related Terminologies" [RFC3753], and the DMM current practices and gap analysis [RFC7429]. These include terms such as Mobile Node (MN), Correspondent Node (CN), Home Agent (HA), Home Address (HoA), Care-of-Address (CoA), Local Mobility Anchor (LMA), and Mobile Access Gateway (MAG).

In addition, this document uses the following terms and definitions:

**IP session continuity:** The ability to maintain an ongoing transport interaction by keeping the same local endpoint IP address throughout the lifetime of the IP socket despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may change after closing the IP socket and before opening a new one, but that does not jeopardize the ability of applications using these IP sockets to work flawlessly. Session continuity is essential for mobile hosts to maintain ongoing flows without any interruption [RFC8653].

**Higher layer session continuity:** The ability to maintain an ongoing transport or higher layer (e.g., application) interaction by keeping the session identifiers throughout the lifetime of the session despite the mobile host changing its point of attachment within the IP network topology. This can be achieved by using mechanisms at the transport or higher layers.

**IP address reachability:** The ability to maintain the same IP address for an extended period of time. The IP address stays the same across independent sessions, even in the absence of any session. The IP address may be published in a long-term registry (e.g., DNS) and is made available for serving incoming (e.g., TCP) connections. IP address reachability is essential for mobile hosts to use specific/published IP addresses [RFC8653].

**IP mobility:** Combination of IP address reachability and session continuity.

**Home network of a home address:** the network that has assigned the HoA used as the session identifier by the application running in

an MN. The MN may be running multiple application sessions, and each of these sessions can have a different home network.

**Anchoring (of an IP prefix/address):** An IP prefix, i.e., Home Network Prefix (HNP), or address, i.e., HoA, assigned for use by an MN is topologically anchored to an anchor node when the anchor node is able to advertise a route into the routing infrastructure for the assigned IP prefix. The traffic using the assigned IP address/prefix must traverse the anchor node. We can refer to the function performed by IP anchor node as anchoring, which is a data plane function.

**Location Management (LM) function:** control plane function that keeps and manages the network location information of an MN. The location information may be a binding of the advertised IP address/prefix, e.g., HoA or HNP, to the IP routing address of the MN or of a node that can forward packets destined to the MN.

When the MN is a Mobile Router (MR), the location information will also include the Mobile Network Prefix (MNP), which is the aggregate IP prefix delegated to the MR to assign IP prefixes for use by the Mobile Network Nodes (MNNs) in the mobile network.

In a client-server protocol model, secure (i.e., authenticated and authorized) location query and update messages may be exchanged between a Location Management client (LMc) and a Location Management server (LMs), where the location information can be updated or queried from the LMc. Optionally, there may be a Location Management proxy (LMp) between LMc and LMs.

With separation of control plane and data plane, the LM function is in the control plane. It may be a logical function at the control plane node, control plane anchor, or mobility controller.

It may be distributed or centralized.

**Forwarding Management (FM) function:** packet interception and forwarding to/from the IP address/prefix assigned for use by the MN, based on the internetwork location information, either to the destination or to some other network element that knows how to forward the packets to their destination.

This function may be used to achieve traffic indirection. With separation of control plane and data plane, the FM function may

split into a FM function in the data plane (FM-DP) and a FM function in the control plane (FM-CP).

FM-DP may be distributed with distributed mobility management. It may be a function in a data plane anchor or data plane node.

FM-CP may be distributed or centralized. It may be a function in a control plane node, control plane anchor or mobility controller.

Home Control-Plane Anchor (Home-CPA or H-CPA): The Home-CPA function hosts the mobile node (MN)'s mobility session. There can be more than one mobility session for a mobile node and those sessions may be anchored on the same or different Home-CPA's. The home-CPA will interface with the home-DPA for managing the forwarding state.

Home Data Plane Anchor (Home-DPA or H-DPA): The Home-DPA is the topological anchor for the MN's IP address/ prefix(es). The Home-DPA is chosen by the Home-CPA on a session- basis. The Home-DPA is in the forwarding path for all the mobile node's IP traffic.

Access Control Plane Node (Access-CPN or A-CPN): The Access-CPN is responsible for interfacing with the mobile node's Home-CPA and with the Access-DPN. The Access-CPN has a protocol interface to the Home-CPA.

Access Data Plane Node (Access-DPN or A-DPN): The Access-DPN function is hosted on the first-hop router where the mobile node is attached. This function is not hosted on a layer-2 bridging device such as a eNode(B) or Access Point.

### 3. Distributed Mobility Anchoring

#### 3.1. Configurations for Different Networks

We next describe some configurations with multiple distributed anchors. To cover the widest possible spectrum of scenarios, we consider architectures in which the control and data planes are separated. We analyze where LM and FM functions -- which are specific sub-functions involved in mobility management -- can be placed when looking at the different scenarios with distributed anchors.

### 3.1.1. Network-based DMM

Figure 1 shows a general scenario for network-based distributed mobility management.

The main characteristics of a network-based DMM solution are:

- o There are multiple data plane anchors, each with a FM-DP function.
- o The control plane may either be distributed (not shown in the figure) or centralized (as shown in the figure).
- o The control plane and the data plane (Control Plane Anchor -- CPA -- and Data Plane Anchor -- DPA) may be co-located or not. If the CPA is co-located with the distributed DPAs, then there are multiple co-located CPA-DPA instances (not shown in the figure).
- o An IP prefix/address IP1 (anchored to the DPA with IP address IPa1) is assigned for use to a MN. The MN uses this IP1 address to communicate with CNs (not shown in the figure).
- o The location management (LM) function may be co-located or split (as shown in the figure) into a separate server (LMs) and a client (LMc). In this case, the LMs may be centralized whereas the LMc may be distributed or centralized.

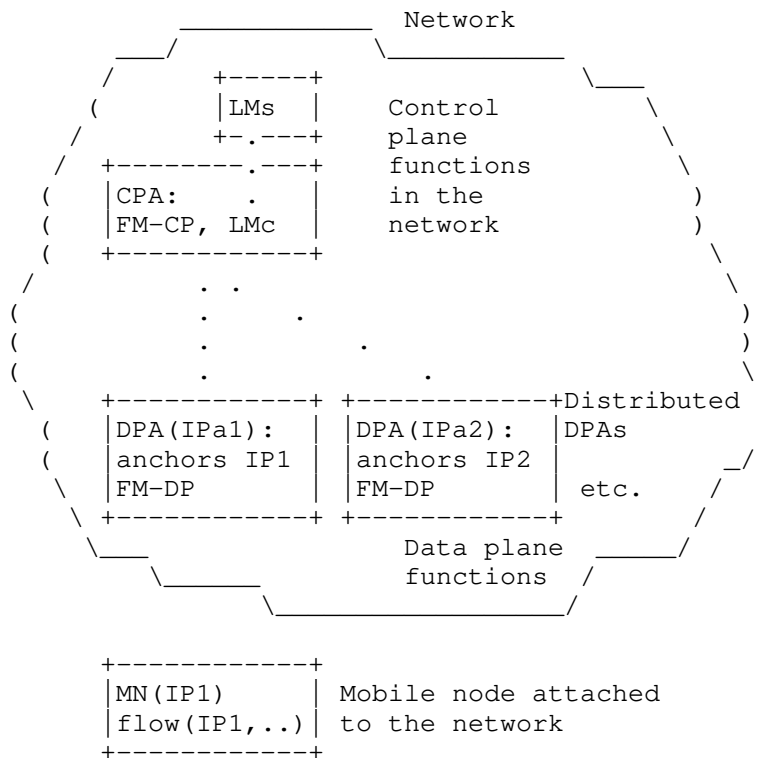


Figure 1: Network-based DMM configuration

### 3.1.2. Client-based DMM

Figure 2 shows a general scenario for client-based distributed mobility management. In this configuration, the mobile node performs Control Plane Node (CPN) and Data Plane Node (DPN) mobility functions, namely the forwarding management and location management (client) roles.

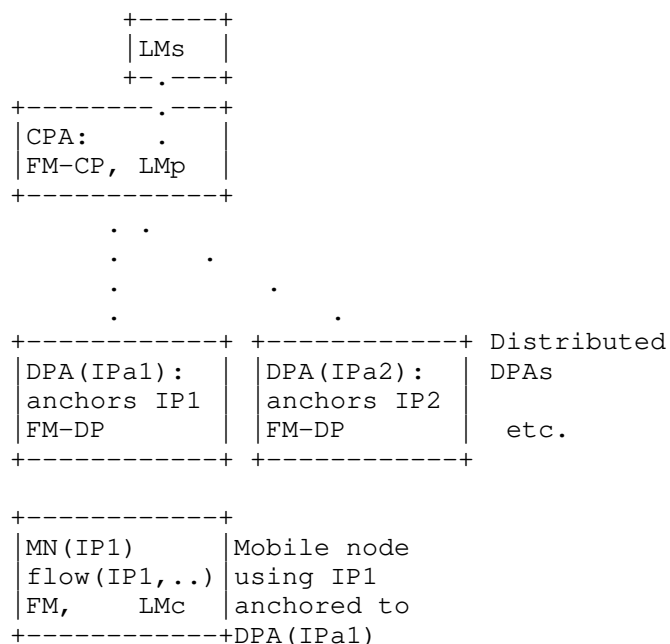


Figure 2: Client-based DMM configuration

#### 4. IP Mobility Handling in Distributed Anchoring Environments - Mobility Support Only When Needed

IP mobility support may be provided only when needed instead of being provided by default. Three cases can be considered:

- o Nomadic case: no address continuity is required. The IP address used by the MN changes after a movement and traffic using the old address is disrupted. If session continuity is required, then it needs to be provided by a solution running at L4 or above.
- o Mobility case, traffic redirection: address continuity is required. When the MN moves, the previous anchor still anchors the traffic using the old IP address, and forwards it to the new MN's location. The MN obtains a new IP address anchored to the new location, and preferably uses it for new communications, established while connected at the new location.
- o Mobility case, anchor relocation: address continuity is required. In this case the route followed by the traffic is optimized, by using some means for traffic indirection to deviate from default routes.

A straightforward choice of mobility anchoring is the following: the MN chooses as source IP address for packets belonging to an IP



flow, an address allocated by the network the MN is attached to when the flow was initiated. As such, traffic belonging to this flow traverses the MN's mobility anchor [I-D.seite-dmm-dma] [I-D.ietf-dmm-pmipv6-dlif].

The IP prefix/address at the MN's side of a flow may be anchored to the Access Router (AR) to which the MN is attached. For example, when a MN attaches to a network (Net1) or moves to a new network (Net2), an IP prefix from the attached network is assigned to the MN's interface. In addition to configuring new link-local addresses, the MN configures from this prefix an IP address which is typically a dynamic IP address (meaning that this address is only used while the MN is attached to this access router, and therefore the IP address configured by the MN dynamically changes when attaching to a different access network). It then uses this IP address when a flow is initiated. Packets from this flow addressed to the MN are simply forwarded according to the forwarding table.

There may be multiple IP prefixes/addresses that an MN can select when initiating a flow. They may be from the same access network or different access networks. The network may advertise these prefixes with cost options [I-D.mccann-dmm-prefixcost] so that the mobile node may choose the one with the least cost. In addition, the IP prefixes/addresses provided by the network may be of different types regarding whether mobility support is supported [RFC8653]. A MN will need to choose which IP prefix/address to use for each flow according to whether it needs IP mobility support or not, using for example the mechanisms described in [RFC8653].

#### 4.1. Nomadic case (no need of IP mobility): Changing to new IP prefix/address

When IP mobility support is not needed for a flow, the LM and FM functions are not utilized so that the configurations in Section 3.1 are simplified as shown in Figure 3.

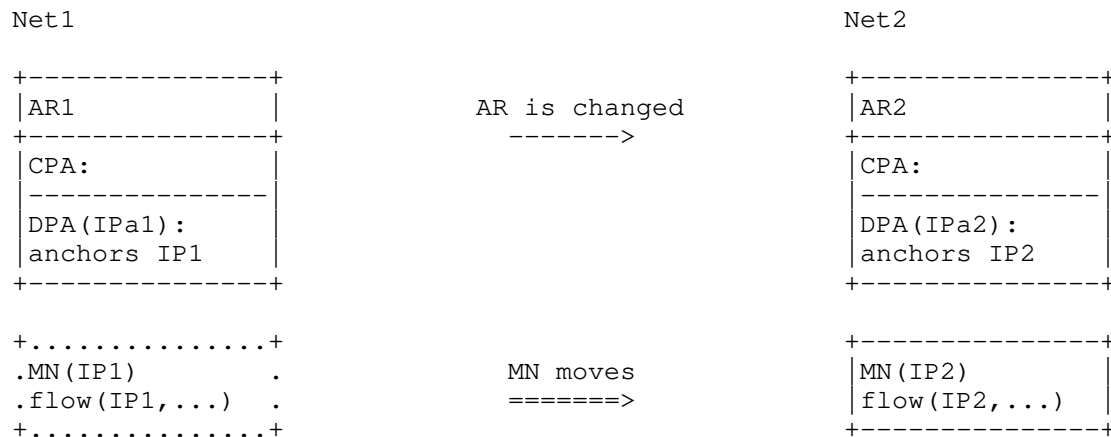


Figure 3: Changing to a new IP address/prefix

When there is no need to provide IP mobility to a flow, the flow may use a new IP address acquired from a new network as the MN moves to the new network.

Regardless of whether IP mobility is needed, if the flow has not terminated before the MN moves to a new network, the flow may subsequently restart using the new IP address assigned from the new network.

When IP session continuity is needed, even if an application flow is ongoing as the MN moves, it may still be desirable for the application flow to change to using the new IP prefix configured in the new network. The application flow may then be closed at IP level and then be restarted using a new IP address configured in the new network. Such a change in the IP address used by the application flow may be enabled using a higher layer mobility support which is not in the scope of this document.

In Figure 3, a flow initiated while the MN was using the IP prefix IP1 -- anchored to a previous access router AR1 in network Net1 -- has terminated before the MN moves to a new network Net2. After moving to Net2, the MN uses the new IP prefix IP2 -- anchored to a new access router AR2 in network Net2 -- to start a new flow. Packets may then be forwarded without requiring IP layer mobility support.

An example call flow is outlined in Figure 4. A MN attaches to AR1, which sends a router advertisement (RA) including information about the prefix assigned to MN, from which MN configures an IP address (IP1). This address is used for new communications, for example with

a correspondent node (CN). If the MN moves to a new network and attaches to AR2, the process is repeated (MN obtains a new IP address, IP2, from AR2). Since the IP address (IP1) configured at the previously visited network is not valid at the current attachment point, and any existing flows have to be reestablished using IP2.

Note that in these scenarios, if there is no mobility support provided by L4 or above, application traffic would stop.

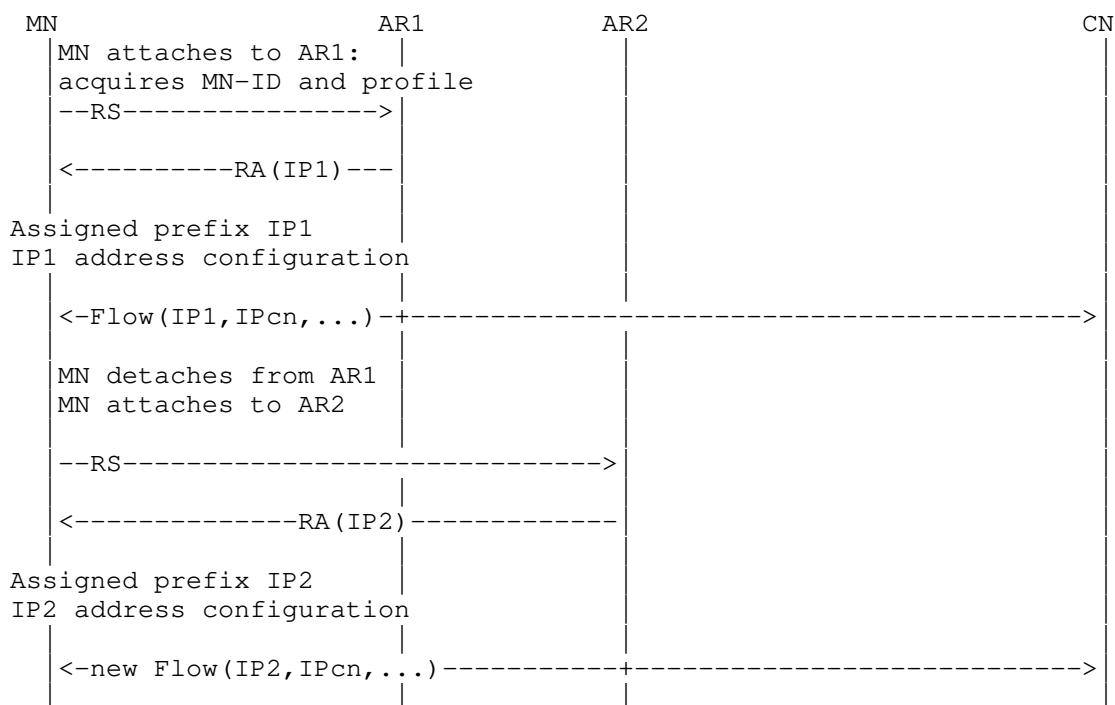


Figure 4: Re-starting a flow with new IP prefix/address

#### 4.2. Mobility case, traffic redirection

When IP mobility is needed for a flow, the LM and FM functions in Section 3.1 are utilized. There are two possible cases: (i) the mobility anchor remains playing that role and forwards traffic to a new locator in the new network, and (ii) the mobility anchor (data plane function) is changed but binds the MN's transferred IP address/prefix. The latter enables optimized routes but requires some data plane node that enforces traffic indirection. Next, we focus on the first case. The second one is addressed in Section 4.3.

Mobility support can be provided by using mobility management methods, such as the several approaches surveyed in the academic papers ([Paper-Distributed.Mobility], [Paper-Distributed.Mobility.PMIP] and [Paper-Distributed.Mobility.Review]). After moving, a certain MN's traffic flow may continue using the IP prefix from the prior network of attachment. Yet, some time later, the application generating this traffic flow may be closed. If the application is started again, the new flow may not need to use the prior network's IP address to avoid having to invoke IP mobility support. This may be the case where a dynamic IP prefix/address, rather than a permanent one, is used. Packets belonging to this flow may then use the new IP prefix (the one allocated in the network where the flow is being initiated). Routing is again kept simpler without employing IP mobility and will remain so as long as the MN which is now in the new network does not move again to another network.

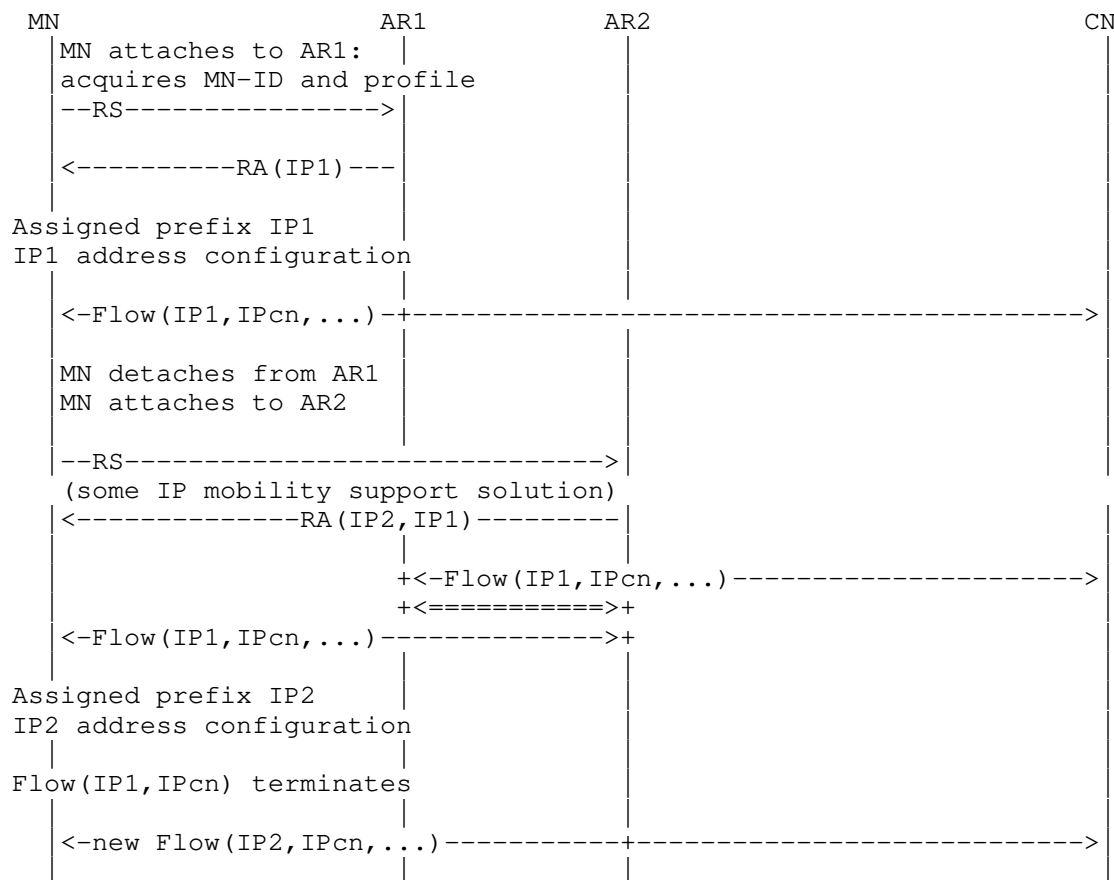


Figure 5: A flow continues to use the IP prefix from its home network after MN has moved to a new network

An example call flow in this case is outlined in Figure 5. In this example, the AR1 plays the role of FM-DP entity and redirects the traffic (e.g., using an IP tunnel) to AR2. Another solution could be to place an FM-DP entity closer to the CN network to perform traffic steering to deviate from default routes (which will bring the packet to AR1 per default routing). The LM and FM functions are implemented as shown in Figure 6.



Figure 6: Anchor redirection

Multiple instances of DPAs (at access routers), which are providing IP prefixes to the MNs, are needed to provide distributed mobility anchoring in an appropriate configuration such as those described in Figure 1 (Section 3.1.1) for network-based distributed mobility or in Figure 2 (Section 3.1.2) for client-based distributed mobility.

#### 4.3. Mobility case, anchor relocation

We focus next on the case where the mobility anchor (data plane function) is changed but binds the MN's transferred IP address/prefix. This enables optimized routes but requires some data plane node that enforces traffic indirection.

IP mobility is invoked to enable IP session continuity for an ongoing flow as the MN moves to a new network. The anchoring of the IP address of the flow is in the home network of the flow (i.e., different from the current network of attachment). A centralized mobility management mechanism may employ indirection from the anchor in the home network to the current network of attachment. Yet it may be difficult to avoid using an unnecessarily long route (when the route between the MN and the CN via the anchor in the home network is significantly longer than the direct route between them). An alternative is to move the IP prefix/address anchoring to the new network.

The IP prefix/address anchoring may move without changing the IP prefix/address of the flow. The LM function in Figure 1 in Section 3.1.1 is implemented as shown in Figure 7.

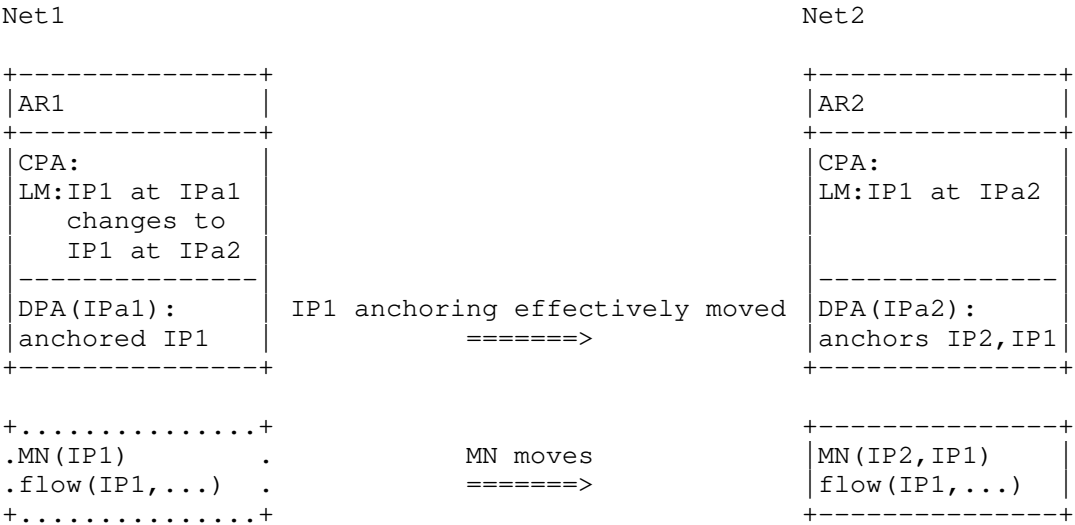


Figure 7: Anchor relocation

As an MN with an ongoing session moves to a new network, the flow may preserve IP session continuity by moving the anchoring of the original IP prefix/address of the flow to the new network.

