

Distributed Mobility Management (DMM)
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

U. Fattore
M. Liebsch
NEC
March 11, 2019

Control-/Data Plane Aspects for N6 Traffic Steering
draft-fattore-dmm-n6-cpdp-trafficsteering-01.txt

Abstract

Current standardization effort on the evolution of the mobile communication system reconsiders the mobile data plane protocol. The IETF DMM Working Group has work that proposes and analyzes various protocols as alternative to the GPRS Tunneling Protocol for User Plane (GTP-U) for an overlay deployment in between the mobile device's assigned data plane anchor and its current radio base station, which are denoted as N9 and N3 interfaces. In the view of some future deployment and the original intent per the very early DMM WG charter, a mobile device's data plane anchor may be highly distributed and re-selected for optimization throughout a mobile device's communication with one or more correspondent services. Such re-configuration has impact on the packet routing in between the mobile device's data plane anchor and the one or multiple data networks hosting the services, which is denoted as N6 interface. This draft proposes and discusses a solution to control, setup and maintain traffic treatment policy on the cellular communication system's N6 interface while taking the UE's PDU session settings per the cellular system's control plane, such as QoS and locator information, into account.

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 12, 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. Terminology	2
2. Introduction	2
3. Positioning of N6 policy control	4
3.1. System architecture for mobile access to data networks	4
3.2. Use cases with demand for N6 traffic treatment policy	7
4. N6 traffic treatment - Requirements and policy types	8
5. Leveraging the mobile control plane for N6 policy control	9
6. N6 endpoints - loose and tight coupling options	11
7. Operations for N6 policy enforcement in a tight coupling scenario	13
7.1. AF/NC-initiated N6 policy enforcement	14
7.2. 3GPP-initiated N6 policy enforcement	16
8. IANA Considerations	20
9. Security Considerations	20
10. Acknowledgments	20
11. Normative References	20
Authors' Addresses	21

1. 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 RFC 2119 [RFC2119].

2. Introduction

Recent releases and deployments of cellular mobile communication systems utilize an overlay on the mobile data plane to forward a mobile device's data packets in between the mobile device and an anchor point, which serves as first hop router to the mobile device. The overlay is realized by the GPRS Tunneling Protocol for user plane

(GTP-U), which is able to carry network-specific attributes in the tunnel protocol headers.

The 3rd Generation Partnership Project (3GPP) is in charge of the cellular mobile communication system's specification and is currently finalizing a 15th release, which has fundamental changes compared to previous releases. Such changes include a clean split between control- and data plane functions, more flexible deployment and re-configuration of data plane anchors, as well as support for local data network (DN) access and multi-homing.

In between a mobile device's current radio base station in the radio access network (RAN) and its data plane anchor, the release 15 specification assumes an overlay per the previous releases utilizing GTP-U. The data plane anchor is denoted as User Plane Function (UPF) to anchor a Packet Data Unit (PDU) Session for the mobile device. This draft abbreviates the UPF, which serves a device's PDU session anchor, as UPF_a. In between a UPF_a and the device's current radio base station, none, one or multiple additional UPFs can be deployed to classify uplink traffic in support of policy-based routing to a particular DN without traversing the UPF_a. This draft denotes such intermediate UPF as UPF_i. Interfaces between a DN and a mobile device's UPF_a is denoted as N6, the interface between a UPF_i and one or multiple UPF_a is denoted as N9, and the interface between a UPF_i and a radio base station is denoted as N3. Whereas regular routing of mobile devices' PDUs is assumed on N6, N9 and N3 deploy a GTP-U overlay with UPF_a, UPF_i and the radio base station serving as tunnel endpoints. This end-to-end architecture is depicted in Figure 1. For a more detailed description of anchor and intermediate UPF and associated deployment and operation, please refer to [I-D.bogineni-dmm-optimized-mobile-user-plane] and the 3GPP specification [TS23.501].

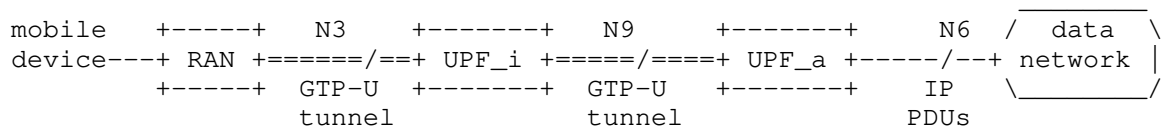


Figure 1: Architecture and interfaces of a 3GPP release 15 data plane in between a data network and a mobile device.

In alignment with the 3GPP's current directions to study data plane protocol candidates which can serve as suitable alternative to GTP-U, the IETF's DMM WG has valuable ongoing individual work that analyzes the GTP-U protocol and derives requirements for an alternative mobile data plane protocol [I-D.hmm-dmm-5g-uplane-analysis], as well as work

that investigates the use of alternative protocol candidates based on SRv6, ID-Locator separation, and locator re-writing in the current release 15 system architecture [I-D.bogineni-dmm-optimized-mobile-user-plane]. The focus of these drafts is on N9 and N3.

In the view of optimization options on the complete end-to-end data plane, [I-D.gundavelli-dmm-mfa] complements other draft and proposes data plane optimization on N6. Such operation is of particular interest when the mobile device's UPF_a is decentralized and deployed close to the device's current radio base station. Such deployment may be preferable for some services, such as edge computing and access to associated edge DNSs, and mitigates the role of the UPF_i and N9 interfaces. In particular the selection and configuration of UPF_i instances can be omitted and associated signaling costs can be saved. However, such deployment strengthens the expectation on IP-based PDU routing on N6, as the serving DN may not be always topologically close to the device and its current UPF_a. Such requirements include QoS support on N6, metering support and traffic steering in case the mobile device's UPF_a changes while its IP address and associated sessions should continue.

The same requirements on N6 apply for multi-homing per [TS23.501] where the mobile device's UPF_a is close to a first DN (DN1) whereas a UPF_i is used to enable access to a second DN (DN2), either through a secondary UPF_a close to DN2 or directly from the UPF_i, without the use of a secondary UPF_a. Since services in both DN address the same IP address of the mobile device (IP_ue) to send downlink traffic, both DN's traffic need to be forwarded to the most suitable (e.g. closest) UPF_a or UPF_i respectively.

This draft focuses on a solution to control, setup and maintain such dedicated routes and additional traffic treatment policy on N6, while taking the UE's PDU session settings per the cellular system's control plane, such as QoS and locator information, into account.

3. Positioning of N6 policy control

This section briefly introduces the relevant mobile system architecture components and interfaces, and covers some high-level use cases which can benefit from data plane policy control on N6 interface endpoints.

3.1. System architecture for mobile access to data networks

The 3GPP's 5G system architecture introduces in the core network a clear control-/user plane separation (CUPS), in order to have flexible deployment of the different functions (e.g., user plane

nodes can scale independently from control plane elements in case of user traffic growth). Again to leverage flexibility and efficiency, the control plane is split in different functions, each offering a specific service, in the so called Service Based Architecture (SBA).

Among all the control plane functions, the Session Management function (SMF) takes care of the session management (session establishment, modification, release), IP allocation and selection of an IP anchor point for the session, as well as traffic steering in between UPFs and radio base stations. In order to manage the user session, the SMF collaborates with other control plane services (e.g., Policy Control Function - PCF - providing policy rules for traffic treatment and monitoring), in particular with the Access and Mobility Management Function (AMF), which manages registration, authentication and authorization and security context. One of the main task of the SMF is to instruct User Plane Functions (UPFs), through N4 interface. When a new session is to be created, the SMF selects one or multiple UPFs for the user traffic and selects one UPF as session anchor (UPF_a). UPF_a acts as a proxy for user traffic, which means all traffic directed to the UE passes through the UPF anchor. Beside the UPF_a, if other UPFs are present (i.e., between the radio base station and the UPF_a), this are deployed as classifiers for user uplink traffic.

In Figure 2 a simplified 5G architecture [TS23.501] is depicted, showing two Data Networks (DN) to whom a user may need a connection. To each Data Network a UPF_a is associated, acting as session anchor and providing to the user an IP address needed for the connection. UPF_a also acts as tunnel termination point, since user traffic is encapsulated on both N3 and N9 interfaces, using the GPRS Tunneling Protocol for User Plane (GTP-U). Whereas, on N6 interface IP PDUs are routed without tunneling.

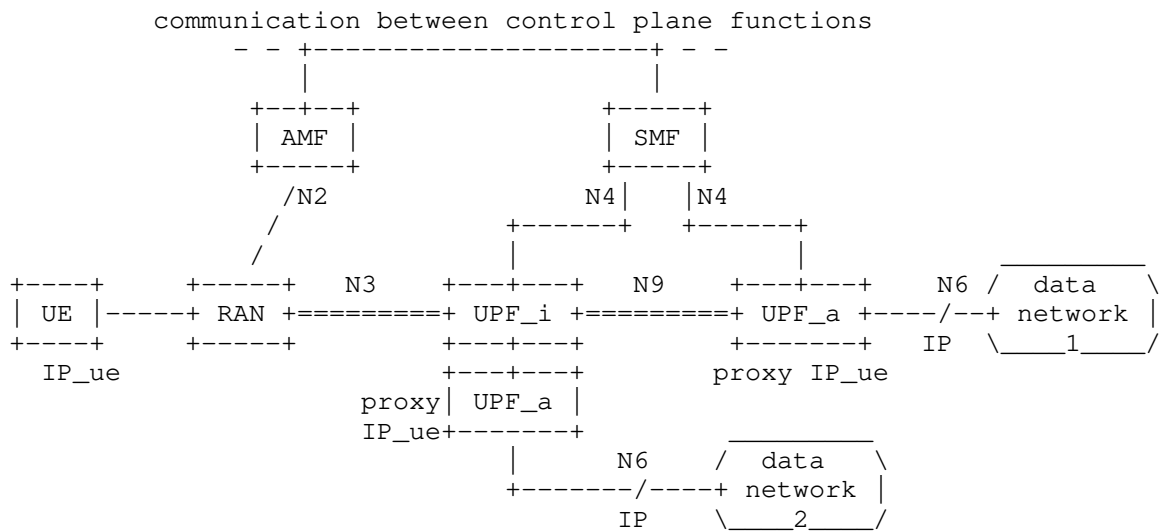


Figure 2: Data plane with a simplified release 15 control plane

Data networks host Application Servers (AS), which provide a services to UEs, and an internal network comprising data plane nodes (DPN), such as routers and switches, to connect the services with the transport network. Both, the transport network and the data network's internal network build the N6 interface, which is depicted in Figure 3. In order to apply traffic treatment policy to uplink traffic in between a UPF and a data network, the UPF receives policies via the N4 interface. For downlink traffic, the AS/DPN should have means to receive traffic treatment policies.

A way to enforce N6 policies to the DPN/AS in a data network is needed. It is evident that this rule must originate from the cellular control plane due to its knowledge about the UE's states, such as its locator or QoS, and when these states are updated or re-configured. Different means to convey and enforce associated traffic treatment policies in a DPN/AS exist, such as the use of routing protocols or control-/data plane configuration protocols.

this can be avoid, giving a traffic steering policy to the DPN in charge for the edge DN.

Concurrent use of multiple data networks: a possible scenario is the one in which a UE collects the desired content from different data networks (e.g., because of Content Delivery Networks - CDN). To optimize routing in this scenario, the downlink traffic should traverse for each data network the optimized path through the UE and not be forced through a (central) UPF_a common to all the data networks. Again, this can be done with policies on N6 interface. This particular use case also highlights the importance to consider optimization on N6, whereas other works focus on N9: considering a UPF_a near the data network, as proposed in other solutions, would not allow multiple DN access in an unique user session and so would not allow for content access on different destinations.

4. N6 traffic treatment - Requirements and policy types

Use cases for traffic treatment on N6 per a data plane policy include cases where the UPF_a is deployed closer at the mobile edge, e.g. to not only access a local data network in the proximity of the UE, but also other data networks sharing the single edge UPF_a. In that case the N6 interface may span some distance in the transport network in between the data network(s) and the UPF_a. Dependent on the expected QoE/QoS of the traffic, traffic treatment policies for QoS differentiation, packet labeling, etc. may apply to the UE's packets on N6. For uplink traffic, the UE's UPF_a can enforce such traffic treatment policies to uplink traffic, where a DPN associated with the data network(s) (e.g. PE router, transit router, router/switch of the data center transport network, TOR switches of Application Servers, etc.) enforces such policies to downlink traffic.

The same need for traffic treatment policies applies to traffic between a UPF_i, which classifies uplink traffic for forwarding to a local data network, and the data network. Downlink traffic from the local data network to the UE should then be forwarded towards the UPF_i, not via the UE's UPF_a.

In advanced scenarios, the SMF may decide to reconfigure the UE's UPFs, e.g. by relocating the UPF_a or a UPF_i while maintaining the UE's IP address (IP_ue) and data sessions using this IP address. In such case, a DPN associated with the one or multiple data networks, which run correspondent services for the UE, must enforce traffic steering policies to downlink traffic to achieve routing of downlink traffic to the UE's current UPF_a or UPF_i respectively.

In summary, traffic treatment policies that apply to a UE's uplink and downlink traffic on N6 include the following types:

- o QoS differentiation and traffic engineering"
- o Packet label push/pop"
- o Metering
- o Traffic steering (e.g. SRv6 rules, locator re-write rules, etc.)
- o E dormancy monitoring rules to initiate paging

Requirements for N6 traffic treatment include the following:

- o Awareness of UE location information (first hop router accuracy, UPF_a/UPF_i) - Set or update DPN policy for traffic steering
- o Awareness of topology - Select and update most suitable UPF (UPF_a/UPF_i) for the communication with a data network, e.g. after UPF changed
- o Availability of initial or updated policies when needed
- o No/Low impact on data traffic (packet loss, re-ordering) when policies are updated - DPNs may request/solicit policies or get notified about initial and updated policies

5. Leveraging the mobile control plane for N6 policy control

Methods for N6 policy control consist in instructing the DPNs with rules for traffic steering, QoS policies enforcing, etc. The solution described in this draft is based on leveraging the mobile control plane, in order to introduce some logic to manage and forward policies to DPNs on N6 interface. To do this, the Application Function (AF) defined in 5GS [TS23.501] is used as binding element in between the cellular network control plane and the data network data plane.

Per [TS23.501], the AF is introduced to inter-work with the Policy Control Function (PCF) in order to condition and contribute to some SMF decisions. This happens with the AF sending specific requests to the PCF and the latter translating those requests in policies for the SMF. Depending on the domain in which the AF is located, a Network Exposure Function (NEF) may be in between to enable the AF collaborating with the other control plane elements of the cellular architecture.

In support of the proposed scenario, the AF can solicit data plane policies from the cellular control plane by sending a request. At reception of the policies, the AF can pass the policies on for

further processing and enforcement in the data network's AS/DPN. In this way, DPNs receive from the control plane policies for the user traffic traversing them. The AF may be co-located with a control function, which utilizes the DMM WG's Forwarding Policy Configuration (FPC) protocol to implement policies in the AS/DPN, or leverage an SDN controller for the selection and configuration of AS/DPN.

The policies defined and forwarded by the AF are based on the status of the mobile network, which the AF can obtain from the SMF. In any moment, in fact, the SMF is in charge of keeping track of the selected UPFs and of monitoring the user session. Based on this information, the AF forwards specific rules to a DPN (e.g., traffic steering rules to make the user's traffic reach the most suitable UPF_a). In some cases (e.g., user mobility), the SMF can also change UPFs for a specific user and in this case the AF will receive updated policies for enforcement in the involved AS/DPN.

Figure 4 shows how the previous architecture evolves with the introduction of the AF.

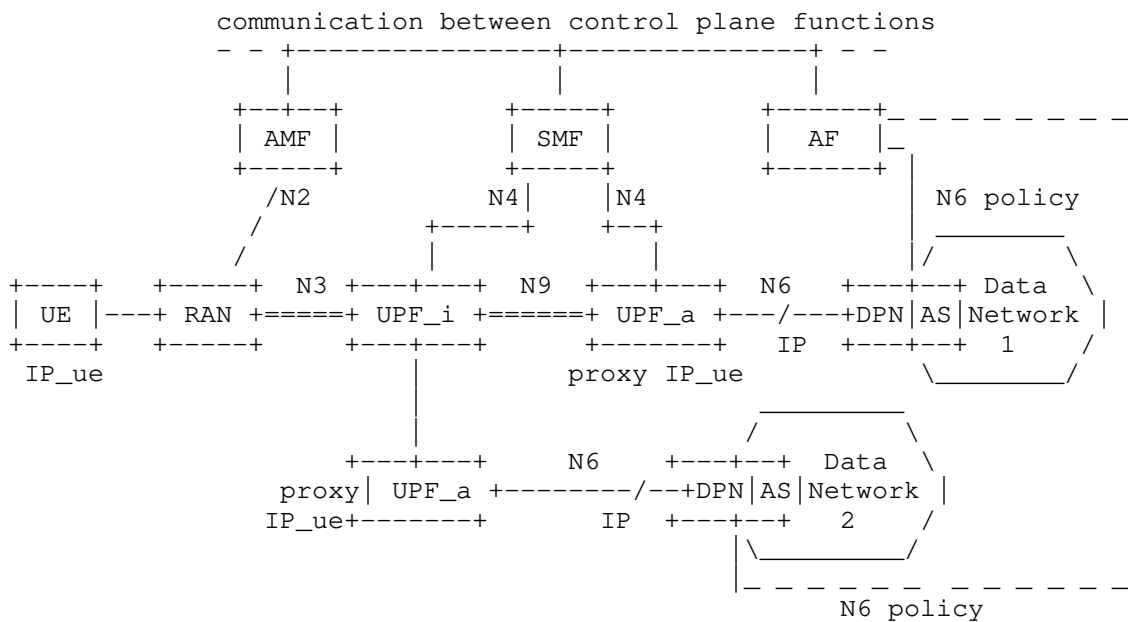


Figure 4: Using AF in control plane for traffic policy enforcement

6. N6 endpoints - loose and tight coupling options

As described in the previous section, we take advantage of the Application Function (AF) to bind the 3GPP's domain functions with those introduced in this draft for N6 policy enforcement. According to [TS23.501], an Application Function may send requests to influence SMF decisions for User Plane (UP) traffic of PDU Sessions (e.g., based on the relocation of an application on the Data Network side, the AF can notify this to the SMF in order to trigger a relocation of UPF(s) from the SMF, to choose a new UPF more suitable for the new Data Network).

In addition, the AF can subscribe to events from the SMF in order to receive notification about UP management events (e.g., when a PDU Session anchor has been established or released).

As defined in [TS23.502], the AF interacts with the PCF/SMF via the NEF or directly and the PCF then forwards requests from the AF towards the SMF as Session Management (SM) Policies. For the sake of simplicity, in this section all the 3GPP's functions apart from the AF are collected under the name of "3GPP's C-PLANE", and the specific service to which the AF interacts in the 3GPP C-PLANE is not relevant for this draft.

