

DMM  
Internet-Draft  
Intended status: Informational  
Expires: September 20, 2016

H. Chan  
X. Wei  
Huawei Technologies  
J. Lee  
Sangmyung University  
S. Jeon  
Instituto de Telecomunicacoes  
F. Templin  
Boeing Research and Technology  
March 19, 2016

Distributed Mobility Anchoring  
draft-chan-dmm-distributed-mobility-anchoring-07

Abstract

This document defines distributed mobility anchoring. Multiple anchors and nodes are configured with appropriate mobility functions and work together to enable mobility solutions. Example solution is mid-session switching of the IP prefix anchor. Without ongoing session requiring session continuity, a flow can be started or re-started using the new IP prefix which is allocated from the new network and is therefore anchored to the new network. With ongoing session, the anchoring of the prior IP prefix may be relocated to the new network to enable session continuity.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 20, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Conventions and Terminology . . . . .	3
3. Distributed anchoring . . . . .	5
3.1. Distributed anchoring configurations . . . . .	5
3.2. Distributed anchoring behaviors and message information elements . . . . .	8
3.2.1. Location management behaviors and message information elements . . . . .	8
3.2.2. Forwarding management behaviors and message information elements . . . . .	9
4. Example mobility solutions with distributed anchoring . . . . .	11
4.1. IP mobility support only when needed . . . . .	11
4.1.1. Not needed: Changing to the new IP prefix/address . . . . .	12
4.1.2. Needed: Providing IP mobility support . . . . .	13
4.2. IP prefix/address anchor switching to the new network . . . . .	15
4.2.1. Centralized control plane . . . . .	16
4.2.2. Hierarchical network . . . . .	19
4.2.3. Hierarchical network with anchoring change . . . . .	21
5. Security Considerations . . . . .	22
6. IANA Considerations . . . . .	23
7. Contributors . . . . .	23
8. References . . . . .	23
8.1. Normative References . . . . .	23
8.2. Informative References . . . . .	25
Authors' Addresses . . . . .	25

## 1. Introduction

A key requirement in distributed mobility management [RFC7333] is to enable traffic to avoid traversing single mobility anchor far from the optimal route. Distributed mobility management solutions do not

make use of centrally deployed mobility anchor [Paper-Distributed.Mobility]. As such, the traffic of a flow SHOULD be able to change from traversing one mobility anchor to traversing another mobility anchor as the mobile node moves, or when changing operation and management requirements call for mobility anchor switching, thus avoiding non-optimal routes. This draft proposes distributed mobility anchoring to enable making such route changes.

Distributed mobility anchoring employs multiple anchors in the data plane. In general, the control plane function may co-located with the data plane function at these distributed anchors but may also be separate from the data plane functions and be centralized. Different configurations (Section 3.1) of distributed anchoring are then possible. Yet the distributed anchors need to have expected behaviors (Section 3.2).

A mobile node (MN) attached to an access router of a network may be allocated an IP prefix which is anchored to that router. It may then use the IP address configured from this prefix as the source IP address to run a flow with its correspondent node (CN). When there are multiple anchors, the flow may need to select the anchor when it is initiated (Section 4). Using an anchor in MN's network of attachment has the advantage that the packets can simply be forwarded according to the forwarding table. Although the anchor is in the MN's network of attachment when the flow was initiated, the MN may later move another network, so that the IP address no longer belongs to the new network of attachment of the MN. Whether the flow needs session continuity will determine how to ensure that the IP address of the flow will be anchored to the new network of attachment. 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 (Section 4.1.1). On the other hand, if the ongoing IP flow cannot cope with such change, the IP address anchoring can be relocated from the original network to the new network (Section 4.2).

## 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].

All general mobility-related terms and their acronyms used in this document are to be interpreted as defined in the Mobile IPv6 base specification [RFC6275], the Proxy Mobile IPv6 specification [RFC5213], and the DMM current practices and gap analysis [RFC7429]. This includes 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 term:

Home network of an application session (or of an HoA): the network that has allocated the IP address (HoA) used for the session identifier by the application running in an MN. An MN may be running multiple application sessions, and each of these sessions can have a different home network.

IP prefix/address anchoring: An IP prefix, i.e., Home Network Prefix (HNP), or address, i.e., Home Address (HoA), allocated to a mobile node is topologically anchored to a node when the anchor node is able to advertise a connected route into the routing infrastructure for the allocated IP prefix.

Internetwork Location Management (LM) function: managing and keeping track of the internetwork location of an MN. The location information may be a binding of the IP advertised 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. It is a control plane function.

In a client-server protocol model, location query and update messages may be exchanged between a Location Management client (LMc) and a Location Management server (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 to 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 indirection. With separation of control plane and data plane, FM 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.

Security Management (SM) function: The security management function controls security mechanisms/protocols providing access control, integrity, authentication, authorization, confidentiality, etc. for the control plane and data plane.

This function resides in all nodes such as control plane anchor, data plane anchor, mobile node, and correspondent node.

### 3. Distributed anchoring

#### 3.1. Distributed anchoring configurations

The mobility functions may be implemented in different configurations of distributed anchoring. Some of these configurations are described in [I-D.siyeon-dmm-deployment-models].

Figure 1 shows 4 configurations. In each configuration, an MN is allocated an IP prefix/address IP1 and is using IP1 to communicate with a correspondent node (CN) not shown in the figure. The flow of this communication session is shown as flow(IP1, ...) which uses IP1 and other parameters. The IP1 is anchored to the data plane anchor (DPA) which has IP prefix/address IPa1. The data plane is distributed so that there may be multiple instances of the DPA (not shown). The control plane may either be distributed or centralized. When the control plane anchor (CPA) co-locates with the distributed DPA there will be multiple instances of the co-located CPA and DPA (not shown).

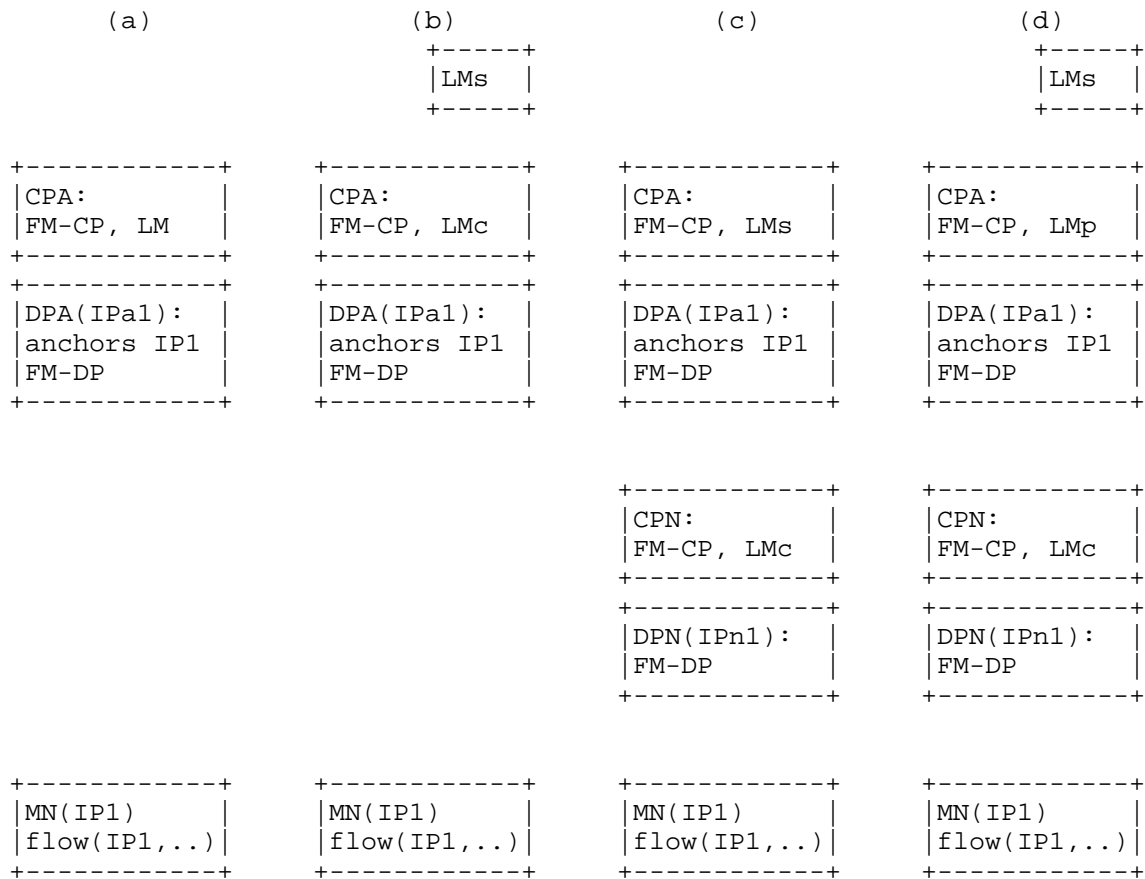


Figure 1. (a) FM-CP and LM at CPA, FM-DP at DPA; (b) Separate LMs, FM-CP and LMc at CPA, FM-DP at DPA; (c) FM-CP and LMs at CPA, FM-DP at DPA, FM-CP and LMc at CPN, FM-DP at DPN; (d) Separate LMs, FM-CP and LMp at CPA, FM-DP at DPA, FM-CP and LMc at CPN, FM-DP at DPN.

In Figure 1(a), both LM and FM co-locate at the anchor. FM-DP is at the DPA whereas LM and FM-CP are at the CPA. Then LM may be distributed or centralized according to whether the CPA is distributed or centralized.

Figure 1(b) differs from Figure 1(a) in that the LM function is split into a server LMs and a client LMc. FM-DP is at the DPA whereas LMc and FM-CP are at the CPA. The LMs may be centralized whereas the LMc may be distributed or centralized according to whether the CPA is distributed or centralized.

In Figure 1(c), FM-DP is at DPA whereas LMs and FM-CP are at the CPA. In addition, there is also FM-DP at a data plane node (DPN), and there are also FM-CP together with LMc at a control plane node (CPN). In the hierarchy, there may be multiple DPN's for each DPA. Again, LMs may be distributed or centralized according to whether the CPA is distributed or centralized. The DPA may co-locate with CPA or may be separated. When separation of data plane and control plane, DPA may be distributed when CPA is centralized.

Figure 1(d) differs from Figure 1(c) in that the LMs is separated out, and a proxy LMP is added between the LMs and LMc. FM-DP is at the DPA whereas LMP and FM-CP are at the CPA. Again, there is also FM-DP at a data plane node (DPN), and there are also FM-CP together with LMc at a control plane node (CPN). The FMs may be centralized whereas the LMP may be distributed or centralized according to whether the CPA is distributed or centralized.

A host-based variant of the mobility function configuration from Figure 1(c) and 1(d) is shown in Figure 2(a) and 2(b).

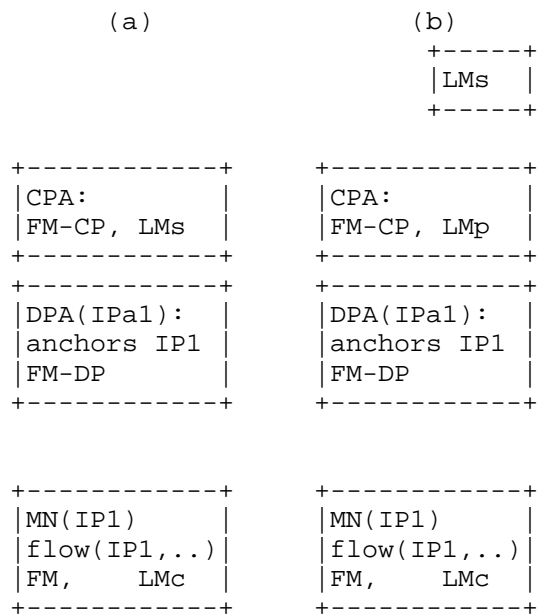


Figure 2. (a) FM-CP and LMs at CPA, FM-DP at DPA, FM and LMc at MN; (b) Separate LMs, FM-CP and LMP at CPA, FM-DP at DPA, FM and LMc at MN.

In Figure 2(a), FM-DP is at DPA whereas LMs and FM-CP are at the CPA. In addition, there is FM and LMc at the MN. The LMs may be

distributed or centralized according to whether the CPA is distributed or centralized.

Figure 2(b) differs from Figure 2(a) in that the LMs is separated out and a proxy is added between the LMs and LMc. FM-DP is at the DPA whereas LMp and FM-CP are at the CPA. In addition, there is FM and LMc at the MN. The FMs may be centralized whereas the LMp may be distributed or centralized according to whether the CPA is distributed or centralized.

### 3.2. Distributed anchoring behaviors and message information elements

The behaviors of distributed anchoring are defined in this section in order that they may work together in expected manners to produce a distributed mobility solution. The needed information elements are passed as message parameters.

#### 3.2.1. Location management behaviors and message information elements

It is seen in (Section 3.1) that

- (1) LMs may be a separate server or may co-locate with CPA;
- (2) LMc may be at CPA, CPN, or MN.

Example LM design may consists of a distributed database of LMs servers in a pool of distributed servers. The prefix of a MN is hosted at a given LMs as the primary location information for this prefix. Peer LMs may exchange the location information with each other. LMc may retrieve a given record or send a given update record to LMs.

Location information behaviors:

- (LM:1) LMc queries LMs about location information for a prefix of MN (pull).  
Parameters:  
IP prefix of MN.
- (LM:2) LMs replies to LMc query about location information for a prefix of MN (pull).  
Parameters:  
IP prefix of MN,  
IP address of FM-DP/DPA/DPN to forward the packets of the flow.
- (LM:3) LMs informs LMc about location information for a prefix of MN (push).  
Parameters:



IP prefix of MN,  
IP address of FM-DP/DPA/DPN to forward the packets of the  
flow.

- (LM:4) LMs joins a LMs pool.  
Parameters:  
IP address of the LMs,  
IP prefixes for which the LMs will host the primary location  
information.
- (LM:5) LMs queries a peer LMs about location information for a  
prefix of MN.  
Parameters:  
IP prefix.
- (LM:6) LMs replies to a peer LMs about location information for a  
prefix of MN (push).  
Parameters:  
IP prefix of MN,  
IP address of FM-DP/DPA/DPN to forward the packets of the  
flow.

### 3.2.2. Forwarding management behaviors and message information elements

It is seen in (Section 3.1) that

- (1) FM-CP may be at CPA, CPN, MN;
- (2) FM-DP may be at DPA, DPN, MN.

The FM behaviors and message information elements are:

- (FM:1) An anchor acts on packets on a per flow basis and performs  
the changes to the forwarding path upon a change of point of  
attachment of a MN:
- (FM:1-1) FM filters the packets up to the granularity of a  
flow.  
Example matching parameters are the 5-tuple of a  
flow.
  - (FM:1-2) FM makes the necessary changes to the forwarding  
path of a flow.  
Example mechanism is through forwarding table  
update activated by DHCPv6-PD.
  - (FM:1-3) FM reverts the previously made changes to the  
forwarding path of a flow when such changes are no

longer needed, e.g., when an ongoing flow requiring session continuity has closed.

Example mechanism is through expiration of DHCPv6-PD.

(FM:2) An anchor may discover and be discovered such as through an anchor registration system:

(FM:2-1) FM registers and authenticates itself with a centralized mobility controller.

Parameters:

IP address of DPA and its CPA;

IP prefix anchored to the DPA.

(FM:2-2) registration reply: acknowledge of registration and echo the input parameters.

(FM:2-3) FM discovers the FM of another IP prefix by querying the mobility controller based on the IP prefix.

Parameters:

IP prefix of MN.

(FM:2-4) when making anchor discovery FM expects the answer parameters as: IP address of DPA to which IP prefix of MN is anchored; IP prefix of the corresponding CPA.

(FM:3) With separation of control plane function and data plane function, these function must work together.

(FM:3-1) CPA/FM-CP sends forwarding table updates to DPA/FM-DP.

Parameters:

new forwarding table entries to add;

expired forwarding table entries to delete.

(FM:3-2) DPA/FM-DP sends to CPA/FM-CP about its status and load.

Parameters:

state of forwarding function being active or not;

loading percentage.

(FM:4) An anchor can buffer packets of a flow in a mobility event:

(FM:4-1) CPA/FM-CP informs DPA/FM-DP to buffer packets of a flow.

Trigger:

MN leaves DPA in a mobility event.

Parameters:

IP prefix of the flow for which packets need to be buffered.

(FM:4-2) CPA/FM-CP on behalf of a new DPA/FM-DP informs CPA/FM-CP of the prior DPA/FM-DP that it is ready to receive any buffered packets of a flow.

Parameters:

destination IP prefix of the flow's packets;

IP address of the new DPA.

#### 4. Example mobility solutions with distributed anchoring

The IP prefix/address at the MN's side of a flow may be anchored at the access router to which the MN is attached. For example, when an MN attaches to a network (Net1) or moves to a new network (Net2), it is allocated an IP prefix from that network. It configures from this prefix an IP address which is typically a dynamic IP address. It then uses this IP address when a flow is initiated. Packets to the MN in this flow are simply forwarded according to the forwarding table.

There may be multiple IP prefixes/addresses to choose from. 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, these IP prefixes/addresses may be of different types regarding whether mobility support is needed [I-D.dmm-ondemand-mobility-api]. A flow will need to choose the appropriate one according to whether it needs IP mobility support.

##### 4.1. IP mobility support only when needed

IP mobility support may be provided only when needed instead of being provided by default. The simplest configuration in this case is shown in Figures 1(a) and 1(b) in Section 3.1 for which the LM and FM functions are utilized only when needed.

A straightforward choice of mobility anchoring is for a flow to use the IP prefix of the network to which the MN is attached when the flow is initiated [I-D.seite-dmm-dma].

#### 4.1.1.1. Not needed: Changing to the 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 configuration from Figures 1(a) and 1(b) in Section 3.1 simplifies to that shown in Figure 3.



Figure 3. Changing to the new IP prefix/address. MN running a flow using IP1 in Net1 changes to running a flow using IP2 in Net2.

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 terminated before the MN moves to a new network, the flow may subsequently restart using the new IP address allocated from the new network.

When session continuity is needed, even if a flow is ongoing as the MN moves, it may still be desirable for the flow to change to using the new IP prefix configured in the new network. The flow may then close and then restart using a new IP address configured in the new network. Such a change in the IP address of the 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 in Net1 has terminated before the MN moves to a new network Net2. After moving to Net2, the MN uses the new IP prefix anchored in Net2 to start a new flow. The packets may then be forwarded without requiring IP layer mobility support.

The call flow is outlined in Figure 4.

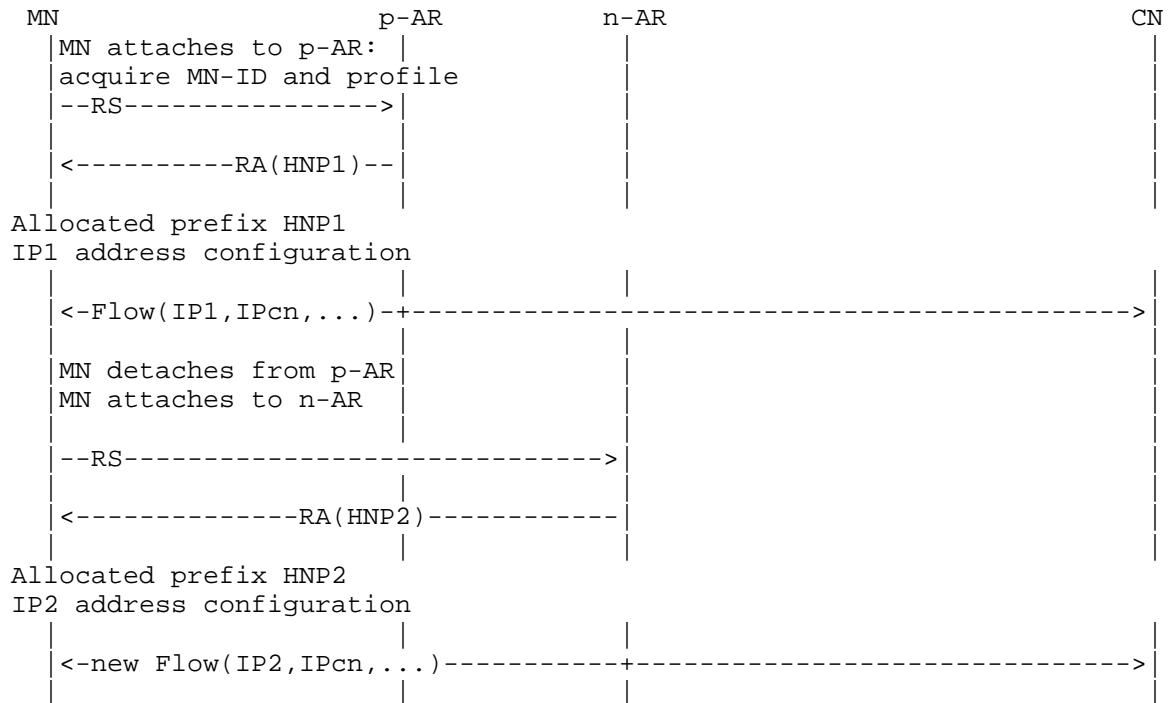


Figure 4. A flow uses the IP allocated from the network at which the MN is attached when the flow is initiated.

The security management function in the anchor node at a new network must allow to assign a valid IP prefix/address to a mobile node.

#### 4.1.2. Needed: Providing IP mobility support

When IP mobility is needed for a flow, the LM and FM functions in Figures 1(a) and 1(b) in Section 3.1 are utilized. The mobility support may be provided by IP prefix anchor switching to the new network to be described in Section 4.2 or by using other mobility management methods ([Paper-Distributed.Mobility.PMIP] and [Paper-Distributed.Mobility.Review]). Then the flow may continue to use the IP prefix from the prior network. Yet some time later, the user application for the 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 permanent IP prefix/address is not used. The flow may then use the new IP prefix 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 has not moved away from that network.

The call flow in this case is outlined in Figure 5.

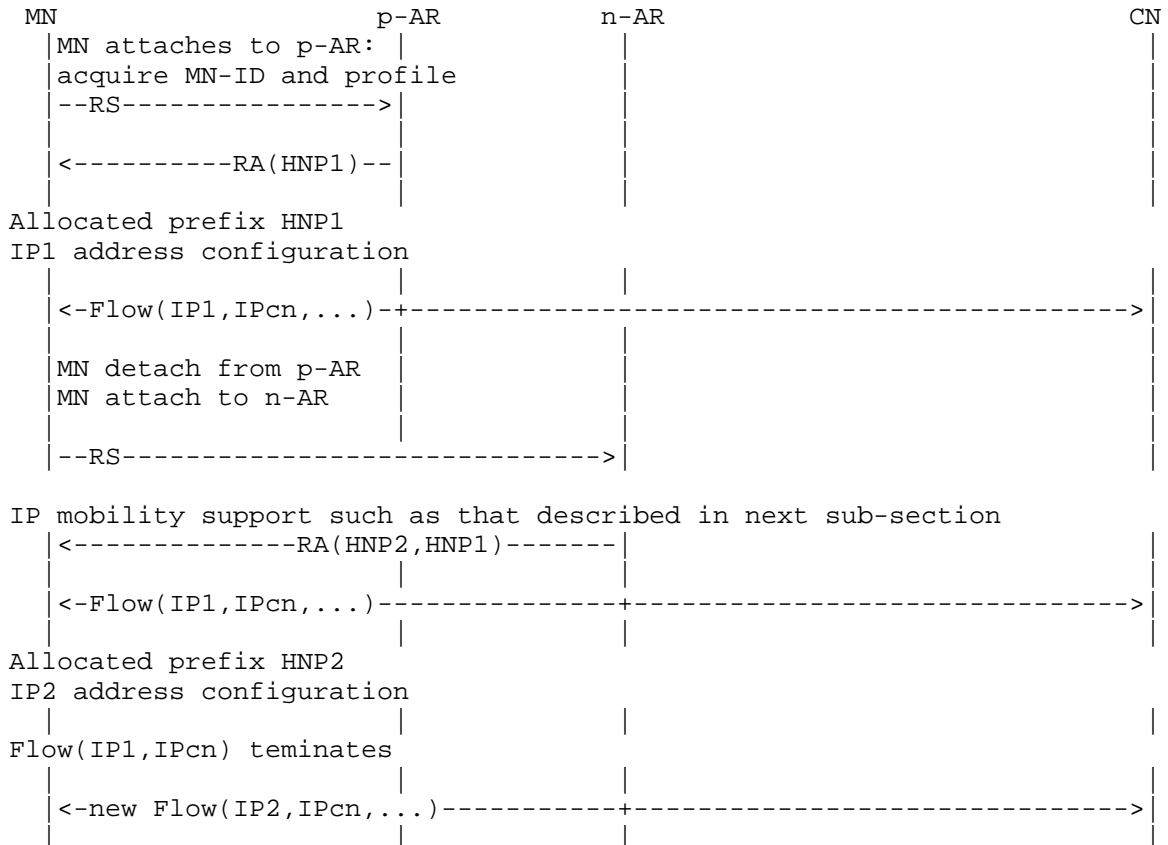


Figure 5. A flow uses the IP allocated from the network at which the MN is attached when the flow is initiated.

To provide IP mobility support with distributed anchoring, the distributed anchors may need to message with each other. When such messaging is needed, the anchors may need to discover each other as described in the FM behaviors and information elements (FM:2) in Section 3.2.2.

Then the anchors need to properly forward the packets of the flows as described in the FM behaviors and information elements (FM:1) in Section 3.2.2.

If there are in-flight packets toward the old anchor while the MN is moving to the new anchor, it may be necessary to buffer these packets and then forward to the new anchor after the old anchor knows that the new anchor is ready. Such are described in the FM behaviors and information elements (FM:4) in Section 3.2.2.

#### 4.2. IP prefix/address anchor switching to the new network

The IP prefix/address anchoring may move without changing the IP prefix/address of the flow. Here the LM and FM functions in Figures 1(a) and 1(b) in Section 3.1 are implemented as shown in Figure 6.

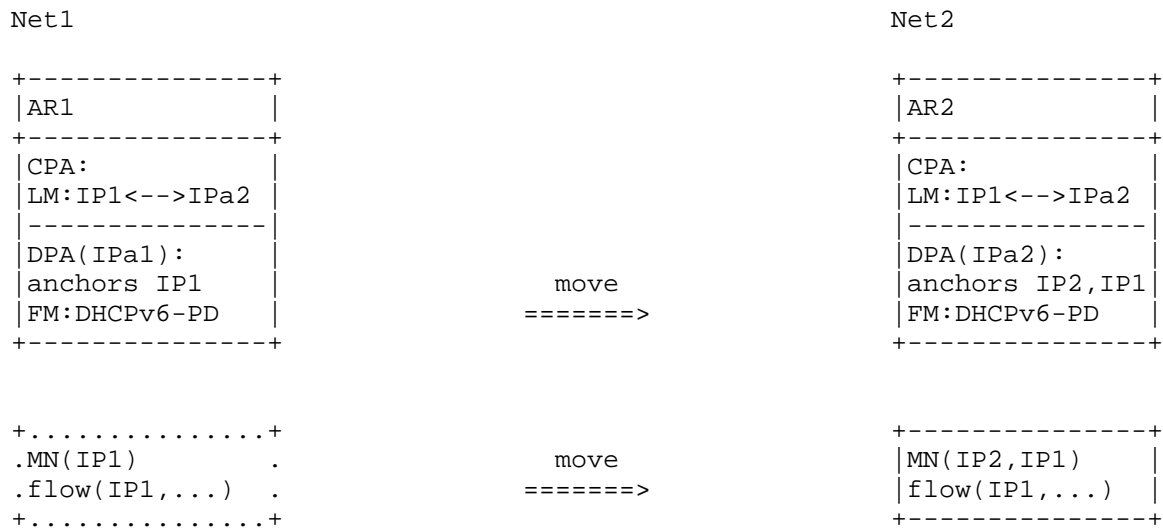


Figure 6. IP prefix/address anchor switching to the new network. MN with flow using IP1 in Net1 continues to run the flow using IP1 as it moves to Net2.

As an MN with an ongoing session moves to a new network, the flow may preserve session continuity by moving the anchoring of the original IP prefix/address of the flow to the new network. An example is in the use of BGP UPDATE messages to change the forwarding table entries as described in [I-D.mccann-dmm-flatarch] and also for 3GPP Evolved Packet Core (EPC) network in [I-D.matsushima-stateless-uplane-vepc]. However, the response time and scalability of using a distributed routing protocol to update forwarding tables may be controversial.

Use of a centralized routing protocol with a centralized control plane as described in Section 4.2.1 will be more scalable.

The location management provides information about which IP prefix from an AR in the original network is being used by a flow in which AR in a new network. Such information needs to be deleted or updated when such flows have closed so that the IP prefix is no longer used in a different network. The LM behaviors are described in Section 3.2.1.

The FM functions are implemented through the DHCPv6-PD protocol. Here the anchor behavior to properly forward the packets for a flow as described in the FM behaviors and information elements FM:1 in Section 3.2.2 is realized by changing the anchor with DHCPv6-PD and also by reverting such changes later after the application has already closed and when the DHCPv6-PD timer expires. If there are in-flight packets toward the old anchor while the MN is moving to the new anchor, it may be necessary to buffer these packets and then forward to the new anchor after the old anchor knows that the new anchor is ready. Such are described in the FM behaviors and information elements FM:4 in Section 3.2.2. The anchors may also need to discover each other as described in the FM behaviors and information elements FM:2.