One way to accomplish such a move is to use a centralized routing protocol, but such a solution may present some scalability concerns and its applicability is typically limited to small networks. One example of this type of solution is described in [I-D.ietf-rtgwg-atn-bgp]. When a MN associates with an anchor the anchor injects the mobile’s prefix into the global routing system. If the MN moves to a new anchor, the old anchor withdraws the /64 and the new anchor injects it instead.

5. Security Considerations

As stated in [RFC7333], "a DMM solution MUST support any security protocols and mechanisms needed to secure the network and to make continuous security improvements". It "MUST NOT introduce new security risks".

There are different potential deployment models of a DMM solution. The present document has presented 3 different scenarios for distributed anchoring: (i) nomadic case, (ii) mobility case with

traffic redirection, and (iii) mobility case with anchor relocation. Each of them has different security requirements, and the actual security mechanisms would depend on the specifics of each solution/scenario.

As general rules, for the first distributed anchoring scenario (nomadic case), no additional security consideration is needed, as this does not involve any additional mechanism at L3. If session connectivity is required, the L4 or above solution used to provide it MUST also provide the required authentication and security.

The second and third distributed anchoring scenarios (mobility case) involve mobility signalling among the mobile node and the control and data plane anchors. The control-plane messages exchanged between these entities MUST be protected using end-to-end security associations with data-integrity and data-origination capabilities. IPsec [RFC8221] ESP in transport mode with mandatory integrity protection SHOULD be used for protecting the signaling messages. IKEv2 [RFC8247] SHOULD be used to set up security associations between the data and control plane anchors. Note that in scenarios in which traffic redirection mechanisms are used to relocate an anchor, authentication and authorization mechanisms MUST be used.

Control-plane functionality MUST apply authorization checks to any commands or updates that are made by the control-plane protocol.

## 6. IANA Considerations

This document presents no IANA considerations.

## 7. Contributors

Alexandre Petrescu and Fred Templin had contributed to earlier versions of this document regarding distributed anchoring for hierarchical network and for network mobility, although these extensions were removed to keep the document within reasonable length.

This document has benefited from other work on mobility support in SDN network, on providing mobility support only when needed, and on mobility support in enterprise network. These works have been referenced. While some of these authors have taken the work to jointly write this document, others have contributed at least indirectly by writing these drafts. The latter include Philippe Bertin, Dapeng Liu, Satoru Matushima, Pierrick Seite, Jouni Korhonen, and Sri Gundavelli.



Some terminology has been incorporated for completeness from draft-ietf-dmm-deployment-models-04 document.

Valuable comments have been received from John Kaippallimalil, ChunShan Xiong, Dapeng Liu, Fred Templin, Paul Kyzivat, Joseph Salowey, Yoshifumi Nishida, Carlos Pignataro, Mirja Kuehlewind, Eric Vyncke, Qin Wu, Warren Kumari, Benjamin Kaduk, Roman Danyliw and Barry Leiba. Dirk von Hugo, Byju Pularikkal, Pierrick Seite have generously provided careful review with helpful corrections and suggestions. Marco Liebsch and Lyle Bertz also performed very detailed and helpful reviews of this document.

## 8. References

### 8.1. 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>>.
- [RFC3753] Manner, J., Ed. and M. Kojo, Ed., "Mobility Related Terminology", RFC 3753, DOI 10.17487/RFC3753, June 2004, <<https://www.rfc-editor.org/info/rfc3753>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <<https://www.rfc-editor.org/info/rfc7333>>.
- [RFC7429] Liu, D., Ed., Zuniga, JC., Ed., Seite, P., Chan, H., and CJ. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis", RFC 7429, DOI 10.17487/RFC7429, January 2015, <<https://www.rfc-editor.org/info/rfc7429>>.
- [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>>.

- [RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <<https://www.rfc-editor.org/info/rfc8221>>.
- [RFC8247] Nir, Y., Kivinen, T., Wouters, P., and D. Migault, "Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8247, DOI 10.17487/RFC8247, September 2017, <<https://www.rfc-editor.org/info/rfc8247>>.

## 8.2. Informative References

- [I-D.ietf-dmm-fpc-cpdp]  
Matsushima, S., Bertz, L., Liebsch, M., Gundavelli, S., Moses, D., and C. Perkins, "Protocol for Forwarding Policy Configuration (FPC) in DMM", draft-ietf-dmm-fpc-cpdp-12 (work in progress), June 2018.
- [I-D.ietf-dmm-pmipv6-dlif]  
Bernardos, C., Oliva, A., Giust, F., Zuniga, J., and A. Mourad, "Proxy Mobile IPv6 extensions for Distributed Mobility Management", draft-ietf-dmm-pmipv6-dlif-05 (work in progress), November 2019.
- [I-D.ietf-rtgwg-atn-bgp]  
Templin, F., Saccone, G., Dawra, G., Lindem, A., and V. Moreno, "A Simple BGP-based Mobile Routing System for the Aeronautical Telecommunications Network", draft-ietf-rtgwg-atn-bgp-05 (work in progress), January 2020.
- [I-D.matsushima-stateless-uplane-vepc]  
Matsushima, S. and R. Wakikawa, "Stateless user-plane architecture for virtualized EPC (vEPC)", draft-matsushima-stateless-uplane-vepc-06 (work in progress), March 2016.
- [I-D.mccann-dmm-prefixcost]  
McCann, P. and J. Kaippallimalil, "Communicating Prefix Cost to Mobile Nodes", draft-mccann-dmm-prefixcost-03 (work in progress), April 2016.
- [I-D.sarikaya-dmm-for-wifi]  
Sarikaya, B. and L. Li, "Distributed Mobility Management Protocol for WiFi Users in Fixed Network", draft-sarikaya-dmm-for-wifi-05 (work in progress), October 2017.

- [I-D.seite-dmm-dma]  
Seite, P., Bertin, P., and J. Lee, "Distributed Mobility Anchoring", draft-seite-dmm-dma-07 (work in progress), February 2014.
- [I-D.yhkim-dmm-enhanced-anchoring]  
Kim, Y. and S. Jeon, "Enhanced Mobility Anchoring in Distributed Mobility Management", draft-yhkim-dmm-enhanced-anchoring-05 (work in progress), July 2016.
- [Paper-Distributed.Mobility]  
Lee, J., Bonnin, J., Seite, P., and H. Chan, "Distributed IP Mobility Management from the Perspective of the IETF: Motivations, Requirements, Approaches, Comparison, and Challenges", IEEE Wireless Communications, October 2013.
- [Paper-Distributed.Mobility.PMIP]  
Chan, H., "Proxy Mobile IP with Distributed Mobility Anchors", Proceedings of GlobeCom Workshop on Seamless Wireless Mobility, December 2010.
- [Paper-Distributed.Mobility.Review]  
Chan, H., Yokota, H., Xie, J., Seite, P., and D. Liu, "Distributed and Dynamic Mobility Management in Mobile Internet: Current Approaches and Issues", February 2011.
- [RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <<https://www.rfc-editor.org/info/rfc6459>>.
- [RFC8653] Yegin, A., Moses, D., and S. Jeon, "On-Demand Mobility Management", RFC 8653, DOI 10.17487/RFC8653, October 2019, <<https://www.rfc-editor.org/info/rfc8653>>.

#### Authors' Addresses

H. Anthony Chan (editor)  
Huawei Technologies  
5340 Legacy Dr. Building 3  
Plano, TX 75024  
USA

Email: [h.a.chan@ieee.org](mailto:h.a.chan@ieee.org)

Xinpeng Wei  
Huawei Technologies  
Xin-Xi Rd. No. 3, Haidian District  
Beijing, 100095  
P. R. China

Email: [weixinpeng@huawei.com](mailto:weixinpeng@huawei.com)

Jong-Hyouk Lee  
Sangmyung University  
31, Sangmyeongdae-gil, Dongnam-gu  
Cheonan 31066  
Republic of Korea

Email: [jonghyouk@smu.ac.kr](mailto:jonghyouk@smu.ac.kr)

Seil Jeon  
Sungkyunkwan University  
2066 Seobu-ro, Jangan-gu  
Suwon, Gyeonggi-do  
Republic of Korea

Email: [seiljeon@skku.edu](mailto:seiljeon@skku.edu)

Carlos J. Bernardos (editor)  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Phone: +34 91624 6236  
Email: [cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)  
URI: <http://www.it.uc3m.es/cjbc/>

DMM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 27 March 2021

S. Matsushima  
SoftBank  
L. Bertz  
Sprint  
M. Liebsch  
NEC  
S. Gundavelli  
Cisco  
D. Moses  
Intel Corporation  
C.E. Perkins  
Futurewei  
23 September 2020

Protocol for Forwarding Policy Configuration (FPC) in DMM  
draft-ietf-dmm-fpc-cpdp-14

Abstract

This document describes a way, called Forwarding Policy Configuration (FPC) to manage the separation of data-plane and control-plane. FPC defines a flexible mobility management system using FPC agent and FPC client functions. A FPC agent provides an abstract interface to the data-plane. The FPC client configures data-plane nodes by using the functions and abstractions provided by the FPC agent for the data-plane nodes. The data-plane abstractions presented in this document are extensible in order to support many different types of mobility management systems and data-plane functions.

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 27 March 2021.

## Copyright Notice

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

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

## Table of Contents

|                                                     |    |
|-----------------------------------------------------|----|
| 1. Introduction . . . . .                           | 3  |
| 2. Terminology . . . . .                            | 4  |
| 3. FPC Design Objectives and Deployment . . . . .   | 6  |
| 4. FPC Mobility Information Model . . . . .         | 9  |
| 4.1. Model Notation and Conventions . . . . .       | 10 |
| 4.2. Templates and Attributes . . . . .             | 12 |
| 4.3. Attribute-Expressions . . . . .                | 13 |
| 4.4. Attribute Value Types . . . . .                | 14 |
| 4.5. Namespace and Format . . . . .                 | 14 |
| 4.6. Configuring Attribute Values . . . . .         | 15 |
| 4.7. Entity Configuration Blocks . . . . .          | 16 |
| 4.8. Information Model Checkpoint . . . . .         | 17 |
| 4.9. Information Model Components . . . . .         | 18 |
| 4.9.1. Topology Information Model . . . . .         | 18 |
| 4.9.2. Service-Group . . . . .                      | 18 |
| 4.9.3. Domain Information Model . . . . .           | 20 |
| 4.9.4. DPN Information Model . . . . .              | 20 |
| 4.9.5. Policy Information Model . . . . .           | 22 |
| 4.9.6. Mobility-Context Information Model . . . . . | 24 |
| 4.9.7. Monitor Information Model . . . . .          | 26 |
| 5. Security Considerations . . . . .                | 28 |
| 6. IANA Considerations . . . . .                    | 28 |
| 7. Work Team Participants . . . . .                 | 28 |
| 8. References . . . . .                             | 28 |
| 8.1. Normative References . . . . .                 | 28 |
| 8.2. Informative References . . . . .               | 28 |
| Appendix A. Implementation Status . . . . .         | 29 |
| Authors' Addresses . . . . .                        | 33 |

## 1. Introduction

This document describes Forwarding Policy Configuration (FPC), a system for managing the separation of control-plane and data-plane. FPC enables flexible mobility management using FPC client and FPC agent functions. A FPC agent exports an abstract interface representing the data-plane. To configure data-plane nodes and functions, the FPC client uses the interface to the data-plane offered by the FPC agent.

Control planes of mobility management systems, or related applications which require data-plane control, can utilize the FPC client at various levels of abstraction. FPC operations are capable of directly configuring a single Data-Plane Node (DPN), as well as multiple DPNs, as determined by the data-plane models exported by the FPC agent.

A FPC agent represents the data-plane operation according to several basic information models. A FPC agent also provides access to Monitors, which produce reports when triggered by events or FPC Client requests regarding Mobility Contexts, DPNs or the Agent.

To manage mobility sessions, the FPC client assembles applicable sets of forwarding policies from the data model, and configures them on the appropriate FPC Agent. The Agent then renders those policies into specific configurations for each DPN at which mobile nodes are attached. The specific protocols and configurations to configure a DPN from a FPC Agent are outside the scope of this document.

A DPN is a logical entity that performs data-plane operations (packet movement and management). It may represent a physical DPN unit, a sub-function of a physical DPN or a collection of physical DPNs (i.e., a "virtual DPN"). A DPN may be virtual -- it may export the FPC DPN Agent interface, but be implemented as software that controls other data-plane hardware or modules that may or may not be FPC-compliant. In this document, DPNs are specified without regard for whether the implementation is virtual or physical. DPNs are connected to provide mobility management systems such as access networks, anchors and domains. The FPC agent interface enables establishment of a topology for the forwarding plane.

When a DPN is mapped to physical data-plane equipment, the FPC client can have complete knowledge of the DPN architecture, and use that information to perform DPN selection for specific sessions. On the other hand, when a virtual DPN is mapped to a collection of physical DPNs, the FPC client cannot select a specific physical DPN because it is hidden by the abstraction; only the FPC Agent can address the specific associated physical DPNs. Network architects have the

flexibility to determine which DPN-selection capabilities are performed by the FPC Agent (distributed) and which by the FPC client (centralized). In this way, overlay networks can be configured without disclosing detailed knowledge of the underlying hardware to the FPC client and applications.

The abstractions in this document are designed to support many different mobility management systems and data-plane functions. The architecture and protocol design of FPC is not tied to specific types of access technologies and mobility protocols.

## 2. Terminology

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

**Attribute Expression:** The definition of a template Property. This includes setting the type, current value, default value and if the attribute is static, i.e. can no longer be changed.

**Domain:** One or more DPNs that form a logical partition of network resources (e.g., a data-plane network under common network administration). A FPC client (e.g., a mobility management system) may utilize a single or multiple domains.

**DPN:** A data-plane node (DPN) is capable of performing data-plane features. For example, DPNs may be switches or routers, regardless of whether they are realized as hardware or purely in software.

**FPC Client:** A FPC Client is integrated with a mobility management system or related application, enabling control over forwarding policy, mobility sessions and DPNs via a FPC Agent.

**Mobility Context:** A Mobility Context contains the data-plane information necessary to efficiently send and receive traffic from a mobile node. This includes policies that are created or modified during the network's operation - in most cases, on a per-flow or per session basis. A Mobility-Context represents the mobility sessions (or flows) which are active



on a mobile node. This includes associated runtime attributes, such as tunnel endpoints, tunnel identifiers, delegated prefix(es), routing information, etc. Mobility-Contexts are associated to specific DPNs. Some pre-defined Policies may apply during mobility signaling requests. The Mobility Context supplies information about the policy settings specific to a mobile node and its flows; this information is often quite dynamic.

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mobility Session: | Traffic to/from a mobile node that is expected to survive reconnection events.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Monitor:          | A reporting mechanism for a list of events that trigger notification messages from a FPC Agent to a FPC Client.                                                                                                                                                                                                                                                                                                                                                                                                |
| Policy:           | A Policy determines the mechanisms for managing specific traffic flows or packets. Policies specify QoS, rewriting rules for packet processing, etc. A Policy consists of one or more rules. Each rule is composed of a Descriptor and Actions. The Descriptor in a rule identifies packets (e.g., traffic flows), and the Actions apply treatments to packets that match the Descriptor in the rule. Policies can apply to Domains, DPNs, Mobile Nodes, Service-Groups, or particular Flows on a Mobile Node. |
| Property:         | An attribute-value pair for an instance of a FPC entity.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Service-Group:    | A set of DPN interfaces that support a specific data-plane purpose, e.g. inbound/outbound, roaming, subnetwork with common specific configuration, etc.                                                                                                                                                                                                                                                                                                                                                        |
| Template:         | A recipe for instantiating FPC entities. Template definitions are accessible (by name or by a key) in an indexed set. A Template is used to create specific instances (e.g., specific policies) by assigning appropriate values into the Template definition via Attribute Expression.                                                                                                                                                                                                                         |

|                        |                                                                                                                                                                                                                                                                                  |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Template Configuration | The process by which a Template is referenced (by name or by key) and Attribute Expressions are created that change the value, default value or static nature of the Attribute, if permitted. If the Template is Extensible, new attributes MAY be added.                        |
| Tenant:                | An operational entity that manages mobility management systems or applications which require data-plane functions. A Tenant defines a global namespace for all entities owned by the Tenant enabling its entities to be used by multiple FPC Clients across multiple FPC Agents. |
| Topology:              | The DPNs and the links between them. For example, access nodes may be assigned to a Service-Group which peers to a Service-Group of anchor nodes.                                                                                                                                |

### 3. FPC Design Objectives and Deployment

Using FPC, mobility control-planes and applications can configure DPNs to perform various mobility management roles as described in [I-D.ietf-dmm-deployment-models]. This fulfills the requirements described in [RFC7333].

This document defines FPC Agent and FPC Client, as well as the information models that they use. The attributes defining those models serve as the protocol elements for the interface between the FPC Agent and the FPC Client.

Mobility control-plane applications integrate features offered by the FPC Client. The FPC Client connects to FPC Agent functions. The Client and the Agent communicate based on information models described in Section 4. The models allow the control-plane to configure forwarding policies on the Agent for data-plane communications with mobile nodes.

Once the Topology of DPN(s) and domains are defined on an Agent for a data plane, the DPNs in the topology are available for further configuration. The FPC Agent connects those DPNs to manage their configurations.

A FPC Agent configures and manages its DPN(s) according to forwarding policies requested and Attributes provided by the FPC Client. Configuration commands used by the FPC agent to configure its DPN node(s) may be specific to the DPN implementation; consequently the

method by which the FPC Agent carries out the specific configuration for its DPN(s) is out of scope for this document. Along with the data models, the FPC Client (on behalf of control-plane and applications) requests that the Agent configures Policies prior to the time when the DPNs start forwarding data for their mobility sessions.

This architecture is illustrated in Figure 1. A FPC Agent may be implemented in a network controller that handles multiple DPNs, or (more simply) an FPC Agent may itself be integrated into a DPN.

This document does not specify a protocol for the FPC interface; it is out of scope.

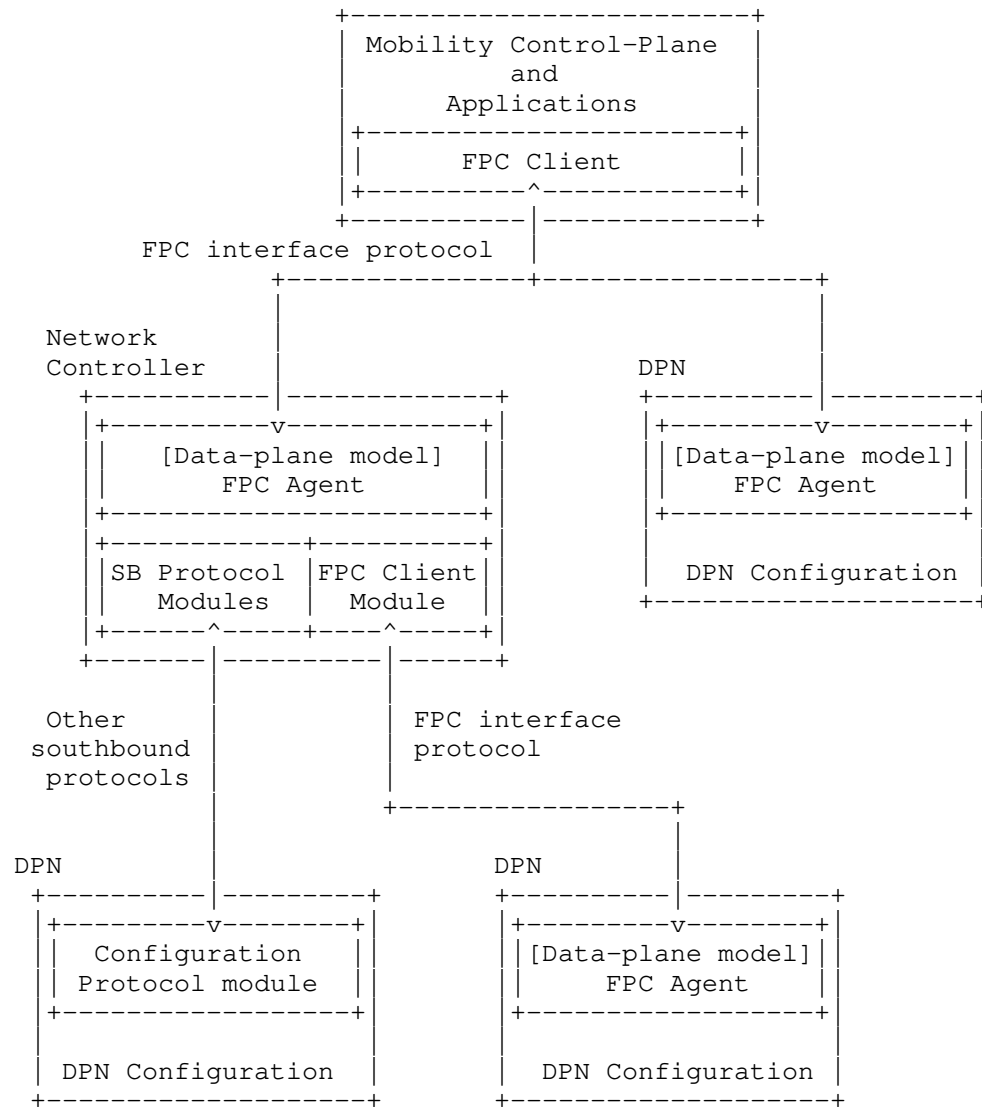


Figure 1: Reference Forwarding Policy Configuration (FPC)  
Architecture

The FPC architecture supports multi-tenancy; a FPC enabled data-plane supports tenants of multiple mobile operator networks and/or applications. It means that the FPC Client of each tenant connects to the FPC Agent and it MUST partition namespace and data for their data-planes. DPNs on the data-plane may fulfill multiple data-plane roles which are defined per session, domain and tenant.

Multi-tenancy permits the partitioning of data-plane entities as well as a common namespace requirement upon FPC Agents and Clients when they use the same Tenant for a common data-plane entity.

FPC information models often configuration to fit the specific needs for DPN management of a mobile node's traffic. The FPC interfaces in Figure 1 are the only interfaces required to handle runtime data in a Mobility Context. The Topology and some Policy FPC models MAY be pre-configured; in that case real-time protocol exchanges are not required for them.

The information model provides an extensibility mechanism through Templates that permits specialization for the needs of a particular vendor's equipment or future extension of the model presented in this specification.

#### 4. FPC Mobility Information Model

The FPC information model includes the following components:

- DPN Information Model,
- Topology Information Model,
- Policy Information Model,
- Mobility-Context, and
- Monitor, as illustrated in Figure 2.

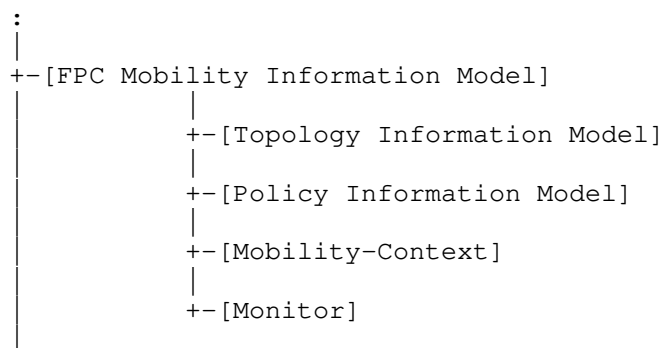


Figure 2: FPC Information Model structure

#### 4.1. Model Notation and Conventions

The following conventions are used to describe the FPC information models.

Information model entities (e.g. DPNs, Rules, etc.) are defined in a hierarchical notation where all entities at the same hierarchical level are located on the same left-justified vertical position sequentially. When entities are composed of sub-entities, the sub-entities appear shifted to the right, as shown in Figure 3.

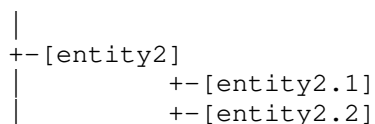


Figure 3: Model Notation - An Example

Some entities have one or more qualifiers placed on the right hand side of the element definition in angle-brackets. Common types include:

List: A collection of entities (some could be duplicated)

Set: A nonempty collection of entities without duplications

Name: A human-readable string

Key: A unique value. We distinguish 3 types of keys:

U-Key: A key unique across all Tenants. U-Key spaces typically

involve the use of registries or language specific mechanisms that guarantee universal uniqueness of values.

G-Key: A key unique within a Tenant

L-Key: A key unique within a local namespace. For example, there may exist interfaces with the same name, e.g. "if0", in two different DPNs but there can only be one "if0" within each DPN (i.e. its local Interface-Key L-Key space).