In order to forward specific policies to the Data Plane Nodes/ Application Servers (DPNs/ASs) associated with each Data Network, a Network Controller (NC) is considered to be co-located with the AF element. The NC performs the selection of a DPN/AS element based on the received information from the C-PLANE. The AF/NC forwards control messages to a DPN/AS through an AFNC-CPUP interface, giving indications to steer the downlink traffic properly and coherently with the UP updates from the 3GPP's side.

Forwarding N6 policies to the N6 endpoints involved (i.e., UPF and DPN) can happen in two different ways:

- 1) Tight coupling scenario: The UPF can enforce policies per the AF/NC decisions. The UPF receives associated policies from the 3GPP's C-PLANE. The corresponding DPN/AS receives the policy via the AFNC-CPUP interface.
- 2) Loose coupling scenario: A separate DPN function is co-located with the UPF. Main policies for N6 traffic treatment do not traverse the 3GPP's C-PLANE but are controlled at both N6 interface endpoints' DPN by the AF/NC via the AFNC-CPUP interface.

In the tight coupling scenario, the N6 interface configurations for the UPF are all enforced through the 3GPP domain. Therefore, the 3GPP's C-PLANE interacts with the AF/NC element through the AFNC_3GPP interface and receives on this interface requests to influence the UP traffic policies. 3GPP decides if enforce those policies on the UPF(s) involved.

The architecture and interfaces involved in this tight coupling scenario are depicted in Figure 5.

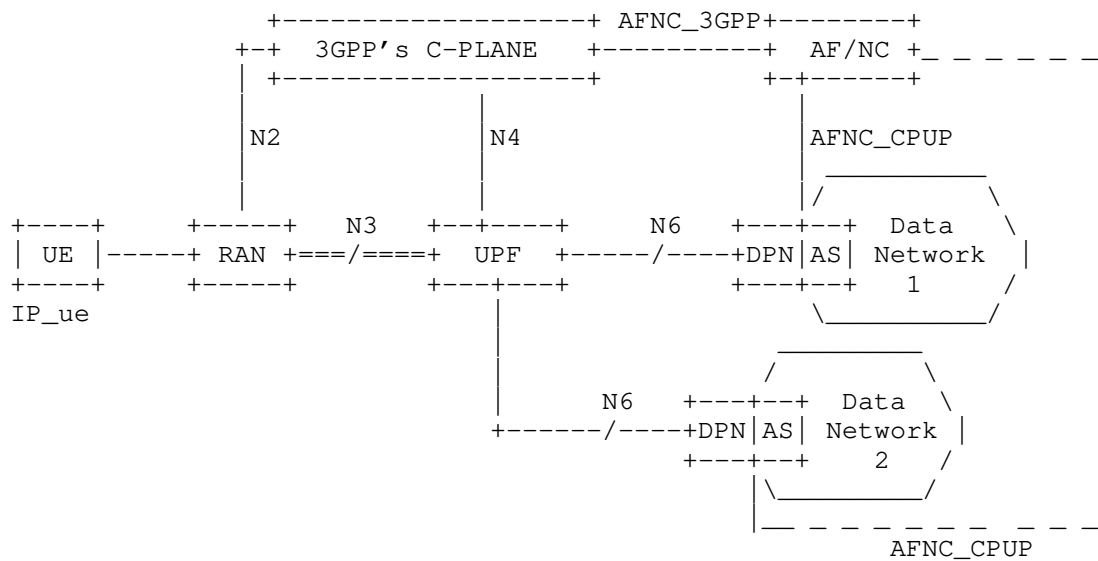


Figure 5: N6 endpoints tight coupling scenario

In Section 7.1 the operation flow and information model for the messages exchanged in this type of coupling are presented and described. Both the cases of a AF/NC-initiated and 3GPP-initiated message flow are considered.

In the loose coupling scenario, an additional DPN element is associated with a UPF and represents a key element to enforce N6 traffic treatment policies on the UPF-side of the N6 interface. This DPN is controlled by the AF/NC through the AFNC_CPUP interface, as depicted in Figure 6.

Loose coupling allows reducing 3GPP's role in the N6 endpoint management, potentially allowing under certain assumptions (e.g., no UPF re-selection is needed), an optimized control of the N6 interface from the AF/NC element, transparently from 3GPP's domain. This kind

of scenario results as an advantage particularly for use cases in which the UPF is deployed in the proximity of the Data Network and far from the 3GPP's C-PLANE (i.e., in a Mobile Edge Computing - MEC - alike scenario).

For particular cases which request 3GPP's C-PLANE involvement (i.e., UPF re-selection or other changes not related to the only N6 endpoint) the AFNC_3GPP is still used for notifications and requests between the AF/NC and the 3GPP's C-PLANE.

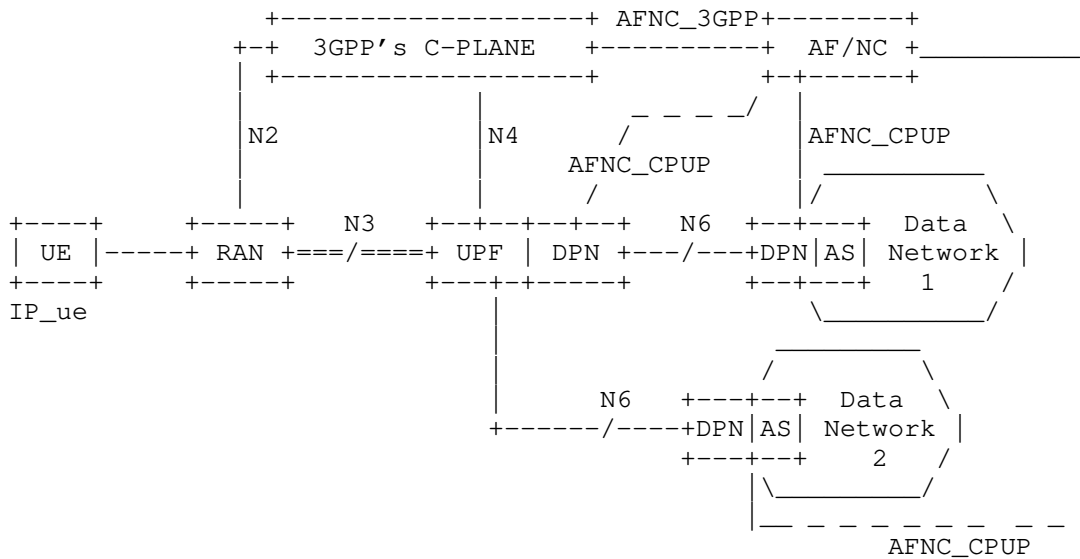


Figure 6: N6 endpoints loose coupling scenario

7. Operations for N6 policy enforcement in a tight coupling scenario

In the following sub-sections, message sequences are shown assuming a tight coupling scenario between N6 interface endpoints, as depicted in Figure 5. Two different operation flows can be distinguished, based on the entity initiating and requesting for the N6 policy. Section 7.1 describes the message sequence in the case of AF/NC-initiated N6 policy request, while Section 7.2 covers the alternative case in which the request for a N6 policy is initiated from the 3GPP's C-PLANE.

In the message sequences, special attention is given to the AFNC_CPUP and AFNC_3GPP interfaces defined in this draft and Information Models for messages exchanged on those interfaces are provided.

7.1. AF/NC-initiated N6 policy enforcement

A N6 policy can be triggered from the AF/NC element and is then forwarded directly to the DPN N6 endpoint (through AFNC_CPUP interface) and indirectly to the UPF N6 endpoint (through AFNC_3GPP interface).

As example, the AF/NC may request updated n6 policies for the following reasons:

- o there is the need of a different QoS to be applied to traffic, which is identified in the request.
- o there is the need for a re-location of the application to a different Data Network and therefore changes for traffic in uplink on the UPF's N6 endpoint should be applied.

Figure 7 depicts the AF/NC-initiated N6 policy enforcement message sequence.

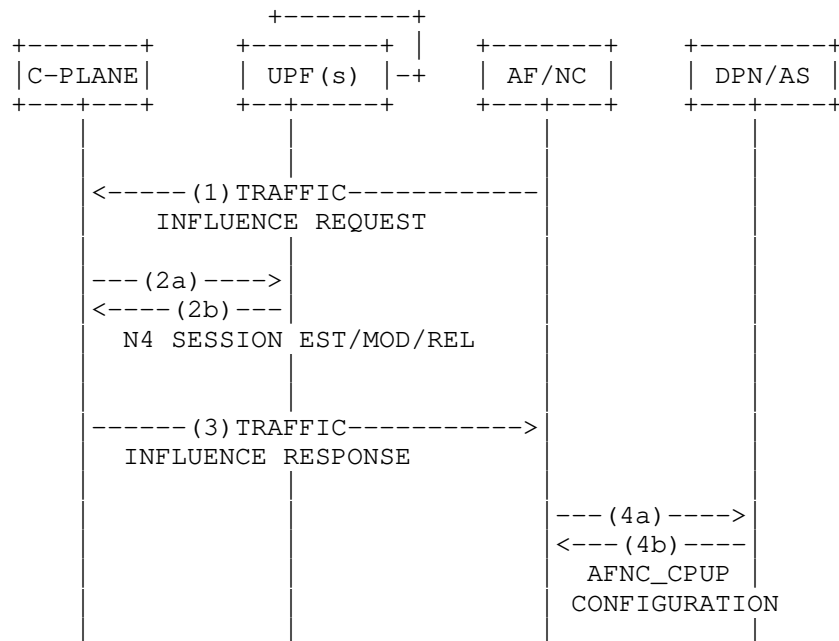


Figure 7: Message flow for AF/NC-initiated N6 policy enforcement

Following, a description for each message is given:

(1) **TRAFFIC INFLUENCE REQUEST:** this message is sent from the AF/NC to the 3GPP's C-PLANE in order to request a modification for UP traffic. The message contains the fields listed in Table 1.

Information model for TRAFFIC INFLUENCE REQUEST message

Message Fields	Description	Notes
Request ID	Identifies the current request in order to match it with following response messages.	-
Traffic Identifier	Identifies the UP traffic which is targeted by the request. Traffic may be identified based on the session, UE-based or even slice-based (i.e., addressing all the traffic belonging to a specific network slice).	3GPP's identifiers defined in [TS23.501] may be used to identify traffic (e.g., DNN for traffic toward a specific Data Network, NSSAI for a specific slice, UE GUTI for a specific user, etc.)
QoS parameters	Contains the QoS parameters for the targeted traffic	-
DPN N6 endpoint	Brings information about the N6 endpoint on the Data Network side.	-

Table 1

Based on the N6 endpoint information, the 3GPP's C-PLANE may take decisions on UPF(s) selection and re-location. For instance, this field could carry a Data Network Access ID (DNAI), identifying a specific Data Network on which the 3GPP's domain could select the best matching UPF (e.g., based on proximity).

(2a) (2b) **N4 SESSION ESTABLISHMENT/MODIFICATION/RELEASE:** this are 3GPP's messages defined in [TS23.502] and used to enforce changing to one or more UPF or to select and configure a new UPF. Through this messages, the N6 policies requested from the AF/NC can be enforced to the UPF(s).

(3) **TRAFFIC INFLUENCE RESPONSE:** this message is sent from the 3GPP's C-PLANE to the AF/NC in order to acknowledge the UP changes made based on the previous request message. The message contains the fields listed in Table 2.

Information model for TRAFFIC INFLUENCE RESPONSE message

Message Fields	Description	Notes
Request ID	Identifies the request message to which this response is referred to.	-
Traffic Identifier	Identifies the UP traffic which is targeted by the request. Traffic may be identified based on the session, UE-based or even slice-based (i.e., addressing all the traffic belonging to a specific network slice).	Traffic actually influenced could differ from the original traffic targeted in the request.
UPF N6 endpoint	Brings information about the N6 endpoint on the 3GPP's side.	-

Table 2

N6 endpoint information on 3GPP's side (e.g., IP address of the N6 endpoint UPF) are used from the AF/NC to set the DPN(s) in order to properly route downlink traffic.

(4a) (4b) **AFNC_CPUP CONFIGURATION:** This message is used to instruct the DPN(s) involved in the UP changes. For instance, in case of UPF re-selection and UPF's N6 endpoint (e.g., IP address) changing, traffic steering rules for downlink traffic need to be enforced to the DPN. The structure of this message is out of the scope of this draft and candidates for managing this interface are already present (e.g., Forwarding Policy Configuration (FPC) defined in [FPC]).

7.2. 3GPP-initiated N6 policy enforcement

A N6 policy can be triggered by the 3GPP domain. In this case, an initial subscription mechanism is needed, in which one or multiple AF subscribe the 3GPP's C-PLANE in order to receive notification about the subscribed events. Some of the events, of which a AF/NC could be interested in, are:

- o re-selection one or multiple UPF(s) from the 3GPP's C-PLANE.
- o changes in the UP traffic QoS parameters.
- o etc.

Figure 8 depicts the message sequence described the AF subscription and a notification from the 3GPP's domain when the specific event occurs.

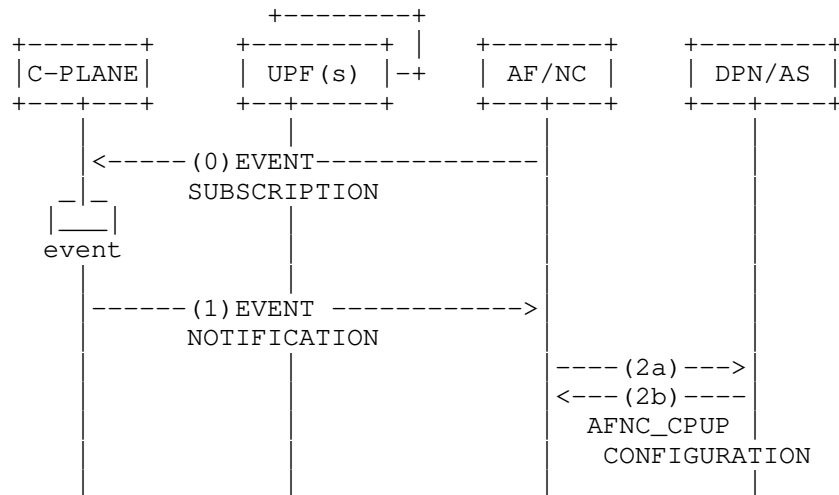


Figure 8: Message flow for 3GPP-initiated N6 policy enforcement

The messages used are here described:

(0) EVENT SUBSCRIPTION: this message is sent from the AF/NC to the 3GPP's C-PLANE in order for the AF/NC to subscribe to some specific UP events. When received from the 3GPP's C-PLANE, all future UP events (e.g., UPF re-selection, changing in UP traffic parameters) which match with the subscription will be notified to the AF/NC. This message fields are listed in Table 3.

Information model for EVENT SUBSCRIPTION message

Message Fields	Description	Notes
Subscription ID	Identifies the subscription in order to then match the resulting notification.	-
Event	Identifies the type of event to which the subscription is referred. For instance, the subscription could refer only to an UPF re-selection event, or may refer to any event for the targeted traffic.	Can be 'all-events' or identify a specific type of event.
Traffic Identifier	Identifies the UP traffic which is targeted by the request. Traffic may be identified based on the session, UE-based or even slice-based (i.e., addressing all the traffic belonging to a specific network slice).	3GPP's identifiers defined in [TS23.501] may be used to identify traffic (e.g., DNN for traffic toward a specific Data Network, NSSAI for a specific slice, UE IP address for a specific user, etc.)

Table 3

(1) EVENT NOTIFICATION: this message is sent from the 3GPP's C-PLANE to the AF/NC, triggered by the subscribed event for the targeted traffic. If no subscription for the specific traffic and event was received before the modification occurs the 3GPP's C-PLANE will not provide any notification for the UP traffic changes. Table 4 lists the field contained in the message.

Information model for EVENT NOTIFICATION message

Message Fields	Description	Notes
Subscription ID	Identifies the subscription message to which this notification is referred to.	-
Traffic Identifier	Identifies the UP traffic which has been change.	Even if there is no notification for traffic which has not been targeted through a subscription, this field may refer to a subset of the traffic targeted in the subscription (e.g., subscription to a specific user traffic and modification of only one PDU sessions for that user).
QoS parameters	Brings information about QoS parameters which have been changed.	-
UPF N6 endpoint	Brings information about the N6 endpoint on the 3GPP's side which have been changed.	-

Table 4

(2a) (2b) AFNC_CPUP CONFIGURATION: This message is used to instruct the DPN(s) involved in the UP changes. For instance, in case of UPF re-selection and UPF's N6 endpoint (e.g., IP address) changing,

traffic steering rules for downlink traffic need to be enforced to the DPN. The structure of this message is anyway out of the scope of this draft and candidates for managing this interface are already present (e.g., Forwarding Polciy Configuration (FPC) defined in [FPC]).

8. IANA Considerations

No IANA action is required for this version of the draft.

9. Security Considerations

Since the solution proposed in this document utilizes the AF to solicit and receive N6 traffic treatment policies from the cellular system's control plane, the trust relationship between the AF and the cellular system's domain matters. In case the AF is located in a different administrative domain, the communication from and to the AF may happen via the system's Network Exposure Functions (NEF). The semantic to request and receive the N6 policy at the AF and in particular the policy types and their descriptions must be aligned to the trust relationship.

Also, the trust relationship between the AF and the DPN/AS matters and a secure direct or indirect (e.g. through an Network Controller) interface, must be ensured.

10. Acknowledgments

The research leading to these results has been partially supported by the H2020-MSCA-ITN-2016 framework under grant agreement number 722788 (SPOTLIGHT).

Authors want to thank Sri Gundavelli, John Kaippallimalil and Shunsuke Homma for their interest and feedback to the use cases and the solution principles for N6 traffic treatment policies.

11. 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>>.

[I-D.hmm-dmm-5g-uplane-analysis]
Homma, S., Miyasaka, T., Matsushima, S., and d. daniel.voyer@bell.ca, "User Plane Protocol and Architectural Analysis on 3GPP 5G System", draft-hmm-dmm-5g-uplane-analysis-02 (work in progress), October 2018.

[I-D.gundavelli-dmm-mfa]

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

[I-D.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.

[FPC]

S.Matsushima, L.Bertz, M.Liebsch, S.Gundavelli, D.Moses, C. Perkins, "Protocol for Forwarding Policy Configuration (FPC) in DMM.", 3GPPTS 23.501, June 2018.

[TS23.501]

3rd Generation Partnership Project (3GPP), "Technical Specification TS23.501, System Architecture for the 5G System, Release 15.", 3GPPTS 23.501, June 2018.

[TS23.502]

3rd Generation Partnership Project (3GPP), "Technical Specification TS23.502, Procedure for the 5G System, Release 15.", 3GPPTS 23.502, June 2018.