The security management function in the anchor node at a new network must allow to assign the original IP prefix/address used by the mobile node at the previous (original) network. As the assigned original IP prefix/address is to be used in the new network, the security management function in the anchor node must allow to advertise the prefix of the original IP address and also allow the mobile node to send and receive data packets with the original IP address.

The security management function in the mobile node must allow to configure the original IP prefix/address used at the previous (original) network when the original IP prefix/address is assigned by the anchor node in the new network. The security management function in the mobile node also allows to use the original IP address for the previous flow in the new network.

#### 4.2.1. Centralized control plane

An example of IP prefix anchor switching is in the case where Net1 and Net2 both belong to the same operator network with separation of control and data planes ([I-D.liu-dmm-deployment-scenario] and [I-D.matsushima-stateless-uplane-vepc]), where the controller may send to the switches/routers the updated information of the forwarding tables with the IP address anchoring of the original IP



prefix/address at AR1 moved to AR2 in the new network. That is, the IP address anchoring in the original network which was advertising the prefix will need to move to the new network. As the anchoring in the new network advertises the prefix of the original IP address in the new network, the forwarding tables will be updated so that packets of the flow will be forwarded according to the updated forwarding tables. The configuration in Figures 1(a) and 1(b) in Section 3.1 for which FM-CP and LM are centralized and FM-DP's are distributed. applies here. Figure 7 shows its implementation where LM is a binding between the original IP prefix/address of the flow and the IP address of the new DPA, whereas FM uses the DHCPv6-PD protocol.

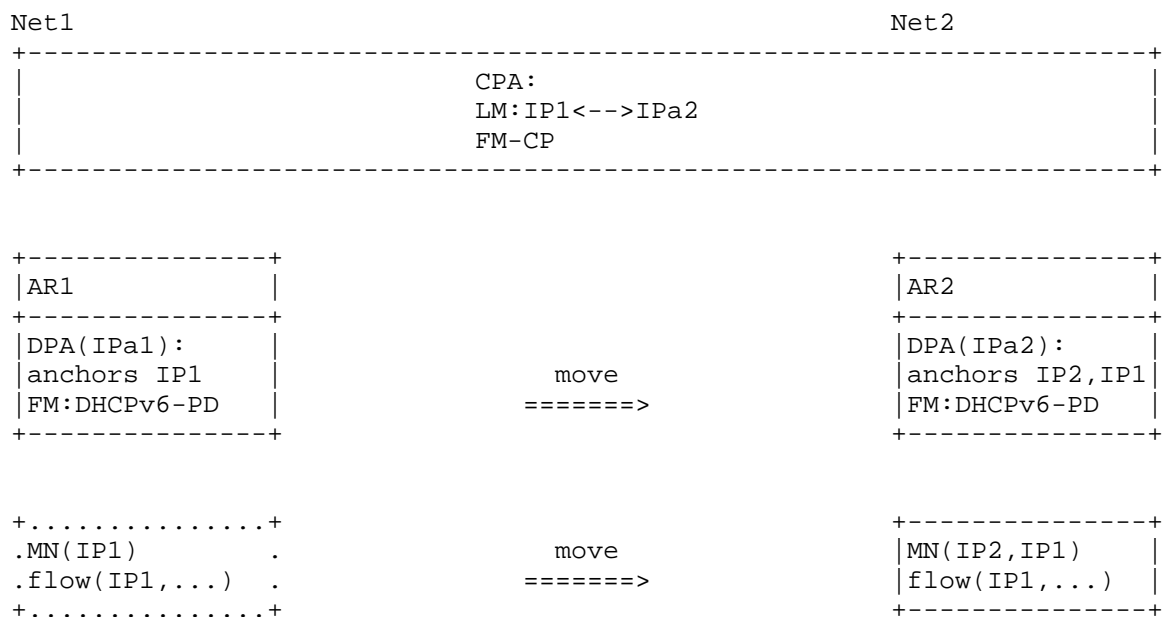


Figure 7. IP prefix/address anchor switching to the new network with LM and FM-CP in a centralized control plane whereas the FM-DP's are distributed.

The call flow in Figure 8 shows that MN is allocated HNP1 when it attaches to the p-AR. A flow running in MN may or may not need IP mobility. If it does, it may continue to use the previous IP prefix. If it does not, it may use a new IP prefix allocated from the new network.

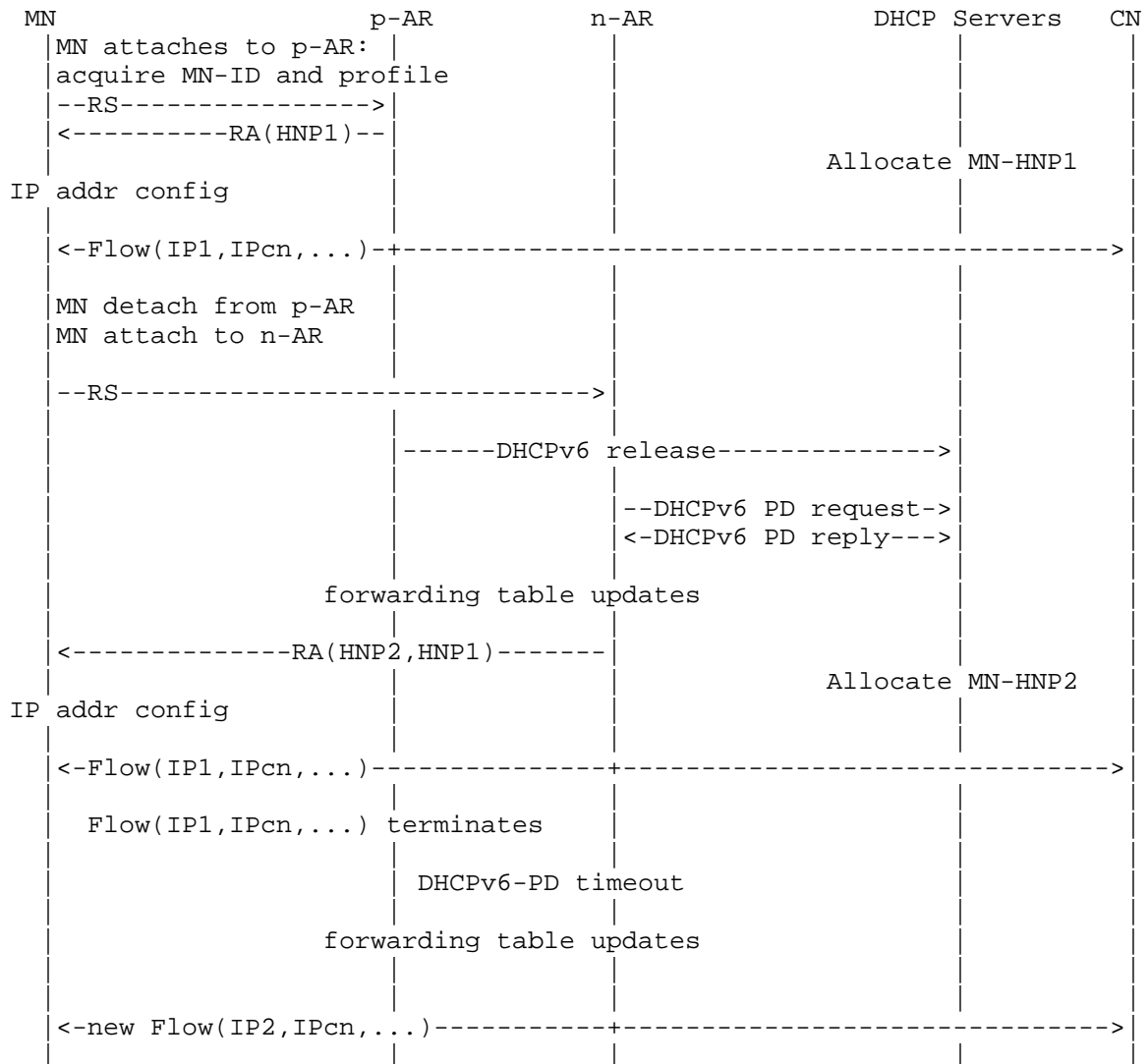


Figure 8. DMM solution. MN with flow using IP1 in Net1 continues to run the flow using IP1 as it moves to Net2.

As the MN moves from p-AR to n-AR, the p-AR as a DHCP client may send a DHCP release message to release the HNP1. It is now necessary for n-AR to learn the IP prefix of the MN from the previous network so that it will be possible for Net2 to allocate both the previous network prefix and the new network prefix. The network may learn the previous prefix in different methods. For example, the MN may

provide its previous network prefix information by including it to the RS message [I-D.jhlee-dmm-dnpp].

Knowing that MN is using HNP1, the n-AR sends to a DHCP server a DHCPv6-PD request to move the HNP1 to n-AR. The server sends to n-AR a DHCPv6-PD reply to move the HNP1. Then BGP route updates will take place here.

In addition, the MN also needs a new HNP in the new network. The n-AR may now send RA to n-AR, with prefix information that includes HNP1 and HNP2. The MN may then continue to use IP1. In addition, the MN is allocated the prefix HNP2 with which it may configure its IP addresses. Now for flows using IP1, packets destined to IP1 will be forwarded to the MN via n-AR.

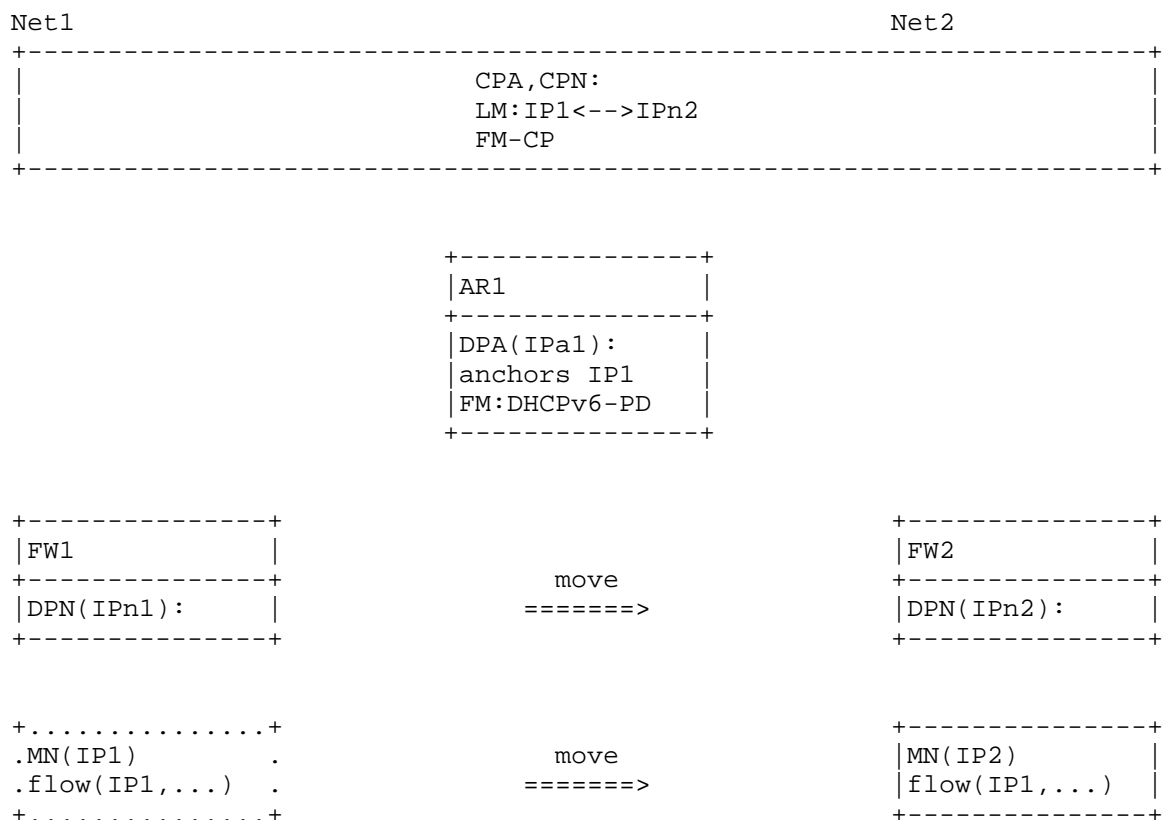
As such flows have terminated and DHCP-PD has timed out, HNP1 goes back to Net1. MN will then be left with HNP2 only, which it will use when it now starts a new flow.

The anchor behavior to properly forward the packets for a flow as described in the FM behaviors and information elements (FM:1) in Section 3.2.2 is realized by changing the anchor with DHCPv6-PD and undoing such changes later when its timer expires and the application has already closed. With the anchors being separated in control and data planes with LMs and FM-CP centralized in the same control plane, messaging between anchors and the discovery of anchors become internal to the control plane. However, the centralized FM-CP needs to communicate with the distributed FM-DP as described as described in the FM behaviors and information elements (FM:3). Such may be realized by the appropriate messages in [I-D.ietf-dmm-fpc-cdpd]. Again, if there are in-flight packets toward the old anchor while the MN is moving to the new anchor, it may be necessary to buffer these packets and then forward to the new anchor after the old anchor knows that the new anchor is ready. The corresponding FM behaviors and information elements (FM:4) are however realized by the internal behavior in the control plane together with signaling between the control plane and distributed data plane.

#### 4.2.2. Hierarchical network

The configuration for a hierarchical network is shown in Figures 1(c) and 1(d) in Section 3.1. With centralized control and with a centralized anchor, LM, CPA, CPN are co-located at the centralized control, and there is an AR with the DPA function supporting multiple forwarding switches (FW's) each with a DPN function. A mobility event in this configuration involving change of FW but not of AR is shown in Figure 9.

The realization of LM may bet the binding between the IP prefix/ address of the flow used by the MN and the IP address of the DPN to which MN has moved. The implementation of FM to enable change of FW without changing AR may be accomplished using tunneling between the AR and the FW as described in [I-D.korhonen-dmm-local-prefix] and in [I-D.templin-aerolink] or using some other L2 mobility mechanism.



Here, the LM behaviors and information elements described in Section 3.2.1 provides information of which IP prefix from its FW needs to be used by a flow using which new FW. The anchor behaviors

to properly forward the packets of a flow described in the FM behaviors and information elements (FM:1) may be realized with PMIPv6 protocol ([I-D.korhonen-dmm-local-prefix]) or with AERO protocol ([I-D.templin-aerolink]) to tunnel between the AR and the FW.

#### 4.2.3. Hierarchical network with anchoring change

The configuration for a hierarchical network is still shown in Figures 1(c) and 1(d) in Section 3.1. Again, with centralized control and with a centralized anchor, LM, CPA, CPN are co-located at the centralized control, and there is an AR with the DPA function supporting multiple forwarding switches (FW's) each with a DPN function. However, the mobility event involving change of FW may also involve a change of AR. Such configuration is shown in Figure 10.

This deployment case involves both a change of anchor from AR1 to AR2 and a network hierarchy AR-FW. It can be realized by a combination of changing the IP prefix/address anchoring from AR1 to AR2 with the mechanism as described in Section 4.2.1 and then forwarding the packets with network hierarchy AR-FW as described in Section 4.2.2.

To change AR, AR1 acting as a DHCP-PD client may exchange message with the DHCP server to release the prefix IP1. Meanwhile, AR2 acting as a DHCP-PD client may exchange message with the DHCP server to delegate the prefix IP1 to AR2.

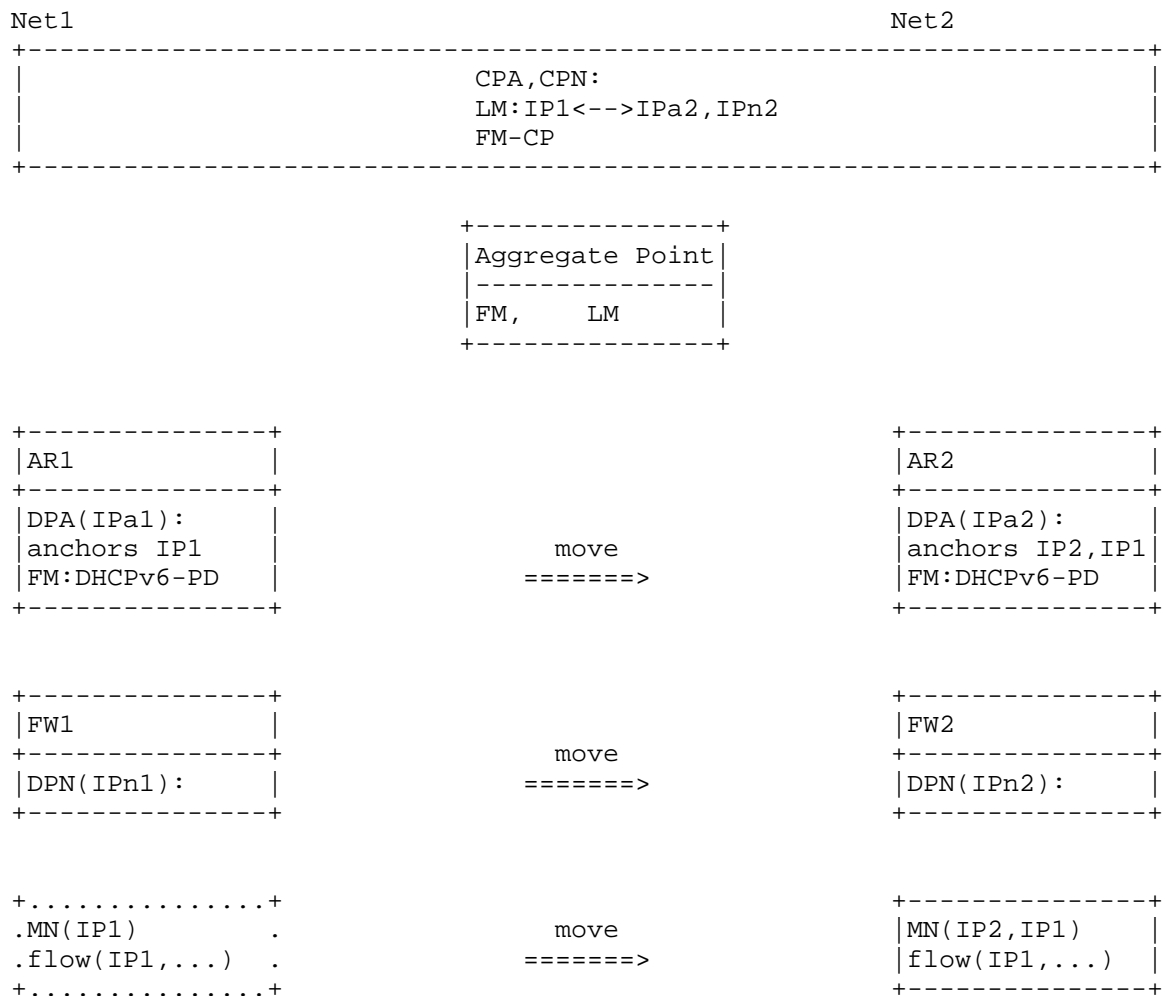


Figure 10. Mobility involving change of IP anchoring in a network with hierarchy in which the IP prefix allocated to the MN is anchored at an Edge Router supporting multiple access routers to which the MN may connect.

## 5. Security Considerations

TBD

## 6. IANA Considerations

This document presents no IANA considerations.

## 7. Contributors

This document has benefited from other work on mobility solutions using BGP update, on mobility support in SDN network, on providing mobility support only when needed, and on mobility support in enterprise network. These work 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 Matsushima, Peter McCann, Pierrick Seite, Jouni Korhonen, and Sri Gundavelli.

Valuable comments have also been received from John Kaippallimil, ChunShan Xiong, and Dapeng Liu.

## 8. References

### 8.1. Normative References

[I-D.dmm-ondemand-mobility-api]

Yegin, A., Kweon, K., Lee, J., Park, J., and D. Moses, "On Demand Mobility API", draft-dmm-ondemand-mobility-api-00 (work in progress), May 2015.

[I-D.ietf-dmm-fpc-cpdp]

Liebsch, M., Matsushima, S., Gundavelli, S., and D. Moses, "Protocol for Forwarding Policy Configuration (FPC) in DMM", draft-ietf-dmm-fpc-cpdp-01 (work in progress), July 2015.

[I-D.jhlee-dmm-dnpp]

Lee, J. and Z. Yan, "Deprecated Network Prefix Provision", draft-jhlee-dmm-dnpp-00 (work in progress), October 2015.

[I-D.korhonen-dmm-local-prefix]

Korhonen, J., Savolainen, T., and S. Gundavelli, "Local Prefix Lifetime Management for Proxy Mobile IPv6", draft-korhonen-dmm-local-prefix-01 (work in progress), July 2013.

[I-D.liu-dmm-deployment-scenario]

Liu, V., Liu, D., Chan, A., Lingli, D., and X. Wei, "Distributed mobility management deployment scenario and architecture", draft-liu-dmm-deployment-scenario-05 (work in progress), October 2015.

- [I-D.matsushima-stateless-uplane-vepc]  
Matsushima, S. and R. Wakikawa, "Stateless user-plane architecture for virtualized EPC (vEPC)", draft-matsushima-stateless-uplane-vepc-05 (work in progress), September 2015.
- [I-D.mccann-dmm-flatarch]  
McCann, P., "Authentication and Mobility Management in a Flat Architecture", draft-mccann-dmm-flatarch-00 (work in progress), March 2012.
- [I-D.mccann-dmm-prefixcost]  
McCann, P. and J. Kaippallimalil, "Communicating Prefix Cost to Mobile Nodes", draft-mccann-dmm-prefixcost-02 (work in progress), October 2015.
- [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.sijeon-dmm-deployment-models]  
Jeon, S. and Y. Kim, "Deployment Models for Distributed Mobility Management", draft-sijeon-dmm-deployment-models-01 (work in progress), October 2015.
- [I-D.templin-aerolink]  
Templin, F., "Asymmetric Extended Route Optimization (AERO)", draft-templin-aerolink-66 (work in progress), February 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<http://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, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc7429>>.

## 8.2. Informative References

- [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.

## Authors' Addresses

H Anthony Chan  
Huawei Technologies  
5340 Legacy Dr. Building 3  
Plano, TX 75024  
USA

Email: [h.a.chan@ieee.org](mailto:h.a.chan@ieee.org)

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

Email: [weixinpeng@huawei.com](mailto:weixinpeng@huawei.com)

Jong-Hyouk Lee  
Sangmyung University  
708 Hannuri Building  
Cheonan 330-720  
Korea

Email: [jonghyouk@smu.ac.kr](mailto:jonghyouk@smu.ac.kr)

Seil Jeon  
Instituto de Telecomunicacoes  
Campus Universitario de Santiago  
Aveiro 3810-193  
Portugal

Email: [seiljeon@av.it.pt](mailto:seiljeon@av.it.pt)

Fred L. Templin  
Boeing Research and Technology  
P.O. Box 3707  
Seattle, WA 98124  
USA

Email: [fltemplin@acm.org](mailto:fltemplin@acm.org)

DMM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 22, 2016

M. Liebsch  
NEC  
S. Matsushima  
SoftBank  
S. Gundavelli  
Cisco  
D. Moses  
Intel Corporation  
L. Bertz  
Sprint  
March 21, 2016

Protocol for Forwarding Policy Configuration (FPC) in DMM  
draft-ietf-dmm-fpc-cpdp-03.txt

Abstract

This specification supports the separation of the Control-Plane for mobility- and session management from the Data-Plane. The protocol semantics abstract the configuration of Data-Plane nodes and applies it between a Client function, which is used by an application of the mobility Control-Plane, and an Agent function, which is associated with the configuration of Data-Plane nodes, according to the Data-Plane rules issued by the mobility Control-Plane. The scope of the rules comprises traffic description and treatment of packets in terms of encapsulation, IP address re-writing and QoS. Additional protocol semantics are described to support the maintenance of the Data-Plane path.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 22, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Conventions and Terminology . . . . .	4
3. Reference Architecture and Deployment Options . . . . .	4
3.1. Architecture for DMM Forwarding Policy Configuration . .	4
3.2. Model 1 . . . . .	6
3.2.1. Role of the FPC Client Function . . . . .	7
3.2.2. Role of the FPC Agent Function . . . . .	7
3.3. Model 2 . . . . .	8
3.3.1. Role of the DMM FPC Client Function . . . . .	8
3.3.2. Role of the DMM FPC Agent Function . . . . .	8
4. Protocol to support Model I . . . . .	9
4.1. Data Structure . . . . .	9
4.2. Protocol Attributes . . . . .	12
4.3. Protocol Messages and Semantics . . . . .	19
4.4. Protocol Operation . . . . .	20
5. Protocol to support Model II . . . . .	29
5.1. Protocol Attributes . . . . .	29
5.2. Protocol Messages and Semantics . . . . .	31
5.3. Protocol Operation . . . . .	33
6. Security Considerations . . . . .	34
7. IANA Considerations . . . . .	34
8. Work Team Participants . . . . .	34
9. References . . . . .	34
9.1. Normative References . . . . .	34
9.2. Informative References . . . . .	35
Appendix A. YANG Data Model for the FPC protocol . . . . .	35
A.1. FPC Base . . . . .	36
A.1.1. FPC Base YANG Model . . . . .	36
A.1.2. FPC Base tree . . . . .	52
A.2. FPC PMIP . . . . .	58
A.2.1. FPC PMIP YANG Model . . . . .	58

A.2.2. FPC PMIP tree . . . . .	61
Authors' Addresses . . . . .	67

## 1. Introduction

One objective of the Distributed Mobility Management (DMM) WG is the separation of the mobility management Control- and Data-Plane to enable flexible deployment, such as decentralized provisioning of Data-Plane nodes (DPN). Data-Plane nodes can be configured to function as an anchor for a registered Mobile Node's (MN) traffic, others can be configured to function as a Mobile Access Gateway (MAG) per the Proxy Mobile IPv6 protocol [RFC5213] or a Foreign Agent (FA) per the Mobile IPv4 protocol [RFC3344]. Requirements for DMM have been described in [RFC7333], whereas best current practices for DMM are documented in [RFC7429].

The Data-Plane must provide a set of functions to the Mobility Control-Plane, such as support for encapsulation, IP address re-writing, QoS differentiation and traffic shaping. In addition, means for traffic description must be provided to complement traffic treatment actions and build unambiguous Data-plane rules. These requirements are met by various transport network components, such as IP switches and routers, though configuration semantics differ between them.

Forwarding Policy Configuration (FPC) per this document enables the configuration of any Data-Plane node and type by the abstraction of configuration details and the use of common configuration semantics. The protocol using the FPC semantics is deployed between a Client function, which is associated with the Mobility Management Control-Plane, and an Agent function. The Agent function enforces the Data-Plane configuration and can be present on a transport network controller or co-located with a Data-Plane node. The Agent applies the generalized configuration semantics to configuration, which is specific to the Data-Plane node and type.

This specification follows a common functional architecture, which utilizes the FPC protocol between the Client and Agent functions, and supports two operational models, Model I and Model II.

A Client supporting Model I interacts with the Agent to build unambiguous rules which are to be enforced in the Data-Plane. An Agent supporting Model I translates a rule, which follows the data model herein, into one or multiple configuration actions to enforce the rule in the Data-Plane.

A Client supporting Model II utilizes a sequence of control messages to interact with the Agent, where each control message has an

unambiguous semantic, e.g. to set up a tunnel interface or to configure a policy route in a Data-Plane node. An Agent supporting Model II performs a configuration action per the semantics of the received control message.

The availability of both operational models enables tailored implementation and deployment of Control-/Data-Plane separation in mobile communication gateways, e.g. by having the Mobility Control-Plane directly communicating to a Data-Plane node as per Model II, or per Model I by the deployment of a Network Controller in between the Mobility Control-Plane and Data-Plane nodes, which are under control of the Network Controller. Support for both the models enables an operator to transition their network in incremental phases.

The architecture and reference interface specified in this document is not tied to any specific Control-Plane protocol that is in use in the mobility network, or to any type of access technology. The mobility protocols in use can be Proxy Mobile IPv6, GTP, IPSec or other protocols; and the access network can be 4G LTE, WiFi, or 5G. These aspects have no direct implication on the FPC interface that is between Control- and Data-Plane nodes.

## 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].

## 3. Reference Architecture and Deployment Options

### 3.1. Architecture for DMM Forwarding Policy Configuration

The DMM Forwarding Policy Configuration (FPC) protocol enables the separation of the mobility management Control-Plane from the Data-Plane and provides the required control and semantics in between these two planes. Figure 1 depicts an exemplary use case where IP traffic between a Correspondent Node (CN) and a Mobile Node (MN) traverses multiple DPNs, each applying policies as per the Control-Plane's request. Policies in the one or multiple DPNs can result in traffic steering according to a host-route, packet scheduling and marking according to a subscriber's QoS profile, or forwarding rules (e.g. encapsulation within GRE or GTP-U tunnel).