Each entity or attribute may be optional (O) or mandatory (M). Entities that are not marked as optional are mandatory.

The following example shows 3 entities:

```
-- Entity1 is a globally unique key, and optionally can have
    an associated Name
-- Entity2 is a list
-- Entity3 is a set and is optional
+
|
+--[entity1] <G-Key> (M), <Name> (O)
+--[entity2] <List>
+--[entity3] <Set> (O)
|
+
```

Figure 4

When expanding entity1 into a modeling language such as YANG it would result in two values: entity1-Key and entity1-Name.

To encourage re-use, FPC defines indexed sets of various entity Templates. Other model elements that need access to an indexed model entity contain an attribute which is always denoted as "entity-Key". When a Key attribute is encountered, the referencing model element may supply attribute values for use when the referenced entity model is instantiated. For example: Figure 5 shows 2 entities:

EntityA definition references an entityB model element.

EntityB model elements are indexed by entityB-Key.

Each EntityB model element has an entityB-Key which allows it to be uniquely identified, and a list of Attributes (or, alternatively, a Type) which specifies its form. This allows a referencing entity to create an instance by supplying entityB-Values to be inserted, in a Settings container.

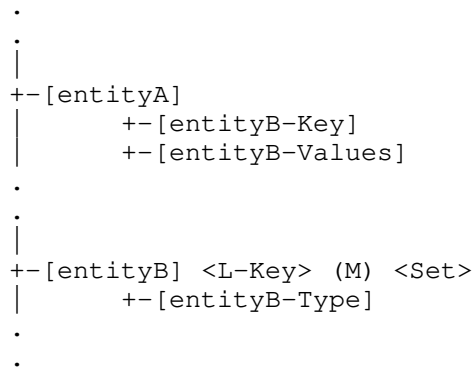


Figure 5: Indexed sets of entities

Indexed sets are specified for each of the following kinds of entities:

- Domain (See Section 4.9.3)
- DPN (See Section 4.9.4)
- Policy (See Section 4.9.5)
- Rule (See Section 4.9.5)
- Descriptor (See Figure 12)
- Action (See Figure 12)
- Service-Group (See Section 4.9.2, and
- Mobility-Context (See Section 4.9.6)

As an example, for a Domain entity, there is a corresponding attribute denoted as "Domain-Key" whose value can be used to determine a reference to the Domain.

#### 4.2. Templates and Attributes

In order to simplify development and maintenance of the needed policies and other objects used by FPC, the Information Models which are presented often have attributes that are not initialized with their final values. When an FPC entity is instantiated according to a template definition, specific values need to be configured for each such attribute. For instance, suppose an entity Template has an Attribute named "IPv4-Address", and also suppose that a FPC Client instantiates the entity and requests that it be installed on a DPN. An IPv4 address will be needed for the value of that Attribute before the entity can be used.



```

+-[Template] <U-Key, Name> (M) <Set>
|   +-[Attributes] <Set> (M)
|   +-[Extensible ~ FALSE]
|   +-[Entity-State ~ Initial]
|   +-[Version]

```

Figure 6: Template entities

**Attributes:** A set of Attribute names MAY be included when defining a Template for instantiating FPC entities.

**Extensible:** Determines whether or not entities instantiated from the Template can be extended with new non-mandatory Attributes not originally defined for the Template. Default value is FALSE. If a Template does not explicitly specify this attribute, the default value is considered to be in effect.

**Entity-State:** Either Initial, PartiallyConfigured, Configured, or Active. Default value is Initial. See Section 4.6 for more information about how the Entity-Status changes during the configuration steps of the Entity.

**Version:** Provides a version tag for the Template.

The Attributes in an Entity Template may be either mandatory or non-mandatory. Attribute values may also be associated with the attributes in the Entity Template. If supplied, the value may be either assigned with a default value that can be reconfigured later, or the value can be assigned with a static value that cannot be reconfigured later (see Section 4.3).

It is possible for a Template to provide values for all of its Attributes, so that no additional values are needed before the entity can be made Active. Any instantiation from a Template MUST have at least one Attribute in order to be a useful entity unless the Template has none.

#### 4.3. Attribute-Expressions

The syntax of the Attribute definition is formatted to make it clear. For every Attribute in the Entity Template, six possibilities are specified as follows:

'[Att-Name: ]' Mandatory Attribute is defined, but template does not provide any configured value.

'[Att-Name: Att-Value]' Mandatory Attribute is defined, and has a

statically configured value.

'[Att-Name: ~ Att-Value]' Mandatory Attribute is defined, and has a default value.

'[Att-Name]' Non-mandatory Attribute may be included but template does not provide any configured value.

'[Att-Name = Att-Value]' Non-mandatory Attribute may be included and has a statically configured value.

'[Att-Name ~ Att-Value]' Non-mandatory Attribute may be included and has a default value.

So, for example, a default value for a non-mandatory IPv4-Address attribute would be denoted by [IPv4-Address ~ 127.0.0.1].

After a FPC Client identifies which additional Attributes have been configured to be included in an instantiated entity, those configured Attributes MUST NOT be deleted by the FPC Agent. Similarly, any statically configured value for an entity Attribute MUST NOT be changed by the FPC Agent.

Whenever there is danger of confusion, the fully qualified Attribute name MUST be used when supplying needed Attribute Values for a structured Attribute.

#### 4.4. Attribute Value Types

For situations in which the type of an attribute value is required, the following syntax is recommended. To declare that an attribute has data type "foo", typecast the attribute name by using the parenthesized data type (foo). So, for instance, [(float) Max-Latency-in-ms:] would indicate that the mandatory Attribute "Max-Latency-in-ms" requires to be configured with a floating point value before the instantiated entity could be used. Similarly, [(float) Max-Latency-in-ms: 9.5] would statically configure a floating point value of 9.5 to the mandatory Attribute "Max-Latency-in-ms".

#### 4.5. Namespace and Format

The identifiers and names in FPC models which reside in the same Tenant must be unique. That uniqueness must be maintained by all Clients, Agents and DPNs that support the Tenant. The Tenant namespace uniqueness MUST be applied to all elements of the tenant model, i.e. Topology, Policy and Mobility models.

When a Policy needs to be applied to Mobility-Contexts in all Tenants on an Agent, the Agent SHOULD define that policy to be visible by all Tenants. In this case, the Agent assigns a unique identifier in the Agent namespace and copies the values to each Tenant. This effectively creates a U-Key although only a G-Key is required within the Tenant.

The notation for identifiers can utilize any format with agreement between data-plane agent and client operators. The formats include but are not limited to Globally Unique IDentifiers (GUIDs), Universally Unique IDentifiers (UUIDs), Fully Qualified Domain Names (FQDNs), Fully Qualified Path Names (FQPNs) and Uniform Resource Identifiers (URIs). The FPC model does not limit the format, which could dictate the choice of FPC protocol. Nevertheless, the identifiers which are used in a Mobility model should be considered to efficiently handle runtime parameters.

#### 4.6. Configuring Attribute Values

Attributes of Information Model components such as policy templates are configured with values as part of FPC configuration operations. There may be several such configuration operations before the template instantiation is fully configured.

Entity-Status indicates when an Entity is usable within a DPN. This permits DPN design tradeoffs amongst local storage (or other resources), over the wire request size and the speed of request processing. For example, DPN designers with constrained systems MAY only house entities whose status is Active which may result in sending over all policy information with a Mobility-Context request. Storing information elements with an entity status of "PartiallyConfigured" on the DPN requires more resources but can result in smaller over the wire FPC communication and request processing efficiency.

When the FPC Client instantiates a Policy from a Template, the Policy-Status is "Initial". When the FPC Client sends the policy to a FPC Agent for installation on a DPN, the Client often will configure appropriate attribute values for the installation, and accordingly changes the Policy-Status to "PartiallyConfigured" or "Configured". The FPC Agent will also configure Domain-specific policies and DPN-specific policies on the DPN. When configured to provide particular services for mobile nodes, the FPC Agent will apply whatever service-specific policies are needed on the DPN. When a mobile node attaches to the network data-plane within the topology under the jurisdiction of a FPC Agent, the Agent may apply policies and settings as appropriate for that mobile node. Finally, when the mobile node launches new flows, or quenches existing flows, the FPC

Agent, on behalf of the FPC Client, applies or deactivates whatever policies and attribute values are appropriate for managing the flows of the mobile node. When a "Configured" policy is de-activated, Policy-Status is changed to be "Active". When an "Active" policy is activated, Policy-Status is changed to be "Configured".

Attribute values in DPN resident Policies may be configured by the FPC Agent as follows:

Domain-Policy-Configuration: Values for Policy attributes that are required for every DPN in the domain.

DPN-Policy-Configuration: Values for Policy attributes that are required for every policy configured on this DPN.

Service-Group-Policy-Configuration: Values for Policy attributes that are required to carry out the intended Service of the Service Group.

MN-Policy-Configuration: Values for Policy attributes that are required for all traffic to/from a particular mobile node.

Service-Data-Flow-Policy-Configuration: Values for Policy attributes that are required for traffic belonging to a particular set of flows on the mobile node.

Any configuration changes MAY also supply updated values for existing default attribute values that may have been previously configured on the DPN resident policy.

Entity blocks describe the format of the policy configurations.

#### 4.7. Entity Configuration Blocks

As described in Section 4.6, a Policy Template may be configured in several stages by configuring default or missing values for Attributes that do not already have statically configured values. A Policy-Configuration is the combination of a Policy-Key (to identify the Policy Template defining the Attributes) and the currently configured Attribute Values to be applied to the Policy Template. Policy-Configurations MAY add attributes to a Template if Extensible is True. They MAY also refine existing attributes by:

- assign new values if the Attribute is not static

- make attributes static if they were not

- make an attribute mandatory

A Policy-Configuration MUST NOT define or refine an attribute twice. More generally, an Entity-Configuration can be defined for any configurable Indexed Set to be the combination of the Entity-Key along with a set of Attribute-Expressions that supply configuration information for the entity's Attributes. Figure 7 shows a schematic representation for such Entity Configuration Blocks.

```
[Entity Configuration Block]
|   +-[Entity-Key] (M)
|   +-[Attribute-Expression] <Set> (M)
```

Figure 7: Entity Configuration Block

This document makes use of the following kinds of Entity Configuration Blocks:

- Descriptor-Configuration
- Action-Configuration
- Rule-Configuration
- Interface-Configuration
- Service-Group-Configuration
- Domain-Policy-Configuration
- DPN-Policy-Configuration
- Policy-Configuration
- MN-Policy-Configuration
- Service-Data-Flow-Policy-Configuration

#### 4.8. Information Model Checkpoint

The Information Model Checkpoint permits Clients and Tenants with common scopes, referred to in this specification as Checkpoint BaseNames, to track the state of provisioned information on an Agent. The Agent records the Checkpoint BaseName and Checkpoint value set by a Client. When a Client attaches to the Agent it can query to determine the amount of work that must be executed to configure the Agent to a specific BaseName / checkpoint revision.

Checkpoints are defined for the following information model components:

Service-Group

DPN Information Model

Domain Information Model

Policy Information Model

#### 4.9. Information Model Components

##### 4.9.1. Topology Information Model

The Topology structure specifies DPNs and the communication paths between them. A network management system can use the Topology to select the most appropriate DPN resources for handling specific session flows.

The Topology structure is illustrated in Figure 8 (for definitions see Section 2):

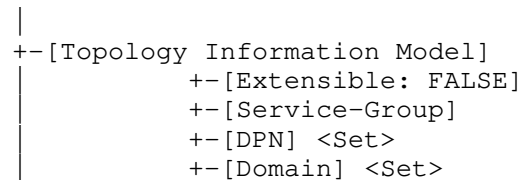


Figure 8: Topology Structure

##### 4.9.2. Service-Group

Service-Group-Set is collection of DPN interfaces serving some data-plane purpose including but not limited to DPN Interface selection to fulfill a Mobility-Context. Each Group contains a list of DPNs (referenced by DPN-Key) and selected interfaces (referenced by Interface-Key). The Interfaces are listed explicitly (rather than referred implicitly by its specific DPN) so that every Interface of a DPN is not required to be part of a Group. The information provided is sufficient to ensure that the Protocol, Settings (stored in the Service-Group-Configuration) and Features relevant to successful interface selection is present in the model.

```

|
|--[Service-Group] <G-Key>, <Name> (0) <Set>
|   |--[Extensible: FALSE]
|   |--[Role] <U-Key>
|   |--[Protocol] <Set>
|   |--[Feature] <Set> (0)
|   |--[Service-Group-Configuration] <Set> (0)
|   |--[DPN-Key] <Set>
|       |--[Referenced-Interface] <Set>
|           |--[Interface-Key] <L-Key>
|           |--[Peer-Service-Group-Key] <Set> (0)

```

Figure 9: Service Group

Each Service-Group element contains the following information:

Service-Group-Key: A unique ID of the Service-Group.

Service-Group-Name: A human-readable display string.

Role: The role (MAG, LMA, etc.) of the device hosting the interfaces of the DPN Group.

Protocol-Set: The set of protocols supported by this interface (e.g., PMIP, S5-GTP, S5-PMIP etc.). The protocol MAY be only its name, e.g. 'gtp', but many protocols implement specific message sets, e.g. s5-pmip, s8-pmip. When the Service-Group supports specific protocol message sub-subsets the Protocol value MUST include this information.

Feature-Set: An optional set of static features which further determine the suitability of the interface to the desired operation.

Service-Group-Configuration-Set: An optional set of configurations that further determine the suitability of an interface for the specific request. For example: SequenceNumber=ON/OFF.

DPN-Key-Set: A key used to identify the DPN.

Referenced-Interface-Set: The DPN Interfaces and peer Service-Groups associated with them. Each entry contains

Interface-Key: A key that is used together with the DPN-Key, to create a key that is refers to a specific DPN interface definition.

Peer-Service-Group-Key: Enables location of the peer Service-Group for this Interface.

#### 4.9.3. Domain Information Model

A Domain-Set represents a group of heterogeneous Topology resources typically sharing a common administrative authority. Other models, outside of the scope of this specification, provide the details for the Domain.

```

|
+--[Domain] <G-Key>, <Name> (O) <Set>
|   +-[Domain-Policy-Configuration] (O) <Set>
|

```

Figure 10: Domain Information Model

Each Domain entry contains the following information:

Domain-Key: Identifies and enables reference to the Domain.

Domain-Name: A human-readable display string naming the Domain.

#### 4.9.4. DPN Information Model

A DPN-Set contains some or all of the DPNs in the Tenant's network. Some of the DPNs in the Set may be identical in functionality and only differ by their Key.

```

|
+--[DPN] <G-Key>, <Name> (O) <Set>
|   +-[Extensible: FALSE]
|   +-[Interface] <L-Key> <Set>
|       +-[Role] <U-Key>
|       +-[Protocol] <Set>
|       +-[Interface-Configuration] <Set> (O)
|   +-[Domain-Key]
|   +-[Service-Group-Key] <Set> (O)
|   +-[DPN-Policy-Configuration] <List> (M)
|   +-[DPN-Resource-Mapping-Reference] (O)
|

```

Figure 11: DPN Information Model

Each DPN entry contains the following information:

DPN-Key: A unique Identifier of the DPN.



DPN-Name: A human-readable display string.

Domain-Key: A Key providing access to the Domain information about the Domain in which the DPN resides.

Interface-Set: The Interface-Set references all interfaces (through which data packets are received and transmitted) available on the DPN. Each Interface makes use of attribute values that are specific to that interface, for example, the MTU size. These do not affect the DPN selection of active or enabled interfaces. Interfaces contain the following information:

Role: The role (MAG, LMA, PGW, AMF, etc.) of the DPN.

Protocol (Set): The set of protocols supported by this interface (e.g., PMIP, S5-GTP, S5-PMIP etc.). The protocol MAY implement specific message sets, e.g. s5-pmip, s8-pmip. When a protocol implements such message sub-subsets the Protocol value MUST include this information.

Interface-Configuration-Set: Configurable settings that further determine the suitability of an interface for the specific request. For example: SequenceNumber=ON/OFF.

Service-Group-Set: The Service-Group-Set references all of the Service-Groups which have been configured using Interfaces hosted on this DPN. The purpose of a Service-Group is not to describe each interface of each DPN, but rather to indicate interface types for use during the DPN selection process, when a DPN with specific interface capabilities is required.

DPN-Policy-Configuration: A list of Policies that have been configured on this DPN. Some may have values for all attributes, and some may require further configuration. Each Policy-Configuration has a key to enable reference to its Policy-Template. Each Policy-Configuration also has been configured to supply missing and non-default values to the desired Attributes defined within the Policy-Template.

DPN-Resource-Mapping-Reference (O): A reference to the underlying implementation, e.g. physical node, software module, etc. that supports this DPN. Further specification of this attribute is out of scope for this document.

#### 4.9.5. Policy Information Model

The Policy Information Model defines and identifies Rules for enforcement at DPNs. A Policy is basically a set of Rules that are to be applied to each incoming or outgoing packet at a DPN interface. Rules comprise Descriptors and a set of Actions. The Descriptors, when evaluated, determine whether or not a set of Actions will be performed on the packet. The Policy structure is independent of a policy context.

In addition to the Policy structure, the Information Model (per Section 4.9.6) defines Mobility-Context. Each Mobility-Context may be configured with appropriate Attribute values, for example depending on the identity of a mobile node.

Traffic descriptions are defined in Descriptors, and treatments are defined separately in Actions. A Rule-Set binds Descriptors and associated Actions by reference, using Descriptor-Key and Action-Key. A Rule-Set is bound to a policy in the Policy-Set (using Policy-Key), and the Policy references the Rule definitions (using Rule-Key).

```

+--[Policy Information Model]
|
+--[Extensible:]
|
+--[Policy-Template] <G-Key> (M) <Set>
|
|   +--[Policy-Configuration] <Set> (O)
|   |
|   |   +--[Rule-Template-Key] <List> (M)
|   |   |
|   |   |   +--[Precedence] (M)
|   |   |
|   +--[Rule-Template] <L-Key> (M) <Set>
|   |
|   |   +--[Descriptor-Match-Type] (M)
|   |   +--[Descriptor-Configuration] <Set> (M)
|   |   |
|   |   |   +--[Direction] (O)
|   |   |
|   |   +--[Action-Configuration] <Set> (M)
|   |   |
|   |   |   +--[Action-Order] (M)
|   |   |
|   |   +--[Rule-Configuration] (O)
|   +--[Descriptor-Template] <L-Key> (M) <Set>
|   |
|   |   +--[Descriptor-Type] (O)
|   |   +--[Attribute-Expression] <Set> (M)
|   +--[Action-Template] <L-Key> (M) <Set>
|   |
|   |   +--[Action-Type] (O)
|   |   +--[Attribute-Expression] <Set> (M)

```

Figure 12: Policy Information Model

The Policy structure defines Policy-Set, Rule-Set, Descriptor-Set, and Action-Set, as follows:

**Policy-Template:** <Set> A set of Policy structures, indexed by Policy-Key, each of which is determined by a list of Rules referenced by their Rule-Key. Each Policy structure contains the following:

**Policy-Key:** Identifies and enables reference to this Policy definition.

**Rule-Template-Key:** Enables reference to a Rule template definition.

**Rule-Precedence:** For each Rule identified by a Rule-Template-Key in the Policy, specifies the order in which that Rule must be applied. The lower the numerical value of Precedence, the higher the rule precedence. Rules with equal precedence MAY be executed in parallel if supported by the DPN. If this value is absent, the rules SHOULD be applied in the order in which they appear in the Policy.

**Rule-Template-Set:** A set of Rule Template definitions indexed by Rule-Key. Each Rule is defined by a list of Descriptors (located by Descriptor-Key) and a list of Actions (located by Action-Key) as follows:

**Rule-Template-Key:** Identifies and enables reference to this Rule definition.

**Descriptor-Match-Type** Indicates whether the evaluation of the Rule proceeds by using conditional-AND, or conditional-OR, on the list of Descriptors.

**Descriptor-Configuration:** References a Descriptor template definition, along with an expression which names the Attributes for this instantiation from the Descriptor-Template and also specifies whether each Attribute of the Descriptor has a default value or a statically configured value, according to the syntax specified in Section 4.2.

**Direction:** Indicates if a rule applies to uplink traffic, to downlink traffic, or to both uplink and downlink traffic. Applying a rule to both uplink and downlink traffic, in case of symmetric rules, eliminates the requirement for a separate entry for each direction. When not present, the direction is implied by the Descriptor's values.

**Action-Configuration:** References an Action Template definition,

along with an expression which names the Attributes for this instantiation from the Action-Template and also specifies whether each Attribute of the Action has a default value or a statically configured value, according to the syntax specified in Section 4.2.

Action-Order: Defines the order in which actions are executed when the associated traffic descriptor selects the packet.

Descriptor-Template-Set: A set of traffic Descriptor Templates, each of which can be evaluated on the incoming or outgoing packet, returning a TRUE or FALSE value, defined as follows:

Descriptor-Template-Key: Identifies and enables reference to this descriptor template definition.

Attribute-Expression: An expression which defines an Attribute in the Descriptor-Template and also specifies whether the Template also defines a default value or a statically configured value for the Attribute of the Descriptor has, according to the syntax specified in Section 4.2.

Descriptor-Type: Identifies the type of descriptor, e.g. an IPv6 traffic selector per [RFC6088].

Action-Template-Set: A set of Action Templates defined as follows:

Action-Template-Key: Identifies and enables reference to this action template definition.

Attribute-Expression: An expression which defines an Attribute in the Action-Template and also specifies whether the Template also defines a default value or a statically configured value for the Attribute of the Action has, according to the syntax specified in Section 4.2.

Action-Type: Identifies the type of an action for unambiguous interpretation of an Action-Value entry.

#### 4.9.6. Mobility-Context Information Model

The Mobility-Context structure holds entries associated with a mobile node and its mobility sessions (flows). It is created on a DPN during the mobile node's registration to manage the mobile node's flows. Flow information is added or deleted from the Mobility-Context as needed to support new flows or to deallocate resources for flows that are deactivated. Descriptors are used to characterize the nature and resource requirement for each flow.

Termination of a Mobility-Context implies termination of all flows represented in the Mobility-Context, e.g. after deregistration of a mobile node. If any Child-Contexts are defined, they are also terminated.

```

+-[Mobility-Context] <G-Key> <Set>
|
|   +-[Extensible:~ FALSE]
|   +-[Delegating-IP-Prefix:] <Set> (0)
|   +-[Parent-Context] (0)
|   +-[Child-Context] <Set> (0)
|   +-[Service-Group-Key] <Set> (0)
|   +-[Mobile-Node]
|   |
|   |   +-[IP-Address] <Set> (0)
|   |   +-[MN-Policy-Configuration] <Set>
|   +-[Domain-Key]
|   |   +-[Domain-Policy-Configuration] <Set>
|   +-[DPN-Key] <Set>
|   |   +-[Role]
|   |   +-[DPN-Policy-Configuration] <Set>
|   |   +-[ServiceDataFlow] <L-Key> <Set> (0)
|   |   |
|   |   |   +-[Service-Group-Key] (0)
|   |   |   +-[Interface-Key] <Set>
|   |   |   +-[ServiceDataFlow-Policy-
|   |   |       Configuration] <Set> (0)
|   |   |
|   |   |   +-[Direction]

```

Figure 13: Mobility-Context Information Model

The Mobility-Context Substructure holds the following entries:

**Mobility-Context-Key:** Identifies a Mobility-Context

**Delegating-IP-Prefix-Set:** Delegated IP Prefixes assigned to the Mobility-Context

**Parent-Context:** If present, a Mobility Context from which the Attributes and Attribute Values of this Mobility Context are inherited.

**Child-Context-Set:** A set of Mobility Contexts which inherit the Attributes and Attribute Values of this Mobility Context.

**Service-Group-Key:** Service-Group(s) used during DPN assignment and re-assignment.

**Mobile-Node:** Attributes specific to the Mobile Node. It contains the following

IP-Address-Set IP addresses assigned to the Mobile Node.

MN-Policy-Configuration-Set For each MN-Policy in the set, a key and relevant information for the Policy Attributes.

Domain-Key: Enables access to a Domain instance.

Domain-Policy-Configuration-Set: For each Domain-Policy in the set, a key and relevant information for the Policy Attributes.

DPN-Key-Set: Enables access to a DPN instance assigned to a specific role, i.e. this is a Set that uses DPN-Key and Role as a compound key to access specific set instances.

Role: Role this DPN fulfills in the Mobility-Context.

DPN-Policy-Configuration-Set: For each DPN-Policy in the set, a key and relevant information for the Policy Attributes.

ServiceDataFlow-Key-Set: Characterizes a traffic flow that has been configured (and provided resources) on the DPN to support data-plane traffic to and from the mobile device.

Service-Group-Key: Enables access to a Service-Group instance.