Authors' Addresses

Umberto Fattore
NEC Laboratories Europe GmbH
Kurfuersten-Anlage 36
D-69115 Heidelberg
Germany

Email: umberto.fattore@neclab.eu

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

Email: marco.liebsch@neclab.eu

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 - Mobility Support Only When Needed	9
4.1. Nomadic case (no need of IP mobility): Changing to new IP 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.

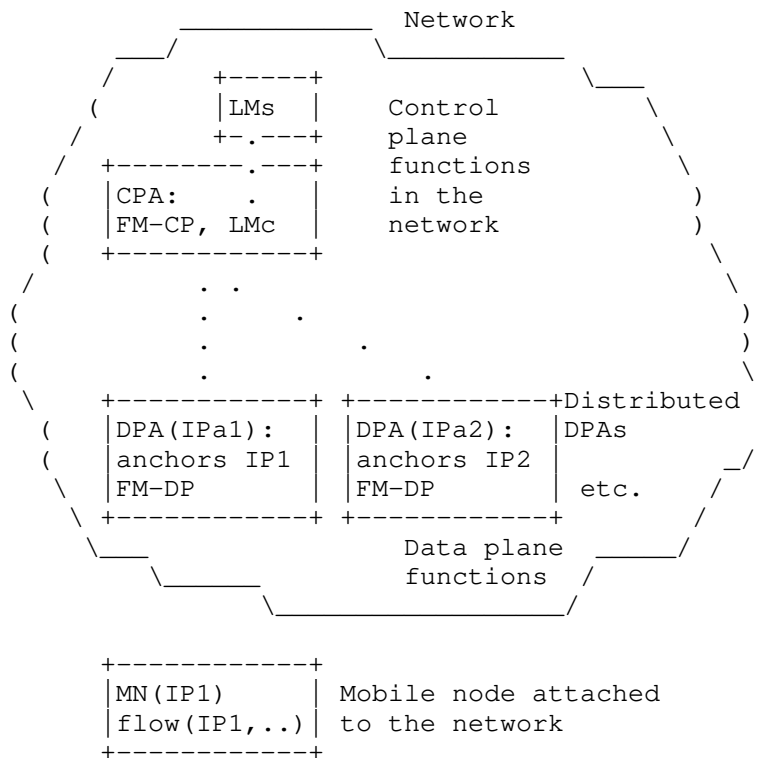


Figure 1: Network-based DMM configuration

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.

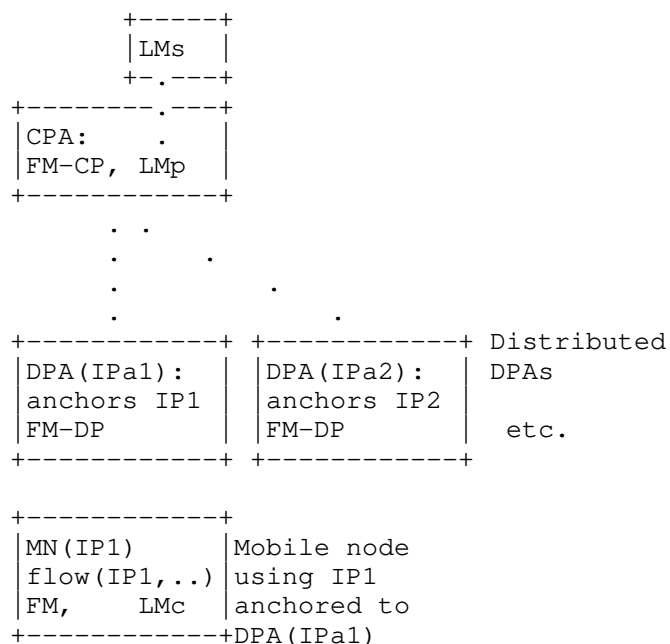


Figure 2: Client-based DMM configuration

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's 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.

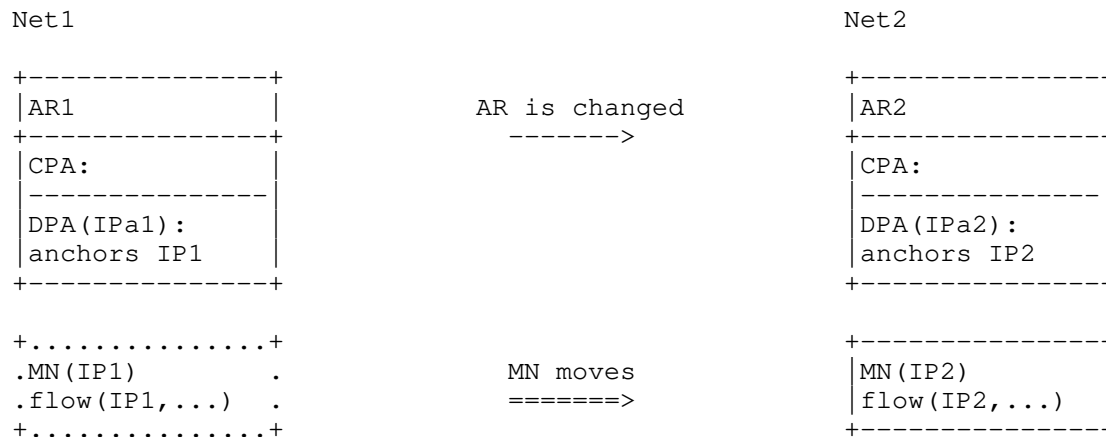


Figure 3: Changing to a new IP address/prefix

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.

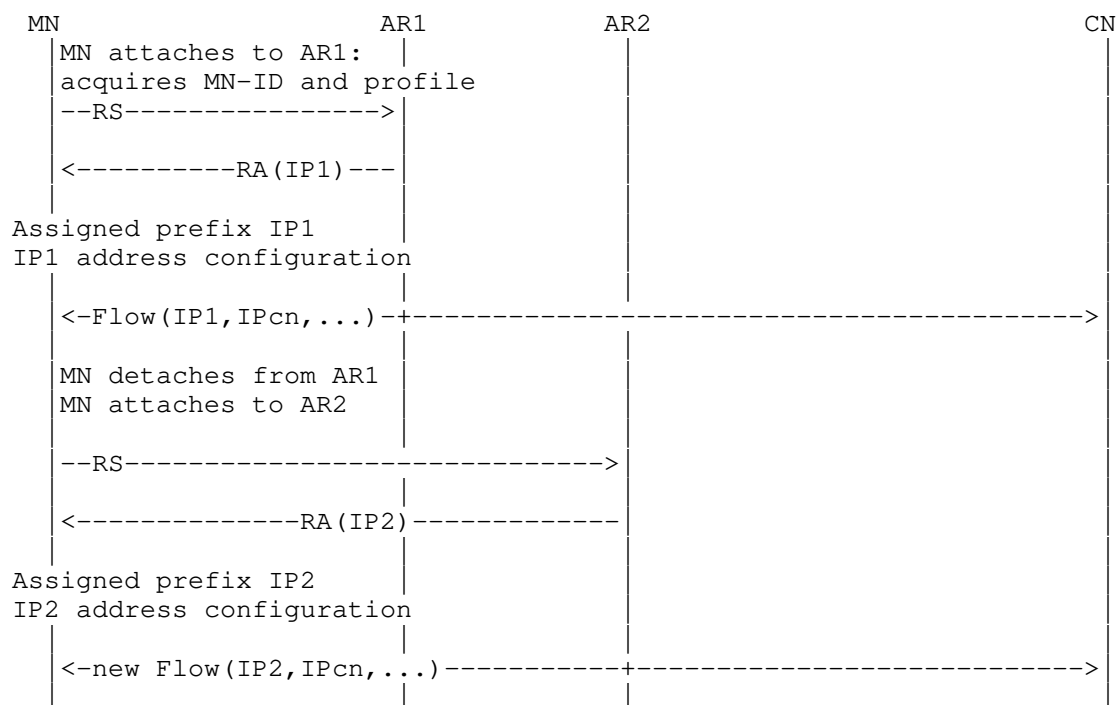


Figure 4: Re-starting a flow with new IP prefix/address

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.

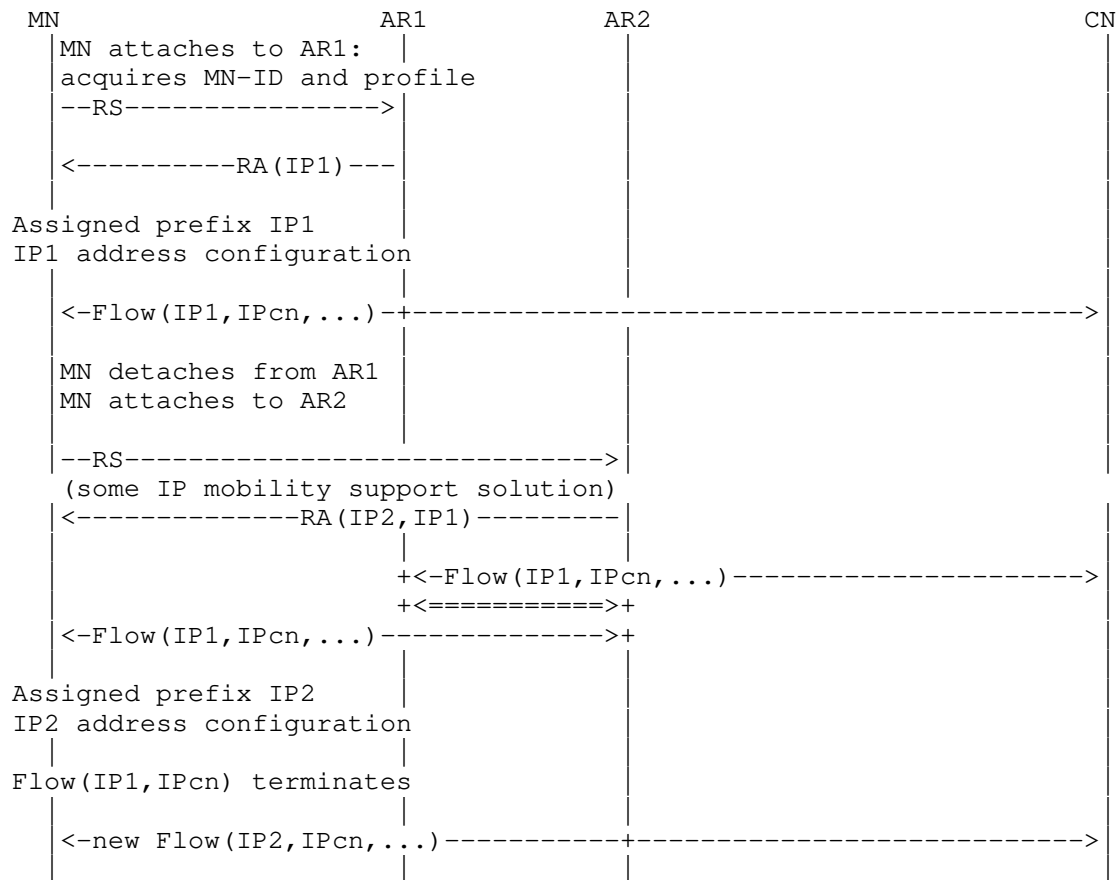


Figure 5: A flow continues to use the IP prefix from its home network after MN has moved to a new network

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.

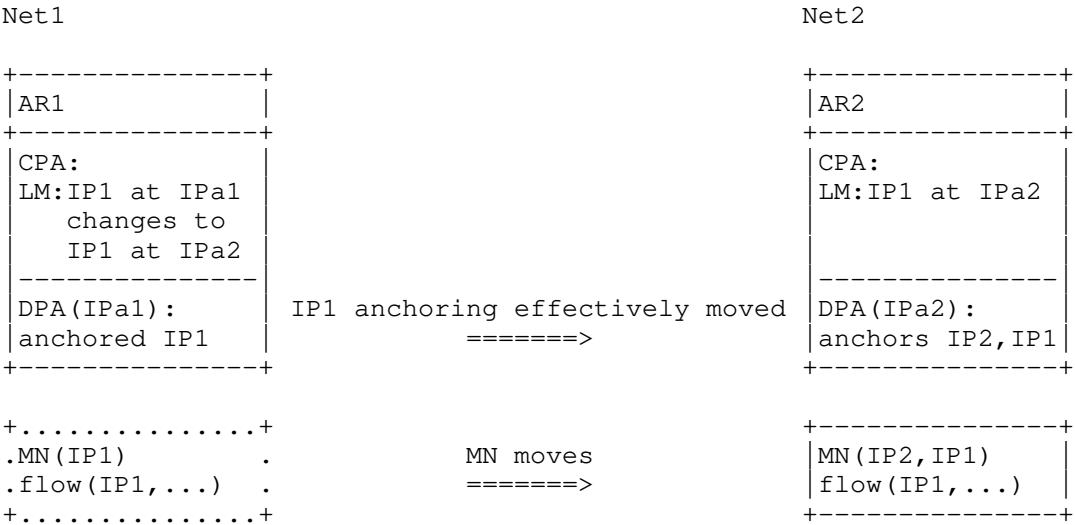


Figure 7: Anchor relocation

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

Xinpeng Wei
Huawei Technologies
Xin-Xi Rd. No. 3, Haidian District
Beijing, 100095
P. R. China

Email: weixinpeng@huawei.com

Jong-Hyouk Lee
Sangmyung University
31, Sangmyeongdae-gil, Dongnam-gu
Cheonan 31066
Republic of Korea

Email: jonghyouk@smu.ac.kr

Seil Jeon
Sungkyunkwan University
2066 Seobu-ro, Jangan-gu
Suwon, Gyeonggi-do
Republic of Korea

Email: 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
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.

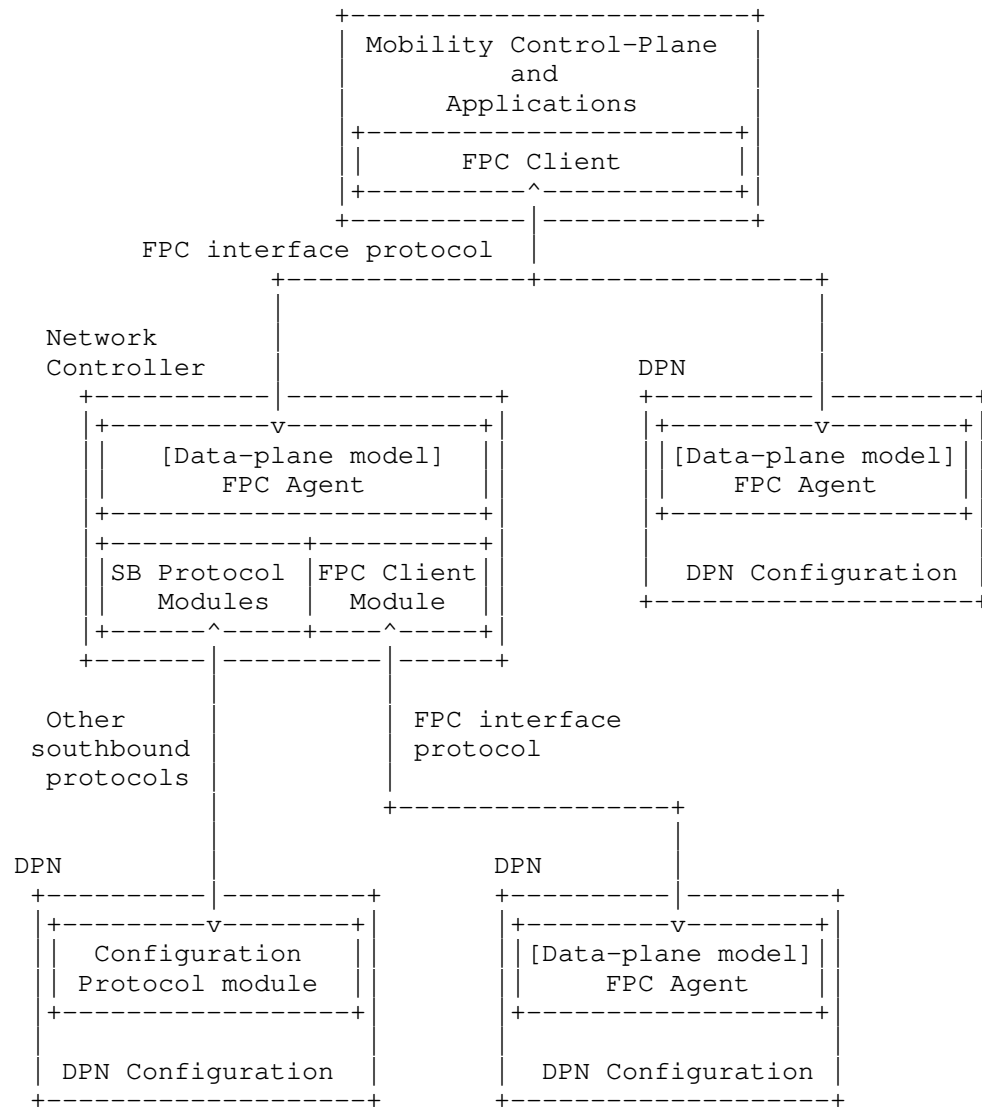


Figure 1: Reference Forwarding Policy Configuration (FPC)
Architecture

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.

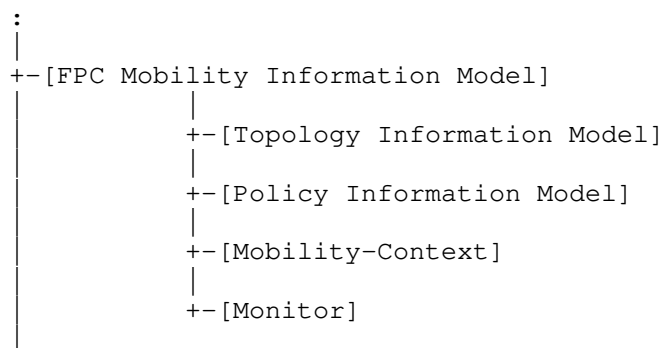


Figure 2: FPC Information Model structure

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.

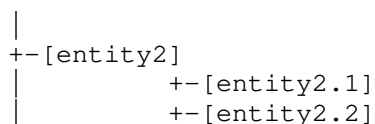


Figure 3: Model Notation - An Example

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.

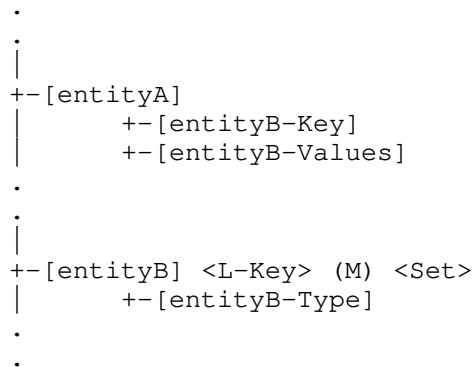


Figure 5: Indexed sets of entities

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):

```
|
+-[Topology Information Model]
|   +-[Extensible: FALSE]
|   +-[Service-Group]
|   +-[DPN] <Set>
|   +-[Domain] <Set>
```

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 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). 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: 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 '*'.