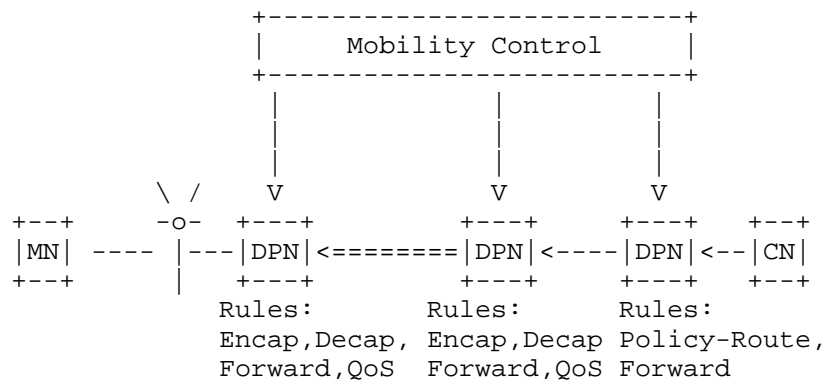


Figure 1: Exemplary illustration of DMM traffic steering and policy enforcement at Data Plane Nodes (DPN)

Mobility Control-Plane functions have the following roles in common:

- o Tracking a mobile node's attachment, detachment from the access network
- o Accept requests to set up and maintain mobility-related Data-Plane paths between DPNs, enforcing QoS and forwarding policies. Such requests are a result of mobility signaling between different Mobility Control-Plane functions.
- o Ensure that required rules to establish and maintain connectivity of an MN with its correspondent nodes are enforced in the Data-Plane.
- o Participate in monitoring the DPNs' operation and support the handling of exceptions, e.g. the detection of a partial DPN failure and the diversion of traffic through a different DPN.
- o Maintain consistency between multiple DPNs which enforce policy rules to ensure connectivity between a MN and its correspondent services.

Mobility Data-Plane functions have the following roles in common:

- o Forward and treat traffic according to the policies and directives sent by the Mobility Control-Plane
- o Provide status information (e.g. load, health, statistics and traffic volume) and events related to service failure upon request

- o Participate in the process of topology acquisition, e.g. by exposing relevant topological and capability information, such as support for QoS differentiation and supported encapsulation protocols

The protocol for DMM FPC applies to the interface between a FPC Client function and a FPC Agent function, as depicted in Figure 2. The FPC Client function is associated with an application function of the mobility management Control-Plane, e.g. a Local Mobility Anchor Control-Plane function per the Proxy Mobile IPv6 protocol. The FPC Agent function processes the FPC protocol semantics and translates them into configuration commands per the DPN's technology. In one example, an FPC Agent can be co-located with a Network Controller, which enforces forwarding rules on a set of Data-plane nodes. In another example, the Agent can be co-located with a Data-Plane node to directly interact with interface management and the router's RIB Manager. The mapping of the common FPC semantics and policy description to the configuration commands of a particular DPN is specific to the DPN's technology and the Agent's implementation.

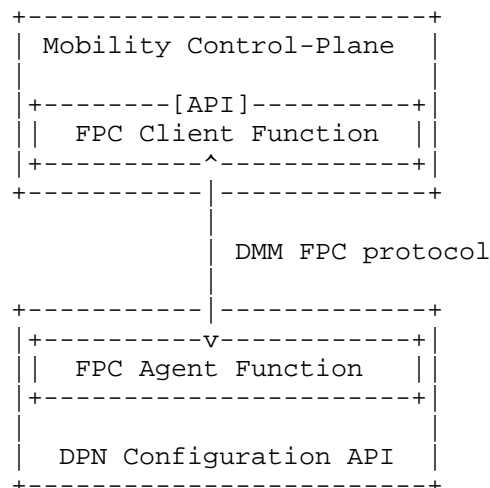


Figure 2: Functional reference architecture for DMM Forwarding Policy Configuration (FPC)

### 3.2. Model 1



### 3.2.1. Role of the FPC Client Function

The FPC Client function, which follows Model I operation, includes the following tasks:

- o Build one or multiple FPC Control messages/attributes to establish, update or delete rules on one or multiple DPN(s) according to the Mobility Control-Plane's directives
- o Apply a DPN's policy rules (encapsulation, address re-write, QoS, traffic monitoring) on the basis of properties bound to logical ports (similar to the bearer concept in cellular networks)
- o Build, modify or delete logical ports as needed
- o Bind associated policy rules as one or multiple properties to a logical port
- o Apply traffic forwarding rules (e.g. per-IP flow, per-MN, per-IP, per-prefix) on the basis traffic descriptions bound to logical ports
- o Send each generated FPC control message to the FPC Agent
- o Keep record of the configured policy rules and interact with the FPC Agent to ensure proper synchronization between Mobility Control-Plane states and rules configured on the FPC Agent
- o Process received Response, Notification and Query messages issued by a FPC Agent and interact with the Control-Plane to act accordingly

### 3.2.2. Role of the FPC Agent Function

The FPC Agent function, which follows Model I operation, includes the following tasks:

- o Process received Control messages issued by a FPC Client Function
- o Apply received rules to local configuration (e.g. encapsulation, NA(P)T, traffic prioritization and scheduling) in the Data-Plane
- o Maintain administrative data as well as operational data, which describes the status of the rules in the Data-Plane
- o Monitor events (e.g. failure, incomplete rule) and issue an associated message to the FPC Client Function (NOTIFICATION, QUERY)

### 3.3. Model 2

#### 3.3.1. Role of the DMM FPC Client Function

The FPC Client function, which follows Model II operation, includes the following tasks:

- o The FPC Client offers a set of services to the mobility control plane entities. These services are for activating/deactivating specific configuration on a Data-Plane node supported by a FPC Agent. These services for example are creation/deletion of a layer-3 tunnel; adding/deleting an IP route;
- o The FPC Client translates the request from the mobile control plane as a FPC message. The message identifies the service name and includes a set of information elements. This message is sent to the FPC Agent over the FPC interface.

#### 3.3.2. Role of the DMM FPC Agent Function

The FPC Agent function, which follows Model II operation, includes the following tasks:

- o FPC Agent offers a set of services to the FPC client. Each of these services have a well-defined meaning and can be invoked by the FPC Client passing a set of parameters. These services for example are creation/deletion of a layer-3 tunnel; adding/deleting an IP route.
- o Any FPC Client can invoke a specific service on the FPC Agent through the use of FPC messaging interface. The interface semantics allow the identification of the service request and for inclusion of the parameters relevant for that service request.
- o FPC Agent processes a FPC message and identifies the service request. The FPC Agent maps the service request to a local configuration and enables that configuration in the forwarding plane. For example, if there is a service request for Tunnel creation including the relevant parameters such as source IP address, destination IP address and encapsulation type, this request will result in the FPC Agent configuring such tunnel configuration on the Data-Plane node.
- o The FPC Agent provides a resulting status code on how the request was executed by the agent.

## 4. Protocol to support Model I

### 4.1. Data Structure

To abstract from configuration details of an IP switch or IP router on the FPC protocol interface, Model I adopts the construct of logical ports to describe rules for D-Plane processing. A port binds one or multiple properties, which describe traffic treatment actions, such as a QoS policy, IP address re-write or packet encapsulation. Which traffic is treated is determined by one or multiple traffic descriptors, which also bind to that port. A group of one or multiple traffic descriptors, one or multiple properties defining traffic treatment actions and the port identifier make a rule. The port identifier serves as key to access the rule.

All traffic arriving at a Data-Plane node and matching a traffic descriptor will be treated per the properties bound to the port the traffic descriptor is also bound to. For example, Traffic Selectors [RFC6088], which can be bound to a port, can identify single or multiple IP flows. Aggregated IP traffic destined toward a given IP address prefix or originated from an address matching a particular IP address range can be described using the Traffic Selector or an IP prefix traffic descriptor per this specification.

In addition to traffic descriptors and traffic treatment actions, which build a Data-Plane processing rule, a port has associated operational data, which tracks the status of rule enforcement in a selected Data-Plane node. A rule can also have administrative data such as its directionality (uni- or bi-directional) and administrative status such as enabled, disabled or virtual. Furthermore, an identifier of the Data-Plane node to which the rule applies is kept in the operational data associated with a port.

When the Client desires specific operational state for the port, it may apply administrative state properties to the port. This, however, may not take immediate effect on the Data-Plane Node. Thus, Client implementations must support situations where differences exist between configured and operational state of a port. A Client can request operational data associated with a particular port from an Agent.

A Client adds, modifies or deletes a rule on an Agent using the FPC protocol messages. The protocol enables a Client to provide additional administrative information about a particular port or a group of ports to the Agent. This includes control of the operation of a rule, e.g. whether a rule associated with a particular port applies only uni-directionally or bi-directionally. In case of bi-directionality, an Agent can apply a rule associated with a single

port in the Data-Plane to both directions. As example, a rule which performs re-writing of an arriving packet's destination IP address from IP\_A to IP\_B matching an associated Traffic Selector, can be enforced in the Data-Plane via an Agent to implicitly consider matching arriving packet's source IP address against IP\_B and re-write the source IP address to IP\_A.

Figure 3 illustrates the generic policy configuration model as used between a FPC Client and a FPC Agent.

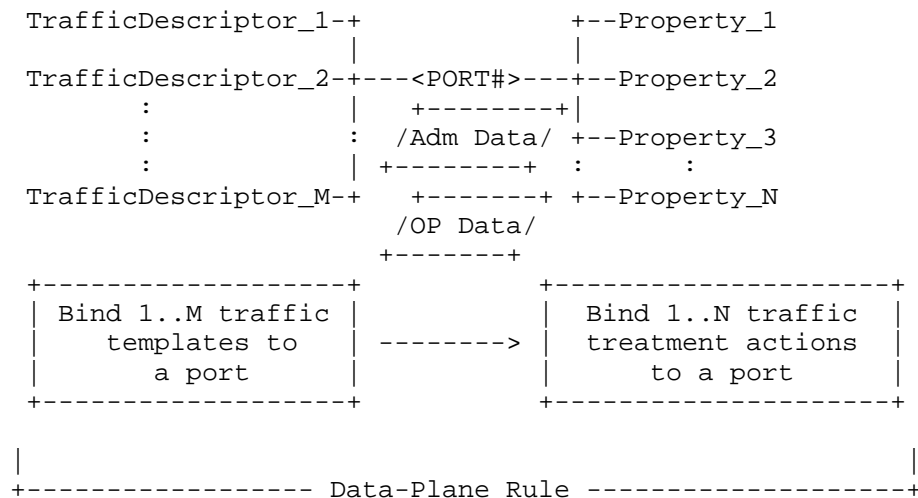


Figure 3: Structure of rules on Client/Agent defining Data-Plane traffic treatment

As depicted in Figure 3, the port represents the anchor of a rule. A Client and Agent use the identifier of a port to access the rule and perform modifications of traffic descriptors or properties. From the viewpoint of packet processing, arriving packets are matched against traffic descriptors and processed according to the treatment actions specified in the list of properties associated with the port.

A Client can assign an existing or new port to a group of ports using a port group identifier. The logic behind grouping multiple ports is up to the Control-Plane. As example, multiple rules associated with a single mobile node can be grouped and identified by the port group identifier. In case the Control-Plane needs to delete all rules associated with the mobile node, the Client can issue a message to delete a port one and identify the group group identifier instead of deleting each port individually. A Client can also apply

administrative properties to a group of ports by adding the port group ID to the FPC message.

A Client can complement a traffic descriptor with a match priority value to allow unambiguous traffic matching on the Data-Plane. If the Client does not provide a match priority value with a traffic descriptor or a group of traffic descriptors have the same priority value, an Agent enforces the rule in the Data-Plane node to enable traffic detection by longest prefix match.

Operational information of a port includes the data listed in the following table:

Admin Data	Format Clarification	Description
DPN_ID	Sect. 4.2	Identifies a Data-Plane node to which the rule applies
PRT_BIDIR	BOOLEAN	Bidirectionality of a port (cleared = unidirectional)
ADMIN_STATUS	[8, admin status]	Requested status for a rule in a Data-Plane node (enabled, disabled, virtual)
SESSION_STATUS	[8, session status]	Status of a session in the Control-Plane (complete, incomplete, outdated)
PRT_GROUP_ID	[32, group id]	Identifies a group of ports to which this port belongs
CLI_ID	Sect. 4.2	Identifies the Client which created this port
AGT_ID	Sect. 4.2	Identifies the Agent which enforces the rule as per this port

Figure 4: Administrative Data associated with a port

Operational Data	Format Clarification	Description
OPER_STATUS	[8, oper status]	Status of a rule in a Data-Plane node (enabled, disabled, virtual)
SERVICE_STATUS	[8, service status]	Ability of an enabled rule to serve traffic (complete, incomplete, outdated)

Figure 5: Operational Data associated with a port

A Client MAY apply an administrative state property to a port indicating the desired operational status of a port, e.g. enabled, disabled or virtual (not intended to serve traffic but used as a template for other ports). Rules specified by an enabled port are enforced in the Data-Plane node. A disabled port on an Agent can be useful for pre-configuration, e.g. other operations can be performed on the port prior to its enablement. Ultimately, a disabled port is intended to be enabled. Virtual ports can serve as a reference to clone new ports, which can then be enabled. When creating a cloned port, the Client can update or add properties to suit the rule that should be enforced in the Data-Plane.

A Client MAY set a Session state for a particular port or group of ports on the Agent to guide the Agent on how to treat local events. As example, an Agent SHOULD refrain from sending an FPC message to the Client as result of a local event, which indicates a missing rule, in case the session state is 'incomplete', as the Agent can expect the Control-Plane to provide the missing rule unsolicited. In case the session state is 'outdated', the Agent MAY notify the Client to update the associated rule on the Agent.

#### 4.2. Protocol Attributes

Protocol messages as per Section 4.3 identify an FPC Client or Agent function, as well as a DPN, and carry traffic descriptor attributes, logical port identification and properties specifying traffic treatment actions. Traffic can be described per-host, in aggregate or per-IP flow. A Client MAY append administrative properties to a message to indicate the desired status of a port to the Agent.

This document specifies attributes from the following categories:

- o Identifier attributes

- o Traffic Descriptors
- o Properties specifying traffic treatment actions
- o Protocol-specific Properties
- o Administrative properties

Attribute	Format Clarification	Description
Identifiers		
PRT_ID	[32,PRT_ID]	Identifies a logical Port
PRT_GROUP_ID	[32,PRT_GROUP_ID]	Identifies a group of logical Ports
PRT_PROP_ID	[32,PRT_ID] [8,PROP_ID]	Identifies a logical Port and one of its properties
PRT_TD_ID	[32,PRT_ID] [8,TD_ID]	Identifies a logical Port and a traffic descriptor that applies to the port
CLI_ID	[16, Carrier ID] [16, Network ID] [32, Client ID]	Identifies an FPC Client function
AGT_ID	[16, Carrier ID] [16, Network ID] [32, Agent ID]	Identifies an FPC Agent function
DPN_ID	[16, Carrier ID] [16, Network ID] [32, DPN ID]	Identifies a Data Plane Node (DPN)
MONITOR_ID	[32, Monitor ID]	Identifies a registered monitor
EVENT_TYPE_ID	[8, Event Type ID]	Identifies an event type
Optional Identifiers		
SERVICE_PATH_ID	[24-bit identifier]	Service Path Identifier

Figure 6: Model I Protocol Attributes: Identifiers

Attribute	Format Clarification	Description
Properties		
PROP_TUN	[type][src][dst]	Property Encapsulation,



		indicates type GRE, IP, GTP
PROP_REWR	[in_src_ip][out_src_ip] [in_dst_ip][out_dst_ip] [in_src_port][out_src_port] [in_dst_port][out_dst_port]	Property NAT defines IP address and port re-write rules
PROP_QOS	[QoS index type][index] [DSCP]	Property QoS refers to single index and DS Code Point to write
PROP_QOS_GBR	[GBR] *[PRT_ID]	Guaranteed Bit Rate and single or multiple PRT_IDs to which the GBR applies when being aggregated
PROP_QOS_MBR	[MBR] *[PRT_ID]	Maximum Bit Rate and single or multiple PRT_IDs to which the MBR applies when being aggregated
PROP_GW	[ip address next hop]	IP address of the Next Hop to which IP packets should be forwarded
PROP_CPY_FORW	[PRT_ID]	Copy IP packets, treat the duplicates per the properties of the referred port
PROP_DROP		Drop IP packet
PROP_CONCAT	[PRT_ID]	Include treatment per the referred port into the rule
Optional Properties		
PROP_NSH	[SERVICE_PATH_ID] [Service Index]	Include NSH

Figure 7: Model I Protocol Attributes: Traffic Treatment Properties

Attribute	Format Clarification	Description
Protocol-specific		
IPIP_CONF		IP-encapsulation configuration attribute
GRE_CONF	[prototype][seq-#] [key]	GRE_encapsulation configuration attribute
GTP_CONF	[TEID_local] [TEID_remote] [seq-#]	GTP-U encapsulation configuration attribute

Figure 8: Model I Protocol Attributes: Protocol-specific

Attribute	Format Clarification	Description
Traffic Descriptor Container		
TD_CONTAINER	[PRT_TD_ID] [8, PRIO] *[traffic descriptor]	Traffic handling priority, One or multiple traffic descriptors

Figure 9: Protocol Attributes: Traffic Description Container

Attribute	Format Clarification	Description
Traffic Descriptors		
TD_DST_IP	[IP address] [Prefix Len]	Aggregated or per-host dst IP address/prefix rule
TD_SRC_IP	[IP address] [Prefix Len]	Aggregated or per-host src IP address/prefix rule
TD_TS	[Traffic Selector]	Traffic Selector, Format as per RFC6088

Figure 10: Protocol Attributes: Traffic Descriptors

Attribute	Format Clarification	Description
Properties		
ADMIN_STATE	[state]	Administrative state: enabled, disabled, virtual
SESSION_STATE	[state]	Session state: complete, incomplete, outdated
CLONE_REF	[PRT_ID]	Cloning of a rule based on referred port ID
ACT_DELAY	[delay]	Delay in ms before an updated rule takes effect at the Agent
PRT_BIDIR	[boolean]	When set, the rule per this port is applied bi-directionally
RESULT	[result]	Result of processing a message: success, failure

Figure 11: Protocol Attributes: Administrative Properties

Attribute	Format Clarification	Description
Monitors and Notification		
MONITOR	Monitor-ID Attribute [REPORT CONFIG]	A Monitor
REPORT_CONFIG	[8, REPORT-TYPE] [TYPE_SPECIFIC_INFO]	The type of report and type-specific configurations
PERIODIC_CONFIG	[32, period]	REPORT-TYPE is PERIODIC, period specifies the report interval (ms)
THRESHOLD_CONFIG	[32, low] [32, hi]	REPORT-TYPE is THRESHOLD, Low Threshold, High Threshold (at least one value required)
SCHEDULED_CONFIG	[32, time]	REPORT-TYPE is SCHEDULED, Time when NOTIFY is sent
EVENTS_CONFIG	*[EVENT_TYPE_ID]	List of Events that trigger the Monitor
DEREG_INFO	*[MONITOR_ID] [boolean]	Monitors to deregister, Boolean (optional) indicates if a successful DEREG triggers a NOTIFY with final data
NOTIFY_INFO	[32, Notification-Id] [MONITOR-ID] [32, TRIGGER] [32, timestamp]	ID used for Client ordering Monitor-ID of the NOTIFY, TRIGGER for the NOTIFY, Timestamp of when the attributes were recorded

Figure 12: Protocol Attributes: Monitor and Notify Attributes

TRIGGERS include but are not limited to the following values:

- o Events specified in the Event List of an EVENTS CONFIG
- o LOW\_THRESHOLD\_CROSSED

- o HIGH\_THRESHOLD\_CROSSED
- o PERIODIC\_REPORT
- o SCHEDULED\_REPORT
- o PROBED
- o DEREG\_FINAL\_VALUE

#### 4.3. Protocol Messages and Semantics

The following table specifies all protocol messages to create and modify a rule by creating and deleting logical Ports, adding and modifying properties and binding traffic descriptors to a port. Furthermore, messages can schedule tasks, such as monitoring, at an Agent or probe the status of the scheduled task from a Client. Additional messages enable the Data-Plane to notify or query the Control-Plane through the Agent and Client functions.

Message	Description
Messages issued by the FPC Client	
PRT_ADD	Add a logical port
PRT_DEL	Delete a logical port
PROP_ADD	Add a property to a logical port
PROP_MOD	Modify a property of a logical port
PROP_DEL	Delete a property from a logical port
TD_ADD	Add traffic descriptor to a logical port
TD_MOD	Modify an existing traffic descriptor
TD_DEL	Delete an existing traffic descriptor
MONITOR_REG	Install a monitor at an Agent. The message includes information about the attribute to monitor and the reporting method.
MONITOR_DEREG	Remove a monitor at an Agent.
PROBE	Probe the status of a registered event
Messages issued by the FPC Agent	
NOTIFY	Notify the Client about the status of a monitored attribute per the reporting method (periodic / event trigger / probed)
QUERY	Query the Client about missing rules/states

Figure 13: Protocol Messages

#### 4.4. Protocol Operation

The following list comprises a more detailed description of each message's semantic.

An FPC Client and Agent MUST identify themselves using the CLI\_ID and AGT\_ID respectively to ensure that for all transactions a recipient of an FPC message can unambiguously identify the sender of the FPC

message. A Client MAY direct the Agent to enforce a rule in a particular DPN by including a DPN\_ID value. Otherwise the Agent selects a suitable DPN to enforce a rule and notifies the Client about the selected DPN using the DPN\_ID.

- o PRT\_ADD - Issued by a Client to add a new logical port at an Agent. An Agent receiving the PRT\_ADD message identifies the new port according to the included port identifier (PRT\_ID). The Agent adds a new port into its conceptual data structures using the port identifier as key. Optionally, the PRT\_ADD message MAY include properties as well as traffic descriptors, which are bound and refer to the new port. This enables a Client to issue a new configuration in a single transaction with an Agent. A Client MAY assign a port to a group of ports and indicate the associated port group identifier (PRT\_GROUP\_ID) in the PRT\_ADD message.
- o PRT\_DEL - Used by a Client to delete a port. An Agent receiving such message MUST delete all properties associated with the identified port.
- o PROP\_ADD - Used by the Client to add a new property to an existing port. The property is unambiguously identified through a property identifier (PRT\_PROP\_ID). All traffic, which is directed to this port is treated according to the existing and newly added property. Optionally, the PROP\_ADD message can include traffic descriptors, which refer to the port to which the properties are bound. This enables a Client to add new rules to the existing port to which the new properties have been bound in a single transaction.
- o PROP\_MOD - Used by a Client to modify an existing property. For example, a tunnel property can be changed to direct traffic to a different tunnel endpoint in case of a mobile node's handover. Optionally, the PROP\_MOD message can include rules descriptions, which refer to the port whose properties are modified. This enables a Client to add new rules to the existing port whose properties have been modified in a single transaction.
- o PROP\_DEL - Used by a Client to delete one or multiple properties, each being identified by a property identifier.
- o TD\_ADD - Used by a Client to add a traffic descriptor to a port. The traffic descriptor SHOULD unambiguously identify aggregated traffic (longest prefix), per host IP traffic or per-flow traffic in the TD\_ADD command and bind the identified traffic to a port. Traffic descriptors are carried in a TD\_CONTAINER, which allows the identification of a traffic description as well as the indication if a traffic handling priority in case the sole traffic

description does not suffice unambiguous traffic matching. An Agent receiving a TD\_ADD command MUST add the traffic descriptor to its local conceptual data structures and apply commands for local configuration to add the new traffic descriptor to the rule on the DPN. Multiple traffic descriptors can bind to the same port. All traffic captured by the traffic descriptor will experience the same treatment per the properties which bind to that port.

- o TD\_MOD - Used by a Client to modify an existing traffic descriptor. An Agent receiving such messages MUST apply commands to the local configuration and update the rule on the DPN accordingly.
- o TD\_DEL - Used to remove an existing traffic descriptor from a port. The Agent receiving such messages MUST delete the identified traffic descriptor from the local configuration and update the rule on the DPN accordingly.
- o MONITOR\_REG - Used by a Client to install a monitor at an Agent. A monitor contains the monitor id, attribute to monitor, and optional reporting configuration. The attribute may be any ID with the exception of MONITOR\_ID and EVENT\_TYPE\_ID. When a Monitor registration is applied, the reporting configuration MUST be applicable to the attribute monitored, e.g. a Monitor using a Threshold configuration cannot be applied to a Port but it can be applied to a numeric Port Property. Four report types are defined: (1) Periodic reporting specifies an interval by which a NOTIFY is sent to the Client, (2) Event reporting specifies a list of EVENT\_TYPE\_IDS that, if they occur and are related to the monitored attribute, will result in sending a NOTIFY to the Client, (3) Scheduled reporting specifies the time (in seconds since Jan 1, 1970) when a NOTIFY for the monitor should be sent to the Client. Once this Monitor's NOTIFY is completed the Monitor is automatically de-registered, (4) Threshold reporting specifies one or both of a low and high threshold. When these values are crossed a corresponding NOTIFY is sent to the Client. All monitored data can be requested by the Client at any time using the PROBE message. Thus, reporting configuration is optional and when not present only PROBE messages may be used for monitoring. If a SCHEDULED or PERIODIC configuration is provided during registration with the time related value (time or period respectively) of 0 a NOTIFY is immediately sent and the monitor is immediately de-registered. This method should when a MONITOR has not been installed, an immediate NOTIFY is sufficient for the Client's needs and the Client has no further need for the monitor to be registered. An Agent may reject a registration if it or the DPN has insufficient resources.



- o MONITOR\_DEREG - Used by a Client to remove a monitor from an Agent. The message identifies one or multiple monitors by including the MONITOR\_ID. The message also includes an optional Boolean value that, when true, will result in NOTIFY messages being sent for the MONITOR\_ID to the Client. When a monitor has a reporting configuration of SCHEDULED it is automatically de-registered after the NOTIFY occurs. An Agent or DPN may temporarily suspend monitoring if insufficient resources exist. In such a case the Agent MUST notify the Client.
- o PROBE - Used by a Client to retrieve information about a previously installed monitor. The PROBE message SHOULD identify one or more monitors by means of including the associated monitor identifier. An Agent receiving a PROBE message SHOULD send the requested information in a single or multiple NOTIFY messages.
- o NOTIFY - Used by an Agent to report the status of a monitor to a Client. This message contains the MONITOR\_ID, a NOTIFICATION\_ID to permit the Client to distinguish amongst many monitoring related requests, a TRIGGER that caused the NOTIFY message, the timestamp of when the monitored information was record for the message along with the value of the monitored attribute.
- o QUERY - Used by an Agent to request an update of port properties via a Client. The Agent adds one or multiple port identifiers to the QUERY message to request all properties associated with the identified port(s). The Agent MAY request the update of particular properties associated with a port by including the property and its identifier. As result of processing a QUERY message, the Client sends one or multiple PROP\_MOD messages with the requested properties to the Agent.

All messages sent from a Client to an Agent MUST be acknowledged by the Agent. The response must include all attributes as well as status information, which indicates the result of processing the message, using the RESULT property. In case the processing of the message results in a failure, the Agent sets the RESULT accordingly and MAY clear the property or traffic descriptor, which caused the failure, in the response.

A Client MAY add a property to a port without providing all required details of the attribute's value. In such case the Agent SHOULD determine the missing details and provide the completed property description back to the Client. In case the Agent cannot determine the missing value of an attribute's value per the Client's request, it leaves the attribute's value cleared in the response and sets the RESULT to failure. As example, the Control-Plane needs to setup a tunnel configuration in the Data-Plane but has to rely on the Agent

to determine the tunnel endpoint which is associated with the DPN that enforces the rule. The Client adds the tunnel property attribute to the FPC message and clears the value of the attribute (e.g. IP address of the local tunnel endpoint). The Agent determines the tunnel endpoint and includes the completed tunnel property in its response to the Client.

The following list provides information on the use and semantics of attributes for traffic treatment:

- o PROP\_TUN - Defines the properties for encapsulation into different tunnel headers. The property includes IP address information of tunnel endpoints as well as a type identifier specifying the encapsulation type. Further attributes may be included to provide information which is relevant for the configuration and initialization of the tunnel.
- o PROP\_REWR - Defines the properties for IP address and port re-write.
- o PROP\_QOS - Defines the QoS properties in terms of a known index type, e.g. LTE's Quality Class Index (QCI), and its value (QCI 1..9), as well as a Differentiated Services Code Point (DSCP) to classify and mark packets. Additional QoS attributes may follow, to define Guaranteed Bit Rate (GBR) and Maximum Bit Rate (MBR) bounds. PROP\_QOS\_GBR and PROP\_QOS\_MBR attributes can apply to a single port or multiple ports. The latter is required to configure aggregate bounds, such as Aggregate Maximum Bit Rate (AMBR), taking traffic, which is forwarded through different ports (hence experiencing different treatment), into account. In such case the GBR/MBR attributes append multiple PRT\_ID attributes to identify the ports which are to be monitored to determine the aggregated view of the bit rate. As alternative to binding a PROP\_QOS\_MBR property to each port whose traffic is to be taken into account for Aggregate Maximum Bitrate (AMBR) metering, a Client can create a separate port with a single PROP\_QOS\_MBR property. Other ports, whose traffic is to be metered per the AMBR, can refer to the port with the PROP\_QOS\_MBR property using the PROP\_CONCAT property. The scope of attributes for QoS is aligned to [RFC7222]. The Allocation and Retention Priority (ARP) as per [RFC7222] is not present in the list of QoS-specific attributes, since ARP is treated and kept in the Control-Plane for granting requests for new resources and QoS, as well as for preempting other QoS configuration, if needed.
- o PROP\_QOS\_GBR - Defines the GBR bound for traffic associated with a port.