Interface-Key-Set: Assigns the selected interface of the DPN.

ServiceDataFlow-Policy-Configuration-Set: For each Policy in the set, a key and relevant information for the Policy Attributes.

Direction: Indicates if the reference Policy applies to uplink or downlink traffic, or to both, uplink- and downlink traffic. Applying a rule to both, uplink- and downlink traffic, in case of symmetric rules, allows omitting a separate entry for each direction. When not present the value is assumed to apply to both directions.

#### 4.9.7. Monitor Information Model

Monitors provide a mechanism to produce reports when events occur. A Monitor will have a target that specifies what is to be watched.

The attribute/entity to be monitored places certain constraints on the configuration that can be specified. For example, a Monitor using a Threshold configuration cannot be applied to a Mobility-Context, because it does not have a threshold. Such a monitor configuration could be applied to a numeric threshold property of a Context.

```

|
+--[Monitor] <G-Key> <List>
|           +-[Extensible:]
|           +-[Target:]
|           +-[Deferrable]
|           +-[Configuration]

```

Figure 14: Monitor Substructure

Monitor-Key: Identifies the Monitor.

Target: Description of what is to be monitored. This can be a Service Data Flow, a Policy installed upon a DPN, values of a Mobility-Context, etc. The target name is the absolute information model path (separated by '/') to the attribute / entity to be monitored.

Deferrable: Indicates that a monitoring report can be delayed up to a defined maximum delay, set in the Agent, for possible bundling with other reports.

Configuration: Determined by the Monitor subtype. The monitor report is specified by the Configuration. Four report types are defined:

- \* "Periodic" reporting specifies an interval by which a notification is sent.
- \* "Event-List" reporting specifies a list of event types that, if they occur and are related to the monitored attribute, will result in sending a notification.
- \* "Scheduled" reporting specifies the time (in seconds since Jan 1, 1970) when a notification for the monitor should be sent. Once this Monitor's notification is completed the Monitor is automatically de-registered.
- \* "Threshold" reporting specifies one or both of a low and high threshold. When these values are crossed a corresponding notification is sent.

## 5. Security Considerations

Detailed protocol implementations for DMM Forwarding Policy Configuration must ensure integrity of the information exchanged between a FPC Client and a FPC Agent. Required Security Associations may be derived from co-located functions, which utilize the FPC Client and FPC Agent respectively.

General usage of FPC MUST consider the following:

FPC Naming Section 4.5 permits arbitrary string values but a user MUST avoid placing sensitive or vulnerable information in those values.

Policies that are very narrow and permit the identification of specific traffic, e.g. that of a single user, SHOULD be avoided.

## 6. IANA Considerations

TBD

## 7. Work Team Participants

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

## 8. References

### 8.1. 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>>.
- [RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, DOI 10.17487/RFC6088, January 2011, <<https://www.rfc-editor.org/info/rfc6088>>.

### 8.2. Informative References



[I-D.bertz-dime-policygroups]

Bertz, L. and M. Bales, "Diameter Policy Groups and Sets", Work in Progress, Internet-Draft, draft-bertz-dime-policygroups-06, 18 June 2018, <<http://www.ietf.org/internet-drafts/draft-bertz-dime-policygroups-06.txt>>.

[I-D.ietf-dmm-deployment-models]

Gundavelli, S. and S. Jeon, "DMM Deployment Models and Architectural Considerations", Work in Progress, Internet-Draft, draft-ietf-dmm-deployment-models-04, 15 May 2018, <<http://www.ietf.org/internet-drafts/draft-ietf-dmm-deployment-models-04.txt>>.

[RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <<https://www.rfc-editor.org/info/rfc7333>>.

#### Appendix A. Implementation Status

Three FPC Agent implementations have been made to date. The first was based upon Version 03 of the draft and followed Model 1. The second follows Version 04 of the document. Both implementations were OpenDaylight plug-ins developed in Java by Sprint. Version 04 is now primarily enhanced by GS Labs. Version 03 was known as fpcagent and version 04's implementation is simply referred to as 'fpc'. A third has been developed on an ONOS Controller for use in MCORD projects.

fpcagent's intent was to provide a proof of concept for FPC Version 03 Model 1 in January 2016 and research various errors, corrections and optimizations that the Agent could make when supporting multiple DPNs.

As the code developed to support OpenFlow and a proprietary DPN from a 3rd party, several of the advantages of a multi-DPN Agent became obvious including the use of machine learning to reduce the number of Flows and Policy entities placed on the DPN. This work has driven new efforts in the DIME WG, namely Diameter Policy Groups [I-D.bertz-dime-policygroups].

A throughput performance of tens per second using various NetConf based solutions in OpenDaylight made fpcagent, based on version 03, undesirable for call processing. The RPC implementation improved throughput by an order of magnitude but was not useful based upon FPC's Version 03 design using two information models. During this time the features of version 04 and its converged model became attractive and the fpcagent project was closed in August 2016. fpcagent will no longer be developed and will remain a proprietary implementation.

The learnings of fpcagent has influenced the second project, fpc. Fpc is also an OpenDaylight project but is an open source release as the Opendaylight FpcAgent plugin ([https://wiki.opendaylight.org/view/Project\\_Proposals:FpcAgent](https://wiki.opendaylight.org/view/Project_Proposals:FpcAgent)). This project is scoped to be a fully compliant FPC Agent that supports multiple DPNs including those that communicate via OpenFlow. The following features present in this draft and others developed by the FPC development team have already led to an order of magnitude improvement.

Migration of non-realtime provisioning of entities such as topology and policy allowed the implementation to focus only on the rpc.

Using only 5 messages and 2 notifications has also reduced implementation time.

Command Sets, an optional feature in this specification, have eliminated 80% of the time spent determining what needs to be done with a Context during a Create or Update operation.

Op Reference is an optional feature modeled after video delivery. It has reduced unnecessary cache lookups. It also has the additional benefit of allowing an Agent to become cacheless and effectively act as a FPC protocol adapter remotely with multi-DPN support or co-located on the DPN in a single-DPN support model.

Multi-tenant support allows for Cache searches to be partitioned for clustering and performance improvements. This has not been capitalized upon by the current implementation but is part of the development roadmap.

Use of Contexts to pre-provision policy has also eliminated any processing of Ports for DPNs which permitted the code for CONFIGURE and CONF\_BUNDLE to be implemented as a simple nested FOR loops (see below).

Initial v04 performance results without code optimizations or tuning allow reliable provisioning of 1K FPC Mobility-Contexts processed per second on a 12 core server. This results in 2x the number of transactions on the southbound interface to a proprietary DPN API on the same machine.

fpc currently supports the following:

- 1 proprietary DPN API

Policy and Topology as defined in this specification using OpenDaylight North Bound Interfaces such as NetConf and RestConf

CONFIG and CONF\_BUNDLE (all operations)

DPN assignment, Tunnel allocations and IPv4 address assignment by the Agent or Client.

Immediate Response is always an OK\_NOTIFY\_FOLLOWS.

```
assignment system (receives rpc call):
  perform basic operation integrity check
  if CONFIG then
    goto assignments
    if assignments was ok then
      send request to activation system
      respond back to client with assignment data
    else
      send back error
    end if
  else if CONF_BUNDLE then
    for each operation in bundles
      goto assignments
      if assignments was ok then
        hold onto data
      else
        return error with the assignments that occurred in
        prior operations (best effort)
      end if
    end for
    send bundles to activation systems
  end if

assignments:
  assign DPN, IPv4 Address and/or tunnel info as required
  if an error occurs undo all assignments in this operation
  return result

activation system:
  build cache according to op-ref and operation type
  for each operation
    for each Context
      for each DPN / direction in Context
        perform actions on DPN according to Command Set
      end for
    end for
  end for
  commit changes to in memory cache
  log transaction for tracking and notification
  (CONFIG_RESULT_NOTIFY)
```

Figure 15: fpc pseudo code

For further information please contact Lyle Bertz who is also a co-author of this document.

NOTE: Tenant support requires binding a Client ID to a Tenant ID (it is a one to many relation) but that is outside of the scope of this specification. Otherwise, the specification is complete in terms of providing sufficient information to implement an Agent.

#### Authors' Addresses

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

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

Lyle Bertz  
6220 Sprint Parkway  
Overland Park KS, 66251,  
United States of America

Email: lylebe551144@gmail.com

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

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

Sri Gundavelli  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
United States of America

Email: sgundave@cisco.com

Danny Moses

Email: danny.moses@intel.com

Charles E. Perkins  
Futurewei Inc.  
2330 Central Expressway  
Santa Clara, CA 95050  
United States of America

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

DMM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 20, 2018

A. Yegin  
Actility  
D. Moses  
Intel  
K. Kweon  
J. Lee  
J. Park  
Samsung  
S. Jeon  
Sungkyunkwan University  
March 19, 2018

On Demand Mobility Management  
draft-ietf-dmm-ondemand-mobility-14

Abstract

Applications differ with respect to whether they need IP session continuity and/or IP address reachability. The network providing the same type of service to any mobile host and any application running on the host yields inefficiencies. This document describes a solution for taking the application needs into account by selectively providing IP session continuity and IP address reachability on a per-socket basis.

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 September 20, 2018.

Copyright Notice

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

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

## Table of Contents

|                                           |    |
|-------------------------------------------|----|
| 1. Introduction                           | 2  |
| 2. Notational Conventions                 | 4  |
| 3. Solution                               | 4  |
| 3.1. Types of IP Addresses                | 4  |
| 3.2. Granularity of Selection             | 5  |
| 3.3. On Demand Nature                     | 6  |
| 3.4. Conveying the Desired Address Type   | 7  |
| 4. Usage example                          | 8  |
| 5. Backwards Compatibility Considerations | 10 |
| 5.1. Applications                         | 10 |
| 5.2. IP Stack in the Mobile Host          | 10 |
| 5.3. Network Infrastructure               | 10 |
| 5.4. Merging this work with RFC5014       | 11 |
| 6. Summary of New Definitions             | 11 |
| 6.1. New APIs                             | 11 |
| 6.2. New Flags                            | 12 |
| 7. Security Considerations                | 13 |
| 8. IANA Considerations                    | 13 |
| 9. Contributors                           | 13 |
| 10. Acknowledgements                      | 13 |
| 11. References                            | 13 |
| 11.1. Normative References                | 13 |
| 11.2. Informative References              | 14 |
| Authors' Addresses                        | 15 |

## 1. Introduction

In the context of Mobile IP [RFC5563][RFC6275][RFC5213][RFC5944], the following two attributes are defined for IP service provided to mobile hosts:

**IP session continuity:** The ability to maintain an ongoing IP session by keeping the same local end-point IP address throughout the session despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may change between two independent IP sessions, but that does not jeopardize its IP



session continuity. IP session continuity is essential for mobile hosts to maintain ongoing flows without any interruption.

IP address reachability: The ability to maintain the same IP address for an extended period of time. The IP address stays the same across independent IP sessions, and even in the absence of any IP session. The IP address may be published in a long-term registry (e.g., DNS), and is made available for serving incoming (e.g., TCP) connections. IP address reachability is essential for mobile hosts to use specific/published IP addresses.

Mobile IP is designed to provide both IP session continuity and IP address reachability to mobile hosts. Architectures utilizing these protocols (e.g., 3GPP, 3GPP2, WIMAX) ensure that any mobile host attached to the compliant networks can enjoy these benefits. Any application running on these mobile hosts is subjected to the same treatment with respect to IP session continuity and IP address reachability.

It should be noted that in reality not every application may need these benefits. IP address reachability is required for applications running as servers (e.g., a web server running on the mobile host). But, a typical client application (e.g., web browser) does not necessarily require IP address reachability. Similarly, IP session continuity is not required for all types of applications either. Applications performing brief communication (e.g., ping) can survive without having IP session continuity support.

Achieving IP session continuity and IP address reachability with Mobile IP incurs some cost. Mobile IP protocol forces the mobile host's IP traffic to traverse a centrally-located router (Home Agent, HA), which incurs additional transmission latency and use of additional network resources, adds to the network CAPEX and OPEX, and decreases the reliability of the network due to the introduction of a single point of failure [RFC7333]. Therefore, IP session continuity and IP address reachability SHOULD be provided only when necessary.

Furthermore, when an application needs session continuity, it may be able to satisfy that need by using a solution above the IP layer, such as MPTCP [RFC6824], SIP mobility [RFC3261], or an application-layer mobility solution. These higher-layer solutions are not subject to the same issues that arise with the use of Mobile IP since they can utilize the most direct data path between the end-points. But, if Mobile IP is being applied to the mobile host, the higher-layer protocols are rendered useless because their operation is inhibited by Mobile IP. Since Mobile IP ensures that the IP address of the mobile host remains fixed (despite the location and movement

of the mobile host), the higher-layer protocols never detect the IP-layer change and never engage in mobility management.

This document proposes a solution for applications running on mobile hosts to indicate whether they need IP session continuity or IP address reachability. The network protocol stack on the mobile host, in conjunction with the network infrastructure, provides the required type of IP service. It is for the benefit of both the users and the network operators not to engage an extra level of service unless it is absolutely necessary. It is expected that applications and networks compliant with this specification will utilize this solution to use network resources more efficiently.

## 2. Notational Conventions

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].

## 3. Solution

### 3.1. Types of IP Addresses

Four types of IP addresses are defined with respect to mobility management.

#### - Fixed IP Address

A Fixed IP address is an address with a guarantee to be valid for a very long time, regardless of whether it is being used in any packet to/from the mobile host, or whether or not the mobile host is connected to the network, or whether it moves from one point-of-attachment to another (with a different IP prefix) while it is connected.

Fixed IP addresses are required by applications that need both IP session continuity and IP address reachability.

#### - Session-lasting IP Address

A session-lasting IP address is an address with a guarantee to be valid throughout the IP session(s) for which it was requested. It is guaranteed to be valid even after the mobile host had moved from one point-of-attachment to another (with a different IP prefix).

Session-lasting IP addresses are required by applications that need IP session continuity but do not need IP address reachability.

- Non-persistent IP Address

This type of IP address does not provide IP session continuity nor IP address reachability. The IP address is created from an IP prefix that is obtained from the serving IP gateway and is not maintained across gateway changes. In other words, the IP prefix may be released and replaced by a new one when the IP gateway changes due to the movement of the mobile host forcing the creation of a new source IP address with the updated allocated IP prefix.

- Graceful Replacement IP Address

In some cases, the network cannot guarantee the validity of the provided IP prefix throughout the duration of the IP session, but can provide a limited graceful period of time in which both the original IP prefix and a new one are valid. This enables the application some flexibility in the transition from the existing source IP address to the new one.

This gracefulness is still better than the non-persistence type of address for applications that can handle a change in their source IP address but require that extra flexibility.

Applications running as servers at a published IP address require a Fixed IP Address. Long-standing applications (e.g., an SSH session) may also require this type of address. Enterprise applications that connect to an enterprise network via virtual LAN require a Fixed IP Address.

Applications with short-lived transient IP sessions can use Session-lasting IP Addresses. For example: Web browsers.

Applications with very short IP sessions, such as DNS clients and instant messengers, can utilize Non-persistent IP Addresses. Even though they could very well use Fixed or Session-lasting IP Addresses, the transmission latency would be minimized when a Non-persistent IP Addresses are used.

Applications that can tolerate a short interruption in connectivity can use the Graceful-replacement IP addresses. For example, a streaming client that has buffering capabilities.

### 3.2. Granularity of Selection

IP address type selection is made on a per-socket granularity. Different parts of the same application may have different needs. For example, the control-plane of an application may require a Fixed

IP Address in order to stay reachable, whereas the data-plane of the same application may be satisfied with a Session-lasting IP Address.

### 3.3. On Demand Nature

At any point in time, a mobile host may have a combination of IP addresses configured. Zero or more Non-persistent, zero or more Session-lasting, zero or more Fixed and zero or more Graceful-Replacement IP addresses may be configured by the IP stack of the host. The combination may be as a result of the host policy, application demand, or a mix of the two.

When an application requires a specific type of IP address and such an address is not already configured on the host, the IP stack SHALL attempt to configure one. For example, a host may not always have a Session-lasting IP address available. When an application requests one, the IP stack SHALL make an attempt to configure one by issuing a request to the network (see Section 3.4 below for more details). If the operation fails, the IP stack SHALL fail the associated socket request and return an error. If successful, a Session-lasting IP Address gets configured on the mobile host. If another socket requests a Session-lasting IP address at a later time, the same IP address may be served to that socket as well. When the last socket using the same configured IP address is closed, the IP address may be released or kept for future applications that may be launched and require a Session-lasting IP address.

In some cases it might be preferable for the mobile host to request a new Session-lasting IP address for a new opening of an IP session (even though one was already assigned to the mobile host by the network and might be in use in a different, already active IP session). It is outside the scope of this specification to define criteria for choosing to use available addresses or choosing to request new ones. It supports both alternatives (and any combination).

It is outside the scope of this specification to define how the host requests a specific type of prefix and how the network indicates the type of prefix in its advertisement or in its reply to a request).

The following are matters of policy, which may be dictated by the host itself, the network operator, or the system architecture standard:

- The initial set of IP addresses configured on the host at boot time.

- Permission to grant various types of IP addresses to a requesting application.
- Determination of a default address type when an application does not make any explicit indication, whether it already supports the required API or it is just a legacy application.

### 3.4. Conveying the Desired Address Type

[RFC5014] introduced the ability of applications to influence the source address selection with the `IPV6_ADDR_PREFERENCE` option at the `IPPROTO_IPV6` level. This option is used with `setsockopt()` and `getsockopt()` calls to set/get address selection preferences.

Extending this further by adding more flags does not work when a request for an address of a certain type results in requiring the IP stack to wait for the network to provide the desired source IP prefix and hence causing the `setsockopt()` call to block until the prefix is allocated (or an error indication from the network is received).

Alternatively a new Socket API is defined - `getsc()` which allows applications to express their desired type of session continuity service. The new `getsc()` API will return an IPv6 address that is associated with the desired session continuity service and with status information indicating whether or not the desired service was provided.

An application that wishes to secure a desired service will call `getsc()` with the service type definition and a place to contain the provided IP address, and call `bind()` to associate that IP address with the Socket (See pseudo-code example in Section 4 below).

When the IP stack is required to use a source IP address of a specified type, it can use an existing address, or request a new IP prefix (of the same type) from the network and create a new one. If the host does not already have an IPv6 prefix of that specific type, it MUST request one from the network.

Using an existing address from an existing prefix is faster but might yield a less optimal route (if a hand-off event occurred after its configuration). On the other hand, acquiring a new IP prefix from the network may be slower due to signaling exchange with the network.

Applications can control the stack's operation by setting a new flag - `ON_NET` flag - which directs the IP stack whether to use a preconfigured source IP address (if exists) or to request a new IPv6 prefix from the current serving network and configure a new IP address.

This new flag is added to the set of flags in the IPV6\_ADDR\_PREFERENCES option at the IPPROTO\_IPV6 level. It is used in setsockopt() to set the desired behavior.

#### 4. Usage example

The following example shows pseudo-code for creating a Stream socket (TCP) with a Session-Lasting source IP address:

```
#include <sys/socket.h>
#include <netinet/in.h>

// Socket information
int s ; // Socket id

// Source information (for setsockopt() and bind())
sockaddr_in6 sourceInfo // my address and port for bind()
in6_addr sourceAddress // will contain the provisioned
                        // source IP address
uint8_t sc_type = IPV6_REQUIRE_SESSION_LASTING_IP ;
                        // For requesting a Session-Lasting
                        // source IP address

// Destination information (for connect())
sockaddr_in6 serverInfo ; // server info for connect()

// Create an IPv6 TCP socket
s = socket(AF_INET6, SOCK_STREAM, 0) ;
if (s!=0) {
    // Handle socket creation error
    // ...
} // if socket creation failed
else {
    // Socket creation is successful
    // The application cannot connect yet, since it wants to use
    // a Session-Lasting source IP address It needs to request
    // the Session-Lasting source IP before connecting
    if (setsockopt(s, &sourceAddress, &sc_type) == 0){
        // setting session continuity to Session Lasting is
        // Successful. sourceAddress now contains the Session-
        // Lasting source IP address

        // Bind to that source IP address
        sourceInfo.sin6_family = AF_INET6 ;
        sourceInfo.sin6_port = 0 // let the stack choose the port
        sourceInfo.sin6_address = sourceAddress ;
        // Use the source address that was
```

```
                                // generated by the setsc() call
if (bind(s, &sourceInfo, sizeof(sourceInfo))==0){
    // Set the desired server's information for connect()
    serverInfo.sin6_family = AF_INET6 ;
    serverInfo.sin6_port = SERVER_PORT_NUM ;
    serverAddress.sin6_addr = SERVER_IPV6_ADDRESS ;

    // Connect to the server
    if (connect(s, &serverInfo, sizeof(serverInfo))==0) {
        // connect successful (3-way handshake has been
        // completed with Session-Lasting source address.
        // Continue application functionality
        // ...
    } // if connect() is successful
    else {
        // connect failed
        // ...
        // Application code that handles connect failure and
        // closes the socket
        // ...
    } // if connect() failed
} // if bind() successful
else {
    // bind() failed
    // ...
    // Application code that handles bind failure and
    // closes the socket
    // ...
} // if bind() failed
} // if setsc() was successful and of a Session-Lasting
  // source IP address was provided
else {
    // application code that does not use Session-lasting IP
    // address. The application may either connect without
    // the desired Session-lasting service, or close the
    // socket...
} // if setsc() failed
} // if socket was created successfully

// The rest of the application's code
// ...
```

## 5. Backwards Compatibility Considerations

Backwards compatibility support is REQUIRED by the following 3 types of entities:

- The Applications on the mobile host
- The IP stack in the mobile host
- The network infrastructure

### 5.1. Applications

Legacy applications that do not support the OnDemand functionality will use the legacy API and will not be able to take advantage of the On-Demand Mobility feature.

Applications using the new OnDemand functionality MUST be aware that they may be executed in legacy environments that do not support it. Such environments may include a legacy IP stack on the mobile host, legacy network infrastructure, or both. In either case, the API will return an error code and the invoking applications may just give up and use legacy calls.

### 5.2. IP Stack in the Mobile Host

New IP stacks MUST continue to support all legacy operations. If an application does not use On-Demand functionality, the IP stack MUST respond in a legacy manner.

If the network infrastructure supports On-Demand functionality, the IP stack SHOULD follow the application request: If the application requests a specific address type, the stack SHOULD forward this request to the network. If the application does not request an address type, the IP stack MUST NOT request an address type and leave it to the network's default behavior to choose the type of the allocated IP prefix. If an IP prefix was already allocated to the host, the IP stack uses it and may not request a new one from the network.

### 5.3. Network Infrastructure

The network infrastructure may or may not support the On-Demand functionality. How the IP stack on the host and the network infrastructure behave in case of a compatibility issue is outside the scope of this API specification.



#### 5.4. Merging this work with RFC5014

[RFC5014] defines new flags that may be used with `setsockopt()` to influence source IP address selection for a socket. The list of flags include: source home address, care-of address, temporary address, public address CGA (Cryptographically Created Address) and non-CGA. When applications require session continuity service and use `setsc()` and `bind()`, they SHOULD NOT set the flags specified in [RFC5014].

However, if an application sets a specific option using `setsockopt()` with one of the flags specified in [RFC5014] and also selects a source IP address using `setsc()` and `bind()` the IP address that was generated by `setsc()` and bound using `bind()` will be the one used by traffic generated using that socket and options set by `setsockopt()` will be ignored.

If `bind()` was not invoked after `setsc()` by the application, the IP address generated by `setsc()` will not be used and traffic generated by the socket will use a source IP address that complies with the options selected by `setsockopt()`.

### 6. Summary of New Definitions

#### 6.1. New APIs

`setsc()` enables applications to request a specific type of source IP address in terms of session continuity. Its definition is:

```
int setsc(int sockfd, in6_addr *sourceAddress, sc_type addressType);
```

Where:

- sockfd - is the socket descriptor of the socket with which a specific address type is associated
- sourceAddress - is a pointer to an area allocated for setsc() to place the generated source IP address of the desired session continuity type
- addressType - Is the desired type of session continuity service. It is a 3-bit field containing one of the following values:
  - 0 - Reserved
  - 1 - FIXED\_IPV6\_ADDRESS
  - 2 - SESSION\_LASTING\_IPV6\_ADDRESS
  - 3 - NON\_PERSISTENT\_IPV6\_ADDRESS
  - 4 - GRACEFUL\_REPLACEMENT\_IPV6\_ADDRESS
  - 5-7 - Reserved

setsc() returns the status of the operation:

- 0 - Address was successfully generated
- EAI\_REQUIREDIPNOTSUPPORTED - the required service type is not supported
- EAI\_REQUIREDIPFAILED - the network could not fulfill the desired request

setsc() MAY block the invoking thread if it triggers the TCP/IP stack to request a new IP prefix from the network to construct the desired source IP address. If an IP prefix with the desired session continuity features already exists (was previously allocated to the mobile host) and the stack is not required to request a new one as a result of setting the IPV6\_REQUIRE\_SRC\_ON\_NET flag (defined below), setsc() MAY return immediately with the constructed IP address and will not block the thread.