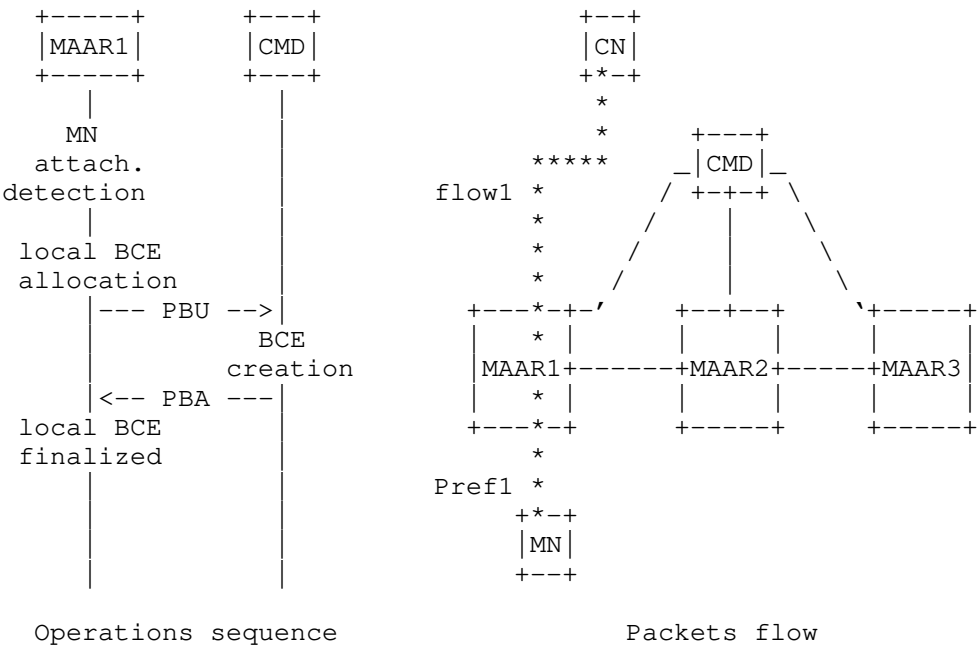


Figure 1: First attachment to the network

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.

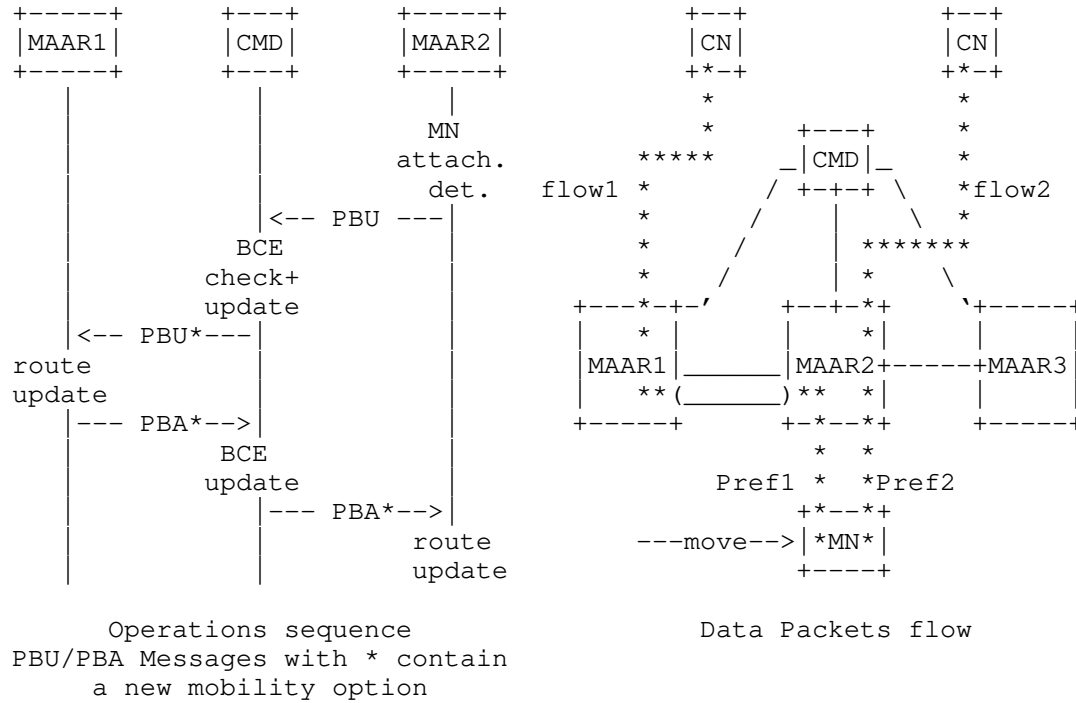


Figure 2: Scenario after a handover, CMD as relay

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.

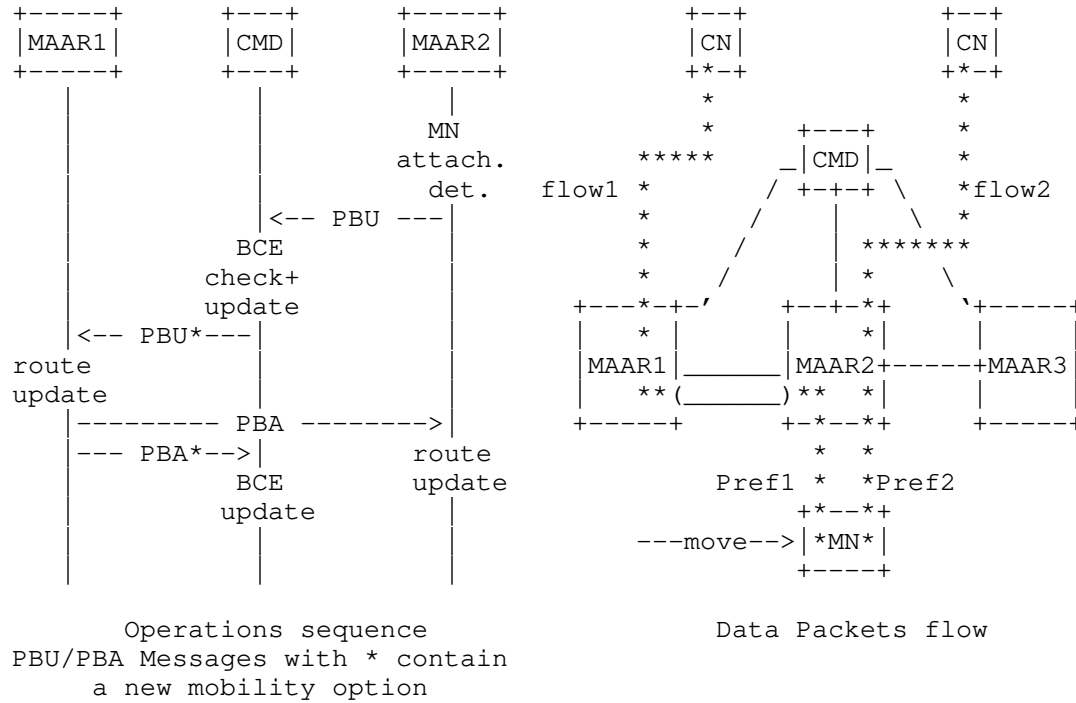


Figure 3: Scenario after a handover, CMD as locator

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.

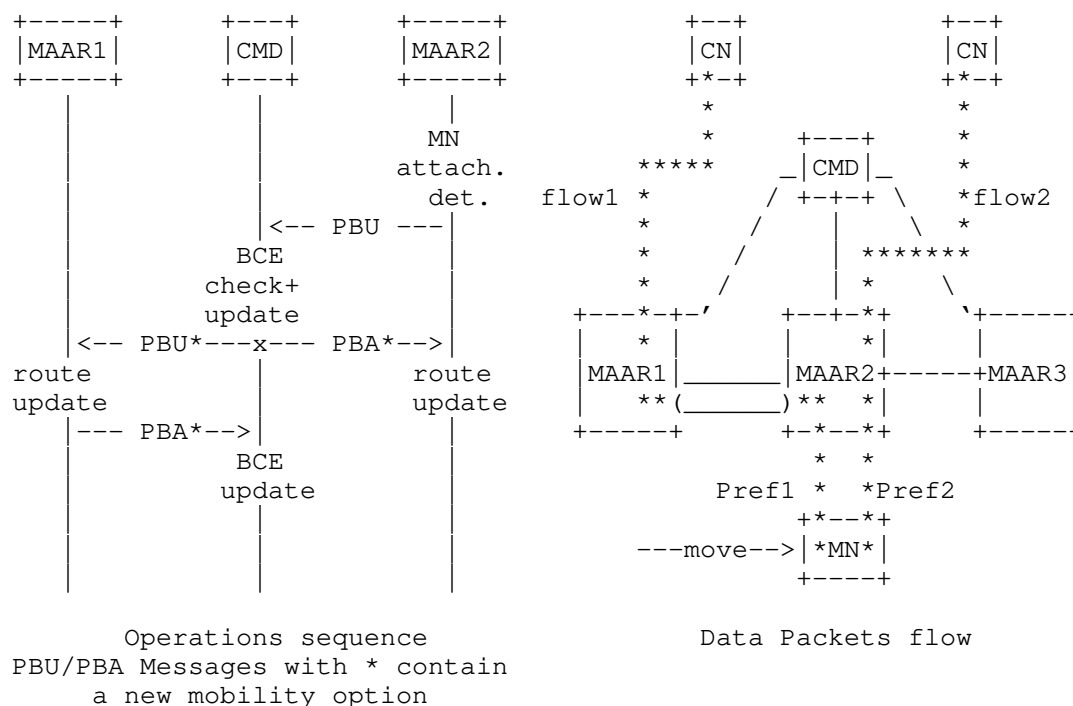


Figure 4: Scenario after a handover, CMD as proxy

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.

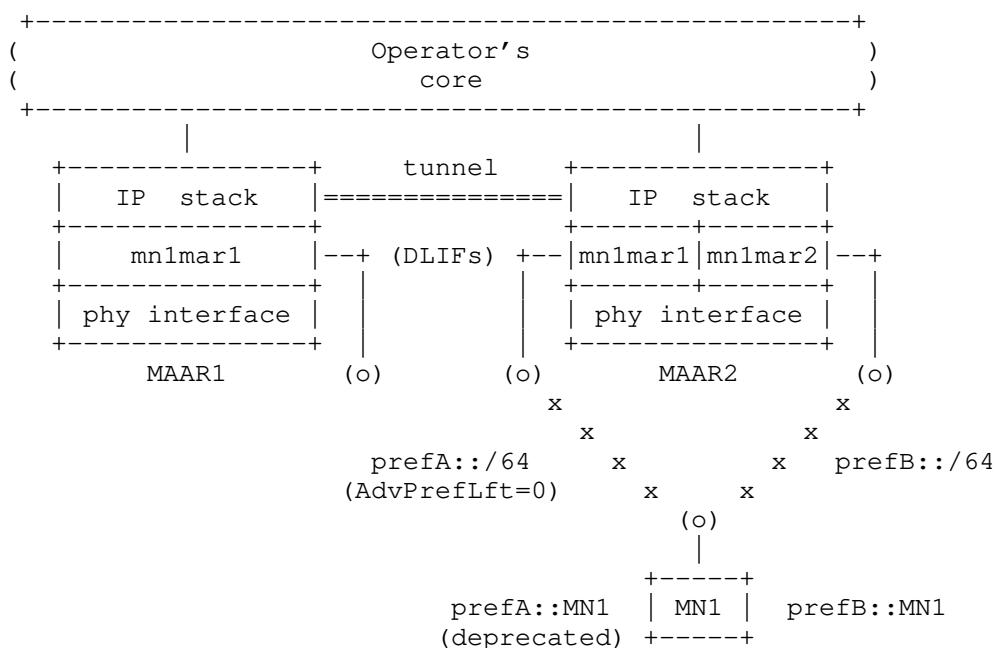


Figure 5: DLIF: exposing multiple routers (one per P-MAAR)

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).

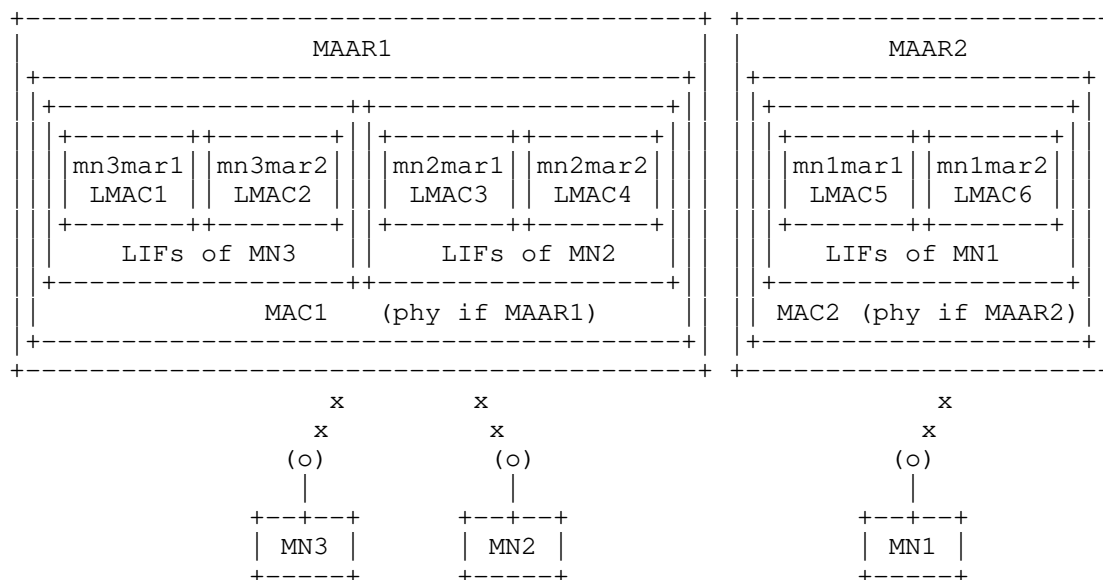


Figure 6: Distributed Logical Interface concept

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

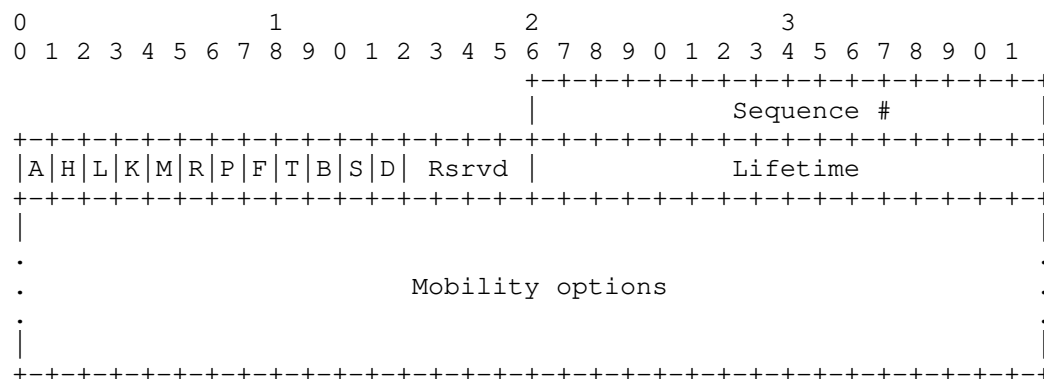
(MAAR1 in the example). These routes are advertised through the distributed logical interface representing the MAAR providing access to the local network (MAAR1 in this example). In this way, if MN1 moves from MAAR1 to MAAR2, any active session that MN1 may have with a node on the local network connected 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.

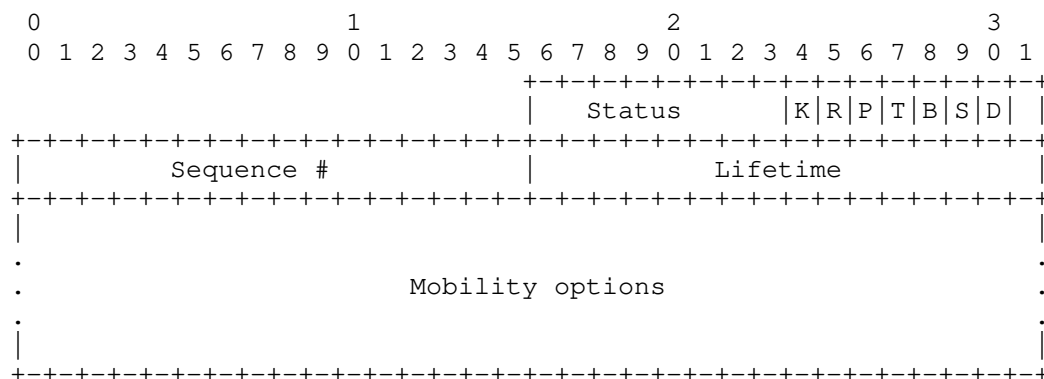
Mobility Options

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

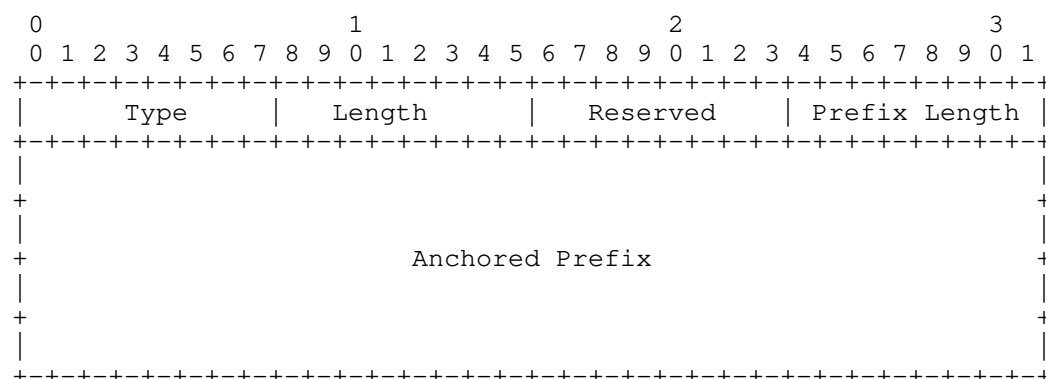
Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2 of [RFC6275]. The MAAR MUST ignore and skip any options 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.

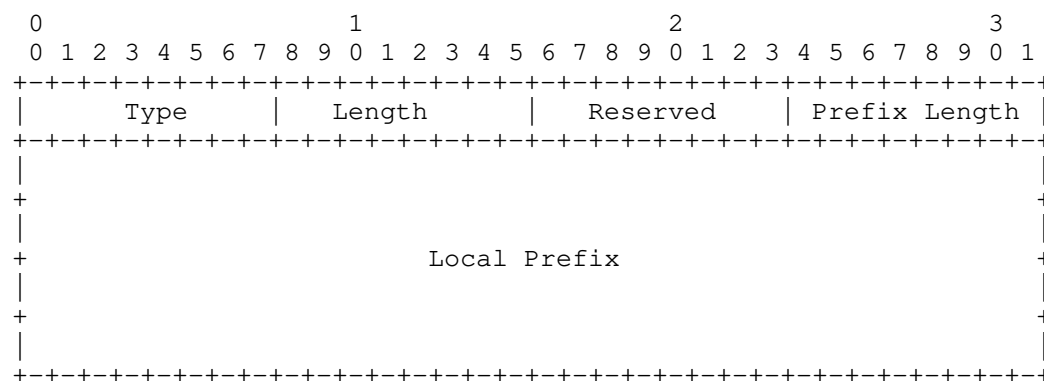
Anchored Prefix

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.