- o PROP\_QOS\_MBR - Defines the MBR bound for traffic associated with a port.
- o PROP\_GW - Defines a Next Hop IP address, to which packets are forwarded. Using this attribute, the Control-Plane can configure a host-route in the Data-Plane to deviate from default routes.
- o PROP\_CPY\_FORW - Refers to a different port and results in treatment of a copy of packets per the properties bound to the referred port.
- o PROP\_DROP - Defines a treatment action to drop packets of traffic associated with a port. As example, this treatment action can be used to enforce gating rules and filter traffic which does not match any traffic descriptor.
- o PROP\_CONCAT - Traffic can be treated per properties bound to concatenated ports. After treatment of traffic according to the properties of a port, additional treatment actions per the properties bound to a separate port, which is referred to in the PROP\_CONCAT property, apply to the traffic. As example, port concatenation can be used to enable AMBR metering to traffic which is associated with multiple other ports.
- o PROP\_NSH - Defines the properties for a Network Service Header (NSH). The header is included to the classified IP flows.

Unlike descriptors, overlapping or contradictory properties cannot be resolved by the Agent. For example, adding address translation related properties and a Drop property to a single port may result in needless activity in the DPN or it may reflect a temporary administrative activity where the port must Drop traffic. Other properties may be intentionally set, e.g. a property that invokes and accounting activity and a Drop property present on the same port. The FPC Client MUST avoid situations where contradictory properties or those that result in unnecessary activity are added to ports. Rather, in such situations, multiple ports MUST be used. In some obvious cases the Agent MAY raise a warning but a contradictory action.

The following list provides information on the use and semantics of administrative properties:

- o ADMIN\_STATE - A Client can apply an administrative state to a port indicating the desired operational status of a port (enabled, disabled, virtual). An Agent, which receives a message without ADMIN\_STATE property, SHOULD consider the port to be 'enabled'.

- o SESSION\_STATE - A Client can indicate to the Agent the status of a rule to serve Data-Plane traffic. A session state 'complete' confirms that a rule is valid and ready to serve Data-Plane traffic. A session state 'incomplete' hints to the Agent that more FPC message will arrive from the Client to complete a rule, whereas session state 'outdated' requires the Agent to solicit an update of the rule from the Client in case a rule with session state 'complete' is desired. An Agent, which receives a message without SESSION\_STATE property, SHOULD assume the session state is 'complete'.
- o CLONE\_REF - Instead of repeatedly sending all properties and traffic descriptors for similar rules, a Client can take a clone of a previously configured rule as base for a new one by using the CLONE\_REF property with a PRT\_ADD message and refer to an existing port. The cloned port will be a copy of the referred port and serve as base for the new port. The cloned port will have its own port identifier, which will also be present in the port identifier portion of the property identifiers. After a cloned port has been created, it represents its own rule without any further dependency on the reference port which served as source to create the clone. A Client MAY apply updates to existing properties of the new port, as well as delete or add properties. Updates to the port in terms of new or changed properties and traffic descriptors MAY already come with the PRT\_ADD message or subsequently using messages to handle properties and traffic descriptors. A Client can use the CLONE\_REF property with messages to handle properties and traffic descriptors to achieve a different result. In such case these messages identify an existing port already and processing the CLONE\_REF property on the receiving Agent will result in a reset of the identified port to match the properties of the port referred to in the CLONE\_REF property.
- o ACT\_DELAY - A Client can use this property to define a delay in ms before an updated rule takes effect at an Agent, e.g. an administrative state 'enabled' will be enforced by the Agent after the delay per the Client's request.
- o PRT\_BIDIR - A Client uses this property to indicate to an Agent to apply a rule associated with a port bi-directionally. In case the PRT\_BIDIR property is absent in a message, the Agent assumes a rule applies uni-directionally.
- o RESULT - An Agent uses this property to signal to the Client in a response the result of processing a message.

Figure 14 illustrates an exemplary session life-cycle based on Proxy Mobile IPv6 registration via MAG Control-Plane function 1 (MAG-C1)

and handover to MAG Control-Plane function 2 (MAG-C2). Edge DPN1 represents the Proxy CoA after attachment, whereas Edge DPN2 serves as Proxy CoA after handover. As exemplary architecture, the FPC Agent and the network control function are assumed to be co-located with the Anchor-DPN, e.g. a Router.

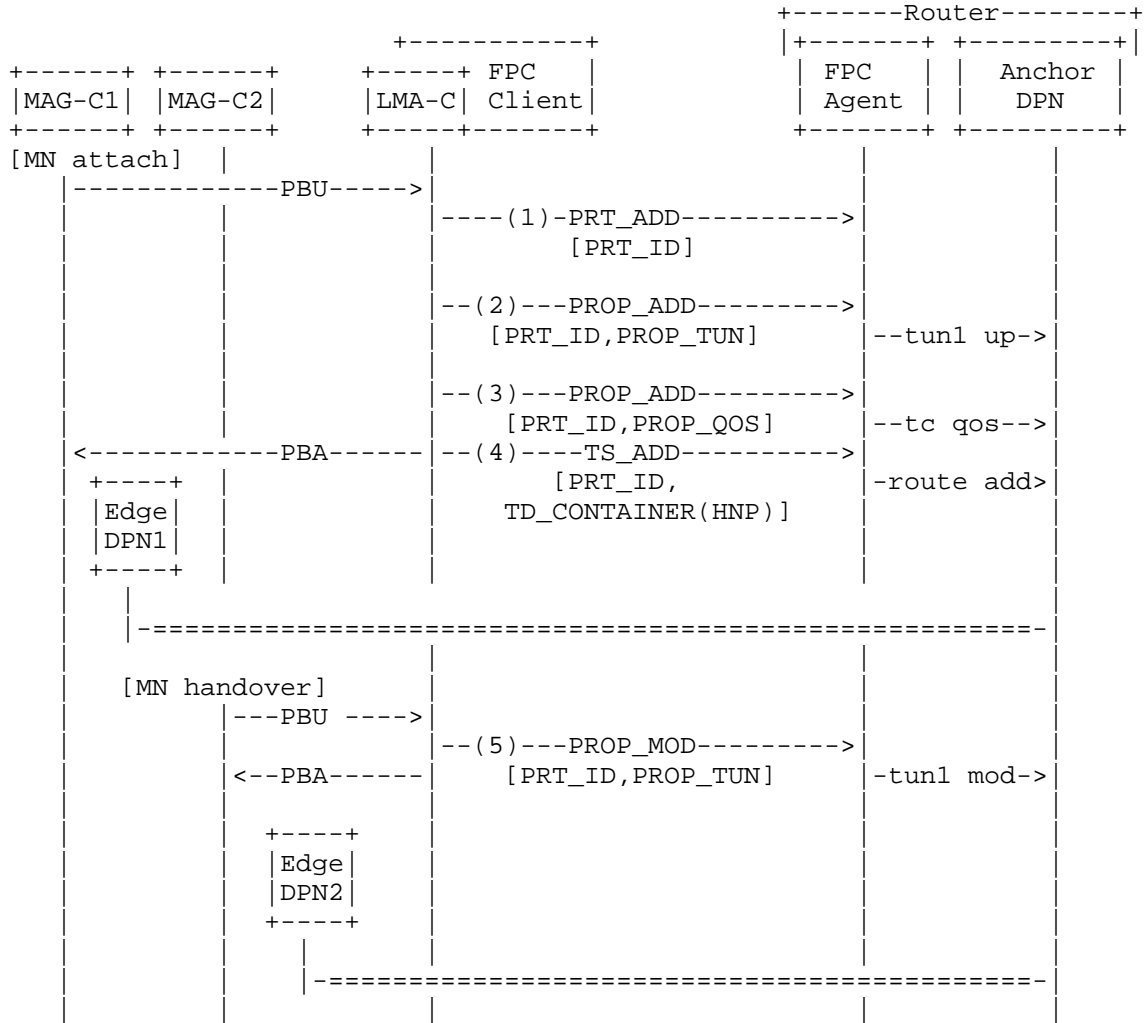


Figure 14: Exemplary Message Sequence (focus on FPC reference point)

After reception of the Proxy Binding Update (PBU) at the LMA Control-Plane function (LMA\_C), the LMA-C selects a suitable DPN, which serves as Data-Plane anchor to the mobile node's (MN) traffic. The

LMA-C adds a new logical port to the DPN to treat the MN's traffic (1) and includes a Port Identifier (PRT\_ID) to the PRT\_ADD command. The LMA-C identifies the selected Anchor DPN by including the associated DPN identifier.

Subsequently, the LMA-C adds properties to the new port. One property is added (2) to specify the forwarding tunnel type and endpoints (Anchor DPN, Edge DPN1). Another property is added (3) to specify the QoS differentiation, which the MN's traffic should experience. At reception of the properties, the FPC Agent utilizes local configuration commands to create the tunnel (tun1) as well as the traffic control (tc) to enable QoS differentiation. After configuration of port properties have been completed, the LMA binds the traffic description for the MN's traffic to the port by sending a TS\_CONTAINER to the Agent and identifying the MN's Nome Network Prefix (HNP) in the traffic descriptor. At the reception of the traffic descriptor, the Agent applies a new route to forward all traffic destined to the MN's HNP to the configured tunnel interface (tun1).

During handover, the LMA-C receives an updating PBU from the handover target MAG-C2. The PBU refers to a new Data-Plane node (Edge DPN2) to represent the new tunnel endpoint. The LMA-C sends a PROP\_MOD message (5) to the Agent to modify the existing tunnel property of the existing port and to update the tunnel endpoint from Edge DPN1 to Edge DPN2. Upon reception of the PROP\_MOD message, the Agent applies updated tunnel property to the local configuration.

To reduce the number of protocol handshakes between the LMA-C and the DPN, the LMA-C can append properties (PROP\_TUN, PROP\_QOS) and traffic descriptor attributes to the PRT\_ADD message, as illustrated in Figure 15.

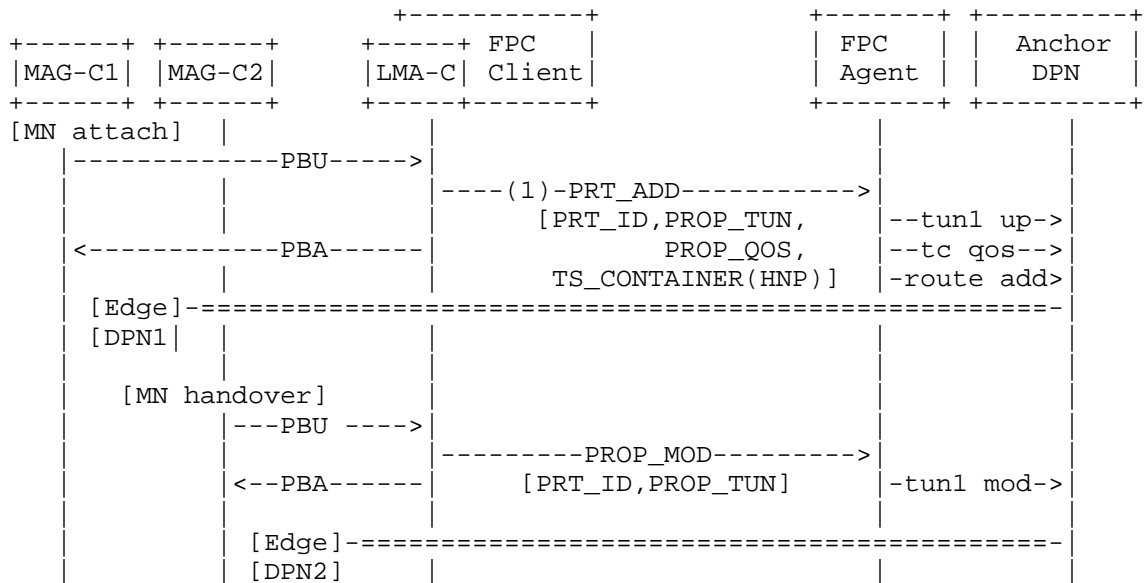


Figure 15: Example: Sequence for Message Aggregation (focus on FPC reference point)

## 5. Protocol to support Model II

### 5.1. Protocol Attributes

Attribute	Format	Description
IP Tunnel Attributes		
TUN_SRC_IP_ADDR	[IP address]	Tunnel Source IP
TUN_DST_IP_ADDR	[IP address]	Tunnel Destination IP
TUN_ENCAP_TYPE	[ENCAP_GRE, ENACP_UDP, ENCAP_IP]	Encapsulation Type
TUN_TYPE_UDP	[SRC_PRT, DST_PRT]	UDP Direction - Source or Destination
TUN_TYPE_GRE	[UPLINK_GRE_KEY, DOWNLINK_GRE_KEY]	GRE Tunnel Type

TUN_IF_MTU	[MTU]	Tunnel Interface MTU
TUN_PAYLOAD_TYPE	[PAYLOAD_IPV4, PAYLOAD_IPV6, PAYLOAD_DUAL]	Tunnel Payload Type
TUN_VENDOR_SPEC_PARAM	[OPAQUE]	Tunnel Vendor Specific Parameters
Route Management Attributes		
INPUT_IF	[IF_INDEX]	Input Interface
OUTPUT_IF	[IF_INDEX]	Output Interface
NEXT_HOP_IP_GW_ADDR	[IP address]	Next Hop IP Gateway Address
TRAFFIC_SELECTOR_ACL	TBD	
DST_IP_SUBNET	[IP prefix]	Destination IP Subnet
DST_IP_SUBNET_MASK	[IP prefix]	Destination IP Subnet Mask
QoS Attributes		
AMBR	[Unsigned Integer (32 bit)]	Aggregate Maximum Bitrate
GBR	[Unsigned Integer (32 bit)]	Guaranteed Bitrate
TCLASS	[Unsigned Integer (32 bit)]	Traffic Class
TFT	TBD	Traffic Flow Template
Optional Attributes		
NSH_HEADER	[Service Path Id]	NSH Header



	Service Index, TFT	
+	-----	+

Figure 16: Model II Protocol Attributes: Traffic Treatment

Attribute	Format	Description
Identifier		
TUNNEL_IF_ID	[IF_INDEX]	Tunnel Interface Identifier
VRF_ID	[Unsigned INT]	VRF Identifier
PBR_ID	[Unsigned INT]	Policy Based Routing Identifier
CTRL_PLANE_ID	IP address	Control-Plane Identifier
CONTEXT_ID	TBD	Context Identifier
QOS_SERVICE_ID	[Unsigned INT]	QoS Service Identifier
SESSION_ID	[Unsigned INT]	Session Identifier
ROUTE_ID	[Unsigned INT]	Route Identifier
Optional Identifiers		
SERVICE_PATH_ID	[24-bit identifier]	Service Path Identifier

Figure 17: Model II Protocol Attributes: Identifiers

## 5.2. Protocol Messages and Semantics

Message	Description
Tunnel Interface Management	
CREATE_TUNNEL_IF	Create a Tunnel Interface
DELETE_TUNNEL_IF	Delete a Tunnel Interface

UPDATE_TUNNEL_PARAMETER	Update a parameter of the specified tunnel
QUERY_TUNNEL_IF	Request Tunnel Interface information
Policy Route Management	
CREATE_POLICY_ROUTE	Create a Policy-based Route
DELETE_POLICY_ROUTE	Deletes a Policy-based Route
ADD_TRAFFIC_SELECTOR	Adds a Traffic Selector to a Policy-based Route
DELETE_TRAFFIC_SELECTOR	Removes a Traffic Selector from a Policy-based Route
QUERY_POLICY_ROUTE	Request Policy Route information
IP Route Management	
CREATE_IP_ROUTE	Create an IP Route
DELETE_IP_ROUTE	Delete an IP Route
QUERY_IP_ROUTE	Request IP Route information
IP QoS Management	
ALLOCATE_QOS_RESOURCES	Allocates QoS Resources, e.g. AMBR, to the specified Session / Context
DEALLOCATE_QOS_RESOURCES	Removes applies QoS Resources from the specified Session / Context
Optional Management	
ADD_NSH_HEADER	Add NSH Header for the classified IP flows
DELETE_NSH_HEADER	Remove NSH Header for the classified IP flows

Figure 18: Model II Protocol Messages

### 5.3. Protocol Operation

The following list comprises a description of each message's semantic.

- CREATE\_TUNNEL\_IF - Message can include TUN\_SRC\_IP\_ADDR, TUN\_DST\_IP\_ADDR, TUN\_ENCAP\_TYPE, TUN\_IF\_ID, TUN\_TYPE\_UDP, TUN\_TYPE\_GRE, TUN\_IF\_MTU, TUN\_PAYLOAD\_TYPE, TUN\_VENDOR\_SPEC\_PARAM, VRF\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- DELETE\_TUNNEL\_IF - Message can include TUN\_SRC\_IP\_ADDR, TUN\_DST\_IP\_ADDR, TUN\_ENCAP\_TYPE, TUN\_IF\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- UPDATE\_TUNNEL\_PARAMETER - Message can include TUN\_SRC\_IP\_ADDR, TUN\_DST\_IP\_ADDR, TUN\_ENCAP\_TYPE, TUN\_IF\_ID, TUN\_IF\_MTU, TUN\_PAYLOAD\_TYPE, TUN\_VENDOR\_SPEC\_PARAM, CTRL\_PLANE\_ID, CONTEXT\_ID.
- QUERY\_TUNNEL\_IF -
- CREATE\_POLICY\_ROUTE - Message can include INPUT\_IF, OUTPUT\_IF, NEXT\_HOP\_IP\_GW\_ADDR, VRF\_ID, PBR\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- DELETE\_POLICY\_ROUTE - Message can include PBR\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- ADD\_TRAFFIC\_SELECTOR - Message can include TRAFFIC\_SELECTOR\_ACL, PBR\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- DELETE\_TRAFFIC\_SELECTOR - Message can include TRAFFIC\_SELECTOR\_ACL, PBR\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- QUERY\_POLICY\_ROUTE -
- CREATE\_IP\_ROUTE - Message can include DST\_IP\_SUBNET, DST\_IP\_SUBNET\_MASK, OUTPUT\_IF, VRF\_ID, ROUTE\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- DELETE\_IP\_ROUTE - Message can include ROUTE\_ID, CTRL\_PLANE\_ID, CONTEXT\_ID.
- QUERY\_IP\_ROUTE -
- ALLOCATE\_QOS\_RESOURCES - Message can include AMBR, GBR, TCLASS, TFT, QOS\_SERVICE\_ID, CONTEXT\_ID.

- o DEALLOCATE\_QOS\_RESOURCES - Message can include Session\_ID, QOS\_SERVICE\_ID, CONTEXT\_ID.
- o ADD\_NSH\_HEADER - Message can include SERVICE\_PATH\_ID, SERVICE\_INDEX, TFT
- o DELETE\_NSH\_HEADER - Message can include SERVICE\_PATH\_ID, SERVICE\_INDEX, TFT

## 6. Security Considerations

Detailed protocol implementations for DMM Forwarding Policy Configuration must ensure integrity of the information exchanged between an FPC Client and an FPC Agent. Required Security Associations may be derived from co-located functions, which utilize the FPC Client and FPC Agent respectively.

## 7. IANA Considerations

This document provides a data model and protocol operation for DMM Forwarding Policy Configuration. YANG models are currently included in the Appendix and will be updated per the next revision of this document to specify the data model as well as to enable an implementation of the FPC protocol using RPC.

No actions from IANA are required. In case the semantics of this specification will be mapped to a particular wire protocol, authors of an associated separate document will approach IANA for the associated action to create a registry or add registry entries.

## 8. 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.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://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, <<http://www.rfc-editor.org/info/rfc6088>>.
- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc7429>>.

## 9.2. Informative References

- [RFC3344] Perkins, C., Ed., "IP Mobility Support for IPv4", RFC 3344, DOI 10.17487/RFC3344, August 2002, <<http://www.rfc-editor.org/info/rfc3344>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<http://www.rfc-editor.org/info/rfc5213>>.
- [RFC7222] Liebsch, M., Seite, P., Yokota, H., Korhonen, J., and S. Gundavelli, "Quality-of-Service Option for Proxy Mobile IPv6", RFC 7222, DOI 10.17487/RFC7222, May 2014, <<http://www.rfc-editor.org/info/rfc7222>>.

## Appendix A. YANG Data Model for the FPC protocol

These modules define Model I YANG definitions. Four modules are defined:

- o ietf-dmm-fpcp-base (fpcp-base) - Defines the base model for Model I FPC as defined in this document
- o ietf-pmip-qos (pmip-qos) - Defines proxy mobile IPv6 QoS parameters per RFC 7222
- o ietf-traffic-selectors-types (traffic-selectors) - Defines Traffic Selectors per RFC 6088

- o ietf-dmm-fpcp-pmip - Augments fpcp-base to include PMIP Traffic Selectors as a Traffic Descriptor subtype and pmip-qos QoS parameters, where applicable, as properties.

Note (2016-03-21): The YANG Data Model does not yet adopt all extensions per this version of the draft and will be updated shortly after the IETF95 meeting.

## A.1. FPC Base

### A.1.1. FPC Base YANG Model

```
module ietf-dmm-fpcp-base {
  namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpcp-base";
  prefix fpcp-base;

  import ietf-inet-types { prefix inet; }

  organization "IETF DMM Working Group";
  contact "Satoru Matsushima <satoru.matsushima@g.softbank.co.jp>";

  description
    "This module contains YANG definition for
    Forwarding Policy Configuration Protocol.(FPCP)";

  revision 2016-01-18 {
    description "Changes based on -01 version of FPCP draft.";
    reference "draft-ietf-dmm-fpc-cpdp-01";
  }

  typedef fpcp-name-type {
    type string;
    description "FPCP common name type";
  }

  typedef fpcp-carrier-id {
    type uint16;
    description "Carrier-ID";
  }

  typedef fpcp-network-id {
    type uint16;
    description "Carrier-ID";
  }

  typedef fpcp-client-id {
    type uint32;
    description "Client-ID";
  }
```

```
}

typedef fpcp-agent-id {
    type uint32;
    description "Agent-ID";
}

typedef fpcp-dpn-id {
    type uint32;
    description "Carrier-ID";
}

typedef fpcp-port-id {
    type uint32;
    description "PRT_ID";
}

typedef fpcp-property-id {
    type uint8;
    description "PRT_PROP_ID";
}

typedef fpcp-rule-id {
    type uint8;
    description "PRT_RULE_ID";
}

typedef fpcp-qos-class-identifier {
    type uint8 {
        range "1..9";
    }
    description "QCI";
}

typedef fpcp-qos-bandwidth {
    type uint32;
    description "Bandwith value in bit per second.";
}

identity tunnel-type {
    description
        "Base identity from which specific use of
        tunnels are derived.";
}

identity fpcp-tunnel-type {
    base "tunnel-type";
    description
```

```
        "Base identity from which specific tunnel
        types in FPCP uses are derived.";
    }

    identity ip-in-ip {
        base "fpcp-tunnel-type";
        description "IP-in-IP tunnel";
    }

    identity gtp {
        base "fpcp-tunnel-type";
        description "GTP-U tunnel";
    }

    identity gre {
        base "fpcp-tunnel-type";
        description "GRE tunnel";
    }

    identity service-function {
        description
        "Base identity from which specific
        service function types are derived.";
    }

    identity ip-protocol {
        description
        "Base identity from which specific
        IP protocol types are derived.";
    }

    identity property-type {
        description
        "Base identity of property";
    }

    identity property-qos {
        base "property-type";
        description
        "QoS property";
    }

    identity property-endpoint {
        base "property-type";
        description
        "Endpoint property";
    }
}
```



```
identity property-type-endpoint {
    base "property-type";
    description
        "Endpoint property";
}

identity qos-type {
    description
        "Base identity from which specific
        uses of QoS types are derived.";
}

identity fpcp-qos-type {
    base "qos-type";
    description
        "Base identity from which specific
        QoS types in FPCP uses are derived.";
}

identity fpcp-qos-type-gbr {
    base "fpcp-qos-type";
    description
        "A QoS Type for Guaranteed Bit Rate (GBR).";
}

identity fpcp-qos-type-mbr {
    base "fpcp-qos-type";
    description
        "A QoS Type for Maximum Bit Rate (MBR).";
}

identity fpcp-qos-index-type {
    base "qos-type";
}

identity fpcp-qos-index {
    base "fpcp-qos-index-type";
}

identity traffic-descriptor-type {
}

identity fpcp-traffic-descriptor {
    base "traffic-descriptor-type";
}

grouping carrier {
```

```
        description "Identify FPCP Carrier";
        leaf carrier-id {
            type fpcp-carrier-id;
            mandatory true;
            description "Carrier ID";
        }
    }

    grouping agent {
        description "AGT_ID to identify FPCP Agent";
        leaf agent-id {
            type fpcp-agent-id;
            description "Agent ID";
        }
    }

    grouping client {
        description "CLI_ID to identify FPCP Client";
        leaf client-id {
            type fpcp-client-id;
            description "Client ID";
        }
    }

    grouping network {
        description "Identify FPCP Network";
        leaf network-id {
            type fpcp-network-id;
            description "Network ID";
        }
    }

    grouping dpn {
        description "Identify FPCP Data-Plane Node";
        leaf dpn-id {
            type fpcp-dpn-id;
            description "DPN ID";
        }
    }

    grouping port {
        description "Identify FPCP Port";
        leaf port-id {
            type fpcp-port-id;
            description "Port-ID";
        }
    }
}
```

```
grouping property {
  description "Identify FPCP Property";
  leaf property-id {
    type fpcp-property-id;
    description "Property-ID";
  }
}

grouping rule {
  description "Identify FPCP Rule";
  leaf rule-id {
    type fpcp-rule-id;
    description "Rule-ID";
  }
}

grouping fpcp-carrier {
  description "Define FPCP network";
  uses carrier;
  uses agent;
  list client {
    key client-id;
    description "List of FPCP Clients";
    leaf name {
      type fpcp-name-type;
      description "Client Name";
    }
    uses client;
  }
  list dpn {
    key dpn-id;
    description "List of FPCP DPNs";
    leaf name {
      type fpcp-name-type;
      description "DPN Name";
    }
    uses dpn;
  }
}

grouping dpn-set {
  description "DPNs which consist a DPN set.";
  leaf name {
    type fpcp-name-type;
    description "DPN set name";
  }
  leaf network {
    type leafref {
```

```
        path "/fpcp-carriers/carrier/network/network-id";
    }
    description "Network-ID which a DPN-set is belonging to.";
}
leaf role {
    type enumeration {
        enum anchor-l3 {
            description "";
        }
        enum anchor-l2 {
            description "";
        }
        enum access {
            description "";
        }
    }
    description "Define DPNs role in data-plane.";
}
list endpoint-dp {
    key local-address;
    description "List of data-plane endpoint properties of a
        set of DPNs.";
    leaf local-address {
        type inet:ip-address;
        description "";
    }
    leaf remote-dpn {
        type leafref {
            path "/fpcp-carriers/carrier/dpn-group/name";
        }
        description "";
    }
    leaf default-tunnel-type {
        type identityref {
            base "fpcp-tunnel-type";
        }
        description "Tunnel Type";
    }
}
grouping dpn-set {
    description "DPNs which consist a DPN set.";
    leaf name {
        type fpcp-name-type;
        description "DPN set name";
    }
    leaf network {
        type leafref {
            path "/fpcp-carriers/carrier/network/network-id";
        }
    }
}
```

```
    }
    description "Network-ID which a DPN-set is belonging to.";
  }
  leaf role {
    type enumeration {
      enum anchor-l3 {
        description "";
      }
      enum anchor-l2 {
        description "";
      }
      enum access {
        description "";
      }
    }
    description "Define DPNs role in data-plane.";
  }
  list endpoint-dp {
    key local-address;
    description "List of data-plane endpoint properties of a
      set of DPNs.";
    leaf local-address {
      type inet:ip-address;
      description "";
    }
    leaf remote-dpn {
      type leafref {
        path "/fpcp-carriers/carrier/dpn-group/name";
      }
      description "";
    }
    leaf default-tunnel-type {
      type identityref {
        base "fpcp-tunnel-type";
      }
      description "Tunnel Type";
    }
  }
  list dpn {
    key dpn-id;
    uses dpn;
    description "DPN list in a DPN set";
  }
}

grouping tunnel-endpoints {
  description
    "PROP_TUN property as a set of tunnel endpoints";
```

```
    leaf tunnel-type {
        type identityref {
            base "fpcp-tunnel-type";
        }
        description "Tunnel Type";
    }
    leaf remote-address {
        type inet:ip-address;
        description "Remote endpoint";
    }
    leaf local-address {
        type inet:ip-address;
        description "Local endpoint";
    }
}

grouping gtp-attributes {
    description
        "GTP_CONF as GTP tunnel specific attributes";
    leaf remote-teid {
        type uint32;
        description "TEID of remote-endpoint";
    }
    leaf local-teid {
        type uint32;
        description "TEID of local-endpoint";
    }
}

grouping gre-attributes {
    description
        "GRE_CONF as GRE tunnel specific attribute";
    leaf key {
        type uint32;
        description "GRE_KEY";
    }
}

grouping rewriting-properties {
    description
        "PROP_REWR. TBD for which type of rewriting functions
        need to be defined";
    leaf type {
        type identityref {
            base service-function;
        }
        description "The type of service-function";
    }
}
```

```
grouping fpcp-qosattribute {
  leaf qci {
    type fpcp-qos-class-identifier;
  }
  leaf attributetype {
    type identityref {
      base fpcp-qos-type;
    }
    description "the attribute type";
  }
  leaf bandwidth {
    type fpcp-qos-bandwidth;
  }
}

grouping fpcp-qos-property {
  description "PROP_QOS";
  leaf name {
    type fpcp-name-type;
  }
  leaf qos-index-type {
    type identityref {
      base "fpcp-qos-index-type";
    }
  }
  choice index-type {
    case qci {
      when "../qos-index-type = 'fpcp-qos-index'";
      container uplink {
        uses fpcp-qosattribute;
      }
      container downlink {
        uses fpcp-qosattribute;
      }
    }
  }
}

grouping traffic-descriptor {
  description
    "Traffic descriptor group collects parameters to
    identify target traffic flow.";

  leaf destination-ip {
    type inet:ip-prefix;
    description "Rule of destination IP";
  }
}
```

```
    leaf source-ip {
      type inet:ip-prefix;
      description "Rule of source IP";
    }
  }

grouping fpcp-traffic-descriptor {
  leaf name {
    type fpcp-name-type;
  }
  leaf traffic-discriptor-type {
    type identityref {
      base "traffic-descriptor-type";
    }
  }

  choice descriptor-type {
    case fpcp-traffic-descriptor {
      when "../descriptor-type = 'fpcp-traffic-descriptor'";
      uses traffic-descriptor;
    }
  }
}

grouping fpcp-forwarding-rule {
  uses rule;
  uses fpcp-traffic-descriptor;
}

grouping fpcp-port-properties {
  description
    "A set of port property attributes";

  uses property;
  list attached-dpns {
    key name;
    leaf name {
      type fpcp-name-type;
      description "DPN group name of which port attached.";
    }
    description "Port attached DPN group list.";
  }
  container endpoints {
    description "Tunnel Endpoint";
    uses tunnel-endpoints;
    choice tunnel {
      description "Tunnel-Type";
    }
  }
}
```



```
        case gtp-u {
            when "tunnel-type = 'gtp'" {
                description "In case of GTP-U is tunnel-type";
            }
            uses gtp-attributes;
        }
        case gre {
            when "tunnel-type = 'gre'" {
                description "In case of GRE is tunnel-type";
            }
            uses gre-attributes;
        }
    }
}
container qos {
    description "QoS Type";
    uses fpcp-qos-property;
    list port-in-aggregated-bandwidth {
        key port-id;
        uses port;
    }
}
container rewriting {
    description "Rewriting function";
    uses rewriting-properties;
}
}

grouping port-field {
    description "Definition of attributes of port field";
    uses port;
    uses carrier;
    uses network;
}

// Container for configurations sets.

container fpcp-carriers {
    description "Attributes set of FPCP network";

    list carrier {
        key carrier-id;
        description "List of carriers";
        leaf name {
            type fpcp-name-type;
            description "FPCP Carrier name";
        }
        uses fpcp-carrier;
    }
}
```

```
    list network {
        key network-id;
        description "List of networks in a carrier.";
        leaf name {
            type fpcp-name-type;
            description "Define visible name of a network.";
        }
        uses network;
    }
    list dpn-group {
        key name;
        description "List of DPN groups in a carrier.";
        uses dpn-set;
    }
    list qos-profile {
        key name;
        uses fpcp-qos-property;
    }
    list traffic-descriptor {
        key name;
        uses fpcp-traffic-descriptor;
    }
}
}
```