## 6.2. New Flags

The following flag is added to the list of flags in the IPV6\_ADDR\_PREFERENCE option at the IPPROTO6 level:

IPV6\_REQUIRE\_SRC\_ON\_NET - set IP stack address allocation behavior

If set, the IP stack will request a new IPv6 prefix of the desired type from the current serving network and configure a new source IP address. If reset, the IP stack will use a preconfigured one if it exists. If there is no preconfigured IP address of the desired type, a new prefix will be requested and used for creating the IP address.

## 7. Security Considerations

The setting of certain IP address type on a given socket may be restricted to privileged applications. For example, a Fixed IP Address may be provided as a premium service and only certain applications may be allowed to use them. Setting and enforcement of such privileges are outside the scope of this document.

## 8. IANA Considerations

This document has no IANA considerations.

## 9. Contributors

This document was merged with [I-D.sijeon-dmm-use-cases-api-source]. We would like to acknowledge the contribution of the following people to that document as well:

Sergio Figueiredo  
Altran Research, France  
Email: sergio.figueiredo@altran.com

Younghan Kim  
Soongsil University, Korea  
Email: younghak@ssu.ac.kr

John Kaippallimalil  
Huawei, USA  
Email: john.kaippallimalil@huawei.com

## 10. Acknowledgements

We would like to thank Wu-chi Feng, Alexandru Petrescu, Jouni Korhonen, Sri Gundavelli, Dave Dolson and Lorenzo Colitti for their valuable comments and suggestions on this work.

## 11. References

### 11.1. 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>>.

- [RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <<https://www.rfc-editor.org/info/rfc5014>>.

## 11.2. Informative References

- [I-D.sijeon-dmm-use-cases-api-source] Jeon, S., Figueiredo, S., Kim, Y., and J. Kaippallimalil, "Use Cases and API Extension for Source IP Address Selection", draft-sijeon-dmm-use-cases-api-source-07 (work in progress), September 2017.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC5563] Leung, K., Dommety, G., Yegani, P., and K. Chowdhury, "WiMAX Forum / 3GPP2 Proxy Mobile IPv4", RFC 5563, DOI 10.17487/RFC5563, February 2010, <<https://www.rfc-editor.org/info/rfc5563>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised", RFC 5944, DOI 10.17487/RFC5944, November 2010, <<https://www.rfc-editor.org/info/rfc5944>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/info/rfc6824>>.
- [RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <<https://www.rfc-editor.org/info/rfc7333>>.

Authors' Addresses

Alper Yegin  
Actility  
Istanbul  
Turkey

Email: [alper.yegin@actility.com](mailto:alper.yegin@actility.com)

Danny Moses  
Intel Corporation  
Petah Tikva  
Israel

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

Kisuk Kweon  
Samsung  
Suwon  
South Korea

Email: [kisuk.kweon@samsung.com](mailto:kisuk.kweon@samsung.com)

Jinsung Lee  
Samsung  
Suwon  
South Korea

Email: [js81.lee@samsung.com](mailto:js81.lee@samsung.com)

Jungshin Park  
Samsung  
Suwon  
South Korea

Email: [shin02.park@samsung.com](mailto:shin02.park@samsung.com)

Seil Jeon  
Sungkyunkwan University  
Suwon  
South Korea

Email: [seiljeon@skku.edu](mailto:seiljeon@skku.edu)

DMM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 31, 2020

A. Yegin  
Actility  
D. Moses  
Intel  
S. Jeon  
Sungkyunkwan University  
July 30, 2019

On Demand Mobility Management  
draft-ietf-dmm-ondemand-mobility-18

Abstract

Applications differ with respect to whether they need session continuity and/or IP address reachability. The network providing the same type of service to any mobile host and any application running on the host yields inefficiencies, as described in [RFC7333]. This document defines a new concept of enabling applications to influence the network's mobility services (session continuity and/or IP address reachability) on a per-Socket basis, and suggests extensions to the networking stack's API to accommodate this concept.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 31, 2020.

Copyright Notice

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

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

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

## Table of Contents

|                                                          |    |
|----------------------------------------------------------|----|
| 1. Introduction . . . . .                                | 2  |
| 2. Notational Conventions . . . . .                      | 4  |
| 3. Solution . . . . .                                    | 4  |
| 3.1. High-level Description . . . . .                    | 4  |
| 3.2. Types of IP Addresses . . . . .                     | 5  |
| 3.3. Granularity of Selection . . . . .                  | 6  |
| 3.4. On Demand Nature . . . . .                          | 6  |
| 4. Backwards Compatibility Considerations . . . . .      | 7  |
| 4.1. Applications . . . . .                              | 8  |
| 4.2. IP Stack in the Mobile Host . . . . .               | 8  |
| 4.3. Network Infrastructure . . . . .                    | 8  |
| 4.4. Merging this work with RFC5014 . . . . .            | 8  |
| 5. Security Considerations . . . . .                     | 9  |
| 6. IANA Considerations . . . . .                         | 10 |
| 7. Contributors . . . . .                                | 10 |
| 8. Acknowledgements . . . . .                            | 10 |
| 9. References . . . . .                                  | 10 |
| 9.1. Normative References . . . . .                      | 10 |
| 9.2. Informative References . . . . .                    | 11 |
| Appendix A. Conveying the Desired Address Type . . . . . | 11 |
| Authors' Addresses . . . . .                             | 12 |

## 1. Introduction

In the context of Mobile IP [RFC5563][RFC6275][RFC5213][RFC5944], the following two attributes are defined for IP service provided to mobile hosts:

### - Session Continuity

The ability to maintain an ongoing transport interaction by keeping the same local end-point IP address throughout the life-time of the IP socket despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may change after closing the IP socket and before opening a new one, but that does not jeopardize the ability of applications using these IP sockets to work flawlessly. Session continuity is essential for mobile hosts to maintain ongoing flows without any interruption.

#### - IP Address Reachability

The ability to maintain the same IP address for an extended period of time. The IP address stays the same across independent sessions, and even in the absence of any session. The IP address may be published in a long-term registry (e.g., DNS), and is made available for serving incoming (e.g., TCP) connections. IP address reachability is essential for mobile hosts to use specific/published IP addresses.

Mobile IP is designed to provide both session continuity and IP address reachability to mobile hosts. Architectures utilizing these protocols (e.g., 3GPP, 3GPP2, WIMAX) ensure that any mobile host attached to the compliant networks can enjoy these benefits. Any application running on these mobile hosts is subjected to the same treatment with respect to session continuity and IP address reachability.

Achieving session continuity and IP address reachability with Mobile IP incurs some cost. Mobile IP protocol forces the mobile host's IP traffic to traverse a centrally-located router (Home Agent, HA), which incurs additional transmission latency and use of additional network resources, adds to the network CAPEX and OPEX, and decreases the reliability of the network due to the introduction of a single point of failure [RFC7333]. Therefore, session continuity and IP address reachability SHOULD be provided only when necessary.

In reality not every application may need these benefits. IP address reachability is required for applications running as servers (e.g., a web server running on the mobile host). But, a typical client application (e.g., web browser) does not necessarily require IP address reachability. Similarly, session continuity is not required for all types of applications either. Applications performing brief communication (e.g., text messaging) can survive without having session continuity support.

Furthermore, when an application needs session continuity, it may be able to satisfy that need by using a solution above the IP layer, such as MPTCP [RFC6824], SIP mobility [RFC3261], or an application-layer mobility solution. These higher-layer solutions are not subject to the same issues that arise with the use of Mobile IP since they can utilize the most direct data path between the end-points. But, if Mobile IP is being applied to the mobile host, the higher-layer protocols are rendered useless because their operation is inhibited by Mobile IP. Since Mobile IP ensures that the IP address of the mobile host remains fixed (despite the location and movement of the mobile host), the higher-layer protocols never detect the IP-layer change and never engage in mobility management.



This document proposes a solution for applications running on mobile hosts to indicate when establishing the network connection ('on demand') whether they need session continuity or IP address reachability. The network protocol stack on the mobile host, in conjunction with the network infrastructure, provides the required type of service. It is for the benefit of both the users and the network operators not to engage an extra level of service unless it is absolutely necessary. It is expected that applications and networks compliant with this specification will utilize this solution to use network resources more efficiently.

## 2. Notational Conventions

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, they appear in all capitals, as shown here.

## 3. Solution

### 3.1. High-level Description

Enabling applications to indicate their mobility service requirements e.g. session continuity and/or IP address reachability, comprises the following steps:

- The application indicates to the network stack (local to the mobile host) the desired mobility service.
- The network stack assigns a source IP address based on an IP prefix with the desired services that was previously provided by the network. If such an IP prefix is not available, the network stack performs the additional steps below.
- The network stack sends a request to the network for a new source IP prefix that is associated with the desired mobility service.
- The network responds with the suitable allocated source IP prefix (or responds with a failure indication).
- If the suitable source IP prefix was allocated, the network stack constructs a source IP address and provides it to the application.

This document specifies the new address types associated with mobility services and details the interaction between the applications and the network stack steps. It uses the Socket

interface as an example for an API between applications and the network stack. Other steps are outside the scope of this document.

### 3.2. Types of IP Addresses

Four types of IP addresses are defined with respect to mobility management.

#### - Fixed IP Address

A Fixed IP address is an address with a guarantee to be valid for a very long time, regardless of whether it is being used in any packet to/from the mobile host, or whether or not the mobile host is connected to the network, or whether it moves from one point-of-attachment to another (with a different IP prefix) while it is connected.

Fixed IP addresses are required by applications that need both session continuity and IP address reachability.

#### - Session-lasting IP Address

A session-lasting IP address is an address with a guarantee to be valid throughout the life-time of the socket(s) for which it was requested. It is guaranteed to be valid even after the mobile host had moved from one point-of-attachment to another (with a different IP prefix).

Session-lasting IP addresses are required by applications that need session continuity but do not need IP address reachability.

#### - Non-persistent IP Address

This type of IP address has no guarantee to exist after a mobile host moves from one point-of-attachment to another, and therefore, no session continuity nor IP address reachability are provided. The IP address is created from an IP prefix that is obtained from the serving IP gateway and is not maintained across gateway changes. In other words, the IP prefix may be released and replaced by a new one when the IP gateway changes due to the movement of the mobile host forcing the creation of a new source IP address with the updated allocated IP prefix.

#### - Graceful Replacement IP Address

In some cases, the network cannot guarantee the validity of the provided IP prefix throughout the duration of the opened socket, but can provide a limited graceful period of time in which both the

original IP prefix and a new one are valid. This enables the application some flexibility in the transition from the existing source IP address to the new one.

This gracefulness is still better than the non-persistence type of address for applications that can handle a change in their source IP address but require that extra flexibility.

Applications running as servers at a published IP address require a Fixed IP Address. Long-standing applications (e.g., an SSH session) may also require this type of address. Enterprise applications that connect to an enterprise network via virtual LAN require a Fixed IP Address.

Applications with short-lived transient sessions can use Session-lasting IP Addresses. For example: Web browsers.

Applications with very short sessions, such as DNS clients and instant messengers, can utilize Non-persistent IP Addresses. Even though they could very well use Fixed or Session-lasting IP Addresses, the transmission latency would be minimized when a Non-persistent IP Addresses are used.

Applications that can tolerate a short interruption in connectivity can use the Graceful-replacement IP addresses. For example, a streaming client that has buffering capabilities.

### 3.3. Granularity of Selection

IP address type selection is made on a per-socket granularity. Different parts of the same application may have different needs. For example, the control-plane of an application may require a Fixed IP Address in order to stay reachable, whereas the data-plane of the same application may be satisfied with a Session-lasting IP Address.

### 3.4. On Demand Nature

At any point in time, a mobile host may have a combination of IP addresses configured. Zero or more Fixed, zero or more Session-lasting, zero or more Non-persistent and zero or more Graceful-Replacement IP addresses may be configured by the IP stack of the host. The combination may be as a result of the host policy, application demand, or a mix of the two.

When an application requires a specific type of IP address and such an address is not already configured on the host, the IP stack SHALL attempt to configure one. For example, a host may not always have a Session-lasting IP address available. When an application requests

one, the IP stack SHALL make an attempt to configure one by issuing a request to the network. If the operation fails, the IP stack SHALL fail the associated socket request and return an error. If successful, a Session-lasting IP Address gets configured on the mobile host. If another socket requests a Session-lasting IP address at a later time, the same IP address may be served to that socket as well. When the last socket using the same configured IP address is closed, the IP address may be released or kept for future applications that may be launched and require a Session-lasting IP address.

In some cases it might be preferable for the mobile host to request a new Session-lasting IP address for a new opening of an IP socket (even though one was already assigned to the mobile host by the network and might be in use in a different, already active IP sockets). It is outside the scope of this specification to define criteria for choosing to use available addresses or choosing to request new ones. It supports both alternatives (and any combination).

It is outside the scope of this specification to define how the host requests a specific type of prefix and how the network indicates the type of prefix in its advertisement or in its reply to a request.

The following are matters of policy, which may be dictated by the host itself, the network operator, or the system architecture standard:

- The initial set of IP addresses configured on the host at boot time.
- Permission to grant various types of IP addresses to a requesting application.
- Determination of a default address type when an application does not make any explicit indication, whether it already supports the required API or it is just a legacy application.

#### 4. Backwards Compatibility Considerations

Backwards compatibility support is REQUIRED by the following 3 types of entities:

- The Applications on the mobile host
- The IP stack in the mobile host
- The network infrastructure

#### 4.1. Applications

Legacy applications that do not support the On-Demand functionality will use the legacy API and will not be able to take advantage of the On-Demand Mobility feature.

Applications using the new On-Demand functionality should be aware that they may be executed in legacy environments that do not support it. Such environments may include a legacy IP stack on the mobile host, legacy network infrastructure, or both. In either case, the API will return an error code and the invoking applications may just give up and use legacy calls.

#### 4.2. IP Stack in the Mobile Host

New IP stacks (that implement On Demand functionality) MUST continue to support all legacy operations. If an application does not use On-Demand functionality, the IP stack MUST respond in a legacy manner.

If the network infrastructure supports On-Demand functionality, the IP stack SHOULD follow the application request: If the application requests a specific address type, the stack SHOULD forward this request to the network. If the application does not request an address type, the IP stack MUST NOT request an address type and leave it to the network's default behavior to choose the type of the allocated IP prefix. If an IP prefix was already allocated to the host, the IP stack uses it and may not request a new one from the network.

#### 4.3. Network Infrastructure

The network infrastructure may or may not support the On-Demand functionality. How the IP stack on the host and the network infrastructure behave in case of a compatibility issue is outside the scope of this API specification.

#### 4.4. Merging this work with RFC5014

[RFC5014] defines new flags that may be used with `setsockopt()` to influence source IP address selection for a socket. The list of flags include: source home address, care-of address, temporary address, public address CGA (Cryptographically Created Address) and non-CGA. When applications require session continuity service, they SHOULD NOT set the flags specified in [RFC5014].

However, if an application erroneously performs a combination of (1) Use `setsockopt()` to set a specific option (using one of the flags specified in [RFC5014]) and (2) Selects a source IP address type, the

IP stack will fulfill the request specified by (2) and ignore the flags set by (1).

## 5. Security Considerations

The different service types (session continuity types and address reachability) associated with the allocated IP address types, may be associated with different costs. The cost to the operator for enabling a type of service, and the cost to applications using a selected service. A malicious application may use these to generate extra billing of a mobile subscriber, and/or impose costly services on the mobile operator. When costly services are limited, malicious applications may exhaust them, preventing other applications on the same mobile host from being able to use them.

Mobile hosts that enables such service options, should provide capabilities for ensuring that only authorized applications can use the costly (or limited) service types.

The ability to select service types requires the exchange of the association of source IP prefixes and their corresponding service types, between the mobile host and mobile network. Exposing these associations may provide information to passive attackers even if the traffic that is used with these addresses is encrypted.

To avoid profiling an application according to the type of IP addresses, it is expected that prefixes provided by the mobile operator are associated to various type of addresses over time. As a result, the type of address could not be associated to the prefix, making application profiling based on the type of address harder.

The application or the OS should ensure that IP addresses regularly change to limit IP tracking by a passive observer. The application should regularly set the On Demand flag. The application should be able to ensure that session lasting IP addresses are regularly changed by setting a lifetime for example handled by the application. In addition, the application should consider the use of graceful replacement IP addresses.

Similarly, the OS may also associated IP addresses with a lifetime. Upon receiving a request for a given type of IP address, after some time, the OS should request a new address to the network even if it already has one IP address available with the requested type. This includes any type of IP address. IP addresses of type graceful replacement or non persistent should be regularly renewed by the OS.

The lifetime of an IP address may be expressed in number of seconds or in number of bytes sent through this IP address.

## 6. IANA Considerations

This document has no IANA considerations.

## 7. Contributors

This document was merged with [I-D.sijeon-dmm-use-cases-api-source]. We would like to acknowledge the contribution of the following people to that document as well:

Sergio Figueiredo  
Altran Research, France  
Email: sergio.figueiredo@altran.com

Younghan Kim  
Soongsil University, Korea  
Email: younghak@ssu.ac.kr

John Kaippallimalil  
Huawei, USA  
Email: john.kaippallimalil@huawei.com

## 8. Acknowledgements

We would like to thank Wu-chi Feng, Alexandru Petrescu, Jouni Korhonen, Sri Gundavelli, Dave Dolson Lorenzo Colitti and Daniel Migault for their valuable comments and suggestions on this work.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <<https://www.rfc-editor.org/info/rfc5014>>.
- [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>>.

## 9.2. Informative References

- [I-D.sijeon-dmm-use-cases-api-source]  
Jeon, S., Figueiredo, S., Kim, Y., and J. Kaippallimalil,  
"Use Cases and API Extension for Source IP Address  
Selection", draft-sijeon-dmm-use-cases-api-source-07 (work  
in progress), September 2017.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,  
A., Peterson, J., Sparks, R., Handley, M., and E.  
Schooler, "SIP: Session Initiation Protocol", RFC 3261,  
DOI 10.17487/RFC3261, June 2002,  
<<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V.,  
Chowdhury, K., and B. Patil, "Proxy Mobile IPv6",  
RFC 5213, DOI 10.17487/RFC5213, August 2008,  
<<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC5563] Leung, K., Dommety, G., Yegani, P., and K. Chowdhury,  
"WiMAX Forum / 3GPP2 Proxy Mobile IPv4", RFC 5563,  
DOI 10.17487/RFC5563, February 2010,  
<<https://www.rfc-editor.org/info/rfc5563>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised",  
RFC 5944, DOI 10.17487/RFC5944, November 2010,  
<<https://www.rfc-editor.org/info/rfc5944>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility  
Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July  
2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure,  
"TCP Extensions for Multipath Operation with Multiple  
Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013,  
<<https://www.rfc-editor.org/info/rfc6824>>.
- [RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J.  
Korhonen, "Requirements for Distributed Mobility  
Management", RFC 7333, DOI 10.17487/RFC7333, August 2014,  
<<https://www.rfc-editor.org/info/rfc7333>>.

## Appendix A. Conveying the Desired Address Type

Following are some suggestions of possible extensions to the Socket  
API for enabling applications to convey their session continuity and  
address reachability requirements.



[RFC5014] introduced the ability of applications to influence the source address selection with the `IPV6_ADDR_PREFERENCE` option at the `IPPROTO_IPV6` level. This option is used with `setsockopt()` and `getsockopt()` calls to set/get address selection preferences.

One alternative is to extend the definition of the `IPV6_ADDR_PREFERENCE` option with flags that express the invoker's desire. An "OnDemand" field could contain one of the values: `FIXED_IP_ADDRESS`, `SESSION_LASTING_IP_ADDRESS`, `NON_PERSISTENT_IP_ADDRESS` or `GRACEFUL_REPLACEMENT_IP_ADDRESS`.

Another alternative is to define a new Socket function used by the invoker to convey its desire. This enables the implementation of two behaviors of Socket functions: The existing "`setsockoptp()`" is a function that returns after executing, and the new "`setsc()`" (Set Service Continuity) function that may initiate a request for the desired service, and wait until the network responds with the allocated resources, before returning to the invoker.

After obtaining an IP address with the desired behavior the application can call the `bind()` Socket function to associate that received IP address with the socket.

#### Authors' Addresses

Alper Yegin  
Actility  
Istanbul  
Turkey

Email: [alper.yegin@actility.com](mailto:alper.yegin@actility.com)

Danny Moses  
Intel Corporation  
Petah Tikva  
Israel

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

Seil Jeon  
Sungkyunkwan University  
Suwon  
South Korea

Email: [seiljeon@skku.edu](mailto:seiljeon@skku.edu)

DMM Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: September 9, 2020

CJ. Bernardos  
A. de la Oliva  
UC3M  
F. Giust  
Athonet  
JC. Zuniga  
SIGFOX  
A. Mourad  
InterDigital  
March 8, 2020

Proxy Mobile IPv6 extensions for Distributed Mobility Management  
draft-ietf-dmm-pmipv6-dlif-06

## Abstract

Distributed Mobility Management solutions allow for setting up networks so that traffic is distributed in an optimal way and does not rely on centrally deployed anchors to provide IP mobility support.

There are many different approaches to address Distributed Mobility Management, as for example extending network-based mobility protocols (like Proxy Mobile IPv6), or client-based mobility protocols (like Mobile IPv6), among others. This document follows the former approach and proposes a solution based on Proxy Mobile IPv6 in which mobility sessions are anchored at the last IP hop router (called mobility anchor and access router). The mobility anchor and access router is an enhanced access router which is also able to operate as a local mobility anchor or mobility access gateway, on a per prefix basis. The document focuses on the required extensions to effectively support simultaneously anchoring several flows at different distributed gateways.

## Requirements Language

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.

## 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 September 9, 2020.

#### Copyright Notice

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

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

#### Table of Contents

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 1. Introduction . . . . .                                       | 3  |
| 2. Terminology . . . . .                                        | 4  |
| 3. PMIPv6 DMM extensions . . . . .                              | 6  |
| 3.1. Initial registration . . . . .                             | 7  |
| 3.2. The CMD as PBU/PBA relay . . . . .                         | 8  |
| 3.3. The CMD as MAAR locator . . . . .                          | 11 |
| 3.4. The CMD as MAAR proxy . . . . .                            | 12 |
| 3.5. De-registration . . . . .                                  | 13 |
| 3.6. Retransmissions and Rate Limiting . . . . .                | 14 |
| 3.7. The Distributed Logical Interface (DLIF) concept . . . . . | 14 |
| 4. Message Format . . . . .                                     | 18 |
| 4.1. Proxy Binding Update . . . . .                             | 18 |
| 4.2. Proxy Binding Acknowledgment . . . . .                     | 19 |
| 4.3. Anchored Prefix Option . . . . .                           | 19 |
| 4.4. Local Prefix Option . . . . .                              | 21 |
| 4.5. Previous MAAR Option . . . . .                             | 22 |
| 4.6. Serving MAAR Option . . . . .                              | 23 |
| 4.7. DLIF Link-local Address Option . . . . .                   | 24 |
| 4.8. DLIF Link-layer Address Option . . . . .                   | 25 |

|                                       |    |
|---------------------------------------|----|
| 5. IANA Considerations . . . . .      | 26 |
| 6. Security Considerations . . . . .  | 26 |
| 7. Acknowledgments . . . . .          | 27 |
| 8. References . . . . .               | 27 |
| 8.1. Normative References . . . . .   | 27 |
| 8.2. Informative References . . . . . | 28 |
| Authors' Addresses . . . . .          | 28 |

## 1. Introduction

The Distributed Mobility Management (DMM) paradigm aims at minimizing the impact of currently standardized mobility management solutions which are centralized (at least to a considerable extent) [RFC7333].

Current IP mobility solutions, standardized with the names of Mobile IPv6 [RFC6275], or Proxy Mobile IPv6 (PMIPv6) [RFC5213], just to cite the two most relevant examples, offer mobility support at the cost of handling operations at a cardinal point, the mobility anchor (i.e., the home agent for Mobile IPv6, and the local mobility anchor for Proxy Mobile IPv6), and burdening it with data forwarding and control mechanisms for a great amount of users. As stated in [RFC7333], centralized mobility solutions are prone to several problems and limitations: longer (sub-optimal) routing paths, scalability problems, signaling overhead (and most likely a longer associated handover latency), more complex network deployment, higher vulnerability due to the existence of a potential single point of failure, and lack of granularity of the mobility management service (i.e., mobility is offered on a per-node basis, not being possible to define finer granularity policies, as for example per-application).