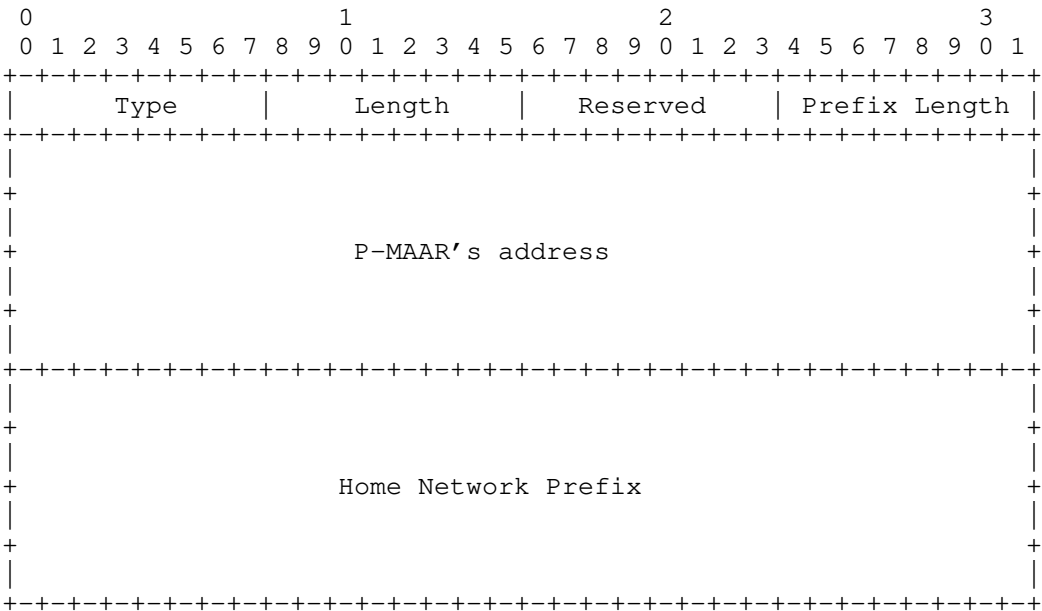
Local Prefix

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.

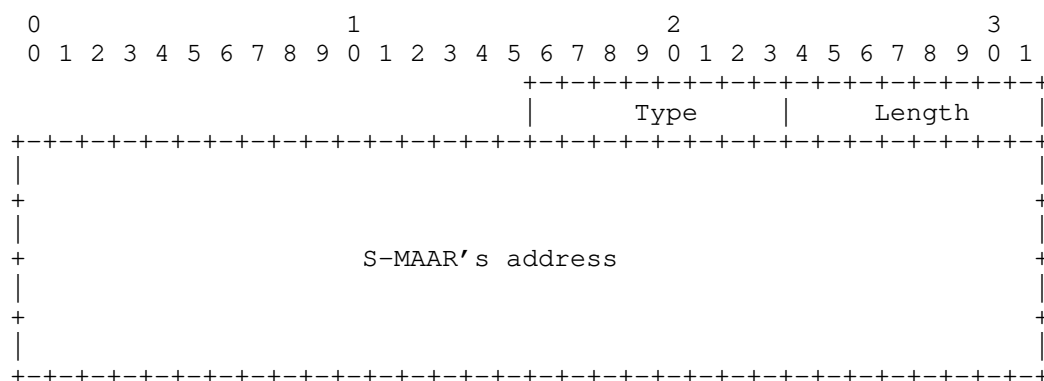
Home Network Prefix

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.

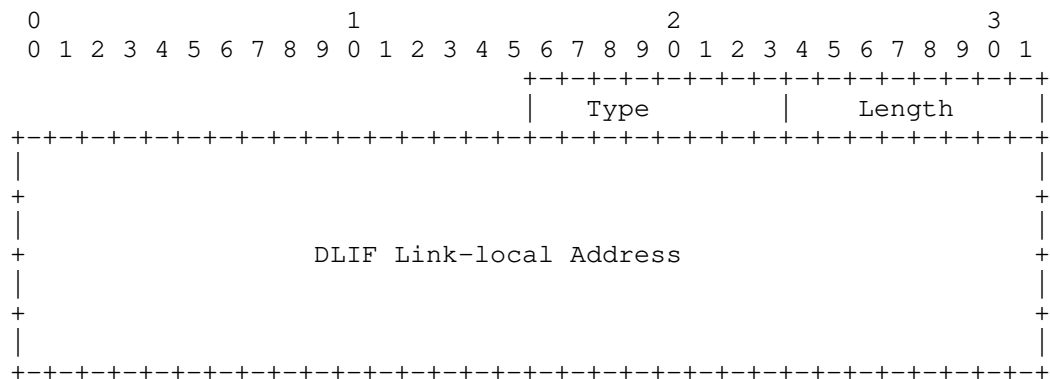
Serving MAAR's address

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.

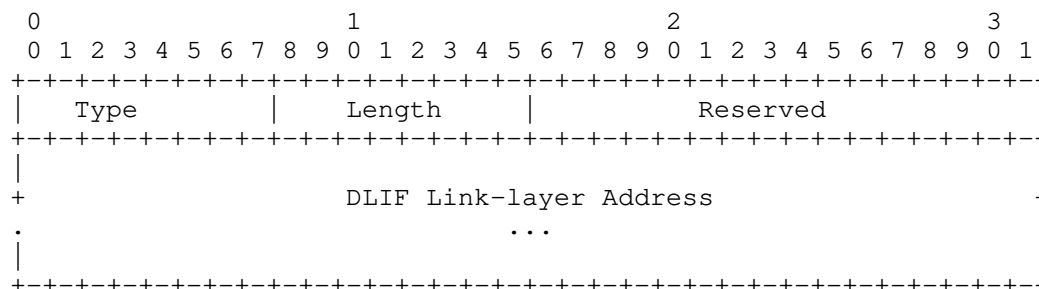
DLIF Link-local Address

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
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
URI: <http://www.it.uc3m.es/aoliva/>

Fabio Giust
Athonet S.r.l.

Email: fabio.giust.2011@ieee.org

Juan Carlos Zuniga
SIGFOX
425 rue Jean Rostand
Labège 31670
France

Email: j.c.zuniga@ieee.org
URI: <http://www.sigfox.com/>

Alain Mourad
InterDigital Europe

Email: 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.

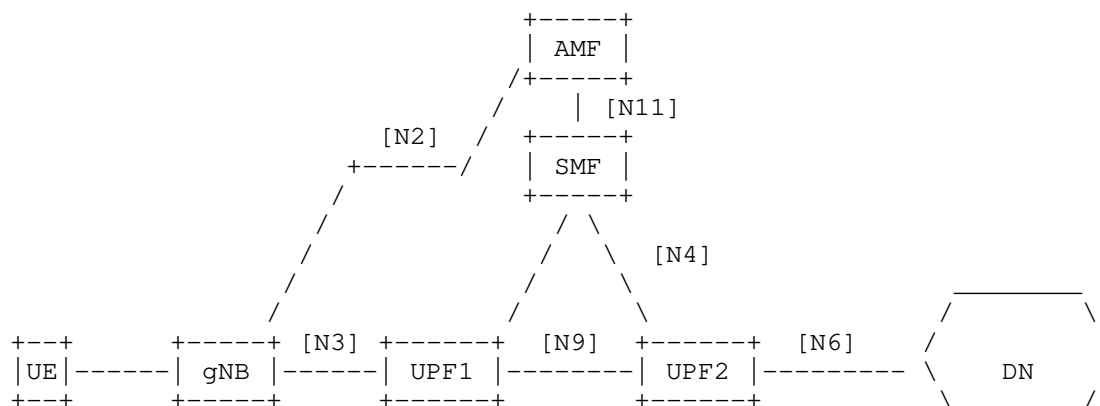


Figure 1: 3GPP 5G Reference Architecture

- * 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.

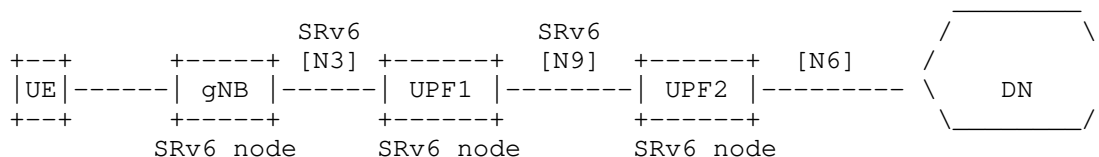


Figure 2: Traditional mode - example topology

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.

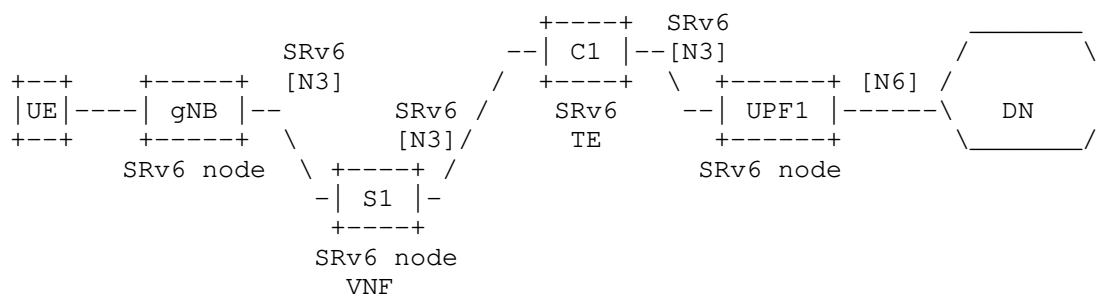


Figure 3: Enhanced mode - Example topology

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.

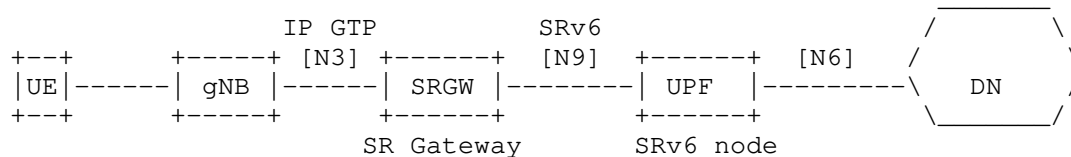


Figure 4: Example topology for interworking

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.

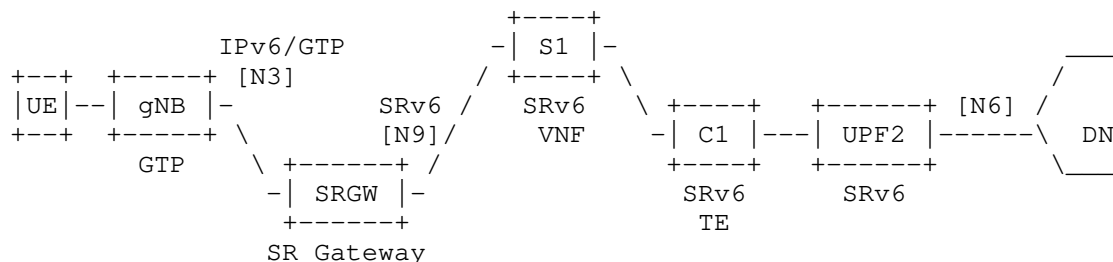


Figure 5: Enhanced mode with unchanged gNB IPv6/GTP behavior

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.

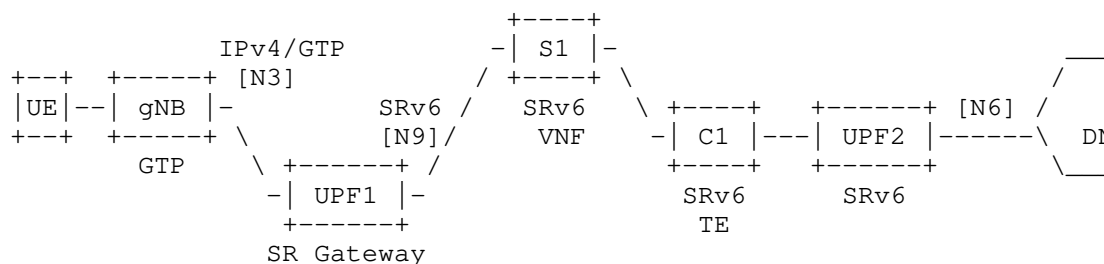


Figure 6: Enhanced mode with unchanged gNB IPv4/GTP behavior

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.

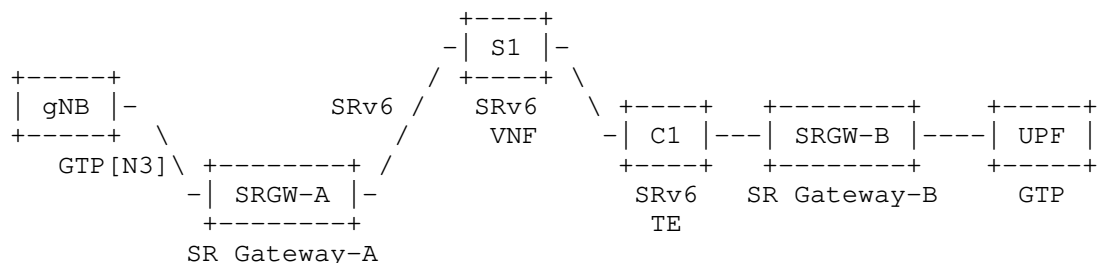


Figure 7: Example topology for SRv6 Drop-in mode

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.

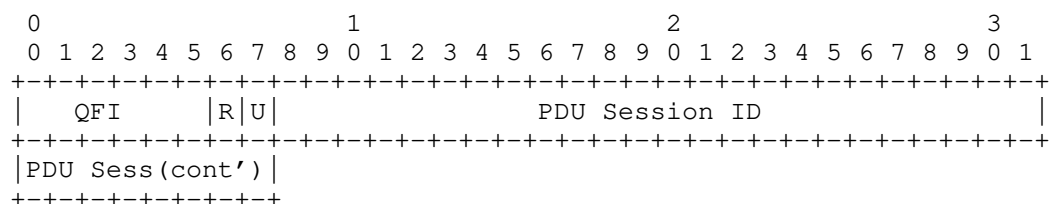


Figure 8: Args.Mob.Session format

- * 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:

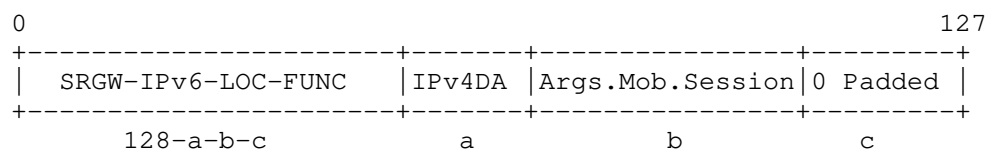


Figure 9: End.M.GTP4.E SID Encoding

The IPv6 Source Address has the following format:

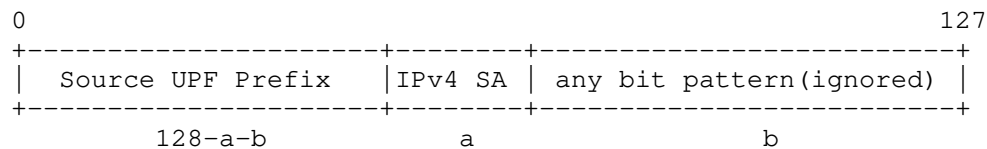


Figure 10: IPv6 SA Encoding for End.M.GTP4.E

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:

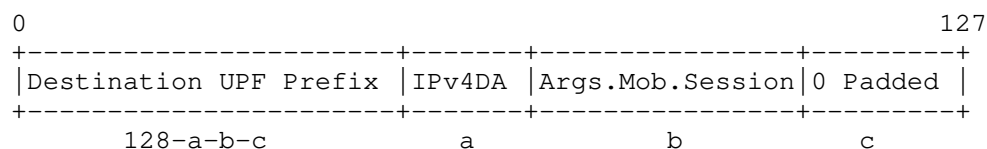


Figure 11: H.M.GTP4.D SID Encoding

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:

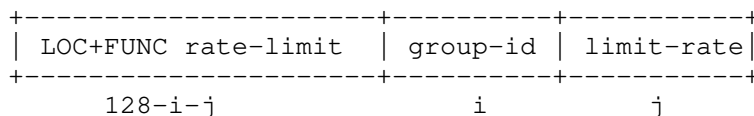


Figure 12: End.Limit: Rate limiting behavior argument format

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

Tetsuya Murakami Arrcus, Inc. United States of America

Email: tetsuya.ietf@gmail.com

14. References

14.1. Normative References

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

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

14.2. Informative References

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

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

Authors' Addresses

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

Clarence Filsfils
Cisco Systems, Inc.
Belgium
Email: cf@cisco.com

Miya Kohno
Cisco Systems, Inc.
Japan
Email: mkohno@cisco.com

Pablo Camarillo Garvia (editor)
Cisco Systems, Inc.
Spain
Email: pcamaril@cisco.com

Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Charles E. Perkins
Lupin Lodge
20600 Aldercroft Heights Rd.
Los Gatos, CA 95033
United States of America
Email: charliep@computer.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 October 2022

F. L. Templin, Ed.
G. Saccone
Boeing Research & Technology
G. Dawra
LinkedIn
A. Lindem
V. Moreno
Cisco Systems, Inc.
19 April 2022

A Simple BGP-based Mobile Routing System for the Aeronautical
Telecommunications Network
draft-ietf-rtgwg-atn-bgp-17

Abstract

The International Civil Aviation Organization (ICAO) is investigating mobile routing solutions for a worldwide Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS). The ATN/IPS will eventually replace existing communication services with an IP-based service supporting pervasive Air Traffic Management (ATM) for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. This informational document describes a simple and extensible mobile routing service based on industry-standard BGP to address the ATN/IPS requirements.

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 21 October 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	2
2. Terminology	7
3. ATN/IPS Routing System	9
4. ATN/IPS (Radio) Access Network (ANET) Model	14
5. ATN/IPS Route Optimization	16
6. BGP Protocol Considerations	19
7. Stub AS Mobile Routing Services	21
8. Implementation Status	21
9. IANA Considerations	21
10. Security Considerations	21
10.1. Public Key Infrastructure (PKI) Considerations	22
11. Acknowledgements	23
12. References	23
12.1. Normative References	23
12.2. Informative References	24
Appendix A. BGP Convergence Considerations	26
Appendix B. Change Log	26
Authors' Addresses	27

1. Introduction

The worldwide Air Traffic Management (ATM) system today uses a service known as Aeronautical Telecommunications Network based on Open Systems Interconnection (ATN/OSI). The service is used to augment controller to pilot voice communications with rudimentary short text command and control messages. The service has seen successful deployment in a limited set of worldwide ATM domains.