// Port Entries

```
container port-entries {
    config false;
    description
        "This container binds set of traffic-descriptor and
        port properties to a port and lists them as a port entry.";

    list port-entry {
        key port-id;
        description "List of port entries";
        uses port-field;

        list property {
            key property-id;
            description "Attributes set of properties";
            uses fpcp-port-properties;
        }

        list forwarding-rule {
            key rule-id;
            description "Rule and traffic-descriptor";
            uses fpcp-forwarding-rule;
        }
    }
}
```

```

    }
  }
}

// PRT_ADD

rpc port_add {
  description "PRT_ADD";
  input {
    list adding-port {
      description "Ports that are added to an agent";
      uses port-field;
      list forwarding-rule {
        key rule-id;
        description "Rule and traffic-descriptor";
        uses fpcp-forwarding-rule;
      }
      list property {
        key property-id;
        description "Attributes set of properties";
        uses fpcp-port-properties;
      }
    }
  }
}

// PRT_DEL

rpc port_delete {
  description "PRT_DEL";
  input {
    list deleting-port {
      description "Ports that are deleted from an agent";
      uses port-field;
    }
  }
}

// PROP_ADD

rpc port_property_add {
  description "PROP_ADD";
  input {
    list adding-property {
      description "Properties that are added to an agent";
      uses port-field;
    }
  }
}

```

```

        list property {
            key property-id;
            description "Attributes set of properties";
            uses fpcp-port-properties;
        }
    }
}

// PROP_MOD

rpc port_property_modify {
    description "PROP_MOD";
    input {
        list modifying-property {
            description
                "Properties that are modified in an agent";
            uses port-field;

            list property {
                key property-id;
                description "Attributes set of properties";
                uses fpcp-port-properties;
            }
        }
    }
}

// PROP_DEL

rpc port_property_delete {
    description "PROP_DEL";
    input {
        list deleting-property {
            description
                "Target port/property-id of deleting properties";
            uses port-field;

            leaf property-id {
                type fpcp-property-id;
                mandatory true;
                description "Property ID";
            }
        }
    }
}

```

```
// RULE_ADD

rpc rule_add {
  description
    "TBD for input parameters of which RULE_ADD includes
    but now just traffic-descriptor.";
  input {
    list adding-rule {
      description "Rules that are added to an agent";
      uses port-field;

      list forwarding-rule {
        description "Added rule";
        uses fpcp-forwarding-rule;
      }
    }
  }
}

// RULE_MOD

rpc rule_modify {
  description
    "TBD for input parameters of which RULE_MOD includes
    but now just traffic-descriptor.";
  input {
    list modifying-rule {
      description "Rules that are modified in an agent";
      uses port-field;

      list forwarding-rule {
        description "Modified rule";
        uses fpcp-forwarding-rule;
      }
    }
  }
}

// RULE_DEL

rpc rule_delete {
  description
    "TBD for input parameters of which RULE_DEL includes
    but now just traffic-descriptor.";
  input {
    list deleting-rule {
      description "Rules that are deleted from an agent";
      uses port-field;
    }
  }
}
```

```

        list target-rule {
            description "Deleting rules";
            leaf target-rule-id {
                type fpcp-rule-id;
                mandatory true;
                description "Rule ID";
            }
        }
    }
}

// EVENT_REG

rpc event_register {
    description
        "TBD for registered parameters included in EVENT_REG.";
}

// PROBE

rpc probe {
    description
        "TBD for retrieved parameters included in PROBE.";
}

// NOTIFY

notification notify {
    description
        "TBD for which status and event are reported to client.";
}
}

```

Figure 19: FPC YANG base

## A.1.2. FPC Base tree

```

module: ietf-dmm-fpcp-base
  +--rw fpcp-carriers
  |   +--rw carrier* [carrier-id]
  |   |   +--rw name?                               fpcp-name-type
  |   |   +--rw carrier-id                           fpcp-carrier-id
  |   |   +--rw agent-id?                             fpcp-agent-id
  |   +--rw client* [client-id]
  |   |   +--rw name?                               fpcp-name-type

```

```

|   |--rw client-id      fpcp-client-id
+--rw dpn* [dpn-id]
|   |--rw name?         fpcp-name-type
|   |--rw dpn-id        fpcp-dpn-id
+--rw network* [network-id]
|   |--rw name?         fpcp-name-type
|   |--rw network-id    fpcp-network-id
+--rw dpn-group* [name]
|   |--rw name          fpcp-name-type
|   |--rw network?      -> /fpcp-carriers/carrier/network/network-id
|   |--rw role?         enumeration
+--rw endpoint-dp* [local-address]
|   |--rw local-address  inet:ip-address
|   |--rw remote-dpn?   -> /fpcp-carriers/carrier/dpn-group/name
|   |--rw default-tunnel-type? identityref
+--rw dpn* [dpn-id]
|   |--rw dpn-id        fpcp-dpn-id
+--rw qos-profile* [name]
|   |--rw name          fpcp-name-type
|   |--rw qos-index-type? identityref
|   |--rw (index-type)?
|   |   |--:(qci)
|   |   |   |--rw uplink
|   |   |   |   |--rw qci?          fpcp-qos-class-identifier
|   |   |   |   |--rw attributetype? identityref
|   |   |   |   |--rw bandwidth?    fpcp-qos-bandwidth
|   |   |   |--rw downlink
|   |   |   |   |--rw qci?          fpcp-qos-class-identifier
|   |   |   |   |--rw attributetype? identityref
|   |   |   |   |--rw bandwidth?    fpcp-qos-bandwidth
+--rw traffic-descriptor* [name]
|   |--rw name          fpcp-name-type
|   |--rw traffic-descriptor-type? identityref
|   |--rw (descriptor-type)?
|   |   |--:(fpcp-traffic-descriptor)
|   |   |   |--rw destination-ip?    inet:ip-prefix
|   |   |   |--rw source-ip?         inet:ip-prefix
+--ro port-entries
+--ro port-entry* [port-id]
|   |--ro port-id        fpcp-port-id
|   |--ro carrier-id      fpcp-carrier-id
|   |--ro network-id?     fpcp-network-id
+--ro property* [property-id]
|   |--ro property-id     fpcp-property-id
|   |--ro attached-dpns* [name]
|   |   |--ro name        fpcp-name-type
+--ro endpoints
|   |--ro tunnel-type?    identityref

```

```

| | | | | +---ro remote-address?    inet:ip-address
| | | | | +---ro local-address?    inet:ip-address
| | | | | +---ro (tunnel)?
| | | | | |   +---:(gtp-u)
| | | | | | |   +---ro remote-teid?    uint32
| | | | | | |   +---ro local-teid?    uint32
| | | | | |   +---:(gre)
| | | | | |   +---ro key?              uint32
| | | | | +---ro qos
| | | | | |   +---ro name?              fpcp-name-type
| | | | | |   +---ro qos-index-type?    identityref
| | | | | |   +---ro (index-type)?
| | | | | | |   +---:(qci)
| | | | | | |   +---ro uplink
| | | | | | | |   +---ro qci?          fpcp-qos-class-identifier
| | | | | | | |   +---ro attributetype? identityref
| | | | | | | |   +---ro bandwidth?    fpcp-qos-bandwidth
| | | | | | |   +---ro downlink
| | | | | | | |   +---ro qci?          fpcp-qos-class-identifier
| | | | | | | |   +---ro attributetype? identityref
| | | | | | | |   +---ro bandwidth?    fpcp-qos-bandwidth
| | | | | |   +---ro port-in-aggregated-bandwidth* [port-id]
| | | | | |   +---ro port-id          fpcp-port-id
| | | | | +---ro rewriting
| | | | | |   +---ro type?              identityref
+---ro forwarding-rule* [rule-id]
+---ro rule-id              fpcp-rule-id
+---ro name?                fpcp-name-type
+---ro traffic-descriptor-type? identityref
+---ro (descriptor-type)?
+---:(fpcp-traffic-descriptor)
+---ro destination-ip?      inet:ip-prefix
+---ro source-ip?           inet:ip-prefix

```

rpcs:

```

+---x port_add
|   +---w input
|   |   +---w adding-port*
|   |   |   +---w port-id?          fpcp-port-id
|   |   |   +---w carrier-id        fpcp-carrier-id
|   |   |   +---w network-id?       fpcp-network-id
|   |   |   +---w forwarding-rule* [rule-id]
|   |   |   |   +---w rule-id          fpcp-rule-id
|   |   |   |   +---w name?          fpcp-name-type
|   |   |   |   +---w traffic-descriptor-type? identityref
|   |   |   |   +---w (descriptor-type)?
|   |   |   |   |   +---:(fpcp-traffic-descriptor)
|   |   |   |   |   +---w destination-ip?      inet:ip-prefix

```



```

|           +---w source-ip?                inet:ip-prefix
+---w property* [property-id]
|   +---w property-id          fpcp-property-id
|   +---w attached-dpns* [name]
|   |   +---w name            fpcp-name-type
+---w endpoints
|   +---w tunnel-type?         identityref
|   +---w remote-address?     inet:ip-address
|   +---w local-address?      inet:ip-address
+---w (tunnel)?
|   +---:(gtp-u)
|   |   +---w remote-teid?      uint32
|   |   +---w local-teid?      uint32
|   +---:(gre)
|   |   +---w key?              uint32
+---w qos
|   +---w name?                fpcp-name-type
|   +---w qos-index-type?     identityref
|   +---w (index-type)?
|   |   +---:(qci)
|   |   |   +---w uplink
|   |   |   |   +---w qci?          fpcp-qos-class-identifier
|   |   |   |   +---w attributetype? identityref
|   |   |   |   +---w bandwidth?    fpcp-qos-bandwidth
|   |   |   +---w downlink
|   |   |   |   +---w qci?          fpcp-qos-class-identifier
|   |   |   |   +---w attributetype? identityref
|   |   |   |   +---w bandwidth?    fpcp-qos-bandwidth
|   |   +---w port-in-aggregated-bandwidth* [port-id]
|   |   +---w port-id          fpcp-port-id
+---w rewriting
|   +---w type?                identityref
+---x port_delete
|   +---w input
|   |   +---w deleting-port*
|   |   |   +---w port-id?          fpcp-port-id
|   |   |   +---w carrier-id        fpcp-carrier-id
|   |   |   +---w network-id?       fpcp-network-id
+---x port_property_add
|   +---w input
|   |   +---w adding-property*
|   |   |   +---w port-id?          fpcp-port-id
|   |   |   +---w carrier-id        fpcp-carrier-id
|   |   |   +---w network-id?       fpcp-network-id
|   |   +---w property* [property-id]
|   |   |   +---w property-id        fpcp-property-id
|   |   |   +---w attached-dpns* [name]
|   |   |   |   +---w name            fpcp-name-type

```

```

+---w endpoints
|   +---w tunnel-type?      identityref
|   +---w remote-address?   inet:ip-address
|   +---w local-address?    inet:ip-address
|   +---w (tunnel)?
|       +---:(gtp-u)
|           |   +---w remote-teid?      uint32
|           |   +---w local-teid?      uint32
|       +---:(gre)
|           +---w key?                  uint32
+---w qos
|   +---w name?                                fpcp-name-type
|   +---w qos-index-type?                      identityref
|   +---w (index-type)?
|       +---:(qci)
|           +---w uplink
|               |   +---w qci?          fpcp-qos-class-identifier
|               |   +---w attributetype? identityref
|               |   +---w bandwidth?    fpcp-qos-bandwidth
|           +---w downlink
|               |   +---w qci?          fpcp-qos-class-identifier
|               |   +---w attributetype? identityref
|               |   +---w bandwidth?    fpcp-qos-bandwidth
|   +---w port-in-aggregated-bandwidth* [port-id]
|   +---w port-id      fpcp-port-id
+---w rewriting
|   +---w type?      identityref
+---x port_property_modify
+---w input
+---w modifying-property*
+---w port-id?      fpcp-port-id
+---w carrier-id    fpcp-carrier-id
+---w network-id?   fpcp-network-id
+---w property* [property-id]
+---w property-id    fpcp-property-id
+---w attached-dpns* [name]
|   +---w name      fpcp-name-type
+---w endpoints
|   +---w tunnel-type?      identityref
|   +---w remote-address?   inet:ip-address
|   +---w local-address?    inet:ip-address
|   +---w (tunnel)?
|       +---:(gtp-u)
|           |   +---w remote-teid?      uint32
|           |   +---w local-teid?      uint32
|       +---:(gre)
|           +---w key?                  uint32
+---w qos

```

```

      +---w name?                fpcp-name-type
      +---w qos-index-type?      identityref
      +---w (index-type)?
      |   +---:(qci)
      |   |   +---w uplink
      |   |   |   +---w qci?        fpcp-qos-class-identifier
      |   |   |   +---w attributetype? identityref
      |   |   |   +---w bandwidth?  fpcp-qos-bandwidth
      |   |   +---w downlink
      |   |   |   +---w qci?        fpcp-qos-class-identifier
      |   |   |   +---w attributetype? identityref
      |   |   |   +---w bandwidth?  fpcp-qos-bandwidth
      |   +---w port-in-aggregated-bandwidth* [port-id]
      |   +---w port-id          fpcp-port-id
      +---w rewriting
      |   +---w type?            identityref
+---x port_property_delete
  +---w input
    +---w deleting-property*
      +---w port-id?            fpcp-port-id
      +---w carrier-id          fpcp-carrier-id
      +---w network-id?         fpcp-network-id
      +---w property-id         fpcp-property-id
+---x rule_add
  +---w input
    +---w adding-rule*
      +---w port-id?            fpcp-port-id
      +---w carrier-id          fpcp-carrier-id
      +---w network-id?         fpcp-network-id
      +---w forwarding-rule*
        +---w rule-id?          fpcp-rule-id
        +---w name?             fpcp-name-type
        +---w traffic-discriptor-type? identityref
        +---w (descriptor-type)?
          +---:(fpcp-traffic-descriptor)
            +---w destination-ip?    inet:ip-prefix
            +---w source-ip?         inet:ip-prefix
+---x rule_modify
  +---w input
    +---w modifying-rule*
      +---w port-id?            fpcp-port-id
      +---w carrier-id          fpcp-carrier-id
      +---w network-id?         fpcp-network-id
      +---w forwarding-rule*
        +---w rule-id?          fpcp-rule-id
        +---w name?             fpcp-name-type
        +---w traffic-discriptor-type? identityref
        +---w (descriptor-type)?

```

```

|               +---:(fpcp-traffic-descriptor)
|               +---w destination-ip?          inet:ip-prefix
|               +---w source-ip?              inet:ip-prefix
+---x rule_delete
|   +---w input
|   +---w deleting-rule*
|       +---w port-id?          fpcp-port-id
|       +---w carrier-id       fpcp-carrier-id
|       +---w network-id?     fpcp-network-id
|       +---w target-rule*
|           +---w target-rule-id   fpcp-rule-id
+---x event_register
+---x probe
notifications:
+---n notify

```

Figure 20: FPC base tree

## A.2. FPC PMIP

### A.2.1. FPC PMIP YANG Model

```

module ietf-dmm-fpcp-pmip {
  namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpcp-pmip";
  prefix fpcp-pmip;

  import ietf-inet-types { prefix inet; }
  import ietf-dmm-fpcp-base { prefix fpcp-base; }
  import ietf-pmip-qos { prefix qos-pmip; }
  import ietf-traffic-selectors { prefix traffic-selectors; }

  organization "IETF DMM Working Group";
  contact "Satoru Matsushima <satoru.matsushima@g.softbank.co.jp>";

  description
    "This module contains YANG definition for
    Forwarding Policy Configuration Protocol.(FPCP)";

  revision 2016-01-19 {
    description "Changes based on -01 version of FPCP draft.";
    reference "draft-ietf-dmm-fpc-cpdp-01";
  }

  identity fpcp-qos-index-pmip {
    base "fpcp-base:fpcp-qos-index-type";
  }

  identity traffic-selector-mip6 {

```

```

    base "fpcp-base:traffic-descriptor-type";
}

grouping qosattribute-pmip {

    leaf dscp {
        type inet:dscp;
    }

    choice attribute {
        case per-mn-agg-max-dl {
            when "../attributetype = 'Per-MN-Agg-Max-DL-Bit-Rate-type'";
            leaf per-mn-agg-max-dl {
                type qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value;
            }
        }
        case per-mn-agg-max-ul {
            when "../attributetype = 'Per-MN-Agg-Max-UL-Bit-Rate-type'";
            leaf per-mn-agg-max-ul {
                type qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value;
            }
        }
        case per-session-agg-max {
            when "../attributetype = 'Per-Session-Agg-Max-DL-Bit-Rate-type'";
            when "../attributetype = 'Per-Session-Agg-Max-UL-Bit-Rate-type'";

            uses qos-pmip:Per-Session-Agg-Max-Bit-Rate-Value;
        }
        case agg-max-dl {
            when "../attributetype = 'Aggregate-Max-DL-Bit-Rate-type'";
            leaf agg-max-dl {
                type qos-pmip:Aggregate-Max-DL-Bit-Rate-Value;
            }
        }
        case agg-max-ul {
            when "../attributetype = 'Aggregate-Max-UL-Bit-Rate-type'";
            leaf agg-max-ul {
                type qos-pmip:Aggregate-Max-UL-Bit-Rate-Value;
            }
        }
        case gbr-dl {
            when "../attributetype = 'Guaranteed-DL-Bit-Rate-type'";
            leaf gbr-dl {
                type qos-pmip:Guaranteed-DL-Bit-Rate-Value;
            }
        }
        case gbr-ul {
            when "../attributetype = 'Guaranteed-UL-Bit-Rate-type'";

```

```
        leaf gbr-ul {
            type qos-pmip:Guaranteed-UL-Bit-Rate-Value;
        }
    }
}

// Configuration choice augmentation in the fpcp-base under the fpcp-carrier
s/carrier/qosprofile.
augment "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-profile/fp
cp-base:index-type" {
    case pmip {
        when "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-profi
le/fpcp-base:qos-index-type = 'fpcp-qos-index-pmip'";
        uses qosattribute-pmip;
    }
}

// Configuration choice augmentation in the fpcp-base under the fpcp-carrier
s/carrier/traffic-descriptor.
augment "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:traffic-descri
ptor/fpcp-base:descriptor-type" {
    case traffic-selector-mip6 {
        when "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:traffic-d
escriptor/fpcp-base:traffic-descriptor-type = 'traffic-selector-mip6'";
        uses traffic-selectors:traffic-selector;
    }
}

// Operational choice augmentation in the fpcp-base under the port-entries/p
ort-entry/property.
augment "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:property/fpc
p-base:qos/fpcp-base:index-type" {
    case pmip {
        when "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:propert
y/fpcp-base:qos/fpcp-base:qos-index-type = 'fpcp-qos-index-pmip'";
        uses qosattribute-pmip;
    }
}

// Operational choice augmentation in the fpcp-base under the port-entries/p
ort-entry/forwarding-rule.
augment "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:forwarding-r
ule/fpcp-base:descriptor-type" {
    case traffic-selector-mip6 {
        when "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:forward
ing-rule/fpcp-base:traffic-descriptor-type = 'traffic-selector-mip6'";
        uses traffic-selectors:traffic-selector;
    }
}

// RPC choice augmentation in the fpcp-base under "port_add" rpc.
augment "/fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp-base
:property/fpcp-base:qos/fpcp-base:index-type" {
    case pmip {
        when "/fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp
-base:property/fpcp-base:qos/fpcp-base:qos-index-type = 'fpcp-qos-index-pmip'";
        uses qosattribute-pmip;
    }
}

augment "/fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp-base
:forwarding-rule/fpcp-base:descriptor-type" {
```



```

        case traffic-selector-mip6 {
            when "/fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp-
-base:forwarding-rule/fpcp-base:traffic-descriptor-type = 'traffic-selector-mip6
'";
                uses traffic-selectors:traffic-selector;
        }
    }

    // RPC choice augmentation in the fpcp-base under "port_property_add" rpc.
    augment "/fpcp-base:port_property_add/fpcp-base:input/fpcp-base:adding-prope
rty/fpcp-base:property/fpcp-base:qos/fpcp-base:index-type" {
        case pmip {
            when "/fpcp-base:port_property_add/fpcp-base:input/fpcp-base:adding-
property/fpcp-base:property/fpcp-base:qos/fpcp-base:qos-index-type = 'fpcp-qos-i
ndex-pmip'";
                uses qosattribute-pmip;
            }
        }

    // RPC choice augmentation in the fpcp-base under "port_property_modify" rpc
    .
    augment "/fpcp-base:port_property_modify/fpcp-base:input/fpcp-base:modifying
-property/fpcp-base:property/fpcp-base:qos/fpcp-base:index-type" {
        case pmip {
            when "/fpcp-base:port_property_modify/fpcp-base:input/fpcp-base:modi
fying-property/fpcp-base:property/fpcp-base:qos/fpcp-base:qos-index-type = 'fpcp
-qos-index-pmip'";
                uses qosattribute-pmip;
            }
        }

    // RPC choice augmentation in the fpcp-base under "rule_add" rpc.
    augment "/fpcp-base:rule_add/fpcp-base:input/fpcp-base:adding-rule/fpcp-base
:forwarding-rule/fpcp-base:descriptor-type" {
        case traffic-selector-mip6 {
            when "/fpcp-base:rule_add/fpcp-base:input/fpcp-base:adding-rule/fpcp
-base:forwarding-rule/fpcp-base:traffic-descriptor-type = 'traffic-selector-mip6
'";
                uses traffic-selectors:traffic-selector;
            }
        }

    // RPC choice augmentation in the fpcp-base under "rule_modify" rpc.
    augment "/fpcp-base:rule_modify/fpcp-base:input/fpcp-base:modifying-rule/fpc
p-base:forwarding-rule/fpcp-base:descriptor-type" {
        case traffic-selector-mip6 {
            when "/fpcp-base:rule_modify/fpcp-base:input/fpcp-base:modifying-rul
e/fpcp-base:forwarding-rule/fpcp-base:traffic-descriptor-type = 'traffic-selecto
r-mip6'";
                uses traffic-selectors:traffic-selector;
            }
        }
    }
}

```

Figure 21: caption1

#### A.2.2. FPC PMIP tree

```

module: ietf-dmm-fpcp-pmip
augment /fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-profile/fpcp-ba
se:index-type:
    +--:(pmip)

```



+--rw dscp?

inet:dscp

Liebsch, et al.

Expires September 22, 2016

[Page 61]

```

    +--rw (attribute)?
      +---:(per-mn-agg-max-dl)
        |   +--rw per-mn-agg-max-dl?      qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value
      +---:(per-mn-agg-max-ul)
        |   +--rw per-mn-agg-max-ul?      qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
      +---:(per-session-agg-max)
        |   +--rw max-dl                  uint32
        |   +--rw service-flag            boolean
        |   +--rw exclude-flag            boolean
      +---:(agg-max-dl)
        |   +--rw agg-max-dl?              qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
      +---:(agg-max-ul)
        |   +--rw agg-max-ul?              qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
      +---:(gbr-dl)
        |   +--rw gbr-dl?                  qos-pmip:Guaranteed-DL-Bit-Rate-Value
      +---:(gbr-ul)
        |   +--rw gbr-ul?                  qos-pmip:Guaranteed-UL-Bit-Rate-Value
      +--rw gbr-ul?                        qos-pmip:Guaranteed-UL-Bit-Rate-Value
augment /fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:traffic-descriptor/
fpcp-base:descriptor-type:
  +---:(traffic-selector-mip6)
    +--rw ts-format?                      identityref
    +--rw start-ipsec-spi?                 ipsec-spi
    +--rw end-ipsec-spi?                   ipsec-spi
    +--rw start-source-port?               inet:port-number
    +--rw end-source-port?                 inet:port-number
    +--rw start-destination-port?          inet:port-number
    +--rw end-destination-port?            inet:port-number
    +--rw start-source-address-v4?         inet:ipv4-address
    +--rw end-source-address-v4?          inet:ipv4-address
    +--rw start-destination-address-v4?    inet:ipv4-address
    +--rw end-destination-address-v4?     inet:ipv4-address
    +--rw start-ds?                        inet:dscp
    +--rw end-ds?                          inet:dscp
    +--rw start-protocol?                  uint8
    +--rw end-protocol?                    uint8
    +--rw start-source-address-v6?         inet:ipv6-address
    +--rw end-source-address-v6?          inet:ipv6-address
    +--rw start-destination-address-v6?    inet:ipv6-address
    +--rw end-destination-address-v6?     inet:ipv6-address
    +--rw start-flow-label?                inet:ipv6-flow-label
    +--rw end-flow-label?                  inet:ipv6-flow-label
    +--rw start-traffic-class?             inet:dscp
    +--rw end-traffic-class?               inet:dscp
    +--rw start-next-header?               uint8
    +--rw end-next-header?                 uint8
augment /fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:property/fpcp-base:
qos/fpcp-base:index-type:
  +---:(pmip)
    +--ro dscp?                           inet:dscp
    +--ro (attribute)?