The purpose of Distributed Mobility Management is to overcome the limitations of the traditional centralized mobility management [RFC7333] [RFC7429]; the main concept behind DMM solutions is indeed bringing the mobility anchor closer to the Mobile Node (MN). Following this idea, the central anchor is moved to the edge of the network, being deployed in the default gateway of the mobile node. That is, the first elements that provide IP connectivity to a set of MNs are also the mobility managers for those MNs. In this document, we call these entities Mobility Anchors and Access Routers (MAARs).

This document focuses on network-based DMM, hence the starting point is making PMIPv6 work in a distributed manner [RFC7429]. Mobility is handled by the network without the MNs involvement, but, differently from PMIPv6, when the MN moves from one access network to another, it may also change anchor router, hence requiring signaling between the anchors to retrieve the MN's previous location(s). Also, a key-aspect of network-based DMM, is that a prefix pool belongs exclusively to each MAAR, in the sense that those prefixes are

assigned by the MAAR to the MNs attached to it, and they are routable at that MAAR. Prefixes are assigned to MNs attached a MAAR at that time, but remain with those MNs as mobility occurs, remaining always routable at that MAAR as well as towards the MN itself.

We consider partially distributed schemes, where only the data plane is distributed among access routers similar to MAGs, whereas the control plane is kept centralized towards a cardinal node used as information store, but relieved from any route management and MN's data forwarding task.

## 2. Terminology

The following terms used in this document are defined in the Proxy Mobile IPv6 specification [RFC5213]:

Local Mobility Anchor (LMA)

Mobile Access Gateway (MAG)

Mobile Node (MN)

Binding Cache Entry (BCE)

Proxy Care-of Address (P-CoA)

Proxy Binding Update (PBU)

Proxy Binding Acknowledgement (PBA)

The following terms are used in this document:

**Home Control-Plane Anchor (Home-CPA or H-CPA):** The Home-CPA function hosts the mobile node (MN)'s mobility session. There can be more than one mobility session for a mobile node and those sessions may be anchored on the same or different Home-CPA's. The home-CPA will interface with the home-DPA for managing the forwarding state.

**Home Data Plane Anchor (Home-DPA or H-DPA):** The Home-DPA is the topological anchor for the MN's IP address/ prefix(es). The Home-DPA is chosen by the Home-CPA on a session- basis. The Home-DPA is in the forwarding path for all the mobile node's IP traffic.

Access Control Plane Node (Access-CPN or A-CPN): The Access-CPN is responsible for interfacing with the mobile node's Home-CPA and with the Access-DPN. The Access-CPN has a protocol interface to the Home-CPA.

Access Data Plane Node (Access-DPN or A-DPN): The Access-DPN function is hosted on the first-hop router where the mobile node is attached. This function is not hosted on a layer-2 bridging device such as a eNode(B) or Access Point.

The following terms are defined and used in this document:

MAAR (Mobility Anchor and Access Router). First hop router where the mobile nodes attach to. It also plays the role of mobility manager for the IPv6 prefixes it anchors, running the functionalities of PMIP's MAG and LMA. Depending on the prefix, it plays the role of Access-DPN, Home-DPA and Access-CPN.

CMD (Central Mobility Database). The node that stores the BCEs allocated for the MNs in the mobility domain. It plays the role of Home-CPA.

P-MAAR (Previous MAAR). When a MN moves to a new point of attachment a new MAAR might be allocated as its anchor point for future IPv6 prefixes. The MAAR that served the MN prior to new attachment becomes the P-MAAR. It is still the anchor point for the IPv6 prefixes it had allocated to the MN in the past and serves as the Home-DPA for flows using these prefixes. There might be several P-MAARs serving a MN when the MN is frequently switching points of attachment while maintaining long-lasting flows.

S-MAAR (Serving MAAR). The MAAR which the MN is currently attached to. Depending on the prefix, it plays the role of Access-DPN, Home-DPA and Access-CPN.

Anchoring MAAR. A MAAR anchoring an IPv6 prefix used by an MN.

DLIF (Distributed Logical Interface). It is a logical interface at the IP stack of the MAAR. For each active prefix used by the MN, the S-MAAR has a DLIF configured (associated to each MAAR still anchoring flows). In this way, an S-MAAR exposes itself towards each MN as multiple routers, one as itself and one per P-MAAR.

### 3. PMIPv6 DMM extensions

The solution consists of de-coupling the entities that participate in the data and the control planes: the data plane becomes distributed and managed by the MAARs near the edge of the network, while the control plane, besides those on the MAARs, relies on a central entity called Central Mobility Database (CMD). In the proposed architecture, the hierarchy present in PMIPv6 between LMA and MAG is preserved, but with the following substantial variations:

- o The LMA is relieved from the data forwarding role, only the Binding Cache and its management operations are maintained. Hence the LMA is renamed into CMD, which is therefore a Home-CPA. Also, the CMD is able to send and parse both PBU and PBA messages.
- o The MAG is enriched with the LMA functionalities, hence the name Mobility Anchor and Access Router (MAAR). It maintains a local Binding Cache for the MNs that are attached to it and it is able to send and parse PBU and PBA messages.
- o The binding cache will be extended to include information regarding P-MAARs where the mobile node was anchored and still retains active data sessions.
- o Each MAAR has a unique set of global prefixes (which are configurable), that can be allocated by the MAAR to the MNs, but must be exclusive to that MAAR, i.e. no other MAAR can allocate the same prefixes.

The MAARs leverage the CMD to access and update information related to the MNs, stored as mobility sessions; hence, a centralized node maintains a global view of the network status. The CMD is queried whenever a MN is detected to join/leave the mobility domain. It might be a fresh attachment, a detachment or a handover, but as MAARs are not aware of past information related to a mobility session, they contact the CMD to retrieve the data of interest and eventually take the appropriate action. The procedure adopted for the query and the message exchange sequence might vary to optimize the update latency and/or the signaling overhead. Here is presented one method for the initial registration, and three different approaches for updating the mobility sessions using PBUs and PBAs. Each approach assigns a different role to the CMD:

- o The CMD is a PBU/PBA relay;
- o The CMD is only a MAAR locator;
- o The CMD is a PBU/PBA proxy.

The solution described in this document allows performing per-prefix anchoring decisions, to support e.g., some flows to be anchored at a central Home-DPA (like a traditional LMA) or to enable an application to switch to the locally anchored prefix to gain route optimization, as indicated in [RFC8563]. This type of per-prefix treatment would potentially require additional extensions to the MAARs and signaling between the MAARs and the MNs to convey the per-flow anchor preference (central, distributed), which are not covered in this document.

Note that a MN may move across different MAARs, which might result in several P-MAARs existing at a given moment of time, each of them anchoring a different prefix used by the MN.

### 3.1. Initial registration

Initial registration is performed when an MN attaches to a network for the first time (rather than attaching to a new network after moving from a previous one).

In this description (shown in Figure 1), it is assumed that:

1. The MN is attaching to MAAR1.
2. The MN is authorized to attach to the network.

Upon MN attachment, the following operations take place:

1. MAAR1 assigns a global IPv6 prefix from its own prefix pool to the MN (Pref1). It also stores this prefix (Pref1) in the locally allocated temporary Binding Cache Entry (BCE).
2. MAAR1 sends a PBU [RFC5213] with Pref1 and the MN's MN-ID to the CMD.
3. Since this is an initial registration, the CMD stores a BCE containing as primary fields the MN-ID, Pref1 and MAAR1's address as a Proxy-CoA.
4. The CMD replies with a PBA with the usual options defined in PMIPv6 [RFC5213], meaning that the MN's registration is fresh and no past status is available.
5. MAAR1 stores the BCE described in (1) and unicasts a Router Advertisement (RA) to the MN with Pref1.
6. The MN uses Pref1 to configure an IPv6 address (IP1) (e.g., with stateless auto-configuration, SLAAC).



Note that:

- 1. Alternative IPv6 auto-configuration mechanisms can also be used, though this document describes the SLAAC-based one.
- 2. IP1 is routable at MAAR1, in the sense that it is on the path of packets addressed to the MN.
- 3. MAAR1 acts as a plain router for packets destined to the MN, as no encapsulation nor special handling takes place.

In the diagram shown in Figure 1 (and subsequent diagrams), the flow of packets is presented using '\*'.

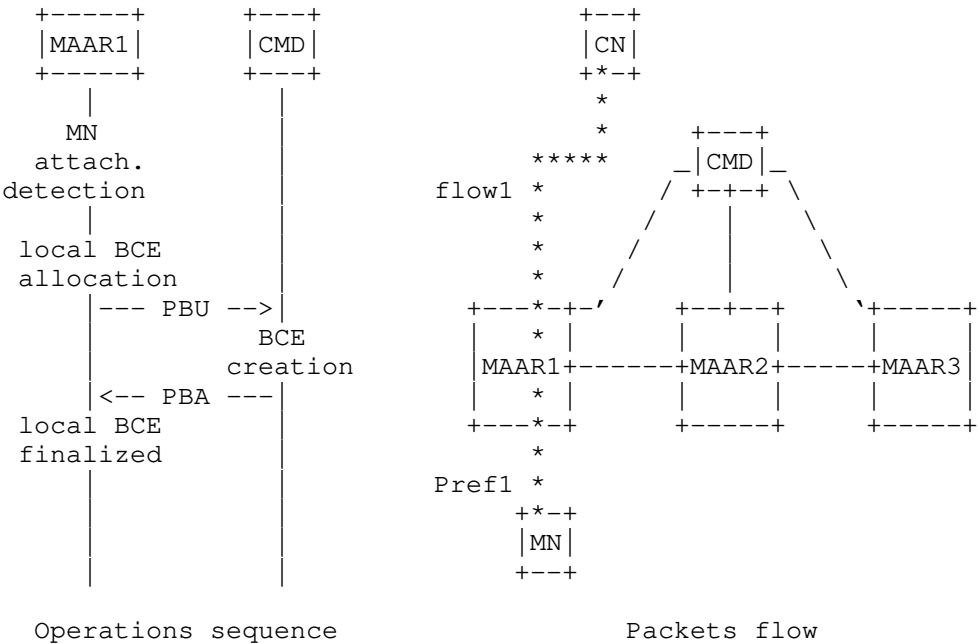


Figure 1: First attachment to the network

Note that the registration process does not change regardless of the CMD's modes (relay, locator or proxy) described next. The procedure is depicted in Figure 1.

3.2. The CMD as PBU/PBA relay

Upon MN mobility, if the CMD behaves as PBU/PBA relay, the following operations take place:

1. When the MN moves from its current point of attachment and attaches to MAAR2 (now the S-MAAR), MAAR2 reserves an IPv6 prefix (Pref2), it stores a temporary BCE, and it sends a PBU to the CMD for registration.
2. Upon PBU reception and BC lookup, the CMD retrieves an already existing entry for the MN, binding the MN-ID to its former location; thus, the CMD forwards the PBU to the MAAR indicated as Proxy CoA (MAAR1), including a new mobility option to communicate the S-MAAR's global address to MAAR1, defined as Serving MAAR Option in Section 4.6. The CMD updates the P-CoA field in the BCE related to the MN with the S-MAAR's address.
3. Upon PBU reception, MAAR1 can install a tunnel on its side towards MAAR2 and the related routes for Pref1. Then MAAR1 replies to the CMD with a PBA (including the option mentioned before) to ensure that the new location has successfully changed, containing the prefix anchored at MAAR1 in the Home Network Prefix option.
4. The CMD, after receiving the PBA, updates the BCE populating an instance of the P-MAAR list. The P-MAAR list is an additional field on the BCE that contains an element for each P-MAAR involved in the MN's mobility session. The list element contains the P-MAAR's global address and the prefix it has delegated. Also, the CMD sends a PBA to the new S-MAAR, containing the previous Proxy-CoA and the prefix anchored to it embedded into a new mobility option called Previous MAAR Option (defined in Section 4.5), so that, upon PBA arrival, a bi-directional tunnel can be established between the two MAARs and new routes are set appropriately to recover the IP flow(s) carrying Pref1.
5. Now packets destined to Pref1 are first received by MAAR1, encapsulated into the tunnel and forwarded to MAAR2, which finally delivers them to their destination. In uplink, when the MN transmits packets using Pref1 as source address, they are sent to MAAR2, as it is MN's new default gateway, then tunneled to MAAR1 which routes them towards the next hop to destination. Conversely, packets carrying Pref2 are routed by MAAR2 without any special packet handling both for uplink and downlink.

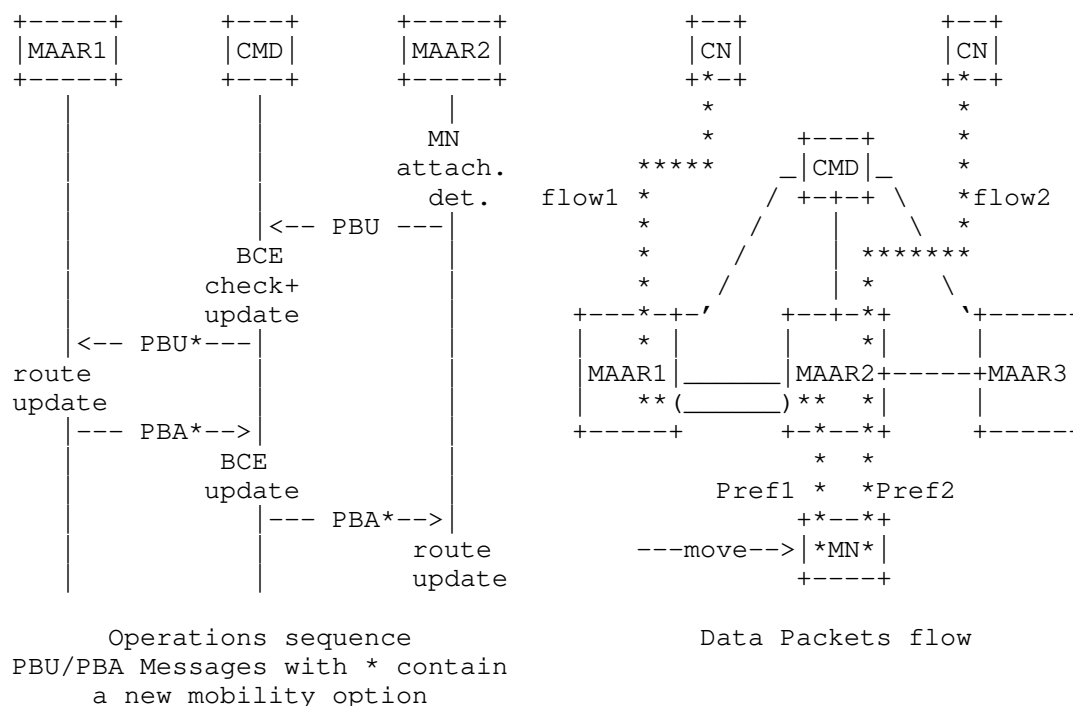


Figure 2: Scenario after a handover, CMD as relay

For MN's next movements the process is repeated except the number of P-MAARs involved increases (accordingly to the number of prefixes that the MN wishes to maintain). Indeed, once the CMD receives the first PBU from the new S-MAAR, it forwards copies of the PBU to all the P-MAARs indicated in the BCE, namely the one registered as current P-CoA (i.e., the MAAR prior to handover) plus the ones in the P-MAARs list. They reply with a PBA to the CMD, which aggregates them into a single one to notify the S-MAAR, that finally can establish the tunnels with the P-MAARs.

It should be noted that this design separates the mobility management at the prefix granularity, and it can be tuned in order to erase old mobility sessions when not required, while the MN is reachable through the latest prefix acquired. Moreover, the latency associated to the mobility update is bound to the PBA sent by the furthest P-MAAR, in terms of RTT, that takes the longest time to reach the CMD. The drawback can be mitigated introducing a timeout at the CMD, by which, after its expiration, all the PBAs so far collected are transmitted, and the remaining are sent later upon their arrival. Note that in this case the S-MAAR might receive multiple PBAs from the CMD in response to a PBU. The CMD SHOULD follow the

retransmissions and rate limiting considerations described in Section 3.6, especially when aggregating and relaying PBAs.

When there are multiple previous MAARs, e.g.,  $k$  MAARs, a single PBU received by the CMD triggers  $k$  outgoing packets from a single incoming packet. This may lead to packet bursts originated from the CMD, albeit to different targets. Pacing mechanisms **MUST** be introduced to avoid bursts on the outgoing link.

### 3.3. The CMD as MAAR locator

The handover latency experienced in the approach shown before can be reduced if the P-MAARs are allowed to signal directly their information to the new S-MAAR. This procedure reflects what was described in Section 3.2 up to the moment the P-MAAR receives the PBU with the S-MAAR option. At that point a P-MAAR is aware of the new MN's location (because of the S-MAAR's address in the S-MAAR option), and, besides sending a PBA to the CMD, it also sends a PBA to the S-MAAR including the prefix it is anchoring. This latter PBA does not need to include new options, as the prefix is embedded in the HNP option and the P-MAAR's address is taken from the message's source address. The CMD is relieved from forwarding the PBA to the S-MAAR, as the latter receives a copy directly from the P-MAAR with the necessary information to build the tunnels and set the appropriate routes. Figure 3 illustrates the new message sequence, while the data forwarding is unaltered.

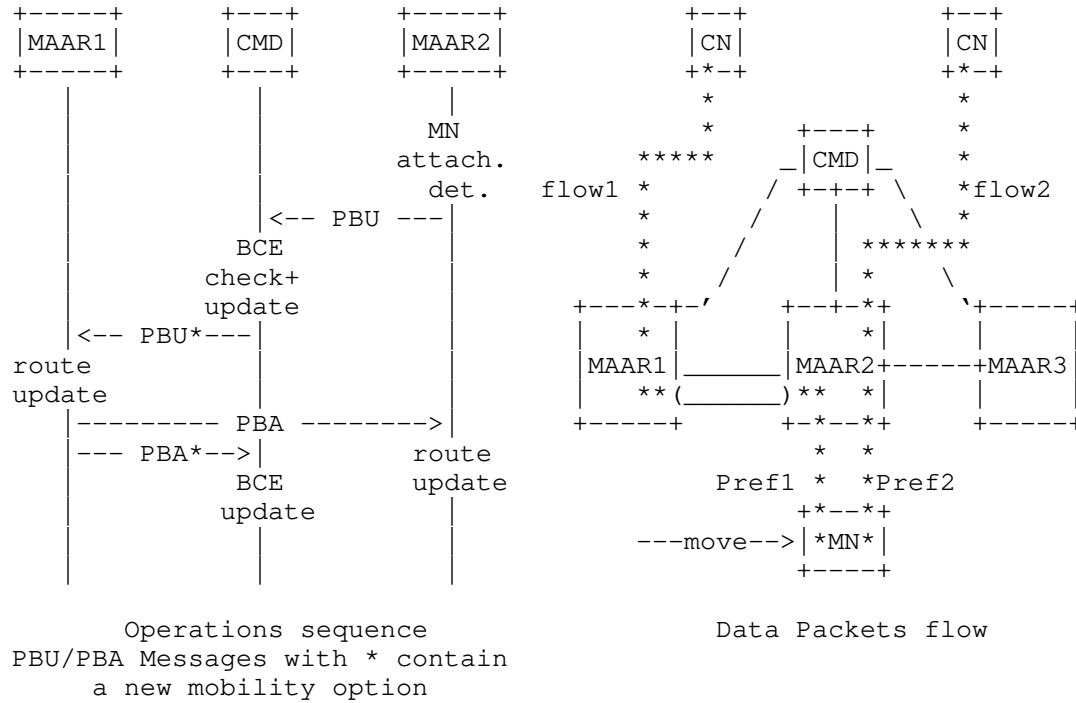


Figure 3: Scenario after a handover, CMD as locator

### 3.4. The CMD as MAAR proxy

A further enhancement of previous solutions can be achieved when the CMD sends the PBA to the new S-MAAR before notifying the P-MAARs of the location change. Indeed, when the CMD receives the PBU for the new registration, it is already in possession of all the information that the new S-MAAR requires to set up the tunnels and the routes. Thus the PBA is sent to the S-MAAR immediately after a PBU is received, including also in this case the P-MAAR option. In parallel, a PBU is sent by the CMD to the P-MAARs containing the S-MAAR option, to notify them about the new MN's location, so they receive the information to establish the tunnels and routes on their side. When P-MAARs complete the update, they send a PBA to the CMD to indicate that the operation is concluded and the information is updated in all network nodes. This procedure is obtained from the first one re-arranging the order of the messages, but the parameters communicated are the same. This scheme is depicted in Figure 4, where, again, the data forwarding is kept untouched.

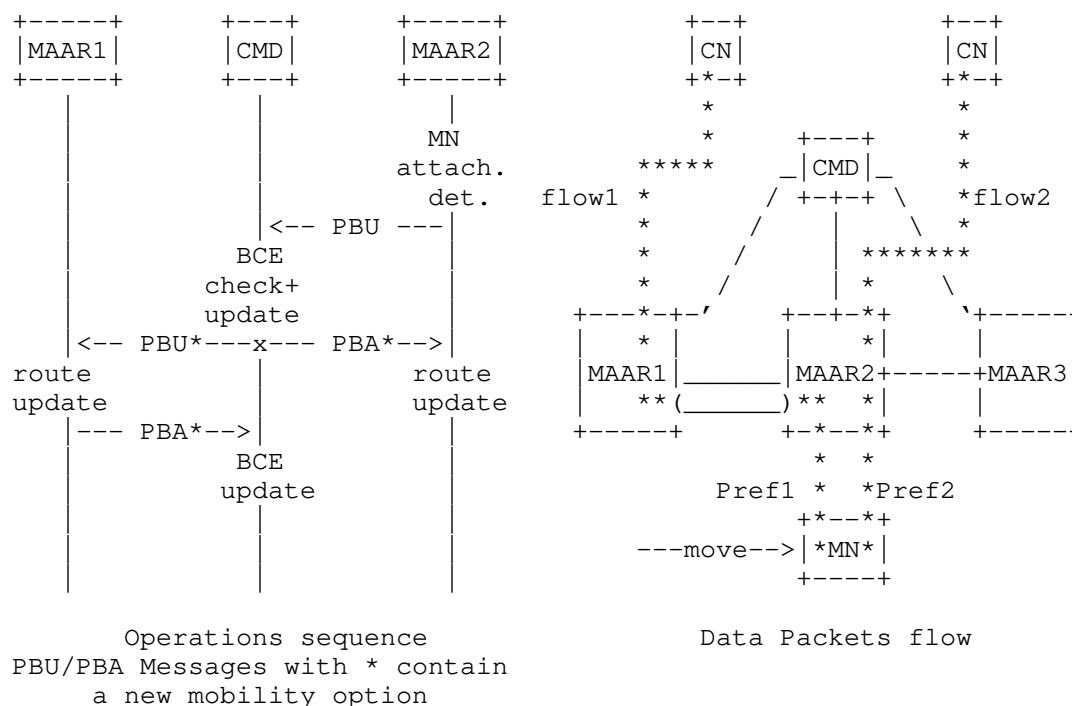


Figure 4: Scenario after a handover, CMD as proxy

### 3.5. De-registration

The de-registration mechanism devised for PMIPv6 cannot be used as-is in this solution. The reason for this is that each MAAR handles an independent mobility session (i.e., a single or a set of prefixes) for a given MN, whereas the aggregated session is stored at the CMD. Indeed, if a previous MAAR initiates a de-registration procedure, because the MN is no longer present on the MAAR's access link, it removes the routing state for that (those) prefix(es), that would be deleted by the CMD as well, hence defeating any prefix continuity attempt. The simplest approach to overcome this limitation is to deny a P-MAAR to de-register a prefix, that is, allowing only a serving MAAR to de-register the whole MN session. This can be achieved by first removing any layer-2 detachment event, so that de-registration is triggered only when the binding lifetime expires, hence providing a guard interval for the MN to connect to a new MAAR. Then, a change in the MAAR operations is required, and at this stage two possible solutions can be deployed:

- o A previous MAAR stops the BCE timer upon receiving a PBU from the CMD containing a "Serving MAAR" option. In this way only the

Serving MAAR is allowed to de-register the mobility session, arguing that the MN definitely left the domain.

- o Previous MAARs can, upon BCE expiry, send de-registration messages to the CMD, which, instead of acknowledging the message with a 0 lifetime, sends back a PBA with a non-zero lifetime, hence re-newing the session, if the MN is still connected to the domain.

### 3.6. Retransmissions and Rate Limiting

When sending PBUs, the node sending them (the CMD or S-MAAR) SHOULD make use of the timeout also to deal with missing PBAs (to retransmit PBUs). The INITIAL\_BINDACK\_TIMEOUT [RFC6275] SHOULD be used for configuring the retransmission timer. The retransmissions by the node MUST use an exponential backoff process in which the timeout period is doubled upon each retransmission, until either the node receives a response or the timeout period reaches the value MAX\_BINDACK\_TIMEOUT [RFC6275]. The node MAY continue to send these messages at this slower rate indefinitely. The node MUST NOT send PBU messages to a particular node more than MAX\_UPDATE\_RATE times within a second [RFC6275].