The International Civil Aviation Organization (ICAO) is now undertaking the development of a next-generation replacement for ATN/OSI known as Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) [ATN][ATN-IPS]. ATN/IPS will eventually

provide an IPv6-based [RFC8200] service supporting pervasive ATM for Air Traffic Controllers (ATC), Airline Operations Controllers (AOC), and all commercial aircraft worldwide. As part of the ATN/IPS undertaking, a new mobile routing service will be needed. This document presents an approach based on the Border Gateway Protocol (BGP) [RFC4271].

Aircraft communicate via wireless aviation data links that typically support much lower data rates than terrestrial wireless and wired-line communications. For example, some Very High Frequency (VHF)-based data links only support data rates on the order of 32Kbps and an emerging L-Band data link that is expected to play a key role in future aeronautical communications only supports rates on the order of 1Mbps. Although satellite data links can provide much higher data rates during optimal conditions, like any other aviation data link they are subject to errors, delay, disruption, signal intermittence, degradation due to atmospheric conditions, etc. The well-connected ground domain ATN/IPS network should therefore treat each safety-of-flight critical packet produced by (or destined to) an aircraft as a precious commodity and strive for an optimized service that provides the highest possible degree of reliability. Furthermore, continuous performance-intensive control messaging services such as BGP peering sessions must be carried only over the well-connected ground domain ATN/IPS network and never over low-end aviation data links.

The ATN/IPS is an IP-based overlay network configured over one or more Internetworking underlays ("INETs") maintained by aeronautical network service providers such as ARINC, SITA and Inmarsat. The Overlay Multilink Network Interface (OMNI) [I-D.templin-6man-omni] uses an adaptation layer encapsulation to create a Non-Broadcast, Multiple Access (NBMA) virtual link spanning the entire ATN/IPS. Each aircraft connects to the OMNI link via an OMNI interface configured over the aircraft's underlying physical and/or virtual access network interfaces.

Each underlying INET comprises one or more "partitions" where all nodes within a partition can exchange packets with all other nodes, i.e., the partition is connected internally. There is no requirement that each INET partition uses the same IP protocol version nor has consistent IP addressing plans in comparison with other partitions. Instead, the OMNI link sees each partition as a "segment" of a link-layer topology concatenated by a service known as the OMNI Adaptation Layer (OAL) [I-D.templin-6man-omni] based on IPv6 encapsulation [RFC2473].

The IPv6 addressing architecture provides different classes of addresses, including Global Unicast Addresses (GUAs), Unique Local Addresses (ULAs) and Link-Local Addresses (LLAs) [RFC4291][RFC4193].

The ATN/IPS receives an IPv6 GUA Mobility Service Prefix (MSP) from an Internet assigned numbers authority, and each aircraft will receive a Mobile Network Prefix (MNP) delegation from the MSP that accompanies the aircraft wherever it travels. ATCs and AOCs will likewise receive MNPs, but they would typically appear in static (not mobile) deployments such as air traffic control towers, airline headquarters, etc. (Note that while IPv6 GUAs are assumed for ATN/IPS, IPv4 with public/private address could also be used.)

The adaptation layer uses ULAs in the source and destination addresses of adaptation layer IPv6 encapsulation headers. Each ULA includes an MNP in the interface identifier ("MNP-ULA"), as discussed in [I-D.templin-6man-omni]. Due to MNP delegation policies and random node mobility properties, MNP-ULAs are generally not aggregable in the BGP routing service and are represented as many more-specific prefixes instead of a smaller number of aggregated prefixes.

In addition, BGP routing service infrastructure nodes configure administratively-assigned ULAs ("ADM-ULA") that are statically-assigned and derived from a shorter ADM-ULA prefix assigned to their BGP network partitions. Unlike MNP-ULAs, the ADM-ULAs are persistently present and unchanging in the routing system. The BGP routing services therefore establish forwarding table entries based on these MNP-ULAs and ADM-ULAs instead of based on the GUA MNPs themselves. However, nodes set the 40-bit Global ID and 16-bit Subnet ID to 0 when they advertise MNP-ULAs in BGP routing exchanges and/or install MNP-ULAs in forwarding tables.

Both ADM-ULAs and MNP-ULAs are used by the OAL for nested encapsulation where the inner IPv6 packet is encapsulated in an IPv6 adaptation layer header with ULA source and destination addresses, which is then encapsulated in an IP header specific to the underlying Internetwork that will carry the actual packet transmission. A high level ATN/IPS network diagram is shown in Figure 1:

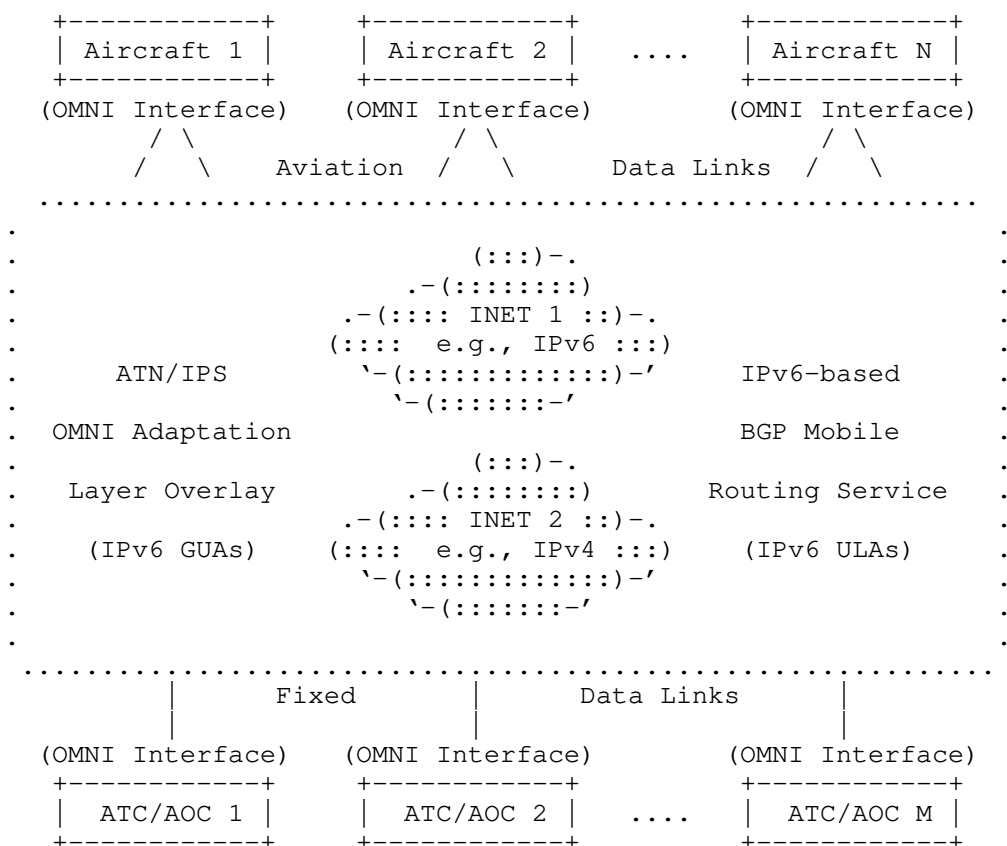


Figure 1: ATN/IPS Network Diagram

Connexion By Boeing [CBB] was an early aviation mobile routing service based on dynamic updates in the global public Internet BGP routing system. Practical experience with the approach has shown that frequent injections and withdrawals of prefixes in the Internet routing system can result in excessive BGP update messaging, slow routing table convergence times, and extended outages when no route is available. This is due to both conservative default BGP protocol timing parameters (see Section 6) and the complex peering interconnections of BGP routers within the global Internet infrastructure. The situation is further exacerbated by frequent aircraft mobility events that each result in BGP updates that must be propagated to all BGP routers in the Internet that carry a full routing table.

We therefore consider an approach using a BGP overlay network routing system where a private BGP routing protocol instance is maintained between ATN/IPS Autonomous System (AS) Border Routers (ASBRs). The private BGP instance does not interact with the native BGP routing systems in underlying INETs, and BGP updates are unidirectional from "stub" ASBRs (s-ASBRs) to a small set of "core" ASBRs (c-ASBRs) in a hub-and-spokes topology. No extensions to the BGP protocol are necessary, and BGP routing is based on (intermediate-layer) ULAs instead of upper- or lower-layer public/private IP prefixes. This allows ASBRs to perform adaptation layer forwarding based on intermediate layer IPv6 header information instead of network layer forwarding based on upper layer IP header information or link layer forwarding based on lower layer IP header information.

The s-ASBRs for each stub AS connect to a small number of c-ASBRs via dedicated high speed links and/or secured tunnels (e.g., IPsec [RFC4301], WireGuard [WG], etc.) over the underlying INET. Neighboring ASBRs should use also such IP layer security encapsulations over direct physical links to ensure INET layer security.

The s-ASBRs engage in external BGP (eBGP) peerings with their respective c-ASBRs, and only maintain routing table entries for the MNP-ULAs currently active within the stub AS. The s-ASBRs send BGP updates for MNP-ULA injections or withdrawals to c-ASBRs but do not receive any BGP updates from c-ASBRs. Instead, the s-ASBRs maintain default routes with their c-ASBRs as the next hop, and therefore hold only partial topology information.

The c-ASBRs connect to other c-ASBRs within the same partition using internal BGP (iBGP) peerings over which they collaboratively maintain a full routing table for all active MNP-ULAs currently in service within the partition. Therefore, only the c-ASBRs maintain a full BGP routing table and never send any BGP updates to s-ASBRs. This simple routing model therefore greatly reduces the number of BGP updates that need to be synchronized among peers, and the number is reduced further still when intradomain routing changes within stub ASes are processed within the AS instead of being propagated to the core. BGP Route Reflectors (RRs) [RFC4456] can also be used to support increased scaling properties.

When there are multiple INET partitions, the c-ASBRs of each partition use eBGP to peer with the c-ASBRs of other partitions so that the full set of ULAs for all partitions are known globally among all of the c-ASBRs. Each c/s-ASBR further configures an ADM-ULA which is taken from an ADM-ULA prefix assigned to each partition, as well as static forwarding table entries for all other OMNI link partition prefixes. Both ADM-ULAs and MNP-ULAs are used by the OAL

for nested encapsulation where the inner IPv6 packet is encapsulated in an IPv6 OAL header with ULA source and destination addresses, which is then encapsulated in an IP header specific to the INET partition.

With these intra- and inter-INET BGP peerings in place, a forwarding plane spanning tree is established that properly covers the entire operating domain. All nodes in the network can be visited using strict spanning tree hops, but in many instances this may result in longer paths than are necessary. AERO [I-D.templin-6man-aero] provides an example service for discovering and utilizing (route-optimized) shortcuts that do not always follow strict spanning tree paths.

The remainder of this document discusses the proposed BGP-based ATN/IPS mobile routing service.

2. Terminology

The terms Autonomous System (AS) and Autonomous System Border Router (ASBR) are the same as defined in [RFC4271].

The following terms are defined for the purposes of this document:

Air Traffic Management (ATM)

The worldwide service for coordinating safe aviation operations.

Air Traffic Controller (ATC)

A government agent responsible for coordinating with aircraft within a defined operational region via voice and/or data Command and Control messaging.

Airline Operations Controller (AOC)

An airline agent responsible for tracking and coordinating with aircraft within their fleet.

Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS)

A future aviation network for ATCs and AOCs to coordinate with all aircraft operating worldwide. The ATN/IPS will be an IPv6-based overlay network service that connects access networks via tunneling over one or more Internetworking underlays.

Internetworking underlay ("INET")

A wide-area network that supports overlay network tunneling and connects Radio Access Networks to the rest of the ATN/IPS. Example INET service providers for civil aviation include ARINC, SITA and Inmarsat.

(Radio) Access Network ("ANET")

An aviation radio data link service provider's network, including radio transmitters and receivers as well as supporting ground-domain infrastructure needed to convey a customer's data packets to outside INETs. The term ANET is intended in the same spirit as for radio-based Internet service provider networks (e.g., cellular operators), but can also refer to ground-domain networks that connect AOCs and ATCs.

partition (or "segment")

A fully-connected internal subnetwork of an INET in which all nodes can communicate with all other nodes within the same partition using the same IP protocol version and addressing plan. Each INET consists of one or more partitions.

Overlay Multilink Network Interface (OMNI)

A virtual layer 2 bridging service that presents an ATN/IPS overlay unified link view even though the underlay may consist of multiple INET partitions. The OMNI virtual link is manifested through nested encapsulation in which original IP packets from the ATN/IPS are first encapsulated in ULA-addressed IPv6 headers which are then forwarded to the next hop using INET encapsulation if necessary. Forwarding over the OMNI virtual link is therefore based on ULAs instead of the original IP addresses. In this way, packets sent from a source can be conveyed over the OMNI virtual link even though there may be many underlying INET partitions in the path to the destination.

OMNI Adaptation Layer (OAL)

A middle layer below the IP layer but above the INET layer that applies IP-in-IPv6 encapsulation prior to INET encapsulation. The IPv6 encapsulation header inserted by the OAL uses ULAs instead of GUAs. End systems that configure OMNI interfaces act as OAL ingress and egress points, while intermediate systems with OMNI interfaces act as OAL forwarding nodes. There may be zero, one or many intermediate nodes between the OAL ingress and egress, but the upper layer IPv6 Hop Limit is not decremented during (OAL layer) forwarding. Further details on OMNI and the OAL are found in [I-D.templin-6man-omni].

OAL Autonomous System (OAL AS)

A "hub-of-hubs" autonomous system maintained through peerings between the core autonomous systems of different OMNI virtual link partitions.

Core Autonomous System Border Router (c-ASBR)

A BGP router located in the hub of the INET partition hub-and-spokes overlay network topology.

Core Autonomous System (Core AS)

The "hub" autonomous system maintained by all c-ASBRs within the same partition.

Stub Autonomous System Border Router (s-ASBR)

A BGP router configured as a spoke in the INET partition hub-and-spokes overlay network topology.

Stub Autonomous System (Stub AS)

A logical grouping that includes all Clients currently associated with a given s-ASBR.

Client

An ATC, AOC or aircraft that connects to the ATN/IPS as a leaf node. The Client could be a singleton host, or a router that connects a mobile or fixed network.

Proxy/Server

An ANET/INET border node that acts as a transparent intermediary between Clients and s-ASBRs. From the Client's perspective, the Proxy/Server presents the appearance that the Client is communicating directly with the s-ASBR. From the s-ASBR's perspective, the Proxy/Server presents the appearance that the s-ASBR is communicating directly with the Client.

Mobile Network Prefix (MNP)

An IPv6 prefix that is delegated to any ATN/IPS end system, including ATCs, AOCs, and aircraft.

Mobility Service Prefix (MSP)

An aggregated IP prefix assigned to the ATN/IPS by an Internet assigned numbers authority, and from which all MNPs are delegated (e.g., up to 2^{32} IPv6 /56 MNPs could be delegated from a /24 MSP).

3. ATN/IPS Routing System

The ATN/IPS routing system comprises a private BGP instance coordinated in an overlay network via tunnels between neighboring ASBRs over one or more underlying INETs. The ATN/IPS routing system interacts with underlying INET BGP routing systems only through the static advertisement of a small and unchanging set of MSPs instead of the full dynamically changing set of MNPs.

Within each INET partition, each s-ASBR connects a stub AS to the INET partition core using a distinct stub AS Number (ASN). Each s-ASBR further uses eBGP to peer with one or more c-ASBRs. All c-ASBRs are members of the INET partition core AS, and use a shared

core ASN. Unique ASNs are assigned according to the standard 32-bit ASN format [RFC4271][RFC6793]. Since the BGP instance does not connect with any INET BGP routing systems, the ASNs can be assigned from the [RFC6996] 32-bit ASN space which reserves 94,967,295 numbers for private use. The ASNs must be allocated and managed by an ATN/IPS assigned numbers authority established by ICAO, which must ensure that ASNs are responsibly distributed without duplication and/or overlap.

The c-ASBRs use iBGP to maintain a synchronized consistent view of all active MNP-ULAs currently in service within the INET partition. Figure 2 below represents the reference INET partition deployment. (Note that the figure shows details for only two s-ASBRs (s-ASBR1 and s-ASBR2) due to space constraints, but the other s-ASBRs should be understood to have similar Stub AS, MNP and eBGP peering arrangements.) The solution described in this document is flexible enough to extend to these topologies.

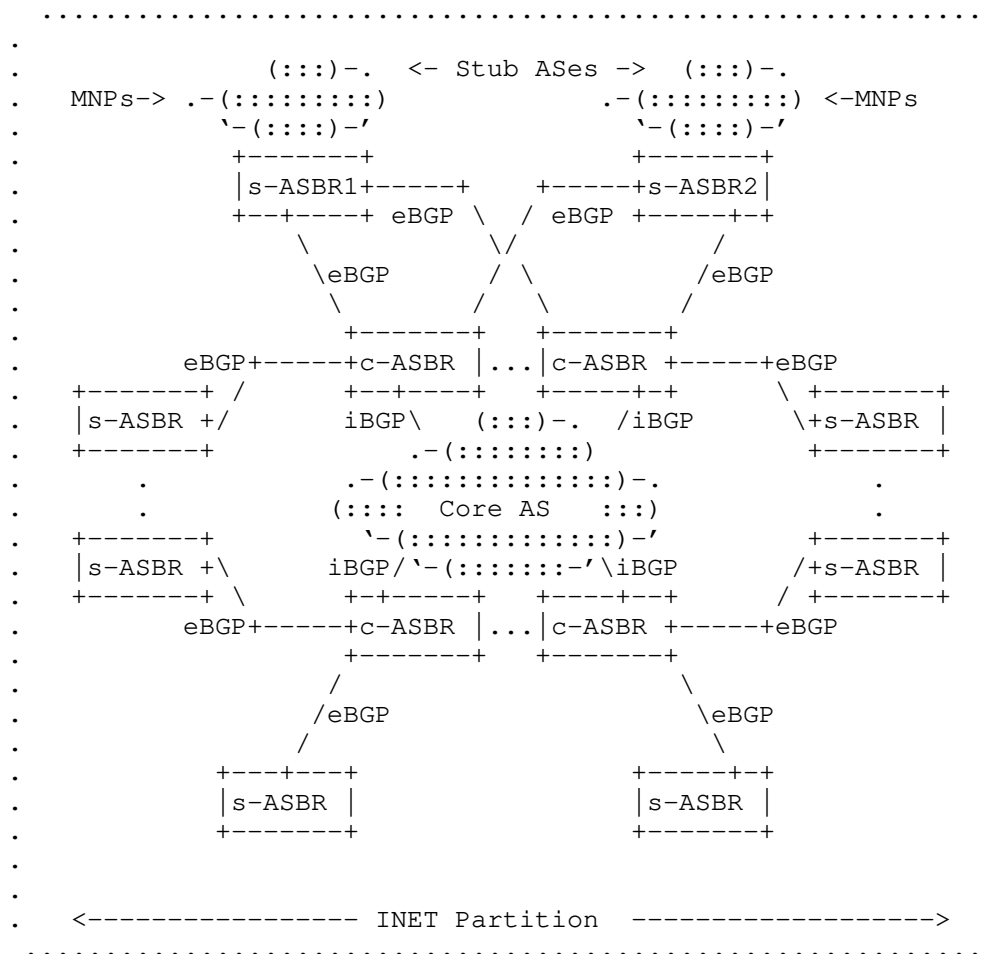


Figure 2: INET Partition Reference Deployment

In the reference deployment, each s-ASBR maintains routes for active MNP-ULAs that currently belong to its stub AS. In response to "Inter-domain" mobility events, each s-ASBR dynamically announces new MNP-ULAs and withdraws departed MNP-ULAs in its eBGP updates to c-ASBRs. Since ATN/IPS end systems are expected to remain within the same stub AS for extended timeframes, however, intra-domain mobility events (such as an aircraft handing off between cell towers) are handled within the stub AS instead of being propagated as inter-domain eBGP updates.