```

```

    +---:(per-mn-agg-max-dl)
    |   +---ro per-mn-agg-max-dl?      qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value
    +---:(per-mn-agg-max-ul)
    |   +---ro per-mn-agg-max-ul?      qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
    +---:(per-session-agg-max)
    |   +---ro max-dl                  uint32
    |   +---ro service-flag            boolean
    |   +---ro exclude-flag            boolean
    +---:(agg-max-dl)
    |   +---ro agg-max-dl?              qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
    +---:(agg-max-ul)
    |   +---ro agg-max-ul?              qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
    +---:(gbr-dl)
    |   +---ro gbr-dl?                  qos-pmip:Guaranteed-DL-Bit-Rate-Value
    +---:(gbr-ul)
    |   +---ro gbr-ul?                  qos-pmip:Guaranteed-UL-Bit-Rate-Value
augment /fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:forwarding-rule/f
pcp-base:descriptor-type:
    +---:(traffic-selector-mip6)
    +---ro ts-format?                  identityref
    +---ro start-ipsec-spi?             ipsec-spi
    +---ro end-ipsec-spi?               ipsec-spi
    +---ro start-source-port?           inet:port-number
    +---ro end-source-port?             inet:port-number
    +---ro start-destination-port?      inet:port-number
    +---ro end-destination-port?        inet:port-number
    +---ro start-source-address-v4?     inet:ipv4-address
    +---ro end-source-address-v4?       inet:ipv4-address
    +---ro start-destination-address-v4? inet:ipv4-address
    +---ro end-destination-address-v4?  inet:ipv4-address
    +---ro start-ds?                    inet:dscp
    +---ro end-ds?                      inet:dscp
    +---ro start-protocol?              uint8
    +---ro end-protocol?                uint8
    +---ro start-source-address-v6?     inet:ipv6-address
    +---ro end-source-address-v6?       inet:ipv6-address
    +---ro start-destination-address-v6? inet:ipv6-address
    +---ro end-destination-address-v6?  inet:ipv6-address
    +---ro start-flow-label?            inet:ipv6-flow-label
    +---ro end-flow-label?              inet:ipv6-flow-label
    +---ro start-traffic-class?         inet:dscp
    +---ro end-traffic-class?           inet:dscp
    +---ro start-next-header?           uint8
    +---ro end-next-header?             uint8
augment /fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp-base:prop
erty/fpcp-base:qos/fpcp-base:index-type:
    +---:(pmip)
    +---- dscp?                        inet:dscp
    +---- (attribute)?
    +---:(per-mn-agg-max-dl)

```

```

    | +---- per-mn-agg-max-dl?    qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value
+---:(per-mn-agg-max-ul)
    | +---- per-mn-agg-max-ul?    qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
+---:(per-session-agg-max)
    | +---- max-dl                uint32
    | +---- service-flag          boolean
    | +---- exclude-flag          boolean
+---:(agg-max-dl)
    | +---- agg-max-dl?           qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
+---:(agg-max-ul)
    | +---- agg-max-ul?           qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
+---:(gbr-dl)
    | +---- gbr-dl?               qos-pmip:Guaranteed-DL-Bit-Rate-Value
+---:(gbr-ul)
    | +---- gbr-ul?               qos-pmip:Guaranteed-UL-Bit-Rate-Value
augment /fpcp-base:port_add/fpcp-base:input/fpcp-base:adding-port/fpcp-base:forw
arding-rule/fpcp-base:descriptor-type:
+---:(traffic-selector-mip6)
    +---- ts-format?              identityref
    +---- start-ipsec-spi?         ipsec-spi
    +---- end-ipsec-spi?           ipsec-spi
    +---- start-source-port?       inet:port-number
    +---- end-source-port?         inet:port-number
    +---- start-destination-port?  inet:port-number
    +---- end-destination-port?    inet:port-number
    +---- start-source-address-v4? inet:ipv4-address
    +---- end-source-address-v4?   inet:ipv4-address
    +---- start-destination-address-v4? inet:ipv4-address
    +---- end-destination-address-v4? inet:ipv4-address
    +---- start-ds?                inet:dscp
    +---- end-ds?                  inet:dscp
    +---- start-protocol?          uint8
    +---- end-protocol?            uint8
    +---- start-source-address-v6? inet:ipv6-address
    +---- end-source-address-v6?   inet:ipv6-address
    +---- start-destination-address-v6? inet:ipv6-address
    +---- end-destination-address-v6? inet:ipv6-address
    +---- start-flow-label?        inet:ipv6-flow-label
    +---- end-flow-label?          inet:ipv6-flow-label
    +---- start-traffic-class?     inet:dscp
    +---- end-traffic-class?       inet:dscp
    +---- start-next-header?       uint8
    +---- end-next-header?         uint8
augment /fpcp-base:port_property_add/fpcp-base:input/fpcp-base:adding-property/f
pcp-base:property/fpcp-base:qos/fpcp-base:index-type:
+---:(pmip)
    +---- dscp?                    inet:dscp
    +---- (attribute)?
        +---:(per-mn-agg-max-dl)
            | +---- per-mn-agg-max-dl?    qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value

```

```

    +---:(per-mn-agg-max-ul)
    |   +----- per-mn-agg-max-ul?      qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
    +---:(per-session-agg-max)
    |   +----- max-dl                  uint32
    |   +----- service-flag            boolean
    |   +----- exclude-flag            boolean
    +---:(agg-max-dl)
    |   +----- agg-max-dl?              qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
    +---:(agg-max-ul)
    |   +----- agg-max-ul?              qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
    +---:(gbr-dl)
    |   +----- gbr-dl?                  qos-pmip:Guaranteed-DL-Bit-Rate-Value
    +---:(gbr-ul)
    |   +----- gbr-ul?                  qos-pmip:Guaranteed-UL-Bit-Rate-Value
augment /fpcp-base:port_property_modify/fpcp-base:input/fpcp-base:modifying-prop
erty/fpcp-base:property/fpcp-base:qos/fpcp-base:index-type:
    +---:(pmip)
    |   +----- dscp?                    inet:dscp
    |   +----- (attribute)?
    |       +---:(per-mn-agg-max-dl)
    |       |   +----- per-mn-agg-max-dl?      qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value
    |       +---:(per-mn-agg-max-ul)
    |       |   +----- per-mn-agg-max-ul?      qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value
    |       +---:(per-session-agg-max)
    |       |   +----- max-dl                  uint32
    |       |   +----- service-flag            boolean
    |       |   +----- exclude-flag            boolean
    |       +---:(agg-max-dl)
    |       |   +----- agg-max-dl?              qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
    |       +---:(agg-max-ul)
    |       |   +----- agg-max-ul?              qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
    |       +---:(gbr-dl)
    |       |   +----- gbr-dl?                  qos-pmip:Guaranteed-DL-Bit-Rate-Value
    |       +---:(gbr-ul)
    |       |   +----- gbr-ul?                  qos-pmip:Guaranteed-UL-Bit-Rate-Value
augment /fpcp-base:rule_add/fpcp-base:input/fpcp-base:adding-rule/fpcp-base:forw
arding-rule/fpcp-base:descriptor-type:
    +---:(traffic-selector-mip6)
    |   +----- ts-format?                identityref
    |   +----- start-ipsec-spi?          ipsec-spi
    |   +----- end-ipsec-spi?            ipsec-spi
    |   +----- start-source-port?        inet:port-number
    |   +----- end-source-port?          inet:port-number
    |   +----- start-destination-port?   inet:port-number
    |   +----- end-destination-port?     inet:port-number
    |   +----- start-source-address-v4?  inet:ipv4-address
    |   +----- end-source-address-v4?    inet:ipv4-address
    |   +----- start-destination-address-v4? inet:ipv4-address
    |   +----- end-destination-address-v4? inet:ipv4-address
    |   +----- start-ds?                  inet:dscp

```

```

+----- end-ds?                               inet:dscp
+----- start-protocol?                       uint8
+----- end-protocol?                         uint8
+----- start-source-address-v6?             inet:ipv6-address
+----- end-source-address-v6?               inet:ipv6-address
+----- start-destination-address-v6?        inet:ipv6-address
+----- end-destination-address-v6?          inet:ipv6-address
+----- start-flow-label?                     inet:ipv6-flow-label
+----- end-flow-label?                       inet:ipv6-flow-label
+----- start-traffic-class?                  inet:dscp
+----- end-traffic-class?                    inet:dscp
+----- start-next-header?                     uint8
+----- end-next-header?                       uint8
augment /fpcp-base:rule_modify/fpcp-base:input/fpcp-base:modifying-rule/fpcp-base:forwarding-rule/fpcp-base:descriptor-type:
+---:(traffic-selector-mip6)
+----- ts-format?                           identityref
+----- start-ipsec-spi?                      ipsec-spi
+----- end-ipsec-spi?                        ipsec-spi
+----- start-source-port?                    inet:port-number
+----- end-source-port?                      inet:port-number
+----- start-destination-port?               inet:port-number
+----- end-destination-port?                 inet:port-number
+----- start-source-address-v4?              inet:ipv4-address
+----- end-source-address-v4?                inet:ipv4-address
+----- start-destination-address-v4?         inet:ipv4-address
+----- end-destination-address-v4?           inet:ipv4-address
+----- start-ds?                             inet:dscp
+----- end-ds?                               inet:dscp
+----- start-protocol?                       uint8
+----- end-protocol?                         uint8
+----- start-source-address-v6?             inet:ipv6-address
+----- end-source-address-v6?               inet:ipv6-address
+----- start-destination-address-v6?         inet:ipv6-address
+----- end-destination-address-v6?           inet:ipv6-address
+----- start-flow-label?                     inet:ipv6-flow-label
+----- end-flow-label?                       inet:ipv6-flow-label
+----- start-traffic-class?                  inet:dscp
+----- end-traffic-class?                    inet:dscp
+----- start-next-header?                     uint8
+----- end-next-header?                       uint8

```

Figure 22: FPC PMIP tree

Authors' Addresses

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

Phone: +49 6221 4342146  
Email: [liebsch@neclab.eu](mailto:liebsch@neclab.eu)

Satoru Matsushima  
SoftBank  
1-9-1, Higashi-Shimbashi, Minato-Ku  
Tokyo 105-7322  
Japan

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

Sri Gundavelli  
Cisco  
170 West Tasman Drive  
San Jose, CA 95134  
USA

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

Danny Moses

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

Lyle Bertz  
6220 Sprint Parkway  
Overland Park KS, 66251  
USA

Email: [lyleb551144@gmail.com](mailto:lyleb551144@gmail.com)

DMM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 21, 2016

A. Yegin  
Unaffiliated  
K. Kweon  
J. Lee  
J. Park  
Samsung  
D. Moses  
Intel  
February 18, 2016

On Demand Mobility Management  
draft-ietf-dmm-ondemand-mobility-02

Abstract

Applications differ with respect to whether they need IP session continuity and/or IP address reachability. The network providing the same type of service to any mobile host and any application running on the host yields inefficiencies. This document describes a solution for taking the application needs into account in selectively providing IP session continuity and IP address reachability on a per-socket basis.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 21, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents



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

## Table of Contents

1. Introduction . . . . .	2
2. Notational Conventions . . . . .	4
3. Solution . . . . .	4
3.1. Types of IP Addresses . . . . .	4
3.2. Granularity of Selection . . . . .	5
3.3. On Demand Nature . . . . .	5
3.4. Conveying the Selection . . . . .	6
4. Backwards Compatibility Considerations . . . . .	8
4.1. Applications . . . . .	8
4.2. IP Stack in the Mobile Host . . . . .	8
4.3. Network Infrastructure . . . . .	8
5. Security Considerations . . . . .	9
6. IANA Considerations . . . . .	9
7. Acknowledgements . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Informative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

In the context of Mobile IP [RFC5563][RFC6275][RFC5213][RFC5944], following two attributes are defined for the IP service provided to the mobile hosts:

**IP session continuity:** The ability to maintain an ongoing IP session by keeping the same local end-point IP address throughout the session despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may change between two independent IP sessions, but that does not jeopardize the IP session continuity. IP session continuity is essential for mobile hosts to maintain ongoing flows without any interruption.

**IP address reachability:** The ability to maintain the same IP address for an extended period of time. The IP address stays the same across independent IP sessions, and even in the absence of any IP session. The IP address may be published in a long-term registry (e.g., DNS), and it is made available for serving incoming (e.g., TCP)

connections. IP address reachability is essential for mobile hosts to use specific/published IP addresses.

Mobile IP is designed to provide both IP session continuity and IP address reachability to mobile hosts. Architectures utilizing these protocols (e.g., 3GPP, 3GPP2, WIMAX) ensure that any mobile host attached to the compliant networks can enjoy these benefits. Any application running on these mobile hosts is subjected to the same treatment with respect to the IP session continuity and IP address reachability.

It should be noted that in reality not every application may need those benefits. IP address reachability is required for applications running as servers (e.g., a web server running on the mobile host). But, a typical client application (e.g., web browser) does not necessarily require IP address reachability. Similarly, IP session continuity is not required for all types of applications either. Applications performing brief communication (e.g., DNS client) can survive without having IP session continuity support.

Achieving IP session continuity and IP address reachability by using Mobile IP incurs some cost. Mobile IP protocol forces the mobile host's IP traffic to traverse a centrally-located router (Home Agent, HA), which incurs additional transmission latency and use of additional network resources, adds to the network CAPEX and OPEX, and decreases the reliability of the network due to the introduction of a single point of failure [I-D.ietf-dmm-requirements]. Therefore, IP session continuity and IP address reachability should be provided only when needed.

Furthermore, when an application needs session continuity, it may be able to satisfy that need by using a solution above the IP layer, such as MPTCP [RFC6824], SIP mobility [RFC3261], or an application-layer mobility solution. Those higher-layer solutions are not subject to the same issues that arise with the use of Mobile IP since they can utilize the most direct data path between the end-points. But, if Mobile IP is being applied to the mobile host, those higher-layer protocols are rendered useless because their operation is inhibited by the Mobile IP. Since Mobile IP ensures the IP address of the mobile host remains fixed (despite the location and movement of the mobile host), the higher-layer protocols never detect the IP-layer change and never engage in mobility management.

This document proposes a solution for the applications running on the mobile host to indicate whether they need IP session continuity or IP address reachability. The network protocol stack on the mobile host, in conjunction with the network infrastructure, would provide the required type of IP service. It is for the benefit of both the users

and the network operators not to engage an extra level of service unless it is absolutely necessary. So it is expected that applications and networks compliant with this specification would utilize this solution to use network resources more efficiently.

## 2. Notational Conventions

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

## 3. Solution

### 3.1. Types of IP Addresses

Three types of IP addresses are defined with respect to the mobility management.

#### - Fixed IP Address

This is what standard Mobile IP provides with a Home Address (HoA). The mobile host is configured a HoA from a centrally-located Home Network. Both IP session continuity and IP address reachability are provided to the mobile host with the help of a router in the Home Network (Home Agent, HA). This router acts as an anchor for the IP address of the mobile host.

#### - Sustained IP Address

This type of IP address provides IP session continuity but not IP address reachability. It is achieved by ensuring that the IP address used at the beginning of the session remains usable despite the movement of the mobile host. The IP address may change after the termination of the IP session(s), therefore it does not exhibit persistence.

A sustained IP address may be configured and maintained by using access network anchoring, corresponding network anchoring, or some other solution.

#### - Nomadic IP Address

This type of IP address provides neither IP session continuity nor IP address reachability. The IP address is obtained from the serving IP gateway and it is not maintained across gateway changes. In other words, the IP address may be released and replaced by a new IP address when the IP gateway changes due to the movement of the mobile host.

Applications running as servers at a published IP address require a Fixed IP Address. Long-standing applications (e.g., an SSH session) may also require this type of address. Those applications could use a Sustained IP Address, but that can produce sub-optimal results if the mobile host ends up far from the anchor gateway. Enterprise applications that connect to an enterprise network via virtual LAN require a Fixed IP Address.

Applications with short-lived transient IP sessions can use Sustained IP Addresses. For example: Web browsers.

Applications with very short IP sessions, such as DNS client and instant messengers, can utilize Nomadic IP Addresses. Even though they could very well use a Fixed or Sustained IP Addresses, the transmission latency would be minimized when a Nomadic IP Address is used.

### 3.2. Granularity of Selection

The IP address type selection is made on a per-socket granularity. Different parts of the same application may have different needs. For example, control-plane of an application may require a Fixed IP Address in order to stay reachable, whereas data-plane of the same application may be satisfied with a Sustained IP Address.

### 3.3. On Demand Nature

At any point in time, a mobile host may have a combination of IP addresses configured. Zero or more Nomadic, zero or more Sustained, and zero or more Fixed IP addresses may be configured on the IP stack of the host. The combination may be as a result of the host policy, application demand, or a mix of the two.

When the application requires a specific type of IP address and such an IP address is not already configured on the host, then the IP stack shall attempt to configure one. For example, a host may not always have a Fixed IP address available as such an address is rarely used. In case an application requests one, then the IP stack shall make an attempt to configure one using Mobile IP. If Mobile IP protocol is not available on the stack, or if its operation fails, then the IP stack shall fail the associated socket request. In case of successful Mobile IP operation, a Fixed IP Address gets configured on the mobile host. If another socket requests a Fixed IP address at a later time, then the same IP address may be served to that socket as well. When the last socket using the requested IP address is closed, the IP address may be released or kept for future applications that may be launched and require a Fixed IP address.

The following are matters of policy, which may be dictated by the host itself, the network operator, or the system architecture standard:

- The initial set of IP addresses configured on the host at the boot time.
- Permission to grant various types of IP addresses to a requesting application.
- Determination of a default address type when an application does not make any explicit indication, whether it already supports the required API or it is just a legacy application.

### 3.4. Conveying the Selection

The selection of the address type is conveyed from the applications to the IP stack in a way to influence the source address selection algorithm [RFC6724].

The current source address selection algorithm operates on the available set of IP addresses when selecting an address. According to the proposed solution, if the requested type IP address is not available at the time of the request, then the IP stack shall make an attempt to configure one such IP address. The selected IP address shall be compliant with the requested IP address type, whether it is selected among available addresses or dynamically configured. In the absence of a matching type (because it is not available and not configurable on demand), the source address selection algorithm shall return an empty set.

A Socket API-based interface for enabling applications to influence the source address selection algorithm is described in [RFC5014]. That specification defines IPV6\_ADDR\_PREFERENCES option at the IPPROTO\_IPV6 level. That option can be used with setsockopt() and getsockopt() calls to set and get address selection preferences.

Furthermore, that RFC also specifies two flags that relate to IP mobility management: IPV6\_PREFER\_SRC\_HOME and IPV6\_PREFER\_SRC\_COA. These flags are used for influencing the source address selection to prefer either a Home Address or a Care-of Address.

Unfortunately, these flags do not satisfy the aforementioned needs due to the following reasons, therefore new flags are proposed in this document:

- Current flags indicate a "preference" whereas there is a need for indicating "requirement". Source address selection algorithm does

not have to produce an IP address compliant with the "preference" , but it has to produce an IP address compliant with the "requirement".

- Current flags influence the selection made among available IP addresses. The new flags force the IP stack to configure a compliant IP address if none is available at the time of the request.

- The Home vs. Care-of Address distinction is not sufficient to capture the three different types of IP addresses described in Section 2.1.

The following new flags are defined in this document and they shall be used with Socket API in compliance with the [RFC5014]:

```
IPV6_REQUIRE_FIXED_IP /* Require a Fixed IP address as source */
```

```
IPV6_REQUIRE_SUSTAINED_IP /* Require a Sustained IP address as source */
```

```
IPV6_REQUIRE_NOMADIC_IP /* Require a Nomadic IP address as source */
```

Only one of these flags may be set on the same socket. If an application attempts to set more than one flag, the most recent setting will be the one in effect.

When any of these new flags is used, then the IPV6\_PREFER\_SRC\_HOME and IPV6\_PREFER\_SRC\_COA flags, if used, shall be ignored.

These new flags are used with `setsockopt()/getsockopt()`, `getaddrinfo()`, and `inet6_is_srcaddr()` functions [RFC5014]. Similar with the `setsockopt()/getsockopt()` calls, `getaddrinfo()` call shall also trigger configuration of the required type IP address, if one is not already available. When the new flags are used with `getaddrinfo()` and the triggered configuration fails, the `getaddrinfo()` call shall ignore that failure (i.e., not return an error code to indicate that failure). Only the `setsockopt()` shall return an error when configuration of the requested type IP address fails.

The following new error codes are also defined in the document and will be used in the Socket API in compliance with [RFC5014].

```
EAI_REQUIREDIPNOTSUPPORTED /* The network does not support the ability to request that specific IP address type */
```

```
EAI_REQUIREDIPFAILED /* The network could not assign that specific IP address type */
```

#### 4. Backwards Compatibility Considerations

Backwards compatibility support is required by the following 3 types of entities:

- The Applications on the mobile host
- The IP stack in the mobile host
- The network infrastructure

##### 4.1. Applications

Legacy applications that do not support the new flags will use the legacy API to the IP stack and will not enjoy On-Demand Mobility feature.

Applications using the new flags must be aware that they may be executed in environments that do not support On-Demand Mobility feature. Such environments may include legacy IP stack in the mobile host, legacy network infrastructure, or both. In either case, the API will return an error code and the invoking applications must respond with using legacy calls without On-Demand Mobility feature.

##### 4.2. IP Stack in the Mobile Host

New IP stacks must continue to support all legacy operations. If an application does not use On-Demand Mobility feature, the IP stack must respond in a legacy manner.

If the network infrastructure supports On-Demand Mobility feature, the IP stack may still request specific types of source IP address transparently to legacy applications. This may be useful for environments in which both legacy and new applications are executed.

The definition of what type of addresses to request and how they are assigned to legacy applications are outside of the scope of this specification.

##### 4.3. Network Infrastructure

The network infrastructure may or may not support the On-Demand Mobility feature. How the IP stack on the host and the network infrastructure behave in case of a compatibility issue is outside the scope of this API specification.

## 5. Security Considerations

The setting of certain IP address type on a given socket may be restricted to privileged applications. For example, a Fixed IP Address may be provided as a premium service and only certain applications may be allowed to use them. Setting and enforcement of such privileges are outside the scope of this document.

## 6. IANA Considerations

TBD

## 7. Acknowledgements

We would like to thank Alexandru Petrescu, John Kaippallimalil, Jouni Korhonen, Seil Jeon, and Sri Gundavelli for their valuable comments and suggestions on this work.

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <<http://www.rfc-editor.org/info/rfc5014>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<http://www.rfc-editor.org/info/rfc6724>>.

### 8.2. Informative References

- [I-D.ietf-dmm-requirements] Chan, A., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", draft-ietf-dmm-requirements-17 (work in progress), June 2014.



- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<http://www.rfc-editor.org/info/rfc5213>>.
- [RFC5563] Leung, K., Dommety, G., Yegani, P., and K. Chowdhury, "WiMAX Forum / 3GPP2 Proxy Mobile IPv4", RFC 5563, DOI 10.17487/RFC5563, February 2010, <<http://www.rfc-editor.org/info/rfc5563>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised", RFC 5944, DOI 10.17487/RFC5944, November 2010, <<http://www.rfc-editor.org/info/rfc5944>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<http://www.rfc-editor.org/info/rfc6275>>.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <<http://www.rfc-editor.org/info/rfc6824>>.

## Authors' Addresses

Alper Yegin  
Unaffiliated  
Istanbul  
Turkey

Email: [alper.yegin@yegin.org](mailto:alper.yegin@yegin.org)

Kisuk Kweon  
Samsung  
Suwon  
South Korea

Email: [kisuk.kweon@samsung.com](mailto:kisuk.kweon@samsung.com)

Jinsung Lee  
Samsung  
Suwon  
South Korea

Email: [js81.lee@samsung.com](mailto:js81.lee@samsung.com)

Jungshin Park  
Samsung  
Suwon  
South Korea

Email: [shin02.park@samsung.com](mailto:shin02.park@samsung.com)

Danny Moses  
Intel Corporation  
Petah Tikva  
Israel

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

DMM Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 10, 2016

D. Moses  
Intel  
A. Yegin  
December 8, 2015

DHCPv6 Extension for On Demand Mobility exposure  
draft-moses-dmm-dhcp-ondemand-mobility-02

Abstract

Applications differ with respect to whether or not they need IP session continuity and/or IP address reachability. Networks providing the same type of service to any mobile host and any application running on the host yields inefficiencies. This document describes extensions to the DHCPv6 protocol to enable mobile hosts to indicate the required mobility service type associated with a requested IP address, and networks to indicate the type of mobility service associated with the allocated IP address in return.

Status of This Memo

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

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

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

This Internet-Draft will expire on June 10, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Notational Conventions . . . . .	3
3. IPv6 Continuity Service Option . . . . .	3
3.1. Source IPv6 Address Type Specification . . . . .	4
3.2. IPv6 Prefix Type Specification . . . . .	5
4. Anchor Preference Option . . . . .	6
5. Security Considerations . . . . .	7
6. IANA Considerations . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

[I-D.ietf-dmm-ondemand-mobility] defines different types of mobility-associated services provided by access networks to mobile hosts with regards to maintaining IPv6 address continuity after an event of the host moving to different locations with different points of attachments within the IP network topology. It further specifies means for applications to convey to the IP stack in the mobile host, their requirements regarding these services.

This document defines extensions to the DHCPv6 protocol ([RFC3315]) in the form of a new DHCP option that specifies the type of mobility services associated with an IPv6 address. The IP stack in a mobile host uses the DHCP client to communicate the type of mobility service it wishes to receive from the network. The DHCP server in the network uses this option to convey the type of service that is guaranteed with the assigned IPv6 address in return.

This new option also extends the ability of mobile routers to specify desired mobility service in a request for IPv6 proxies (as specified in [RFC3633]), and delegating routers to convey the type of mobility service that is committed with the allocated IPv6 proxies in return.

In a distributed mobility management environment, there are multiple Mobility Anchors (as specified in [TBD reference to the Distributed Mobility Management architecture RFC]). In some use-cases, mobile hosts may wish to indicate to the network, preference of the serving Mobility Anchor. This document specifies a new DHCPv6 option that is used by DHCPv6 clients to convey this preference.

## 2. Notational Conventions

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

## 3. IPv6 Continuity Service Option

The IPv6 Continuity Service option is used to specify the type of continuity service associated with a source IPv6 address or IPv6 prefix. The IPv6 Continuity Service option must be encapsulated in the IAAddr-options field of the IA Address option when associated with an IPv6 address, and in the IAPrefix-options field of the IA\_PD prefix option when associated with an IPv6 prefix.