### 3.7. The Distributed Logical Interface (DLIF) concept

One of the main challenges of a network-based DMM solution is how to allow a mobile node to simultaneously send/receive traffic which is anchored at different MAARs, and how to influence the mobile node's selection process of its source IPv6 address for a new flow, without requiring special support from the mobile node's IP stack. This document defines the Distributed Logical Interface (DLIF), which is a software construct in the MAAR that allows to easily hide the change of associated anchors from the mobile node.

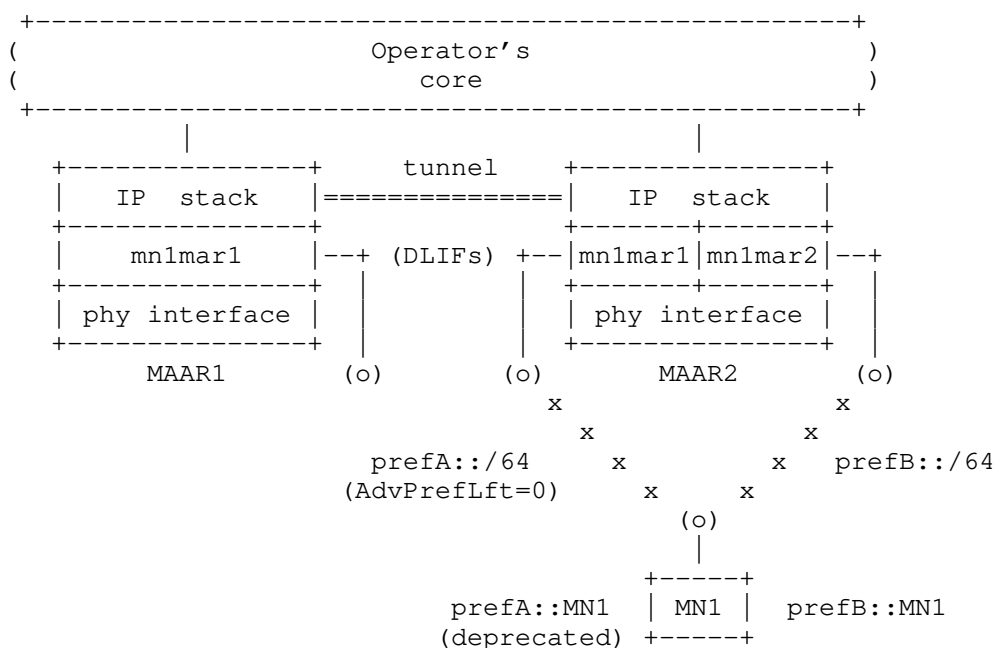


Figure 5: DLIF: exposing multiple routers (one per P-MAAR)

The basic idea of the DLIF concept is the following: each serving MAAR exposes itself towards a given MN as multiple routers, one per P-MAAR associated to the MN. Let's consider the example shown in Figure 5, MN1 initially attaches to MAAR1, configuring an IPv6 address (prefA::MN1) from a prefix locally anchored at MAAR1 (prefA::/64). At this stage, MAAR1 plays both the role of anchoring and serving MAAR, and also behaves as a plain IPv6 access router. MAAR1 creates a distributed logical interface to communicate (point-to-point link) with MN1, exposing itself as a (logical) router with a specific MAC and IPv6 addresses (e.g., prefA::MAAR1/64 and fe80::MAAR1/64) using the DLIF mn1mar1. As explained below, these addresses represent the "logical" identity of MAAR1 towards MN1, and will "follow" the mobile node while roaming within the domain (note that the place where all this information is maintained and updated is out-of-scope of this draft; potential examples are to keep it on the home subscriber server -- HSS -- or the user's profile).

If MN1 moves and attaches to a different MAAR of the domain (MAAR2 in the example of Figure 5), this MAAR will create a new logical interface (mn1mar2) to expose itself towards MN1, providing it with a locally anchored prefix (prefB::/64). In this case, since the MN1 has another active IPv6 address anchored at a MAAR1, MAAR2 also needs to create an additional logical interface configured to resemble the



one used by MAAR1 to communicate with MN1. In this example, there is only one P-MAAR (in addition to MAAR2, which is the serving one): MAAR1, so only the logical interface mn1mar1 is created, but the same process would be repeated in case there were more P-MAARs involved. In order to maintain the prefix anchored at MAAR1 reachable, a tunnel between MAAR1 and MAAR2 is established and the routing is modified accordingly. The PBU/PBA signaling is used to set-up the bi-directional tunnel between MAAR1 and MAAR2, and it might also be used to convey to MAAR2 the information about the prefix(es) anchored at MAAR1 and about the addresses of the associated DLIF (i.e., mn1mar1).

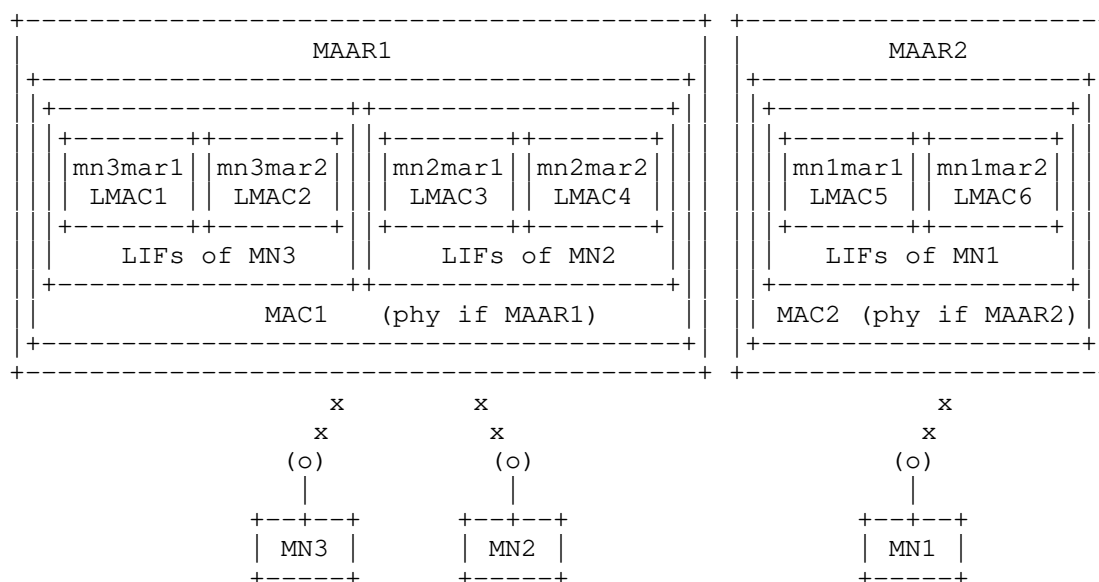


Figure 6: Distributed Logical Interface concept

Figure 6 shows the logical interface concept in more detail. The figure shows two MAARs and three MNs. MAAR1 is currently serving MN2 and MN3, while MAAR2 is serving MN1. Note that a serving MAAR always plays the role of anchoring MAAR for the attached (served) MNs. Each MAAR has one single physical wireless interface as depicted in this example.

As introduced before, each MN always "sees" multiple logical routers -- one per anchoring MAAR -- independently of its currently serving MAAR. From the point of view of the MN, these MAARs are portrayed as different routers, although the MN is physically attached to one single interface. The way this is achieved is by the serving MAAR configuring different logical interfaces. Focusing on MN1, it is currently attached to MAAR2 (i.e., MAAR2 is its serving MAAR) and,

therefore, it has configured an IPv6 address from MAAR2's pool (e.g., prefB::/64). MAAR2 has set-up a logical interface (mnlmar2) on top of its wireless physical interface (phy if MAAR2) which is used to serve MN1. This interface has a logical MAC address (LMAC6), different from the hardware MAC address (MAC2) of the physical interface of MAAR2. Over the mnlmar2 interface, MAAR2 advertises its locally anchored prefix prefB::/64. Before attaching to MAAR2, MN1 was attached to MAAR1, configuring also an address locally anchored at that MAAR, which is still being used by MN1 in active communications. MN1 keeps "seeing" an interface connecting to MAAR1, as if it were directly connected to the two MAARs. This is achieved by the serving MAAR (MAAR2) configuring an additional distributed logical interface: mnlmar1, which behaves as the logical interface configured by MAAR1 when MN1 was attached to it. This means that both the MAC and IPv6 addresses configured on this logical interface remain the same regardless of the physical MAAR which is serving the MN. The information required by a serving MAAR to properly configure this logical interfaces can be obtained in different ways: as part of the information conveyed in the PBA, from an external database (e.g., the HSS) or by other means. As shown in the figure, each MAAR may have several logical interfaces associated to each attached MN, having always at least one (since a serving MAAR is also an anchoring MAAR for the attached MN).

In order to enforce the use of the prefix locally anchored at the serving MAAR, the router advertisements sent over those logical interfaces playing the role of anchoring MAARs (different from the serving one) include a zero preferred prefix lifetime (and a non-zero valid prefix lifetime, so the prefix remains valid, while being deprecated). The goal is to deprecate the prefixes delegated by these MAARs (so that they will no longer be serving the MN). Note that on-going communications may keep on using those addresses, even if they are deprecated, so this only affects the establishment of new sessions.

The distributed logical interface concept also enables the following use case: suppose that access to a local IP network is provided by a given MAAR (e.g., MAAR1 in the example shown in Figure 5) and that the resources available at that network cannot be reached from outside the local network (e.g., cannot be accessed by an MN attached to MAAR2). This is similar to the local IP access scenario considered by 3GPP, where a local gateway node is selected for sessions requiring access to services provided locally (instead of going through a central gateway). The goal is to allow an MN to be able to roam while still being able to have connectivity to this local IP network. The solution adopted to support this case makes use of RFC 4191 [RFC4191] more specific routes when the MN moves to a MAAR different from the one providing access to the local IP network

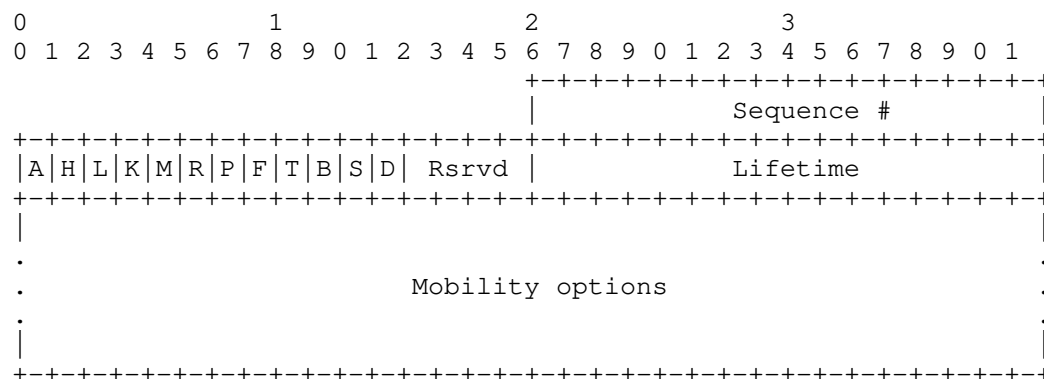
(MAAR1 in the example). These routes are advertised through the distributed logical interface representing the MAAR providing access to the local network (MAAR1 in this example). In this way, if MN1 moves from MAAR1 to MAAR2, any active session that MN1 may have with a node on the local network connected to MAAR1 will survive via the tunnel between MAAR1 and MAAR2. Also, any potential future connection attempt towards the local network will be supported, even though MN1 is no longer attached to MAAR1.

#### 4. Message Format

This section defines extensions to the Proxy Mobile IPv6 [RFC5213] protocol messages.

##### 4.1. Proxy Binding Update

A new flag (D) is included in the Proxy Binding Update to indicate that the Proxy Binding Update is coming from a MAAR or a CMD and not from a mobile access gateway. The rest of the Proxy Binding Update format remains the same as defined in [RFC5213].



##### DMM Flag (D)

The D Flag is set to indicate to the receiver of the message that the Proxy Binding Update is from a MAAR or a CMD. When an LMA that does not support the extensions described in this document receives a message with the D-Flag set, the PBU in that case MUST NOT be processed by the LMA and an error MUST be returned.

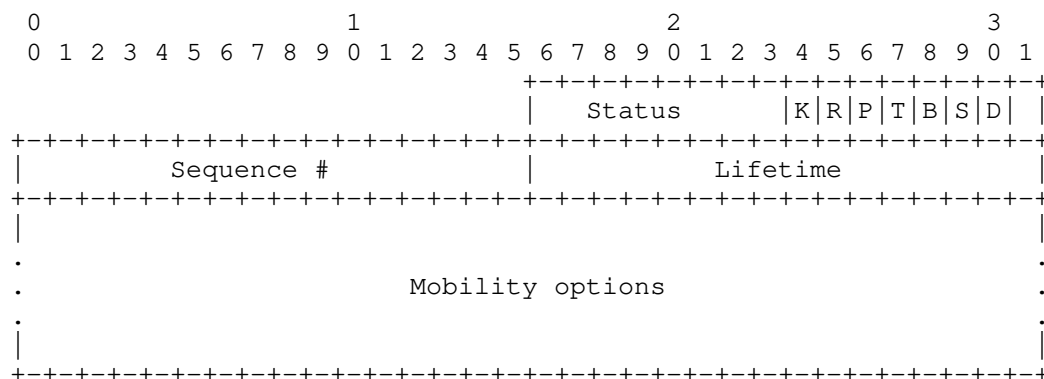
##### Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding

and format of defined options are described in Section 6.2 of [RFC6275]. The receiving node MUST ignore and skip any options that it does not understand.

#### 4.2. Proxy Binding Acknowledgment

A new flag (D) is included in the Proxy Binding Acknowledgment to indicate that the sender supports operating as a MAAR or CMD. The rest of the Proxy Binding Acknowledgment format remains the same as defined in [RFC5213].



##### DMM Flag (D)

The D flag is set to indicate that the sender of the message supports operating as a MAAR or a CMD. When a MAG that does not support the extensions described in this document receives a message with the D-Flag set, it MUST ignore the message and an error MUST be returned.

##### Mobility Options

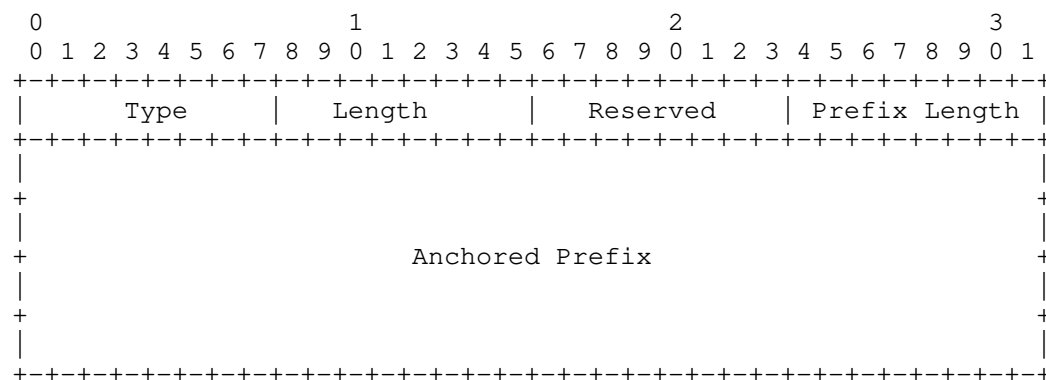
Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2 of [RFC6275]. The MAAR MUST ignore and skip any options that it does not understand.

#### 4.3. Anchored Prefix Option

A new Anchored Prefix option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs and CMDs. Therefore, this option can only appear if the D bit is set in a PBU/PBA. This option is used for exchanging

the mobile node's prefix anchored at the anchoring MAAR. There can be multiple Anchored Prefix options present in the message.

The Anchored Prefix Option has an alignment requirement of  $8n+4$ . Its format is as follows:



#### Type

IANA-1.

#### Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 18.

#### Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

#### Prefix Length

8-bit unsigned integer indicating the prefix length in bits of the IPv6 prefix contained in the option.

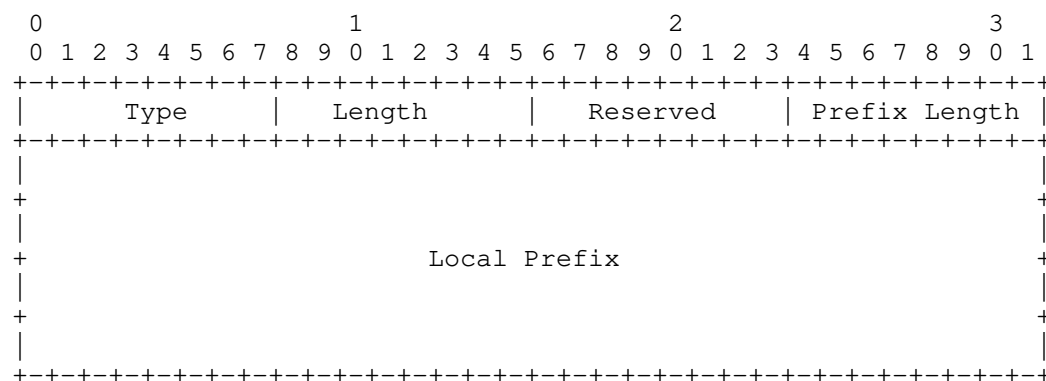
#### Anchored Prefix

A sixteen-octet field containing the mobile node's IPv6 Anchored Prefix. Only the first Prefix Length bits are valid for the Anchored Prefix. The rest of the bits MUST be ignored.

#### 4.4. Local Prefix Option

A new Local Prefix option is defined for use with the Proxy Binding Update and Proxy Binding Acknowledgment messages exchanged between MAARs or between a MAAR and a CMD. Therefore, this option can only appear if the D bit is set in a PBU/PBA. This option is used for exchanging a prefix of a local network that is only reachable via the anchoring MAAR. There can be multiple Local Prefix options present in the message.

The Local Prefix Option has an alignment requirement of  $8n+4$ . Its format is as follows:



##### Type

IANA-2.

##### Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 18.

##### Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

##### Prefix Length

8-bit unsigned integer indicating the prefix length in bits of the IPv6 prefix contained in the option.

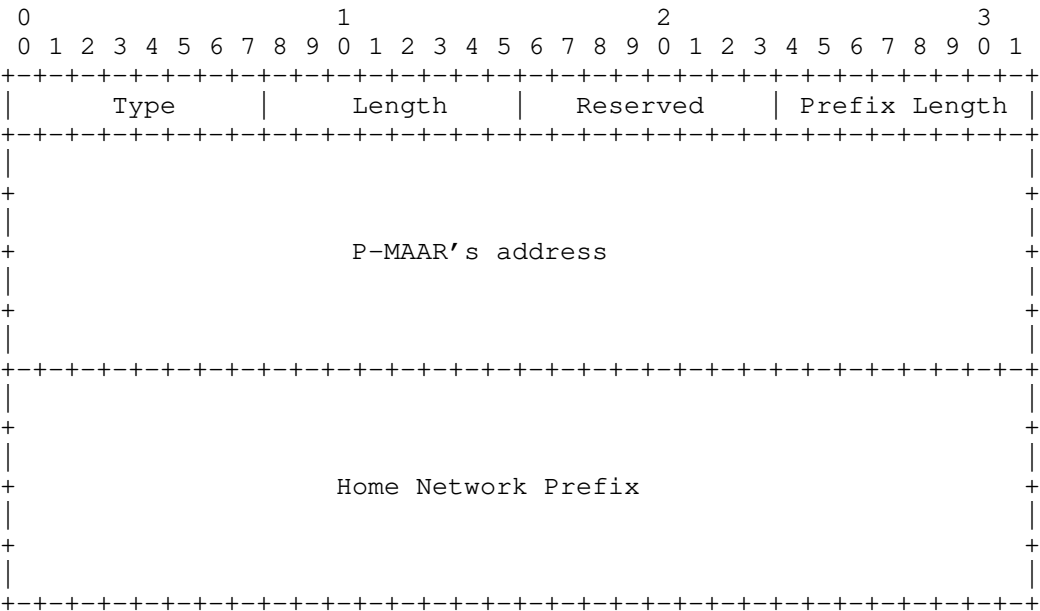
##### Local Prefix

A sixteen-octet field containing the IPv6 Local Prefix. Only the first Prefix Length bits are valid for the IPv6 Local Prefix. The rest of the bits MUST be ignored.

4.5. Previous MAAR Option

This new option is defined for use with the Proxy Binding Acknowledgement messages exchanged by the CMD to a MAAR. This option is used to notify the S-MAAR about the previous MAAR's global address and the prefix anchored to it. There can be multiple Previous MAAR options present in the message. Its format is as follows:

The Previous MAAR Option has an alignment requirement of 8n+4. Its format is as follows:



Type

IANA-3.

Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 34.

Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

#### Prefix Length

8-bit unsigned integer indicating the prefix length in bits of the IPv6 prefix contained in the option.

#### Previous MAAR's address

A sixteen-octet field containing the P-MAAR's IPv6 global address.

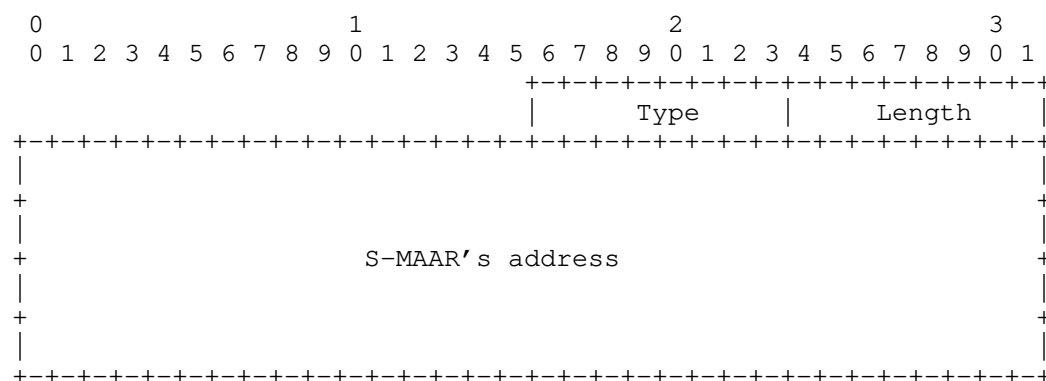
#### Home Network Prefix

A sixteen-octet field containing the mobile node's IPv6 Home Network Prefix. Only the first Prefix Length bits are valid for the mobile node's IPv6 Home Network Prefix. The rest of the bits MUST be ignored.

### 4.6. Serving MAAR Option

This new option is defined for use with the Proxy Binding Update message exchanged between the CMD and a Previous MAAR. This option is used to notify the P-MAAR about the current Serving MAAR's global address. Its format is as follows:

The Serving MAAR Option has an alignment requirement of  $8n+6$ . Its format is as follows:



#### Type

IANA-4.



## Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 16.

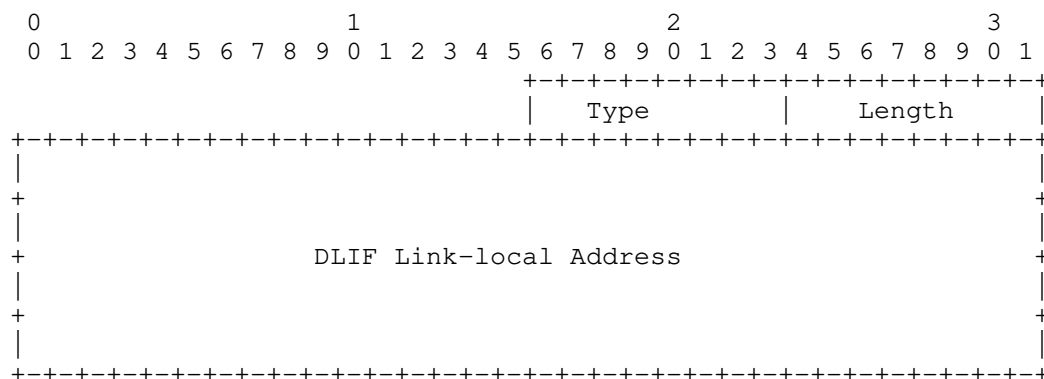
## Serving MAAR's address

A sixteen-octet field containing the S-MAAR's IPv6 global address.

## 4.7. DLIF Link-local Address Option

A new DLIF Link-local Address option is defined for use with the Proxy Binding Acknowledgment message exchanged between MAARs and between a MAAR and a CMD. This option is used for exchanging the link-local address of the DLIF to be configured on the serving MAAR so it resembles the DLIF configured on the P-MAAR.

The DLIF Link-local Address option has an alignment requirement of  $8n+6$ . Its format is as follows:



## Type

IANA-5.

## Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields. This field MUST be set to 16.

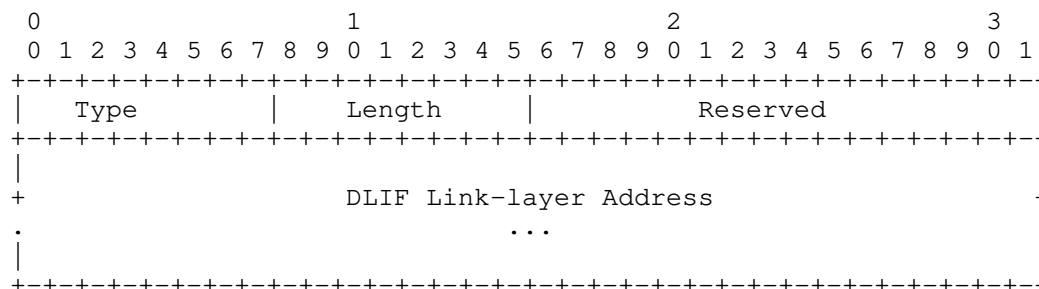
## DLIF Link-local Address

A sixteen-octet field containing the link-local address of the logical interface.

#### 4.8. DLIF Link-layer Address Option

A new DLIF Link-layer Address option is defined for use with the Proxy Binding Acknowledgment message exchanged between MAARs and between a MAAR and a CMD. This option is used for exchanging the link-layer address of the DLIF to be configured on the serving MAAR so it resembles the DLIF configured on the P-MAAR.

The format of the DLIF Link-layer Address option is shown below. Based on the size of the address, the option MUST be aligned appropriately, as per mobility option alignment requirements specified in [RFC6275].



##### Type

IANA-6.

##### Length

8-bit unsigned integer indicating the length of the option in octets, excluding the type and length fields.

##### Reserved

This field is unused for now. The value MUST be initialized to 0 by the sender and MUST be ignored by the receiver.

##### DLIF Link-layer Address

A variable length field containing the link-layer address of the logical interface to be configured on the S-MAAR.

The content and format of this field (including octet and bit ordering) is as specified in Section 4.6 of [RFC4861] for carrying

link-layer addresses. On certain access links, where the link-layer address is not used or cannot be determined, this option cannot be used.

## 5. IANA Considerations

This document defines six new mobility options, the Anchored Prefix Option, the Local Prefix Option, the Previous MAAR Option, the Serving MAAR Option, the DLIF Link-local Address Option and the DLIF Link-layer Address Option. The Type value for these options needs to be assigned from the same numbering space as allocated for the other mobility options in the "Mobility Options" registry defined in <http://www.iana.org/assignments/mobility-parameters>. The required IANA actions are marked as IANA-1 to IANA-6.

This document reserves a new flag (D) in the "Binding Update Flags" and a new flag (D) in the "Binding Acknowledgment Flags" of the "Mobile IPv6 parameters" registry <http://www.iana.org/assignments/mobility-parameters>.

## 6. Security Considerations

The protocol extensions defined in this document share the same security concerns of Proxy Mobile IPv6 [RFC5213]. It is recommended that the signaling messages, Proxy Binding Update and Proxy Binding Acknowledgment, exchanged between the MAARs are protected using IPsec using the established security association between them. This essentially eliminates the threats related to the impersonation of a MAAR.

When the CMD acts as a PBU/PBA relay, the CMD may act as a relay of a single PBU to multiple previous MAARs. In situations of many fast handovers (e.g., with vehicular networks), there may exist multiple previous (e.g., k) MAARs. In this situation, the CMD creates k outgoing packets from a single incoming packet. This bears a certain amplification risk. The CMD MUST use a pacing approach in the outgoing queue to cap the output traffic (i.e., the rate of PBUs sent) to limit this amplification risk.

When the CMD acts as MAAR locator, mobility signaling (PBAs) is exchanged between P-MAARs and current S-MAAR. Hence, security associations are REQUIRED to exist between the involved MAARs (in addition to the ones needed with the CMD).

Since deregistration is performed by timeout, measures SHOULD be implemented to minimize the risks associated to continued resource consumption (DoS attacks), e.g., imposing a limit of the number of P-MAARs associated to a given MN.

The CMD and the participating MAARs MUST be trusted parties, authorized perform all operations relevant to their role.

There are some privacy considerations to consider. While the involved parties trust each other, the signalling involves disclosing information about the previous locations visited by each MN, as well as the active prefixes they are using at a given point of time. Therefore, mechanisms MUST be in place to ensure that MAARs and CMD do not disclose this information to other parties nor use it for other ends than providing the distributed mobility support specified in this document.

## 7. Acknowledgments

The authors would like to thank Dirk von Hugo, John Kaippallimalil, Ines Robles, Joerg Ott, Carlos Pignataro, Vincent Roca, Mirja Kuehlewind, Eric Vyncke, Adam Roach, Benjamin Kaduk and Roman Danyliw for the comments on this document. The authors would also like to thank Marco Liebsch, Dirk von Hugo, Alex Petrescu, Daniel Corujo, Akbar Rahman, Danny Moses, Xinpeng Wei and Satoru Matsushima for their comments and discussion on the documents [I-D.bernardos-dmm-distributed-anchoring] and [I-D.bernardos-dmm-pmip] on which the present document is based.

The authors would also like to thank Lyle Bertz and Danny Moses for their in-deep review of this document and their very valuable comments and suggestions.

## 8. References

### 8.1. 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>>.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<https://www.rfc-editor.org/info/rfc4191>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.

- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", RFC 7333, DOI 10.17487/RFC7333, August 2014, <<https://www.rfc-editor.org/info/rfc7333>>.
- [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>>.

## 8.2. Informative References

- [I-D.bernardos-dmm-distributed-anchoring]  
Bernardos, C. and J. Zuniga, "PMIPv6-based distributed anchoring", draft-bernardos-dmm-distributed-anchoring-09 (work in progress), May 2017.
- [I-D.bernardos-dmm-pmip]  
Bernardos, C., Oliva, A., and F. Giust, "A PMIPv6-based solution for Distributed Mobility Management", draft-bernardos-dmm-pmip-09 (work in progress), September 2017.
- [RFC7429] Liu, D., Ed., Zuniga, JC., Ed., Seite, P., Chan, H., and CJ. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis", RFC 7429, DOI 10.17487/RFC7429, January 2015, <<https://www.rfc-editor.org/info/rfc7429>>.
- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

## Authors' Addresses

Carlos J. Bernardos  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Phone: +34 91624 6236  
Email: [cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)  
URI: <http://www.it.uc3m.es/cjbc/>

Antonio de la Oliva  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Phone: +34 91624 8803  
Email: [aoliva@it.uc3m.es](mailto:aoliva@it.uc3m.es)  
URI: <http://www.it.uc3m.es/aoliva/>

Fabio Giust  
Athonet S.r.l.

Email: [fabio.giust.2011@ieee.org](mailto:fabio.giust.2011@ieee.org)

Juan Carlos Zuniga  
SIGFOX  
425 rue Jean Rostand  
Labège 31670  
France

Email: [j.c.zuniga@ieee.org](mailto:j.c.zuniga@ieee.org)  
URI: <http://www.sigfox.com/>

Alain Mourad  
InterDigital Europe

Email: [Alain.Mourad@InterDigital.com](mailto:Alain.Mourad@InterDigital.com)  
URI: <http://www.InterDigital.com/>

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

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

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 10 November 2022.

Copyright Notice

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

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

## Table of Contents

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



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

## 1. Introduction

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

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

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

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

## 2. Conventions and Terminology

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

### 2.1. Terminology

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

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

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

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

## 2.2. Conventions

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

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

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

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

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

## 2.3. Predefined SRv6 Endpoint Behaviors

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

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

\* End.T: Endpoint with specific IPv6 Table Lookup

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

### 3. Motivation

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

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

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

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

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

### 4. 3GPP Reference Architecture

This section presents a reference architecture and possible deployment scenarios.

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

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

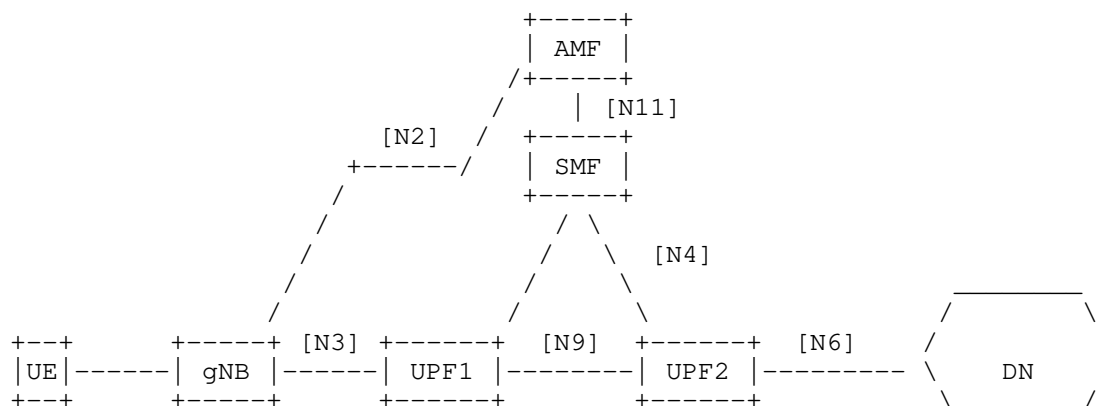


Figure 1: 3GPP 5G Reference Architecture

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

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

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

## 5. User-plane modes

This section introduces an SRv6 based mobile user-plane.

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

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

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

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

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

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

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

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

### 5.1. Traditional mode

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

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

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

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

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

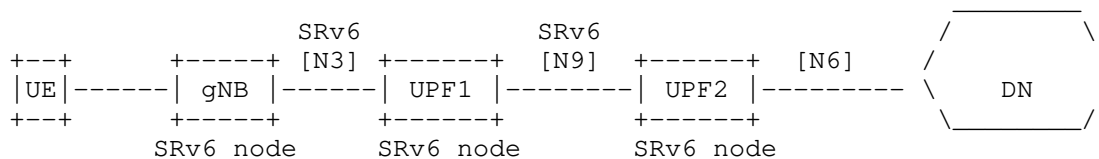


Figure 2: Traditional mode - example topology

#### 5.1.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

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

```

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

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

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

#### 5.1.2. Packet flow - Downlink

The downlink packet flow is as follows:

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

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

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

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

#### 5.2. Enhanced mode

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

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

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

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 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.

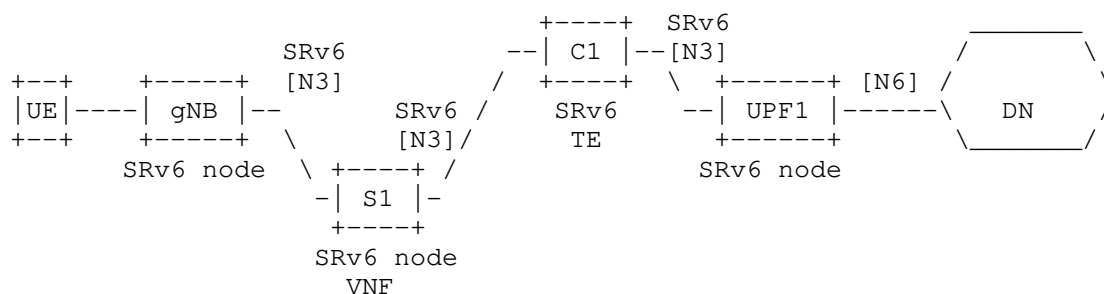


Figure 3: Enhanced mode - Example topology

#### 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.DT2 behaviors.

### 5.2.3. Scalability

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

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

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

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

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

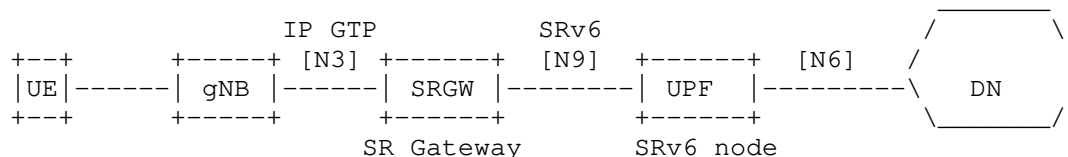


Figure 4: Example topology for interworking

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

#### 5.3.1. Interworking with IPv6 GTP

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

Key points:

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

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

An example topology is shown in Figure 5.

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

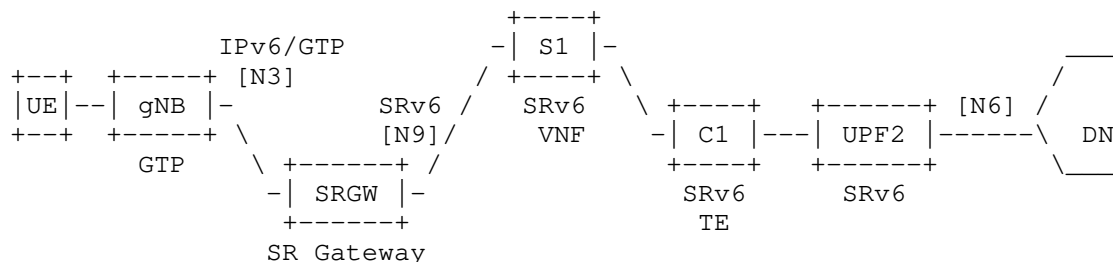


Figure 5: Enhanced mode with unchanged gNB IPv6/GTP behavior

#### 5.3.1.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

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

```

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

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

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

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

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

#### 5.3.1.2. Packet flow - Downlink

The downlink packet flow is as follows:

```

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

```

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

C1 and S1 perform their related Endpoint processing.

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

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

#### 5.3.1.3. Scalability

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

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

#### 5.3.2. Interworking with IPv4 GTP

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

Key points:

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

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

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

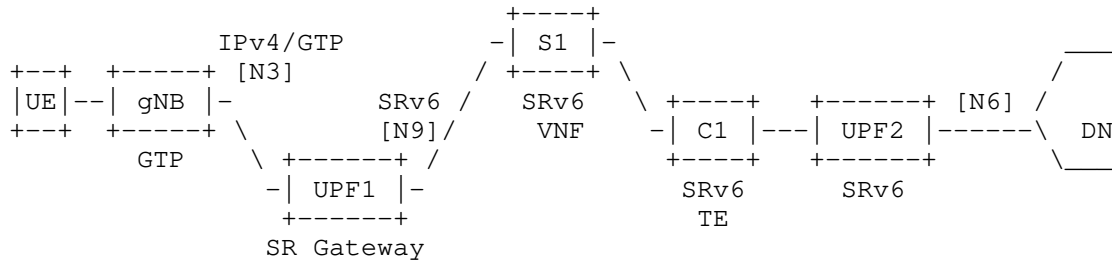


Figure 6: Enhanced mode with unchanged gNB IPv4/GTP behavior

## 5.3.2.1. Packet flow - Uplink

The uplink packet flow is as follows:

```

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

```

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

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

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

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

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

#### 5.3.2.2. Packet flow - Downlink

The downlink packet flow is as follows:

```

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

```

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

The nodes C1 and S1 perform their related Endpoint processing.

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

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

#### 5.3.2.3. Scalability

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

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

### 5.3.3. Extensions to the interworking mechanisms

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

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

### 5.4. SRv6 Drop-in Interworking

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

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

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

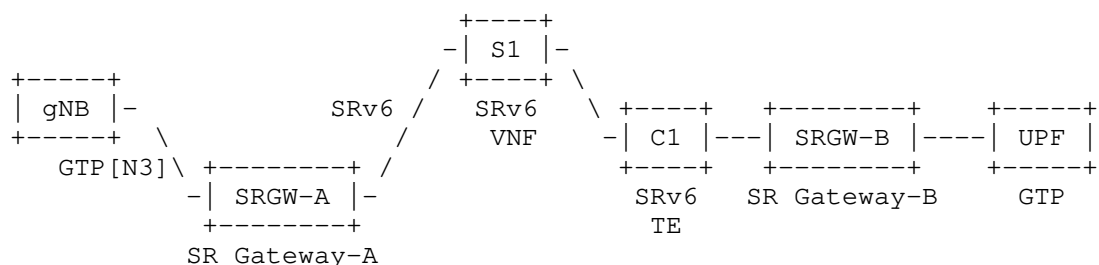


Figure 7: Example topology for SRv6 Drop-in mode

The packet flow of Figure 7 is as follows:



```

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

UPF_out  : (A,Z)

```

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

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

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

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

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

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

## 6. SRv6 Segment Endpoint Mobility Behaviors

## 6.1. Args.Mob.Session

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

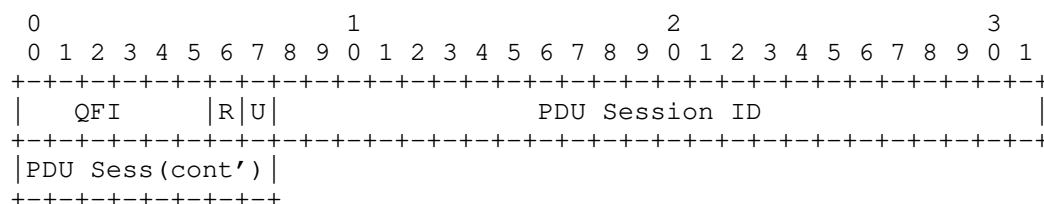


Figure 8: Args.Mob.Session format

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

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

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

## 6.2. End.MAP

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

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

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

Notes: The SIDs in the SRH are not modified.

### 6.3. End.M.GTP6.D

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

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

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

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

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

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

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

#### 6.4. End.M.GTP6.D.Di

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

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

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

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

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

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

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

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

#### 6.5. End.M.GTP6.E

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

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

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

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

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

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

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

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

## 6.6. End.M.GTP4.E

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

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

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

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

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

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

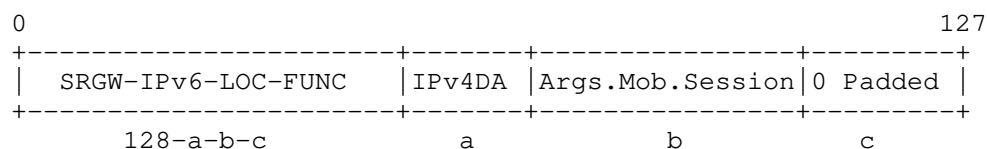


Figure 9: End.M.GTP4.E SID Encoding

The IPv6 Source Address has the following format:

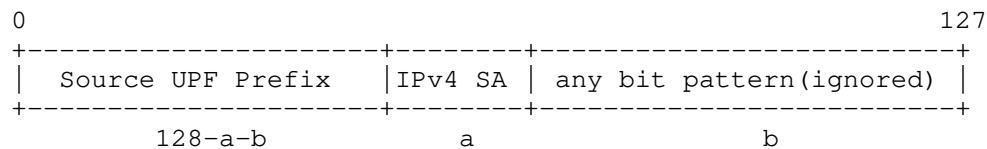


Figure 10: IPv6 SA Encoding for End.M.GTP4.E

#### 6.7. H.M.GTP4.D

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

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

```

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

```

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

The SID B has the following format:

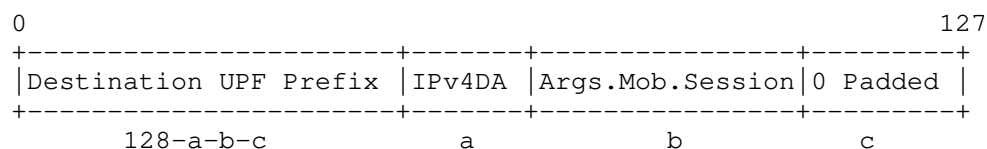


Figure 11: H.M.GTP4.D SID Encoding

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

#### 6.8. End.Limit: Rate Limiting behavior

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

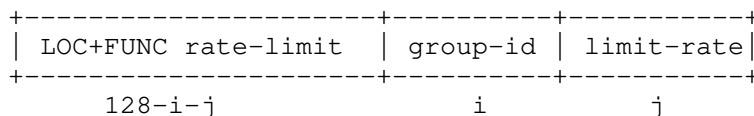


Figure 12: End.Limit: Rate limiting behavior argument format

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



## 7. SRv6 supported 3GPP PDU session types

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

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

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

## 8. Network Slicing Considerations

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

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

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

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

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

## 9. Control Plane Considerations

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

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

## 10. Security Considerations

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

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

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

## 11. IANA Considerations

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

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

Table 1: SRv6 Mobile User-plane Endpoint Behavior Types

## 12. Acknowledgements

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

## 13. Contributors

Kentaro Ebisawa Toyota Motor Corporation Japan

Email: [ebisawa@toyota-tokyo.tech](mailto:ebisawa@toyota-tokyo.tech)

Tetsuya Murakami Arrcus, Inc. United States of America

Email: [tetsuya.ietf@gmail.com](mailto:tetsuya.ietf@gmail.com)

## 14. References

### 14.1. Normative References

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

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

#### 14.2. Informative References

- [I-D.ali-spring-network-slicing-building-blocks]  
Ali, Z., Filsfils, C., Camarillo, P., and D. Voyer, "Building blocks for Slicing in Segment Routing Network", Work in Progress, Internet-Draft, draft-ali-spring-network-slicing-building-blocks-04, 21 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ali-spring-network-slicing-building-blocks-04>>.
- [I-D.camarilloelmalky-springdmm-srv6-mob-usecases]  
Garvia, P. C., Filsfils, C., Elmalky, H., Matsushima, S., Voyer, D., Cui, A., and B. Peirens, "SRv6 Mobility Use-Cases", Work in Progress, Internet-Draft, draft-camarilloelmalky-springdmm-srv6-mob-usecases-02, 15 August 2019, <<https://datatracker.ietf.org/doc/html/draft-camarilloelmalky-springdmm-srv6-mob-usecases-02>>.
- [I-D.gundavelli-dmm-mfa]  
Gundavelli, S., Liebsch, M., and S. Matsushima, "Mobility-aware Floating Anchor (MFA)", Work in Progress, Internet-Draft, draft-gundavelli-dmm-mfa-01, 19 September 2018, <<https://datatracker.ietf.org/doc/html/draft-gundavelli-dmm-mfa-01>>.
- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-19, 7 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lsr-flex-algo-19>>.
- [I-D.ietf-spring-segment-routing-central-epe]  
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. 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-05, 10 September 2021,  
<<https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-05>>.
- [I-D.kohno-dmm-srv6mob-arch]  
Kohno, M., Clad, F., Camarillo, P., and Z. Ali,  
"Architecture Discussion on SRv6 Mobile User plane", Work  
in Progress, Internet-Draft, draft-kohno-dmm-srv6mob-arch-  
05, 8 November 2021,  
<<https://datatracker.ietf.org/doc/html/draft-kohno-dmm-srv6mob-arch-05>>.
- [I-D.matsushima-spring-srv6-deployment-status]  
Matsushima, S., Filsfils, C., Ali, Z., Li, Z., Rajaraman,  
K., and A. Dhamija, "SRv6 Implementation and Deployment  
Status", Work in Progress, Internet-Draft, draft-  
matsushima-spring-srv6-deployment-status-15, 5 April 2022,  
<<https://datatracker.ietf.org/doc/html/draft-matsushima-spring-srv6-deployment-status-15>>.
- [I-D.mhkk-dmm-srv6mup-architecture]  
Matsushima, S., Horiba, K., Khan, A., Kawakami, Y.,  
Murakami, T., Patel, K., Kohno, M., Kamata, T., Garvia, P.  
C., Voyer, D., Zadok, S., Meilik, I., Agrawal, A.,  
Perumal, K., and J. Horn, "Segment Routing IPv6 Mobile  
User Plane Architecture for Distributed Mobility  
Management", Work in Progress, Internet-Draft, draft-mhkk-  
dmm-srv6mup-architecture-03, 20 March 2022,  
<<https://datatracker.ietf.org/doc/html/draft-mhkk-dmm-srv6mup-architecture-03>>.
- [I-D.murakami-dmm-user-plane-message-encoding]  
Murakami, T., Matsushima, S., Ebisawa, K., Camarillo, P.,  
and R. Shekhar, "User Plane Message Encoding", Work in  
Progress, Internet-Draft, draft-murakami-dmm-user-plane-  
message-encoding-05, 5 March 2022,  
<<https://datatracker.ietf.org/doc/html/draft-murakami-dmm-user-plane-message-encoding-05>>.
- [TS.29281] 3GPP, "General Packet Radio System (GPRS) Tunnelling  
Protocol User Plane (GTPv1-U)", 3GPP TS 29.281 15.1.0,  
December 2017.

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

## Appendix A. Implementations

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

Additionally, a full implementation of this document is available in Linux Foundation FD.io VPP project since release 20.05. More information available here: [https://docs.fd.io/vpp/20.05/d7/d3c/srv6\\_mobile\\_plugin\\_doc.html](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