Each c-ASBR configures a black-hole route for each of its MSPs. By black-holing the MSPs, the c-ASBR maintains forwarding table entries only for the MNP-ULAs that are currently active. If an arriving packet matches a black-hole route without matching an MNP-ULA, the c-ASBR should drop the packet and may also generate an ICMPv6 Destination Unreachable message [RFC4443], i.e., without forwarding the packet outside of the ATN/IPS overlay based on a less-specific route.

The c-ASBRs do not send BGP updates for MNP-ULAs to s-ASBRs, but instead originate a default route. In this way, s-ASBRs have only partial topology knowledge (i.e., they know only about the active MNP-ULAs currently within their stub ASes) and they forward all other packets to c-ASBRs which have full topology knowledge.

Each s-ASBR and c-ASBR configures an ADM-ULA that is aggregable within an INET partition, and each partition configures a unique ADM-ULA prefix that is permanently announced into the routing system. The core ASes of each INET partition are joined together through external BGP peerings. The c-ASBRs of each partition establish external peerings with the c-ASBRs of other partitions to form a "core-of-cores" OMNI link AS. The OMNI link AS contains the global knowledge of all MNP-ULAs deployed worldwide, and supports ATN/IPS overlay communications between nodes located in different INET partitions by virtue of OAL encapsulation. OMNI link nodes can then navigate to ASBRs by including an ADM-ULA or directly to an end system by including an MNP-ULA in the destination address of an OAL-encapsulated packet (see: [I-D.templin-6man-aero]). Figure 3 shows a reference OAL topology.

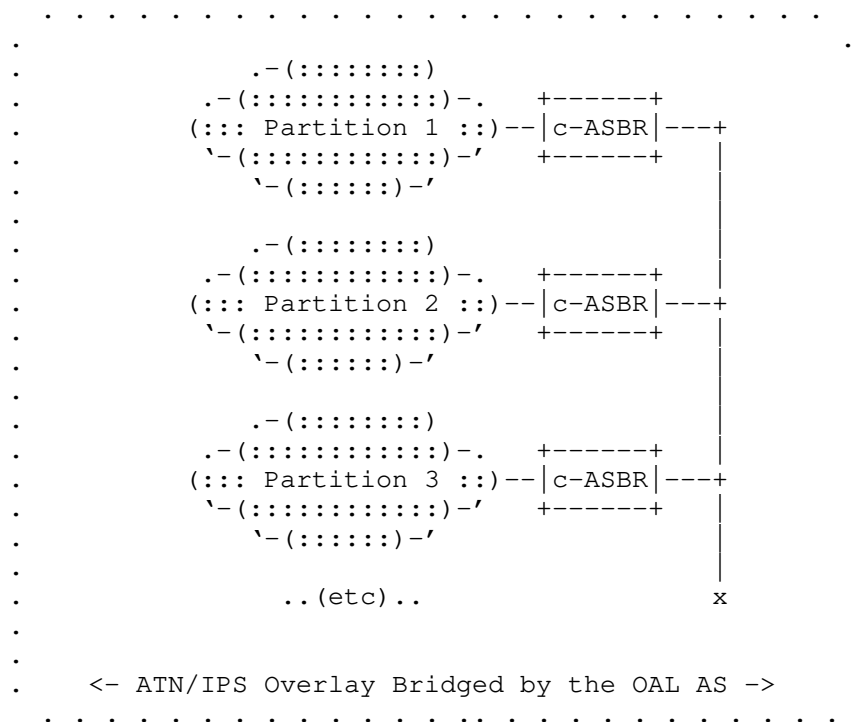


Figure 3: Spanning Partitions with the OAL

Scaling properties of this ATN/IPS routing system are limited by the number of BGP routes that can be carried by the c-ASBRs. A 2015 study showed that BGP routers in the global public Internet at that time carried more than 500K routes with linear growth and no signs of router resource exhaustion [BGP]. A more recent network emulation study also showed that a single c-ASBR can accommodate at least 1M dynamically changing BGP routes even on a lightweight virtual machine. Commercially-available high-performance dedicated router hardware can support many millions of routes.

Therefore, assuming each c-ASBR can carry 1M or more routes, this means that at least 1M ATN/IPS end system MNP-ULAs can be serviced by a single set of c-ASBRs and that number could be further increased by using RRs and/or more powerful routers. Another means of increasing scale would be to assign a different set of c-ASBRs for each set of MSPs. In that case, each s-ASBR still peers with one or more c-ASBRs from each set of c-ASBRs, but the s-ASBR institutes route filters so that it only sends BGP updates to the specific set of c-ASBRs that aggregate the MSP. In this way, each set of c-ASBRs maintains separate routing and forwarding tables so that scaling is distributed

across multiple c-ASBR sets instead of concentrated in a single c-ASBR set. For example, a first c-ASBR set could aggregate an MSP segment A::/32, a second set could aggregate B::/32, a third could aggregate C::/32, etc. The union of all MSP segments would then constitute the collective MSP(s) for the entire ATN/IPS, with potential for supporting many millions of mobile networks or more.

In this way, each set of c-ASBRs services a specific set of MSPs, and each s-ASBR configures MSP-specific routes that list the correct set of c-ASBRs as next hops. This design also allows for natural incremental deployment, and can support initial medium-scale deployments followed by dynamic deployment of additional ATN/IPS infrastructure elements without disturbing the already-deployed base. For example, a few more c-ASBRs could be added if the MNP service demand ever outgrows the initial deployment. For larger-scale applications (such as unmanned air vehicles and terrestrial vehicles) even larger scales can be accommodated by adding more c-ASBRs.

4. ATN/IPS (Radio) Access Network (ANET) Model

(Radio) Access Networks (ANETs) connect end system Clients such as aircraft, ATCs, AOCs etc. to the ATN/IPS routing system. Clients may connect to multiple ANETs at once, for example, when they have both satellite and cellular data links activated simultaneously. Clients configure an Overlay Multilink Network (OMNI) Interface [I-D.templin-6man-omni] over their underlying ANET interfaces as a connection to an NBMA virtual link (manifested by the OAL) that spans the entire ATN/IPS. Clients may further move between ANETs in a manner that is perceived as a network layer mobility event. Clients could therefore employ a multilink/mobility routing service such as those discussed in Section 7.

Clients register all of their active data link connections with their serving s-ASBRs as discussed in Section 3. Clients may connect to s-ASBRs either directly, or via a Proxy/Server at the ANET/INET boundary.

Figure 4 shows the ATN/IPS ANET model where Clients connect to ANETs via aviation data links. Clients register their ANET addresses with a nearby s-ASBR, where the registration process may be brokered by a Proxy/Server at the edge of the ANET.

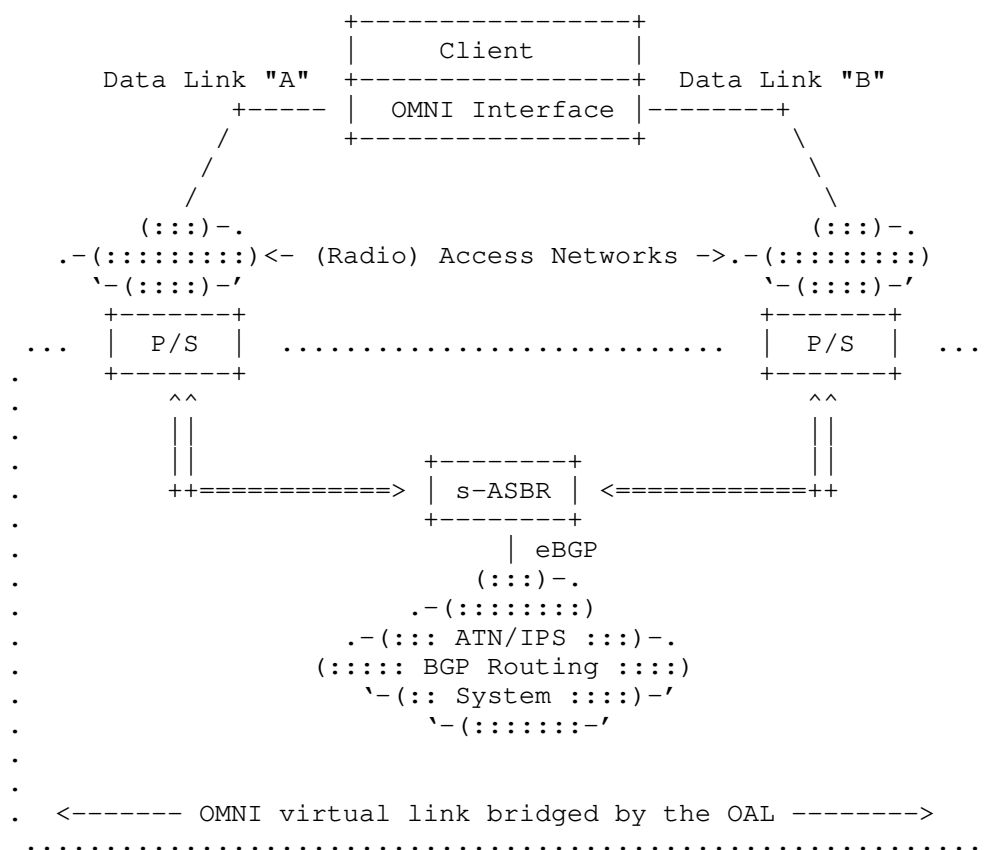


Figure 4: ATN/IPS ANET Architecture

When a Client connects to an ANET it specifies a nearby s-ASBR that it has selected to connect to the ATN/IPS. The login process is transparently brokered by a Proxy/Server at the border of the ANET which then conveys the connection request to the s-ASBR via tunneling across the OMNI virtual link. Each ANET border Proxy/Server is also equally capable of serving in the s-ASBR role so that a first on-link Proxy/Server can be selected as the s-ASBR while all others perform the Proxy/Server role in a hub-and-spokes arrangement. An on-link Proxy/Server is selected to serve the s-ASBR role when it receives a control message from a Client requesting that service.

The Client can coordinate with a network-based s-ASBR over additional ANETs after it has already coordinated with a first-hop Proxy/Server over a first ANET. If the Client connects to multiple ANETs, the s-ASBR will register the individual ANET Proxy/Servers as conduits through which the Client can be reached. The Client then sees the

s-ASBR as the "hub" in a "hub-and-spokes" arrangement with the first-hop Proxy/Servers as spokes. Selection of a network-based s-ASBR is through the discovery methods specified in relevant mobility and virtual link coordination specifications (e.g., see AERO [I-D.templin-6man-aero] and OMNI [I-D.templin-6man-omni]).

The s-ASBR represents all of its active Clients as MNP-ULA routes in the ATN/IPS BGP routing system. The s-ASBR's stub AS is therefore used only to advertise the set of MNPs of all its active Clients to its BGP peer c-ASBRs and not to peer with other s-ASBRs (i.e., the stub AS is a logical construct and not a physical one). The s-ASBR injects the MNP-ULAs of its active Clients and withdraws the MNP-ULAs of its departed Clients via BGP updates to c-ASBRs, which further propagate the MNP-ULAs to other c-ASBRs within the OAL AS. Since Clients are expected to remain associated with their current s-ASBR for extended periods, the level of MNP-ULA injections and withdrawals in the BGP routing system will be on the order of the numbers of network joins, leaves and s-ASBR handovers for aircraft operations (see: Section 6). It is important to observe that fine-grained events such as Client mobility and Quality of Service (QoS) signaling are coordinated only by Proxies and the Client's current s-ASBRs, and do not involve other ASBRs in the routing system. In this way, intradomain routing changes within the stub AS are not propagated into the rest of the ATN/IPS BGP routing system.

5. ATN/IPS Route Optimization

ATN/IPS end systems will frequently need to communicate with correspondents associated with other s-ASBRs. In the BGP peering topology discussed in Section 3, this can initially only be accommodated by including multiple extraneous hops and/or spanning tree segments in the forwarding path. In many cases, it would be desirable to establish a "short cut" around this "dogleg" route so that packets can traverse a minimum number of tunneling hops across the OMNI virtual link. ATN/IPS end systems could therefore employ a route optimization service according to the mobility service employed (see: Section 7).

Each s-ASBR provides designated routing services for only a subset of all active Clients, and instead acts as a simple Proxy/Server for other Clients. As a designated router, the s-ASBR advertises the MNPs of each of its active Clients into the ATN/IPS routing system and provides basic (unoptimized) forwarding services when necessary. An s-ASBR could be the first-hop ATN/IPS service access point for some, all or none of a Client's underlying interfaces, while the Client's other underlying interfaces employ the Proxy/Server function of other s-ASBRs. Route optimization allows Client-to-Client communications while bypassing s-ASBR designated routing services whenever possible.

A route optimization example is shown in Figure 5 and Figure 6 below. In the first figure, multiple spanning tree segments between Proxy/Servers and ASBRs are necessary to convey packets between Clients associated with different s-ASBRs. In the second figure, the optimized route tunnels packets directly between Proxy/Servers without involving the ASBRs.

These route optimized paths are established through secured control plane messaging (i.e., over secured tunnels and/or using higher-layer control message authentications) but do not provide lower-layer security for the data plane. Data communications over these route optimized paths should therefore employ higher-layer security.

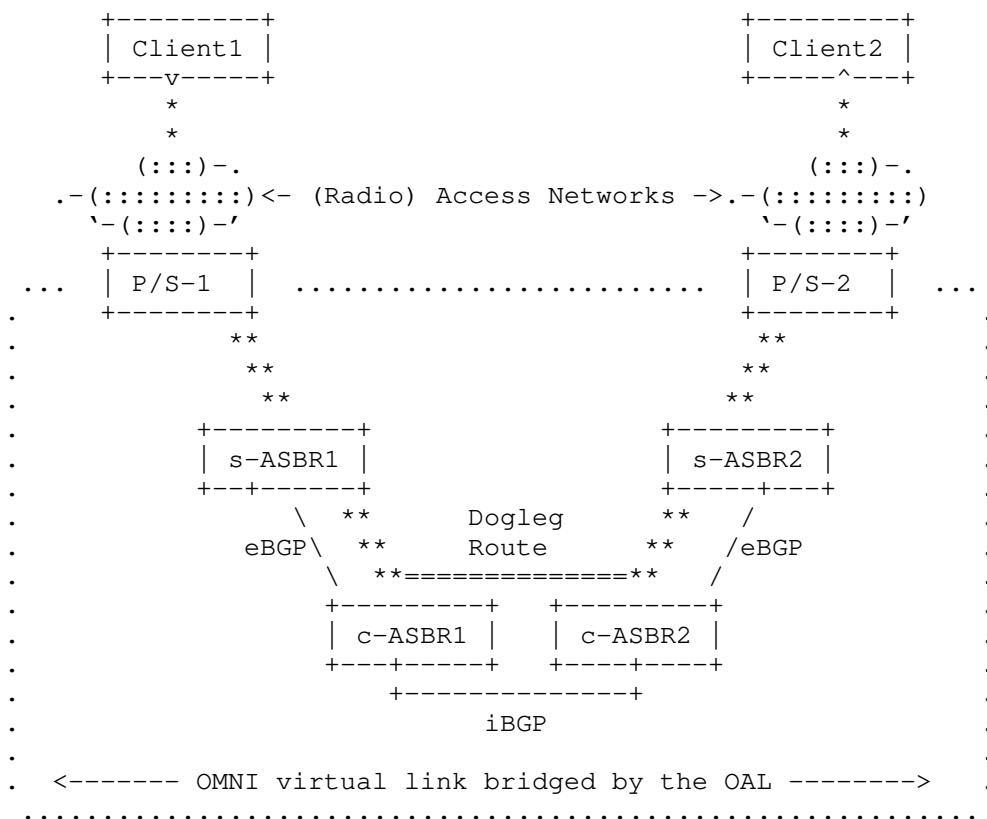


Figure 5: Dogleg Route Before Optimization

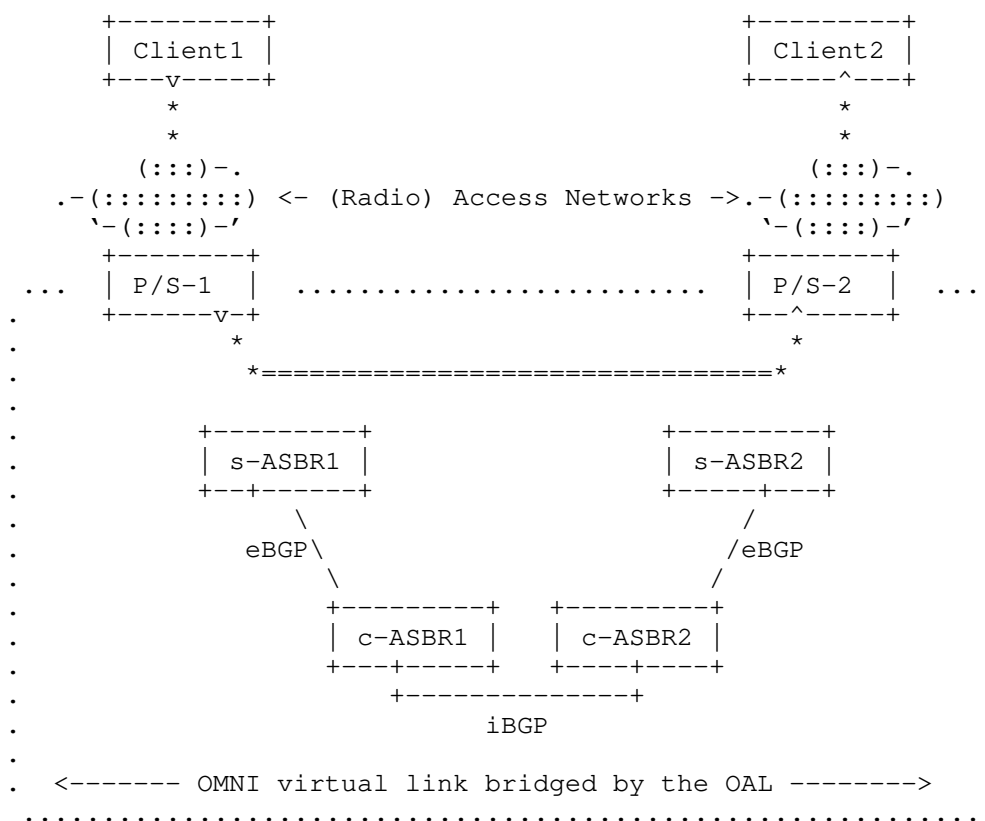


Figure 6: Optimized Route

6. BGP Protocol Considerations

The number of eBGP peering sessions that each c-ASBR must service is proportional to the number of s-ASBRs in its local partition. Network emulations with lightweight virtual machines have shown that a single c-ASBR can service at least 100 eBGP peerings from s-ASBRs that each advertise 10K MNP-ULA routes (i.e., 1M total). It is expected that robust c-ASBRs can service many more peerings than this - possibly by multiple orders of magnitude. But even assuming a conservative limit, the number of s-ASBRs could be increased by also increasing the number of c-ASBRs. Since c-ASBRs also peer with each other using iBGP, however, larger-scale c-ASBR deployments may need to employ an adjunct facility such as BGP Route Reflectors (RRs) [RFC4456].