The format of the IPv6 Continuity Service option is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| OPTION_IPv6_CONTINUITY_SERVICE |          option-length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| service-type |
+-----+-----+-----+-----+

```

option-code      OPTION\_IPv6\_CONTINUITY\_SERVICE (TBD)

option-len       1

service-type    one of the following values:

Nomadic -	a nomadic address or prefix (1)
Sustained -	a sustained address or prefix (2)
Fixed -	a fixed address or prefix (3)
Anytype -	Anyone of the above (0)

This option can appear in one of two contexts: (1) As part of a request to assign a source IPv6 address of the specified mobility service type, and (2) As part of a request to assign an IPv6 prefix of the specified mobility service type.

### 3.1. Source IPv6 Address Type Specification

In this context, the IPv6 Continuity Service option is encapsulated in the IAaddr-options field of the IA Address option.

When in a message sent from a client to a server, the value of the IPv6 Continuity Service option indicates the type of continuity service required for the IPv6 address requested by the client.

When in a message sent from a server to a client, the value of the IPv6 Continuity Service option indicates the type of IP continuity service committed by the network for the associated IPv6 address. The value 'AnyType' cannot appear in a message sent from the server.

Once an IPv6 address type was requested and provided, any subsequent messages involving this address (lease renewal - for example) must include the IPv6 Continuity Service option with the same service type that was assigned by the server during the initial allocation.

If a server received a request to assign an IPv6 address with a specified IPv6 Continuity service, but cannot fulfill the request, it must reply with the [TBD] status.

A server that does not support this option will discard it as well as the IA Address option that had this option encapsulated in one of its IAaddr-options field.

If a client does not receive the requested address, it must resent the request without the desired IPv6 Continuity Service option since it is not supported by the server. In that case, the host of the client cannot assume any IP continuity service behaviour for that address.

A server must not include the IPv6 Continuity Service option in the IAaddr-options field of an IA Address option, if not specifically requested previously by the client to which it is sending a message.

If a client receives an IA Address option from a server with the IPv6 Continuity Service option in the IAaddr-options field, without initially requesting a specific service using this option, it must discard the received IPv6 address.

If the mobile host has no preference regarding the type of continuity service it uses the 'AnyType' value as the specified type of continuity service. The Server will allocate an IPv6 address with some continuity service and must specify the type in IPv6 Continuity Service option encapsulated in the IAaddr-options field of the IA

Address option. The method for selecting the type of continuity service is outside the scope of this specification.

### 3.2. IPv6 Prefix Type Specification

In this context, the IPv6 Continuity Service option is encapsulated in the IAprefix-options field of the IA\_PD prefix option.

When in a message sent from a client to a server, the value of the IPv6 Continuity Service option indicates the type of continuity service required for the IPv6 prefix requested by the client.

When in a message sent from a server to a client, the value of the IPv6 Continuity Service option indicates the type of continuity service committed by the network for the associated IPv6 prefix. The value 'AnyType' cannot appear in a message sent from the server.

Once an IPv6 prefix type was requested and provided, any subsequent messages involving this prefix (lease renewal - for example) must include the IPv6 Continuity Service option with the same service type that was assigned by the server during the initial allocation.

If a server received a request to assign an IPv6 prefix with a specified IPv6 Continuity service, but cannot fulfill the request, it must reply with the [TBD] status.

A server that does not support this option will discard it as well as the IA\_PD Prefix option that had this option encapsulated in one of its IAprefix-options field.

If a client does not receive the requested prefix, it must resent the request without the desired IPv6 Continuity Service option since it is not supported by the server. In that case, the mobile router of the client cannot assume any IP continuity service behaviour for that prefix.

A server must not include the IPv6 Continuity Service option in the IAprefix-options field of an IA\_PD Prefix option, if not specifically requested previously by the client to which it is sending a message.

If a client receives an IA\_PD Prefix option from a server with the IPv6 Continuity Service option in the IAprefix-options field, without initially requesting a specific service using this option, it must discard the received IPv6 prefix.

If the mobile router has no preference regarding the type of continuity service it uses the 'AnyType' value as the specified type of continuity service. The Server will allocate an IPv6 prefix with

some continuity service and must specify the type in IPv6 Continuity Service option encapsulated in the IAprefix-options field of the IA\_PD Prefix option. The method for selecting the type of continuity service is outside the scope of this specification.

#### 4. Anchor Preference Option

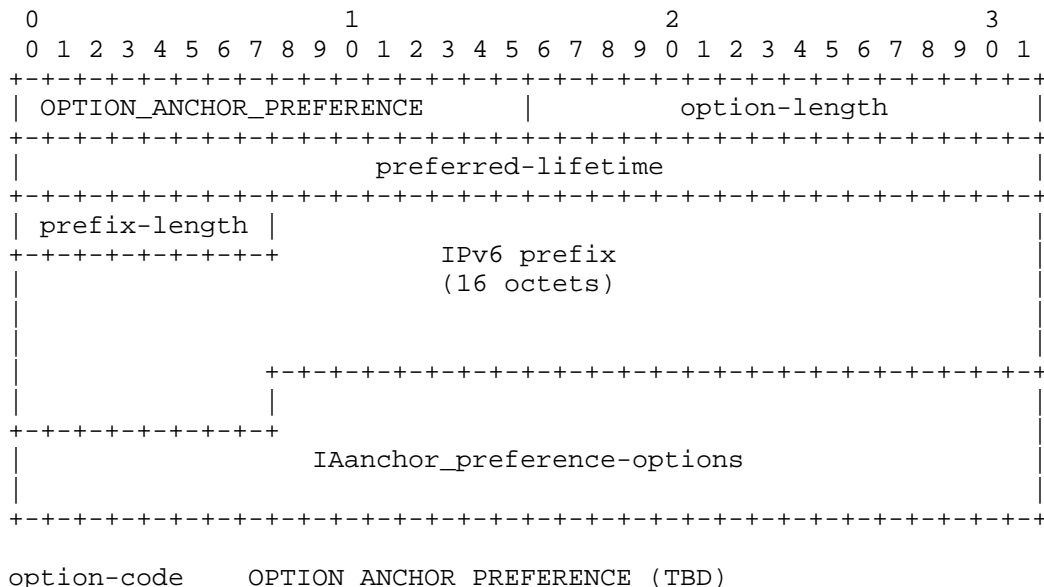
In a distributed mobility management environment that deploys multiple Mobility Anchors, each Mobility Anchor may have a set of IPv6 prefixes that is being used when assigning Sustained or Fixed source IPv6 addresses to hosts.

The selection of the Mobility Anchor that will serve a mobile host is performed by the network at various events like, the event of initial attachment of a mobile host to a network.

The Anchor Preference option enables a host to express its desire to receive a source IPv6 address with a specific IPv6 prefix. This is useful when the mobile host wishes to indicate to the network which Mobility Anchor should be used for anchoring its traffic and ensuring service continuity in the event of handoff between LANs with different IPv6 prefixes.

The network MAY respect this request but is not required to do so.

The format of the Anchor Preference option is:





`option-len`        25 + length of `anchor_preference-options` field

`preferred-lifetime` The preferred lifetime of the IPv6 address whose prefix is requested, expressed in units of seconds

`prefix-length`    The length of this prefix in bits

`IPv6 prefix`       The requested prefix

`IAanchor_preference-option` Options associated with this request

This option is used by the client in a request for a new IPv6 source address. The server replies with an IPv6 address that may or may not have the desired prefix. Subsequent interactions between the client and server regarding this address, must use the the IA address option.

An IPv6 prefix is requested only when the mobile host wishes to be anchored by a specific mobility anchor. The client must also indicate the type of mobility service it requires using the IPv6 Continuity Service option encapsulated in the `IAanchor_preference-options` field of the IA Address option.

When requesting an IPv6 prefix, only the 'Sustained' and 'fixed' types are legal.

The server must assign the IPv6 address of the requested type to the client, even if it does not fulfill the request for the specified prefix.

If a server received a request to use a specific IPv6 prefix and an IPv6 address type, but cannot assign an IPv6 address with that specified IPv6 Continuity it must reply with the [TBD] status.

A server that does not support this option will discard it.

If a client does not receive any address, it must assume that the the option is not supported by the server and use the IA Address option in subsequent requests.

## 5. Security Considerations

There are no specific security considerations for this option.

## 6. IANA Considerations

TBD

## 7. References

### 7.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.

### 7.2. Informative References

- [I-D.ietf-dmm-ondemand-mobility] Yegin, A., Kweon, K., Lee, J., Park, J., and D. Moses, "On Demand Mobility Management", draft-ietf-dmm-ondemand-mobility-01 (work in progress), November 2015.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.

## Authors' Addresses

Danny Moses  
Intel  
Petah Tikva  
Israel

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

Alper Yegin  
Istanbul  
Turkey

Email: [alper.yegin@yegin.org](mailto:alper.yegin@yegin.org)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 15, 2016

B. Pularikkal  
Cisco Systems  
Q. Fu  
H. Deng  
China Mobile  
G. Sundaram  
S. Gundavelli  
Cisco Systems  
March 14, 2016

Virtual CPE Deployment Considerations  
draft-pularikkal-virtual-cpe-00

Abstract

Broadband Service Provider Industry has been gearing towards the adoption of Virtual CPE (vCPE) solutions. The concept of vCPE is build around the idea that the physical CPE device at the customer premises can be simplified by moving some of the key feature functionalities from the physical CPE device to the Service Provider Network. This document starts discussing the drivers behind vCPE adoption followed by Solution level requirements. Two key Architecture models for vCPE, which can address the service provider and subscriber requirements, are covered in this reference document. Document also touches up on some of the key deployment considerations, which can influence the adoption of the vCPE architecture models.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 15, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Solution Requirements for vCPE . . . . .	3
4. Architecture Models for vCPE . . . . .	4
4.1. Virtual CPE Definition . . . . .	4
4.2. Virtual CPE Architecture Model-01 . . . . .	5
4.3. Virtual CPE Architecture Model-02 . . . . .	7
4.4. Virtual CPE Architecture Model-03 . . . . .	9
4.4.1. Forwarding Policy Configuration (FPC) Interface . .	11
5. Deployment Considerations for vCPE . . . . .	11
5.1. Multi-tenancy . . . . .	11
5.2. Tunneling . . . . .	12
5.3. Security . . . . .	12
5.4. Dynamic Service Chaining . . . . .	13
5.5. NAT Traversal . . . . .	13
6. Conclusion . . . . .	14
7. Informative References . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

Broadband Service Providers are constantly looking for opportunities to generate additional revenue streams from their existing broadband infrastructure. In order to achieve this, new value added services need to be created for the end customers. Customer retention is another key focus area for broadband subscribers, where they have been facing competition from Internet content providers on home multi-media services such as broadcast video, video on demand and voice. There is a need to improve the overall end user experience on an ongoing basis to reduce the subscriber churn. In order to achieve these business goals, Broadband Service Providers are starting to

consider the deployment of Virtual CPE based Architectures. There are several factors, which are driving the adoption of vCPE-based solutions. Also the recent technological advancements in cloud computing and software defined networking are expected to further accelerate the adoption vCPE based architectures.

The key aspect of the vCPE solutions is the simplification of the physical CPE device. Such a simplification allows minimizing the feature dependency on CPE vendors for the roll out of new service offerings. Also it reduces the complexities around service provisioning, Service Upgrade Troubleshooting etc. There are multiple architecture options being considered by the industry for vCPE solutions.

Objective of this draft is to serve as a reference material for Broadband Service Operators who are interested in migrating to vCPE based architectures. The document starts with going over some of the key drivers for vCPE solution adoption. Also it covers typical solution level requirements, which needs to be considered while selecting the right architecture models. Document also touches up on some of the key deployment considerations, which can influence the adoption of the vCPE architecture models.

## 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].

## 3. Solution Requirements for vCPE

This section provides a high level summary of solution requirements, which needs to be addressed by Virtual Connected Home Architecture Models. The solution requirements can be broadly classified under the following categories:

(1) Subscriber side requirements: Subscriber in the context of this documentation refers to a homeowner with Broadband connection. These requirements primarily map to the end user experience for a home subscriber in terms of connectivity, quality of experience and value added services.

(2) Broadband Operator side requirements: Operator is the broadband service provider such as Cable MSOs, DSL providers etc. These requirements primarily maps to the business aspects which needs to be covered in the solution in terms of CAPEX, OPEX reduction, service velocity, new revenue generation opportunities etc.

High level requirements under the above two categories are summarised in the table below:

Subscriber Side Requirements	Operator Side requirements
1) Private Home Network	1) Service Velocity
2) Zero Touch Provisioning	2) Simplified CPE
3) Local Bridging	3) Per UE Visibility
4) Local Routing	4) Community Wi-Fi
5) NAT, FW, IDS, Parental Control	5) IP Address Persistence
6) Home Network Analytics	6) UE Attachment/ Detachment detection
7) Self Service Portal	7) Usage based billing
8) Dynamic IP address Assignment	8) Quality of Service
9) Home Network Remote Access	9) NAT Traversal

Figure 1: VCPE Requirements

#### 4. Architecture Models for vCPE

In this document three different architecture models are covered for the Virtual CPE based solutions. This section starts with a definition of what represents a virtual CPE and then gets into the details of the Architecture options, which are available for the implementation of the same.

##### 4.1. Virtual CPE Definition

A virtual CPE (vCPE) is a logical representation of classical CPE functions performed by a physical CPE device. In other words, business logic and feature functionalities which are traditionally embedded in a CPE device is separated from the hardware device and runs in the Service Providers cloud. Concepts of vCPE has basis on the Network Function Virtualization. The business logic and feature functionalities of a CPE device are virtualized and runs as NFV in the cloud. Each simplified physical CPE would have a corresponding virtual CPE function running in the cloud. There are several ways to realize this vCPE instance in the cloud. One approach is to have separate vCPE instance running as a Linux container or micro-VM corresponding to each physical CPE instance. The vCPE may also be implemented as a representational state on aggregation platforms such as broadband network gateways (BNGs). A third approach may rely on a

combination of the BNG representational state and Service function chaining to represent the vCPE instance in the cloud. These Architecture models are covered in the subsequent sub-sections.

#### 4.2. Virtual CPE Architecture Model-01

This model is build around the concept of separate virtualized instance per physical CPE device. In this model Virtual CPE instance handles the control plane as well as the data plane. Each micro VM represents an NFV element of CPE with integrated control and data plane. All feature functionalities get implemented on the NFV element itself. This model does not leverage the dynamic service function chaining capabilities.

A high level Architecture view of this model is provided in figure-below:

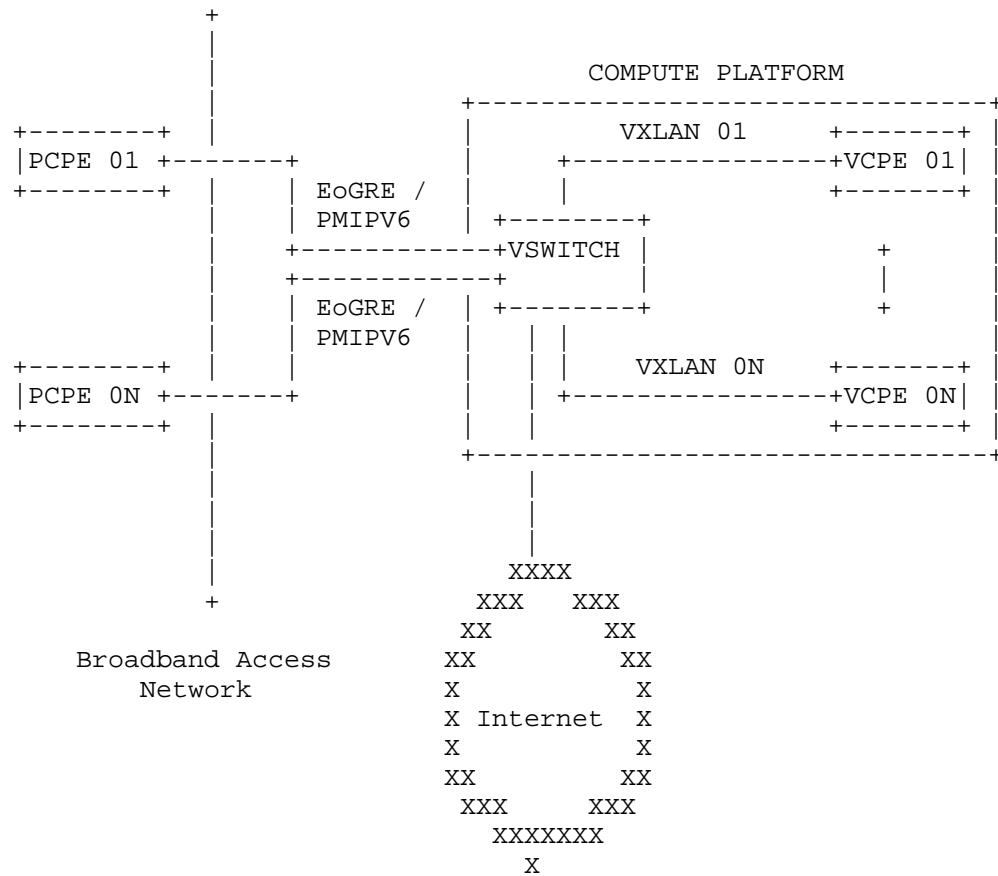


Figure 2: VCPE deployment model-01

P-CPE device performs just the bridging function where the layer-2 traffic between directly connected devices will be simply bridged by the P-CPE. Any layer-3 traffic will be transparently forwarded to the cloud over the tunnel.

In this model all the key cloud based components run on virtualized platforms. These virtual components are deployed on standalone virtual platforms rather than on large scale virtual DC of Service providers. Therefore, the tunnel will be terminated on the vSwitch rather than a tunnel termination GW located at the boarder of the DC.



An implementation could leverage either a vendor specific vSwitch or an Open vSwitch.

Tunnel end points are uniquely identified with the IP address of the P-CPE. The vSwitch maps the de-encapsulated traffic from the tunnels to unique VXLANs and will forward to the corresponding Micro VM instances. Micro VM instances will be responsible for supporting the key functions traditionally performed on physical CPE devices. After the feature processing, V-CPE instance will send the traffic back to the v-Switch over VXLAN tunnels and vSwitch will forward it to external network.

#### 4.3. Virtual CPE Architecture Model-02

In this model vCPE in the cloud is corresponding to each physical CPE is realized by a representational state on a tunnel aggregation platform such as BNG. A provisioned physical CPE in run state is expected to have at least one tunnel established between the physical CPE and the BNG. As long as the PCPE is in run state there will be a CPE session on the BNG which represents the CPE itself. Some of the key CPE features will be running on the BNG while supplementary features and services can be deployed using dynamic service function chaining functions.

A high level Architecture view of this mode2 is provided in figure-below:

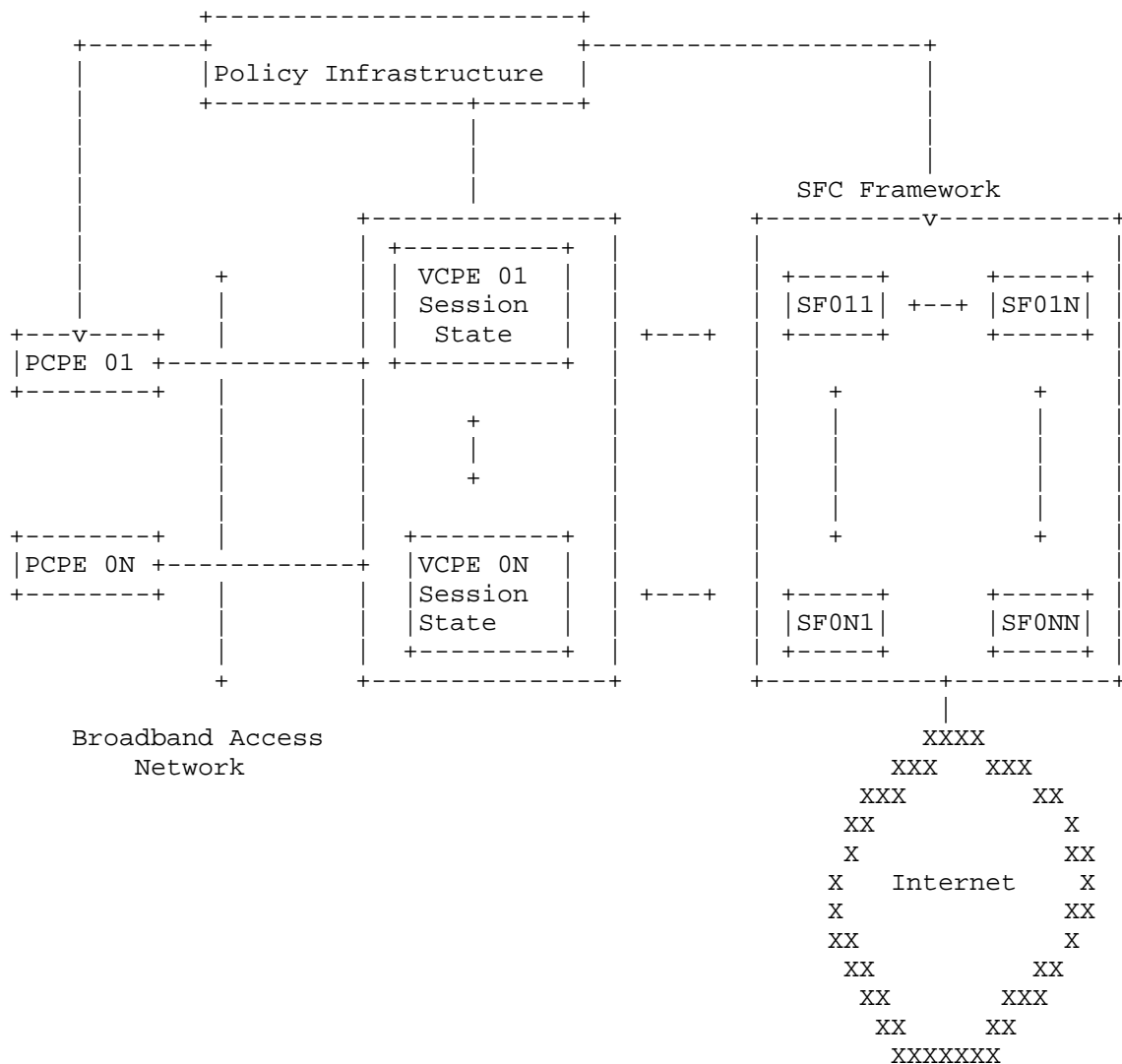


Figure 3: VCPE deployment model-02

In this model as well, P-CPE device performs just the bridging function where the layer-2 traffic between directly connected devices will be simply bridged by the P-CPE. Any layer-3 traffic will be transparently forwarded to the BNG over the tunnel.

There is no need for pre-configuration of the tunnels on BNG. When a P-CPE device become active and gets provisioned it will try to

establish an EoGRE tunnel session with V-CPE. Up on detecting a new P-CPE end point, the BNG would invoke an authorization process for the tunnel end point. It is up to the implementation to decide whether an out of band authentication mechanism is required before establishing v-CPE state on the BNG. If the access network is untrusted, the service provider may decide to overlay the EoGRE tunnel with IPSec encapsulation.

BNG will need to uniquely tag the subscriber flows before forwarding to the SFC framework. This can be accomplished by using some scalable tagging mechanisms such as VXLAN.

#### 4.4. Virtual CPE Architecture Model-03

This is similar to model-02 but leverages split architecture for control plane and data plane for the BNG. This model introduces the concept of a BNG controller, which essentially carries out the control plane functions. Data plane component of the BNG can be a purpose built hardware optimized for scaleable tunnel termination, data encryption and data forwarding. Control plane intelligence of each vCPE resides as a session state on the BNG controller and the data plane intelligence including tunnel termination of each vCPE resides on the BNG-DP system. BNG-CP will leverage the FPC (Forwarding Policy Configuration) interface which is being defined in the DMM working group to instruct the BNG-DP system to establish V-CPE DP states with relevant configuration. Role of FPC interface in this solution is described in the sub-section below. In this model, all basic and supplementary subscriber features will be implemented using a dynamic service function-chaining framework.

A high level Architecture view of this mode3 is provided in figure-below:

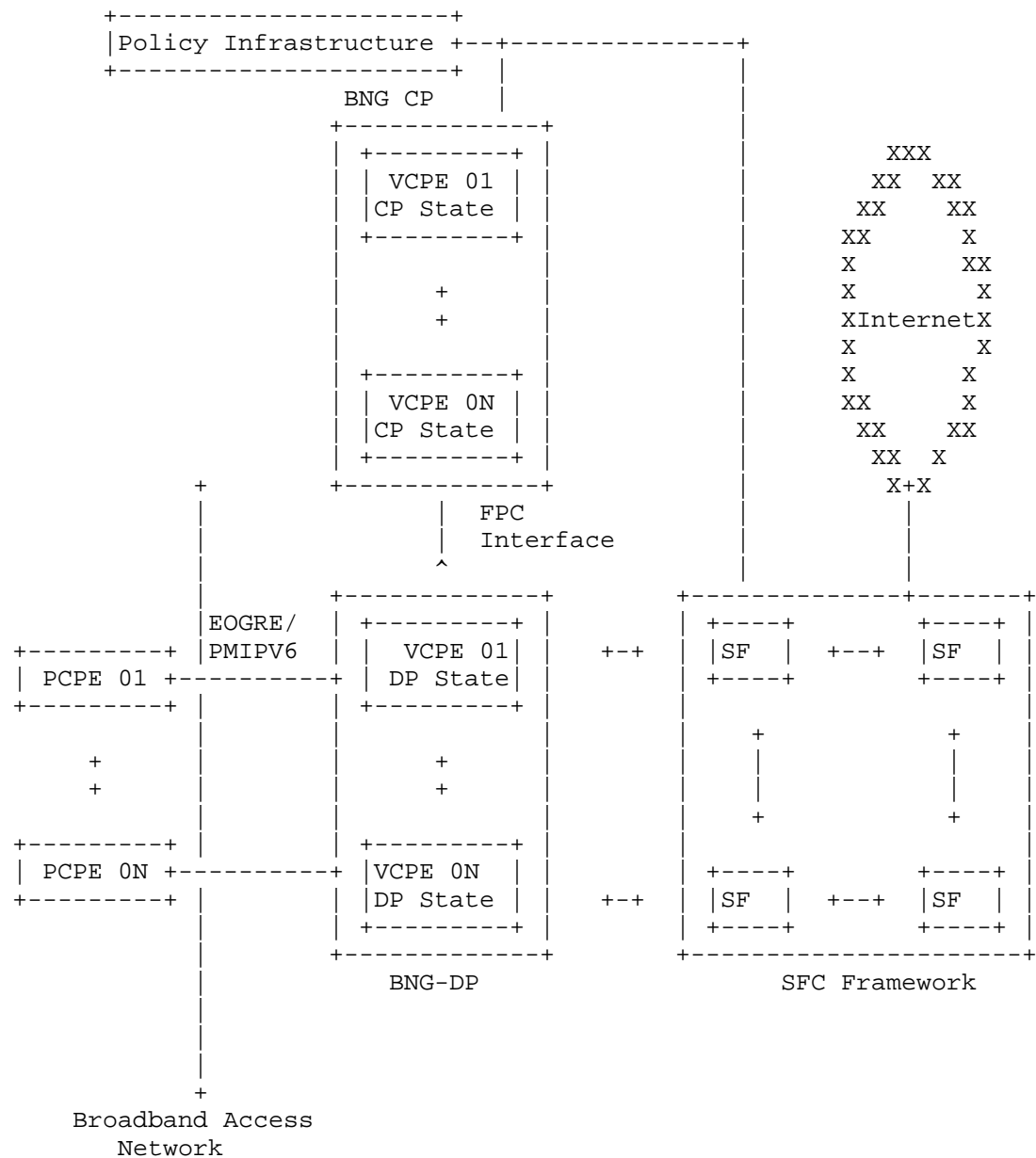


Figure 4: VCPE deployment model-03

#### 4.4.1. Forwarding Policy Configuration (FPC) Interface

FPC Protocol interface defined in the DMM working group enables DMM use cases with Control Plane and data plane separation. In vCPE solution model-03, FPC protocol is applied to the interface between BNG-CP and BNG-DP. FPC interface consists of a client function which resides on the Control Plane System (BNG-CP in this case) and an agent function which resides on the Data Plane System (BNG-DP in this case). FPC defines a standard set of protocol semantics to exchange configuration information from the client to the agent. Agent processes the protocol semantics and translates them into configuration commands as per BNG-DP system technology. FPC client function residing on BNG-CP device will leverage FPC Protocol semantics to provision activate or deactivate the V-CPE DP states on BNG-DP with desired features.

### 5. Deployment Considerations for vCPE

This section at a high level touches up on some of the key deployment characteristics which needs to be considered while selecting the right vCPE architecture

#### 5.1. Multi-tenancy

vCPE represents the abstraction of key functions and features typically performed by classical device into service provider cloud. In order for such a solution to be operationally feasible and profitable, it is important for vCPE architecture to support multi-tenancy. This multi tenancy support needs to scale of the order of hundreds of thousands. From the context of vCPE deployments, the multi-tenancy refers to the logical separation of vCPE instances, which are housed in a common backend infrastructure. This backend infrastructure could consist of virtual elements on a compute platform or physical networking components. It could very well be a combination of virtual and physical components in the service provider cloud. Few of the key areas where multi-tenancy model will have an implication on the operational efficiency of the solution are listed below:

Overlapping IP addressing: Typically home networks are configured with RFC 1918 private address space 192.168.0.0/24. A vCPE solution, which deals with IP address management of the private home network, must support address overlap for these private home subnets.

Tunnel scale: Tunnel termination points in the service provider must support tunnel scale of the order of hundreds of thousands. A vCPE implementation must implement some form of unique tunnel id per

physical CPE to support saleable multi-tenancy for tunnel termination.

Overlapping SSID naming: vCPE framework must be flexible enough to allow home subscribers to configure private SSID names of their choice. Possibility of overlapping SSID names cannot be ruled out as subscribers randomly decide up on their private SSID names. Multi-tenancy solution for a vCPE framework must take into consideration this.

## 5.2. Tunneling

In a vCPE solution, the end subscriber data must be tunneled from the physical CPE towards the vCPE instance in the cloud. Typical home broadband deployments may have community Wi-Fi SSID enabled in addition to subscribers private home SSID. For such cases, the tunnel must be capable of carrying both private and community Wi-Fi SSID traffic in a secured manner. Today there are various tunneling methods being used for community Wi-Fi deployments. Two of the most common tunneling methods in use are EoGRE and PMIPv6. EoGRE is a layer-2 tunneling technology and it does not have a control plane of its own. PMIPv6 is a layer-3 tunneling technology with a well-defined control plane for tunnel management and session management. Either of these tunneling options can be leveraged to carry the private SSID traffic from the home towards the cloud-based vCPE. And both are capable of carrying community Wi-Fi and private home SSID traffic. The choice of the tunneling technology may be influenced by various factors such as simplicity, need for IP address persistence with client roaming, layer-2 forwarding in the data plane to the cloud as opposed to layer-3 forwarding etc.