The number of aircraft in operation at a given time worldwide is likely to be significantly less than 1M, but we will assume this number for a worst-case analysis. Assuming a worst-case average 1 hour flight profile from gate-to-gate with 10 service region transitions per flight, the entire system will need to service at most 10M BGP updates per hour (2778 updates per second). This number is within the realm of the peak BGP update messaging seen in the global public Internet today [BGP2]. Assuming a BGP update message size of 100 bytes (800bits), the total amount of BGP control message traffic to a single c-ASBR will be less than 2.5Mbps which is a nominal rate for modern data links.

Industry standard BGP routers provide configurable parameters with conservative default values. For example, the default hold time is 90 seconds, the default keepalive time is 1/3 of the hold time, and the default MinRouteAdvertisementInterval is 30 seconds for eBGP peers and 5 seconds for iBGP peers (see Section 10 of [RFC4271]). For the simple mobile routing system described herein, these parameters can be set to more aggressive values to support faster neighbor/link failure detection and faster routing protocol convergence times. For example, a hold time of 3 seconds and a MinRouteAdvertisementInterval of 0 seconds for both iBGP and eBGP.

Instead of adjusting BGP default time values, BGP routers can use the Bidirectional Forwarding Detection (BFD) protocol [RFC5880] to quickly detect link failures that don't result in interface state changes, BGP peer failures, and administrative state changes. BFD is important in environments where rapid response to failures is required for routing reconvergence and, hence, communications continuity.

Each c-ASBR will be using eBGP both in the ATN/IPS and the INET with the ATN/IPS unicast IPv6 routes resolving over INET routes. Consequently, c-ASBRs and potentially s-ASBRs will need to support separate local ASes for the two BGP routing domains and routing policy or assure routes are not propagated between the two BGP routing domains. From a conceptual, operational and correctness standpoint, the implementation should provide isolation between the two BGP routing domains (e.g., separate BGP instances).

ADM-ULAs and MNP-ULAs begin with fd00::/8 followed by a pseudo-random 40-bit global ID to form the prefix [ULA]::/48, along with a 16-bit Subnet ID '*' to form the prefix [ULA*]::/64. Each individual address taken from [ULA*]::/64 includes additional routing information in the interface identifier. For example, for the MNP 2001:db8:1:0::/56, the resulting MNP-ULA is [ULA*]:2001:db8:1:0/120, and for the administrative address 1001:2002 the ADM-ULA is [ULA*]:1001:2002/64 (see: [I-D.templin-6man-omni] for further

details). However, MNP-ULA prefixes installed in the BGP routing system always set the Global ID and Subnet ID to 0 (i.e., the "wildcard" subnet) since OMNI link forwarding decisions are based solely on the MNP found in the interface identifier independently of the Global/Subnet IDs.

This gives rise to a BGP routing system that must accommodate large numbers of long and non-aggregable MNP-ULA prefixes as well as moderate numbers of long and semi-aggregable ADM-ULA prefixes. The system is kept stable and scalable through the s-ASBR / c-ASBR hub-and-spokes topology which ensures that mobility-related churn is not exposed to the core.

7. Stub AS Mobile Routing Services

Stub ASes maintain intradomain routing information for mobile node clients, and are responsible for all localized mobility signaling without disturbing the BGP routing system. Clients can enlist the services of a candidate mobility service such as Mobile IPv6 (MIPv6) [RFC6275], LISP [I-D.ietf-lisp-rfc6830bis] or AERO [I-D.templin-6man-aero] according to the service offered by the stub AS. Further details of mobile routing services are out of scope for this document.

8. Implementation Status

The BGP routing topology described in this document has been modeled in realistic network emulations showing that at least 1 million MNP-ULAs can be propagated to each c-ASBR even on lightweight virtual machines. No BGP routing protocol extensions need to be adopted.

9. IANA Considerations

This document does not introduce any IANA considerations.

10. Security Considerations

ATN/IPS ASBRs on the open Internet are susceptible to the same attack profiles as for any Internet nodes. For this reason, ASBRs should employ physical security and/or IP securing mechanisms such as IPsec [RFC4301], WireGuard [WG], etc.

ATN/IPS ASBRs present targets for Distributed Denial of Service (DDoS) attacks. This concern is no different than for any node on the open Internet, where attackers could send spoofed packets to the node at high data rates. This can be mitigated by connecting ATN/IPS ASBRs over dedicated links with no connections to the Internet and/or when ASBR connections to the Internet are only permitted through well-managed firewalls.

ATN/IPS s-ASBRs should institute rate limits to protect low data rate aviation data links from receiving DDoS packet floods.

BGP protocol message exchanges and control message exchanges used for route optimization must be secured to ensure the integrity of the system-wide routing information base. Security is based on IP layer security associations between peers which ensure confidentiality, integrity and authentication over secured tunnels (see above). Higher layer security protection such as TCP-AO [RFC5926] is therefore optional, since it would be redundant with the security provided at lower layers.

Data communications over route optimized paths should employ end-to-end higher-layer security since only the control plane and unoptimized paths are protected by lower-layer security. End-to-end higher-layer security mechanisms include QUIC-TLS [RFC9001], TLS [RFC8446], DTLS [RFC6347], SSH [RFC4251], etc. applied in a manner outside the scope of this document.

This document does not include any new specific requirements for mitigation of DDoS.

10.1. Public Key Infrastructure (PKI) Considerations

In development of the overall ATN/IPS operational concept, ICAO addressed the security concerns in multiple ways to ensure coordination and consistency across the various groups. This also avoided potential duplicative work. Technical provisions related specifically to the operation of ATN/IPS are specified in supporting ATN/IPS standards. However, other considerations such as the establishment of a PKI, were determined to have an impact beyond ATN/IPS. ICAO created a Trust Framework Study Group (TFSG) to define various governance, policy, procedures and overall technical performance requirements for system connectivity and interoperability.

As part of their charter, the TSFG is specifically developing a concept of operations for a common aviation digital trust framework and principles to facilitate an interoperable secure, cyber resilient and seamless exchange of information in a digitally connected

environment. They are also developing governance principles, policy, procedures and requirements for establishing digital identity for a global trust framework that will consider any exchange of information among users of the aviation ecosystem, and to promote these concepts with all relevant stakeholders.

ATN/IPS will take advantage of the developments of TFSG within the overall ATN/IPS operational concept. As such, this will include the usage of the PKI specification resulting from the TFSG.

11. Acknowledgements

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the Boeing Commercial Airplanes (BCA) Internet of Things (IoT) and autonomy programs.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

The following individuals contributed insights that have improved the document: Ahmad Amin, Mach Chen, Russ Housley, Erik Kline, Hubert Kuenig, Tony Li, Gyan Mishra, Alexandre Petrescu, Dave Thaler, Pascal Thubert, Michael Tuxen, Tony Whyman.

12. References

12.1. Normative References

- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [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>>.

12.2. Informative References

- [ATN] Maiolla, V., "The OMNI Interface - An IPv6 Air/Ground Interface for Civil Aviation, IETF Liaison Statement #1676, <https://datatracker.ietf.org/liaison/1676/>", 3 March 2020.
- [ATN-IPS] WG-I, ICAO., "ICAO Document 9896 (Manual on the Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocol), Draft Edition 3 (work-in-progress)", 10 December 2020.
- [BGP] Huston, G., "BGP in 2015, <http://potaroo.net>", January 2016.
- [BGP2] Huston, G., "BGP Instability Report, <http://bgpupdates.potaroo.net/instability/bgpupd.html>", May 2017.
- [CBB] Dul, A., "Global IP Network Mobility using Border Gateway Protocol (BGP), http://www.quark.net/docs/Global_IP_Network_Mobility_using_BGP.pdf", March 2006.

- [I-D.ietf-lisp-rfc6830bis]
Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-36, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-36.txt>>.
- [I-D.templin-6man-aero]
Templin, F. L., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero-42, 9 April 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-aero-42.txt>>.
- [I-D.templin-6man-omni]
Templin, F. L., "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", Work in Progress, Internet-Draft, draft-templin-6man-omni-57, 9 April 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-omni-57.txt>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <<https://www.rfc-editor.org/info/rfc4251>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, DOI 10.17487/RFC5926, June 2010, <<https://www.rfc-editor.org/info/rfc5926>>.
- [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>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.

- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <<https://www.rfc-editor.org/info/rfc6793>>.
- [RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <<https://www.rfc-editor.org/info/rfc6996>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <<https://www.rfc-editor.org/info/rfc9001>>.
- [WG] Donenfeld, J., "WireGuard: Fast, Modern, Secure VPN Tunnel, <https://www.wireguard.com/>", February 2022.

Appendix A. BGP Convergence Considerations

Experimental evidence has shown that BGP convergence time required after an MNP-ULA is asserted at a new location or withdrawn from an old location can be several hundred milliseconds even under optimal AS peering arrangements. This means that packets in flight destined to an MNP-ULA route that has recently been changed can be (mis)delivered to an old s-ASBR after a Client has moved to a new s-ASBR.

To address this issue, the old s-ASBR can maintain temporary state for a "departed" Client that includes an OAL address for the new s-ASBR. The OAL address never changes since ASBRs are fixed infrastructure elements that never move. Hence, packets arriving at the old s-ASBR can be forwarded to the new s-ASBR while the BGP routing system is still undergoing reconvergence. Therefore, as long as the Client associates with the new s-ASBR before it departs from the old s-ASBR (while informing the old s-ASBR of its new location) packets in flight during the BGP reconvergence window are accommodated without loss.

Appendix B. Change Log

<< RFC Editor - remove prior to publication >>

Differences from earlier versions:

* Submit for RFC publication.

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: fltemplin@acm.org

Greg Saccone
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: gregory.t.saccone@boeing.com

Gaurav Dawra
LinkedIn
United States of America
Email: gdawra.ietf@gmail.com

Acee Lindem
Cisco Systems, Inc.
United States of America
Email: acee@cisco.com

Victor Moreno
Cisco Systems, Inc.
United States of America
Email: vimoreno@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 2, 2019

F. Templin, Ed.
G. Saccone
Boeing Research & Technology
G. Dawra
LinkedIn
A. Lindem
V. Moreno
Cisco Systems, Inc.
January 29, 2019

Scalable De-Aggregation for Overlays Using the Border Gateway Protocol
(BGP)

draft-templin-rtgwg-scalable-bgp-01.txt

Abstract

The Border Gateway Protocol (BGP) has well-known limitations in terms of the numbers of routes that can be carried and stability of the routing system. This is especially true when mobile nodes frequently change their network attachment points, which in the past has resulted in excessive announcements and withdrawals of de-aggregated prefixes. This document discusses a means of accommodating scalable de-aggregation of IPv6 prefixes for overlay networks using BGP.

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 August 2, 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. Overview and Analysis	2
3. Opportunities and Limitations	4
4. Use Cases	4
5. Implementation Status	4
6. IANA Considerations	5
7. Security Considerations	5
8. Acknowledgements	5
9. References	5
9.1. Normative References	5
9.2. Informative References	5
Appendix A. Change Log	6
Authors' Addresses	6

1. Introduction

The Border Gateway Protocol (BGP) [RFC4271] has well-known limitations in terms of the numbers of routes that can be carried and the stability of the routing system. This is especially true for routing systems that include mobile nodes that frequently change their network attachment points, which in the past have resulted in excessive announcements and withdrawals of de-aggregated prefixes. This document discusses a means of accommodating scalable de-aggregation of IPv6 prefixes [RFC8200] for overlay networks using BGP.

2. Overview and Analysis

As discussed in [I-D.ietf-rtgwg-atn-bgp] and [I-D.templin-intarea-6706bis], the method for accommodating de-aggregation is to institute an overlay network instance of BGP that is separate and independent from the global Internet BGP routing system. The overlay is presented to the global Internet as a small number of aggregated IPv6 prefixes (also known as Mobility Service Prefixes (MSPs)) that never change. In this way, the Internet BGP routing system sees only stable aggregated MSPs (e.g., 2001:db8::/32)

and is completely unaware of any de-aggregation or mobility-related churn that may be occurring within the overlay.

The overlay is operated by an Overlay Service Provider (OSP), and consists of a core Autonomous System (AS) with core AS Border Routers (c-ASBRs) that connect to stub ASes with stub ASBRs (s-ASBRs) in a hub-and-spokes fashion. Mobile nodes associate with nearby (i.e., regional) stub ASes for extended timeframes, and change to new stub ASes only after movements of significant topological or geographical distance. Mobility-related changes between stub ASes are therefore normally infrequent.

The s-ASBRs use eBGP to announce de-aggregated Mobile Network Prefixes (MNP) of mobile nodes (e.g., 2001:db8:1:2::/64, etc.) to their neighboring c-ASBRs, but do not announce fine-grained mobility events such as a mobile node moving to a new network attachment point. Instead, mobile nodes coordinate with stub ASes using mobility protocols such as MIPv6, LISP, AERO, etc. and stub ASes accommodate these localized mobility events without disturbing the c-ASBRs.

The c-ASBRs originate "default" to their neighboring s-ASBRs but do not announce any MNP routes. In this way, MNP announcements and withdrawals are unidirectional from s-ASBRs to c-ASBRs only, thereby suppressing BGP updates on the reverse path. The c-ASBRs in turn use iBGP to maintain a consistent view of the full topology. BGP Route Reflectors (RRs) [RFC4456] can also be used to support increased c-ASBR scaling.

Each c-ASBR should be able to carry at least as many routes as a typical core router in the global public Internet BGP routing system. Since the number of active routes in the Internet is rapidly approaching 1 million (1M), viable c-ASBRs must be capable of carrying at least 1M MNP routes (this has been proven even for BGP running on lightweight virtual machines). The method for increasing scaling therefore is to divide the MSP into longer sub-MSPs, and to assign a different set of c-ASBRs for each sub-MSP.

For example, the MSP 2001:db8::/32 could be sub-divided into sub-MSPs such as 2001:db8:0010::/44, 2001:db8:0020::/44, 2001:db8:0030::/44, etc. with each sub-MSP assigned to a different set of c-ASBRs. Each s-ASBR peers with at least one member of each c-ASBR set and uses route filters such that BGP updates are only sent to the c-ASBR(s) that aggregate the specific sub-MSP. Then, assuming 1 thousand (1K) or more sub-MSPs (each with its own set of c-ASBRs) the entire BGP overlay routing system should be able to service 1 billion (1B) MNPs or more.

3. Opportunities and Limitations

Since a lightweight virtual machine (e.g., a linux image running quagga in the cloud) can service up to 1M MNPs using BGP, it is likely that dedicated high-performance IPv6 router hardware could support even more. With such dedicated high-performance hardware, the number of MNPs could be increased further.

The deployed numbers of s-ASBRs even for very large overlays should not exceed a c-ASBR's capacity for BGP peering sessions. For example, c-ASBRs should be capable of servicing 1K or more BGP peering sessions, with the upper bound limited by keepalive and update control messaging overhead. Conversely, s-ASBRs should be capable of supporting even more sessions since they only receive keepalives and only send updates for mobile nodes within their local stub ASes.

Mobile nodes should refrain from moving rapidly between stub ASes for no good reason, since the objective is only to reduce routing stretch due to movement of significant distances. OSPs could employ disincentives such as surcharge penalties for gratuitous mobility, but intentional abuse would also yield little reward since only the bad actor (i.e., and not others) would be subject to MNP instability.

Packets sent between mobile nodes that associate with different stub ASes would initially need to be forwarded through the core AS, which presents a forwarding bottleneck. For this reason, a route optimization function is needed to reduce congestion in the core. Since c-ASBRs should be commercial off-the-shelf (COTS) dedicated high-performance IPv6 routers, however, they should not be required to participate directly in any out-of-band route optimization signaling. Instead, route optimization should be coordinated by stub AS network elements and/or the mobile nodes themselves.

4. Use Cases

Use cases include Unmanned Air Systems (UAS) in controlled and uncontrolled airspaces, Intelligent Transportation Systems (ITS) in urban air/ground mobility environments, aviation networks, enterprise mobile device users, and cellular network users. Any other use cases in which an OSP services large numbers of mobile nodes are also in scope.

5. Implementation Status

The arrangement of stub and core ASes described in this document has been implemented using standards-compliant linux operating systems and BGP routing protocol implementations (i.e., quagga). No new code

was included, and all requirements were satisfied through standard configuration options.

6. IANA Considerations

This document does not introduce any IANA considerations.

7. Security Considerations

Security considerations are discussed in the references.

8. Acknowledgements

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

9. References

9.1. Normative References

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <<https://www.rfc-editor.org/info/rfc4456>>.
- [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>>.

9.2. Informative References

- [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-01 (work in progress), January 2019.

[I-D.templin-intarea-6706bis]

Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-intarea-6706bis-03 (work in progress), December 2018.

Appendix A. Change Log

<< RFC Editor - remove prior to publication >>

Changes from -00 to -01:

- o added Route Reflectors
- o introduced term "Overlay Service Provider (OSP)"
- o removed estimate of number of routes for high-performance routers
- o revised text on route optimization
- o added use case and implementation sections

Status as of 01/23/2018:

- o -00 draft published

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Greg Saccone
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: gregory.t.saccone@boeing.com

Gaurav Dawra
LinkedIn
USA

Email: gdawra.ietf@gmail.com

Acee Lindem
Cisco Systems, Inc.
USA

Email: acee@cisco.com

Victor Moreno
Cisco Systems, Inc.
USA

Email: vimoreno@cisco.com