## 5.3. Security

The classical home broadband deployments based up on intelligent physical CPE devices typically provide data privacy and security for the end subscriber content as it gets carried over the access network. A security framework for a vCPE network has to account for the following key aspects:

Subscriber Authentication

Protection against spoofing attacks

Data privacy

Prevention of eaves dropping between subscribers

Security considerations go hand in hand with multi-tenancy requirements as data and meta data from multiple subscribers will be handled by the backend systems.

#### 5.4. Dynamic Service Chaining

One of the key motivation behind a cloud based connected home solution is to find additional revenue generation opportunities through rapid deployment of new services. The implementation of these new services requires a combination of system and network level functions to be applied to the end user traffic flows. Some of these functions may be enabled, by leveraging system level features on the CPE Device Anchor. But in many cases, it makes more sense to offload the feature processing to network function elements, which are external to the CPE Device Anchor (CDA).

Service function chaining (SFC) refers to a collection of network elements connected in a serialized fashion through which a traffic flow will be diverted prior to forwarding to the intended destination. Traditionally these service chains are hard connected there by causing challenges around flexibility and scale.

With dynamic service function chaining approach, the network elements, which perform various service functions, are arranged in grid model. Logical connectivity is established on a per traffic flow basis between the network elements to establish SFC pipeline for a qualified traffic flow. Dynamic SFC addresses the scale and flexibility limitations of the traditional chaining model. A vCPE solution must support the deployment of dynamic service function chaining.

#### 5.5. NAT Traversal

Some vCPE deployments may leverage third party access networks and offer the solution as an overlay. In such cases, there may be requirement to bring up P-CPE behind a NAT router. The vCPE service provider may not have direct control over the NAT router, which is managed by the access network provider. In such cases, a tunnel transport mode, which can traverse NAT, needs to be selected.

EoGRE tunnels do not support NAT traversal, since there is no UDP layer in the encapsulation header. PMIPv6 can support NAT traversal if the right data encapsulation option is selected. If a layer tunneling technology is desired for the implementation where NAT traversal is a requirement, then tunnel transport mechanisms such as L2TPV3 may be explored.

## 6. Conclusion

In this document, the concept of VCPE is illustrated in detail. The basic concept of VCPE is to shift the complicated functions from the pCPE at the customer side to the VCPE at the service provider side. The motivation of such shifting can be concluded as providing quick launched customer defined services, reducing the Capex and Opex of the pCPE, and simplify the maintainance of both pCPE and VCPE. A use cases of community Wi-Fi is proposed for VCPE, which is a typical scenario for DMM. Three models are then discussed for the field deployment of VCPE. And CP/DP interface is suggested to be utilized in the deployment models.

## 7. Informative References

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

### Authors' Addresses

Byju Pularikkal  
Cisco Systems  
170 West Tasman Drive  
San Jose  
United States

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

Qiao Fu  
China Mobile  
Xuanwumenxi Ave. No.32  
Beijing  
China

Email: [fuqiao1@outlook.com](mailto:fuqiao1@outlook.com)

Hui Deng  
China Mobile  
Xuanwumenxi Ave. No.32  
Beijing  
China

Email: [denghui@chinamobile.com](mailto:denghui@chinamobile.com)



Ganesh Sundaram  
Cisco Systems  
170 West Tasman Drive  
San Jose  
United States

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

Sri Gundavelli  
Cisco Systems  
170 West Tasman Drive  
San Jose  
United States

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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 16, 2016

B. Sarikaya  
Huawei  
L. Xue  
Unaffiliated  
March 15, 2016

Distributed Mobility Management Protocol for WiFi Users in Fixed Network  
draft-sarikaya-dmm-for-wifi-04.txt

## Abstract

As networks are moving towards flat architectures, a distributed approach is needed to mobility management. This document defines a distributed mobility management protocol called Distributed Mobility Management for Wi-Fi protocol. The protocol is based on mobility aware virtualized routing system with software-defined network support. Routing is in Layer 2 in the access network and in Layer 3 in the core network. Smart phones access the network over IEEE 802.11 (Wi-Fi) interface and can move in home, hotspot and enterprise buildings.

## Status of This Memo

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

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

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

This Internet-Draft will expire on September 16, 2016.

## Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Overview . . . . .	3
4. Detailed Protocol Operation . . . . .	5
4.1. Layer 2 Mobility in Access Network . . . . .	5
4.2. Layer 3 Mobility and Routing in Core Network . . . . .	6
4.3. Route Establishment . . . . .	8
4.4. Authentication and Charging . . . . .	10
4.4.1. Authentication . . . . .	10
4.4.2. Charging . . . . .	13
5. Multicast Support . . . . .	15
5.1. IPv4 Support for Multicast . . . . .	15
6. IPv4 Support . . . . .	16
7. Security Considerations . . . . .	16
8. IANA Considerations . . . . .	16
9. Acknowledgements . . . . .	16
10. References . . . . .	16
10.1. Normative References . . . . .	16
10.2. Informative references . . . . .	19
Appendix A. YANG and RPC Programs . . . . .	20
A.1. Host Routing Module . . . . .	20
A.2. Route Establishment RPCs . . . . .	20
A.3. get-config RPC procedure for host routes . . . . .	21
A.4. edit-config RPC procedure to create a host route . . . . .	22
Authors' Addresses . . . . .	23

## 1. Introduction

Centralized mobility anchoring has several drawbacks such as single point of failure, routing in a non optimal route, overloading of the centralized data anchor point due to the data traffic increase, low scalability of the centralized route and context management [I-D.ietf-dmm-requirements].

In this document, we define a routing based distributed mobility management protocol. The protocol assumes a flat network architecture as shown in Figure 1. No client software is assumed at the mobile node.

IP level mobility signaling needs to be used even when MN is connected to a home network or a hotspot. Distributed anchors in the protocol are called Unified Gateways and they represent an evolution from the Broadband Network Gateway (BNG) currently in use.

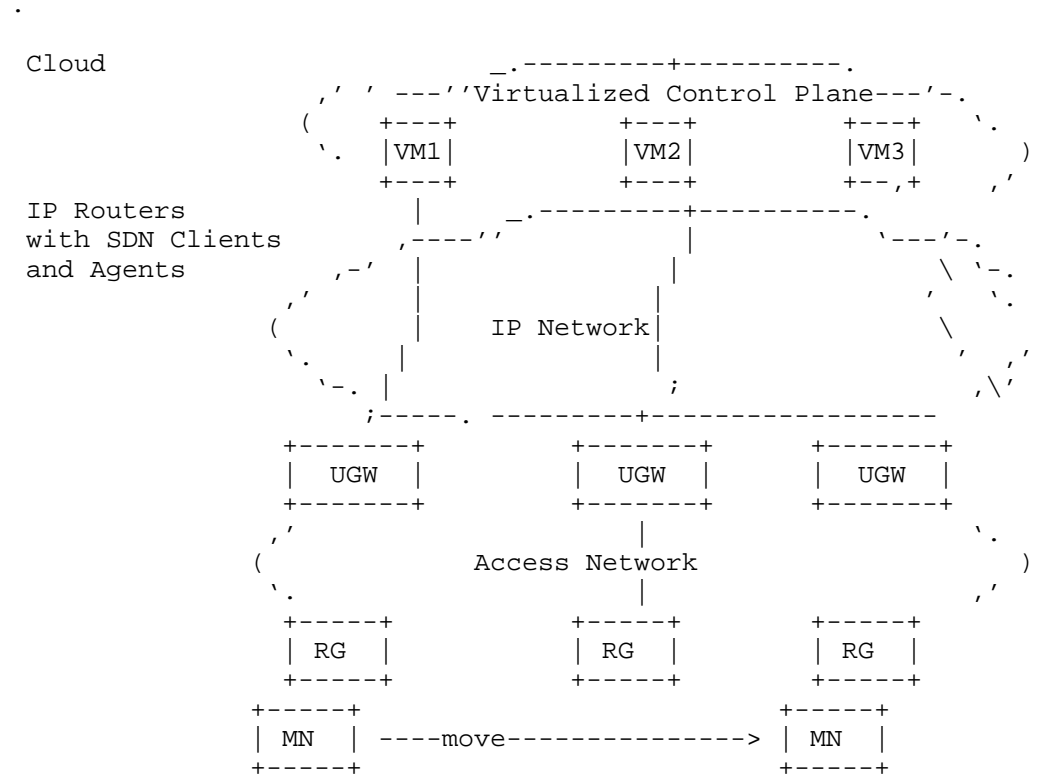


Figure 1: Architecture of DMM for Wi-Fi Protocol

## 2. Terminology

This document uses the terminology defined in [I-D.matsushima-stateless-uplane-vepc].

## 3. Overview

This section presents an overview of the protocol, Distributed Mobility Management for Wi-Fi protocol (DMM4WiFi). See also Figure 1.

Access routers (AR) are Unified Gateways (UGW) that are the access network gateways that behave similarly as Evolved Packet Core (EPC)

Edge Router (EPC-E) in [I-D.matsushima-stateless-uplane-vepc]. UGW is configured an anycast address on the interface facing the Residential Gateway (RG). RGs use this address to forward packets from the users. The fixed access network delivers the packets to geographically closest UGW.

Wi-Fi smart phone, the mobile node (MN) is assigned a unique prefix using either Stateless Address Auto Configuration (SLAAC) or by a DHCP server which could be placed in the cloud. In case of SLAAC, RG is delegated the prefixes by DHCP server using [RFC3633].

Prefix assignments to MNs are consistent with the prefixes assigned to UGWs that are shorter than /64. These prefixes are part of the operator's prefix(es) which could be /32, /24, etc.

The mobile node can move at home or in a hot spot from one Access Point (AP) to another AP and MN mobility will be handled in Layer 2 using IEEE 802.11k and 802.11r. Authentication is handled in Layer 2 using [IEEE-802.11i] and [IEEE-802.11-2007] (as described in Section 4.4).

When MN moves from one UGW into another UGW, IP mobility signaling needs to be introduced. In this document we use Handover Initiate/Handover Acknowledge (HI/HACK) messages defined in [RFC5949]. Handover Initiate message can be initiated by either previous UGW (predictive handover) or the next UGW (reactive UGW). In reactive handover, RG establishes a new connection with the next UGW when MN moves to this RG and provides previous UGW address. This will trigger the next UGW to send HI message to the previous UGW. Previous UGW sends HACK messages which establishes a tunnel between previous and next UGWs. Previous UGW sends packets destined to MN to the new UGW which in turn sends them to MN.

Note that the mobility signaling just described is control plane functionality. Control plane in our document is moved to the cloud, thus mobility signaling happens at the cloud, possibly between two virtual machines (VM).

Upstream packets from MN at the new UGW establish the initial routing path when MN first enters the system. This path needs to be updated as MN moves from one UGW to another, i.e. MN handover. Since MN keeps the prefix initially assigned, after handover, the new upstream path establishment may establish host routes in the upstream routers. This route is refreshed as long as MN stays under the same UGW. Handover signaling and subsequent upstream path establishment is very critical because the downstream packets may need to follow the path that is established for MN.

Software-Defined Networking (SDN) is used in DMM4WiFi in both Layer 2 and Layer 3 routing management. In case of Layer 2 routing, the Open Flow Switch Protocol is used as the south bound interface between the SDN Controller and Layer 2 access network switches. Extensible Messaging and Presence Protocol (XMPP) is used as the north bound interface between the SDN controller and DMM4WiFi application. DMM4WiFi Layer 3 routing is based on SDN controllers manipulating Routing Information Bases (RIB) in a subset of the upstream routers. In this case south bound interface is the NETCONF protocol which is based on the Remote Procedure Call (RPC) protocol and YANG. I2RS architecture is used in this context.

Mobile node generates interface identifier using [RFC7217] in SLAAC. With this method, MN interface identifiers will be different when MN moves from one UGW to another UGW. MN MAY have different IPv6 addresses due to this method of interface identifier generation.

#### 4. Detailed Protocol Operation

In this section, Layer 2 and Layer 3 mobility procedures are explained.

##### 4.1. Layer 2 Mobility in Access Network

In the access network, RG MAC address acts as an identifier for the MN. Access network switches are controlled by SDN. Controller to Switch interface uses a protocol such as Extensible Messaging and Presence Protocol (XMPP)[RFC6121]. XMPP is based on a general subscribe-publish message bus. SDN controller publishes forwarding instructions to the subscribing switch. Forwarding instructions could be Open Flow like match-forward instructions. Open Flow protocol can also be used [ONFv1.5].

Access network is organized as interconnected switches. The switch connected to the RG is called egress switch. The switch connected to the UGW is called ingress switch. IEEE 802.1ad standard for VLAN (Q-in-Q) is used in the access network, where S-VLAN denotes RG groups, and C-VLAN determines traffic classes. One S-VLAN tag is assigned to create one or more VLAN paths between egress and ingress switches.

MN mobility in the access network can be tracked by keeping a table consisting of MN IP address and RG MAC address pairs. In this document SDN controllers keep the mobility table. This table is used to select proper S-VLAN downstream path from ingress switch to egress switch and upstream path from egress switch to ingress switch.

After a new MN with WiFi associates with RG, RG sends an Unsolicited Neighbor Advertisement (NA) message upstream. This NA message is

constructed as per [RFC4861] but the Source Address field is set to a unicast address of MN. NA message is received by SDN controller and it enables SDN controller to update the mobility table. SDN controller selects proper path including S-VLAN and ingress switch to forward the traffic from this MN. The controller establishes the forwarding needed on these switches [UTD-Paper], i.e. Layer 2 route.

The packet eventually reaches the closest UGW due to the anycast addressing used at the access network interfaces. UGW forwards this packet to the upstream router and so on. The upstream router establishes a route for MN in its routing table with MN's prefix and with the UGW as the next hop. Prefixes in those routes get smaller and smaller as the packet moves upstream in the routing hierarchy. The routing protocol used could be BGP or other protocols like IS-IS.

#### 4.2. Layer 3 Mobility and Routing in Core Network

MN moving from one RG to another may eventually require MN moving from one UGW to another. This is Layer 3 mobility.

Predictive handover happens when MN just before leaving the previous RG (pRG) for the next RG (nRG) MN is able to send an 802.11 message containing MN MAC address and nRG MAC address, e.g. learned from beacons to the pRG (called Leave Report in Figure 2. pRG then sends a handover indication message to pUGW providing MN and nRG addresses (called Leave Indication) and this could happen between two respective virtual machines in the cloud. This message results in pUGW getting nUGW information and then sending Handover Initiate message to nUGW, which also could happen in the cloud. nUGW replies with Handover Acknowledge message. pUGW sends any packets destined to MN to nUGW after being alerted by the control plane. MN moves to nRG and nUGW is informed about this from Layer 2 mobility Section 4.1. uGW delivers MN's outstanding packets to MN.

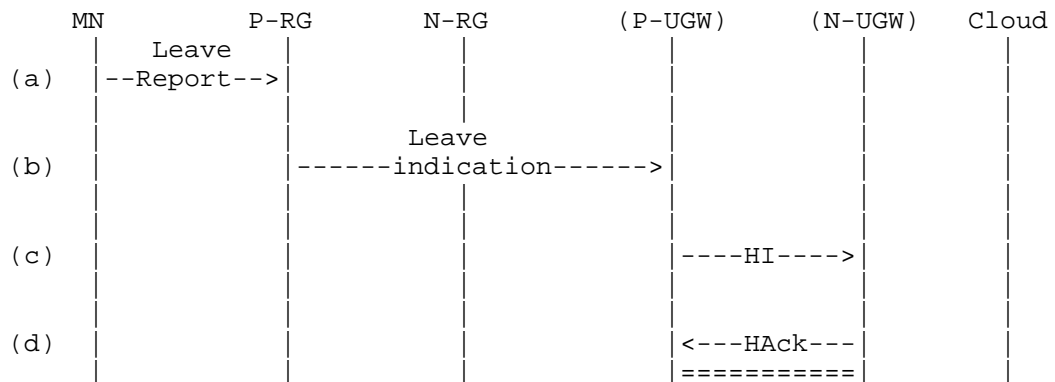


Figure 2: Predictive Handover

Reactive handover happens when MN attaches the new RG from the previous RG (called Join Report in Figure 3). MN is able to signal in 802.11 association messages previous RG MAC address. nUGW receives new association information together with pRG information, possibly in the cloud (called Handover Indication). nUGW finds pUGW address and sends HI message to pUGW, again happening between two virtual machines in the cloud. pUGW after receiving indication from the cloud server delivers any outstanding MN's packets to nUGW which in turn delivers them to MN.

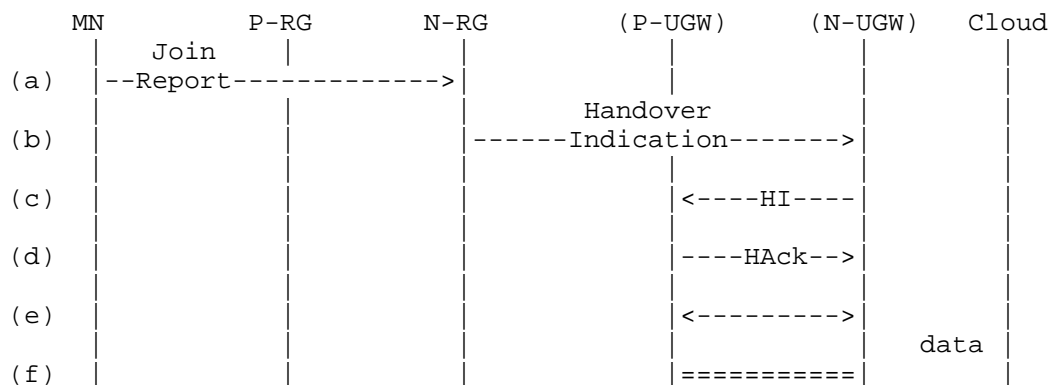


Figure 3: Reactive Handover

Note that Handover Initiate and Handover Acknowledge messages used in this document carry only a subset of parameters defined in [RFC5949]. Also no involvement with the Local Mobility Anchor (LMA) is needed.



#### 4.3. Route Establishment

After handover, SDN route establishment in upstream routers needs to take place. In this case NETCONF protocol [RFC6241] and YANG modeling [RFC6020] are used.

Client and Server exchange their capabilities using NETCONF message layer message called hello messages. Client builds and sends an operation defined in YANG module, encoded in XML, within RPC request message [RFC6244]. Server verifies the contents of the request against the YANG module and then performs the requested operation and then sends a response, encoded in XML, in RPC reply message.

Defining configuration data is the primary focus of YANG. Configuration data is writable (rw - read-write) data that is required to transform a system from its initial default state into its current state. There is also state data (ro - read-only) which is a set of data that has been obtained by the system at runtime. An example is routing table changes made by routing protocols in response to the ongoing traffic.

A YANG module for routing management is given in [I-D.ietf-netmod-routing-cfg]. The core routing data model consists of three YANG modules, ietf-routing, ietf-ipv4-unicast-routing, ietf-ipv6-unicast-routing. The core routing data model has two trees: configuration data and state data trees. "routing-instance" or "rib" trees have to be populated with at least one entry in the device, and additional entries may be configured by a client. Normally the server creates the required item as an entry in state data. Additional entries may be created in the configuration by a client via the NETCONF protocol using RPC messages like edit-config and copy-config.

The user may provide supplemental configuration of system- controlled entries by creating new entries in the configuration with the desired contents. In order to bind these entries with the corresponding entry in the state data list, the key of the configuration entry has to be set to the same value as the key of the state entry.

RPC get message can be used to retrieve all or part of the running configuration data store merged with the device's state data. RPC get-config operation retrieves configuration data only. RPC fib-route message defined in [I-D.ietf-netmod-routing-cfg] retrieves a routing instance for the active route in the Forwarding Information Base (FIB) which is the route that is currently used for sending datagrams to a destination host whose address is passed as an input parameter. So fib-route message plays the role of show route command line interface command.

NETCONF protocol and ietf-routing YANG module can be used for route establishment after handover. As a result for MNs that handover, upstream routing that takes place is not modified up to the lowest level of routers. The lowest level of routers handle the mobility but only proper modifications are needed so that the packets reach the right Unified Gateway, i.e. nUGW.

I2RS Agent as NETCONF Server in nUGW and in pUGW inform the handover to I2RS Clients as NETCONF Client upstream. I2RS Agent at pUGW removes any routing information for MN by first using get-config to retrieve the active route for MN and then an edit-config message with delete operation to delete the active route making sure that the same key is used.

I2RS Agent in nUGW after the handover needs to add a new routing table entry for MN. Due to the topological correctness of MN's prefix, the new route could be a host route. Next this route is propagated upstream. In this case, nUGW starts the process. SDN Controller as I2RS Client knows that MN handover is successfully completed. SDN Controller starts the upstream route establishment process starting with the I2RS Agent at the upstream router. Either a new route or the host route is added with shorter prefix. Route propagation continues until MN's prefix becomes topologically correct at which point route propagation stops.

Route propagation at the lowest level starts with I2RS Agent as NETCONF Server in nUGW informing the handover to I2RS Client as NETCONF Client upstream. I2RS Client then checks any routing information for MN by first using get-config to retrieve the active route for MN to make sure that none exists and MN prefix is topologically incorrect. Next I2RS client issues an edit-config message with create operation to add a host route for the new MN. I2RS Client then informs this route to I2RS Client upstream which creates a similar route at the I2RS Agent upstream.

In Appendix A, we present our experimental work using YANG data modelling language which has its own syntax and NETCONF protocol which is XML-based remote procedure call (RPC) mechanism. HTTP based RESTCONF could also be used in a similar way. Two RPC call examples are given. RPC call in Appendix A.3 shows a get-config filter with rtr0 as the key and it is used to retrieve a specific route with a given destination prefix and next hop address. RPC call in Appendix A.4 shows an example edit-config create operation to create a new route with specific route parameters.

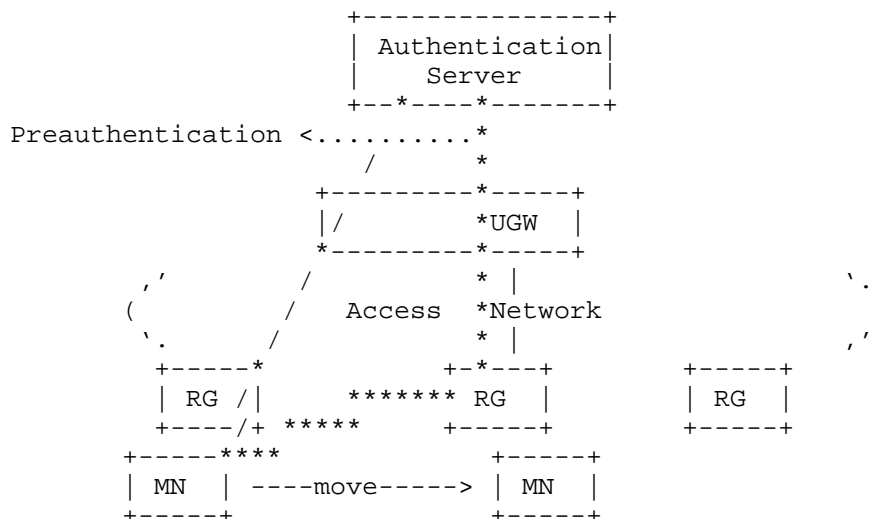
#### 4.4. Authentication and Charging

##### 4.4.1. Authentication

Extensible Authentication Protocol (EAP)[RFC3748] is preferred for MN authentication in IEEE 802.11 (Wi-Fi) network. When a MN tries to connect to the WiFi, it needs to mutually authenticate with the network server first. A successful EAP authentication procedure must result in a Pairwise Master Key(PMK) (defined in [IEEE-802.11i]) for the traffic encryption between the MN and the AR.

When a MN moves at home or in a hot spot from one AP to another AP in the same UGW, it is possible that it may to undergo a full EAP authentication (as defined in[RFC3748]). However, there are simplified several authentication methods (defined in [IEEE-802.11-2007] ):

- o Preauthentication: When The MN supplicant may authenticate with both pRG and nRG at a time. Successful completion of EAP authentication between the MN and nRG establishes a pair of PMKSA on both the MN and nRG. When the MN moves to the nRG, the authentication has already done, which is shown as follows.



##### Preauthentication

- o Cached PMK: The RG reserves the PMK as a result of previous authentication. When the MN is roaming back to the previous RG, if a successful EAP authentication has happened. The MN can retain the 802.11 connection based on PMK information reserved.

When the authentication is handled by the UGW as an Authenticator. When the MN moves to the nRG, a join report packet will be initiated from the MN to nRG for IEEE802.11 connection to the same UGW. The nRG can retain the PMK information from the UGW which is reserved during the successful authentication procedure between the MN and the pRG, as shown in Figure 4.

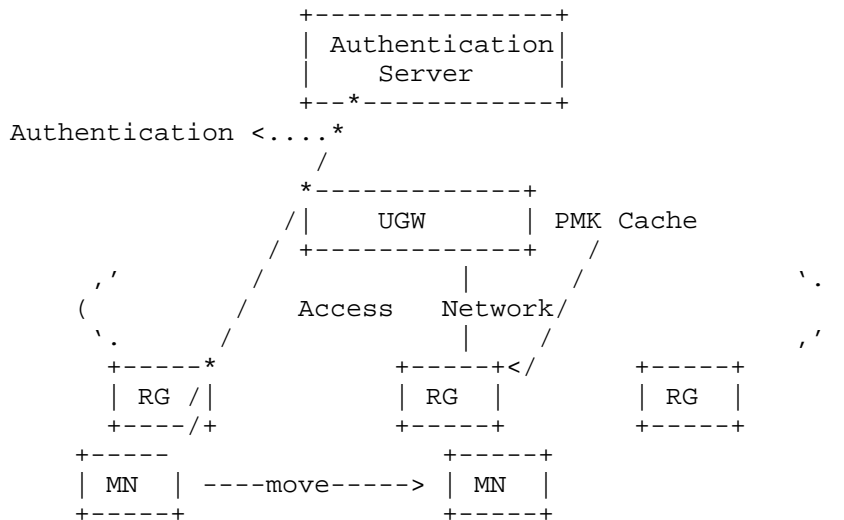
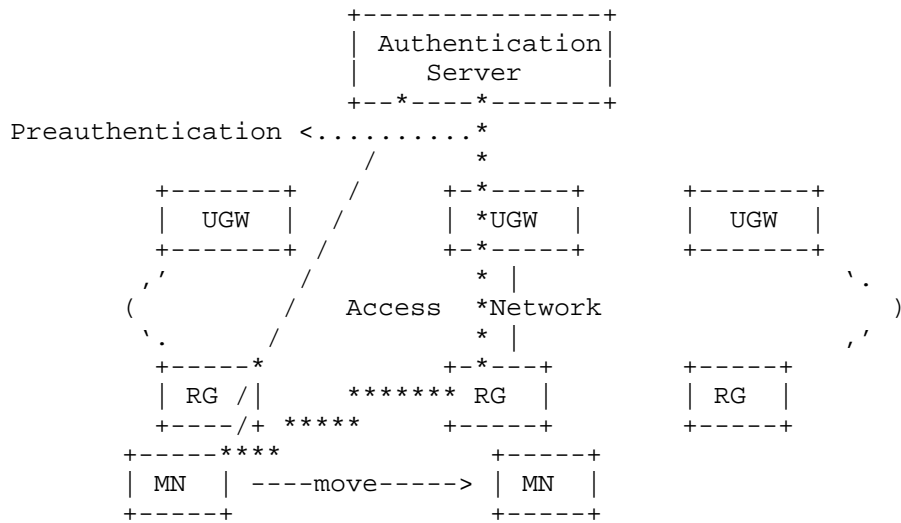


Figure 4: Cached PMK-UGW Authenticator

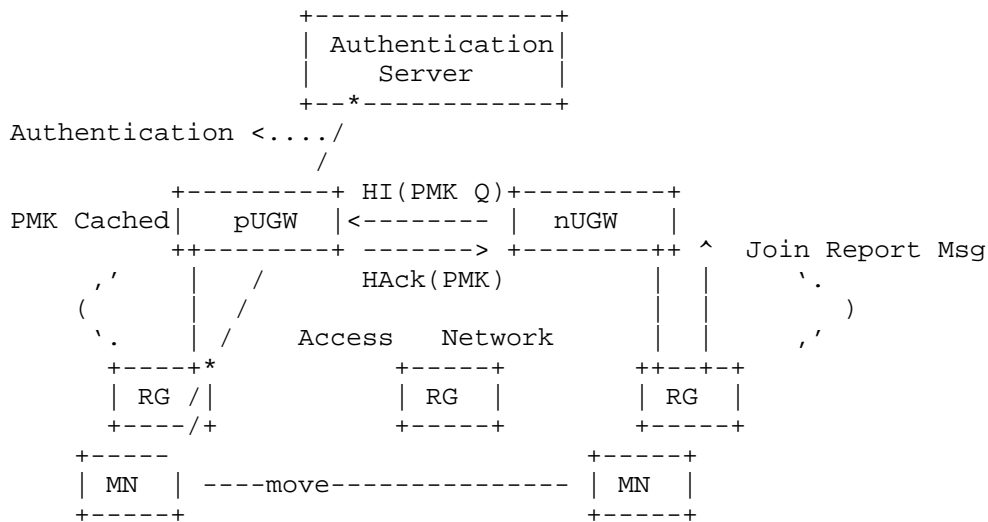
When a MN moves at home or in a hot spot from one AP to another AP in the same UGW, it is possible that it may to undergo a full EAP authentication (as defined in[RFC3748]). However, there are several simple authentication methods (defined in [IEEE-802.11-2007] ):

When MN moves from one UGW into another UGW, a join report packet will be initiated from the MN to nRG for IEEE802.11 connection. It is possible that it may to undergo a full EAP authentication (as defined in[RFC3748]). However, because of service performance and continuity requirement, the operators prefer to avoid the full EAP authentication. There are several simplified authentication methods (defined in [IEEE-802.11-2007] ):

- o Preauthentication: MN supplicant may authenticate with both pRG and nRG at a time. Successful completion of EAP authentication between the MN and nRG establishes a pair of PMKSA on both the MN and nRG. When the MN moves to the nRG, the authentication has already been completed, which is shown as follows.



- o **Cached PMK:** The RG reserves the PMK as a result of previous authentication. When the MN is roaming back to the previous RG, if a successful EAP authentication has happened. The MN can retain the 802.11 connection based on PMK information reserved. When the authentication is handled by the UGW as an Authenticator. When the MN moves to the nRG, a join report packet will be initiated from the MN to nRG for IEEE802.11 connection to nUGW. The nRG can retain the PMK information from the nUGW, the nUGW may can retain the reserved PMK from the pUGW based on HI message.



The above Layer 2 operations do not affect Layer 3. MN does not change the prefix assigned to it initially.

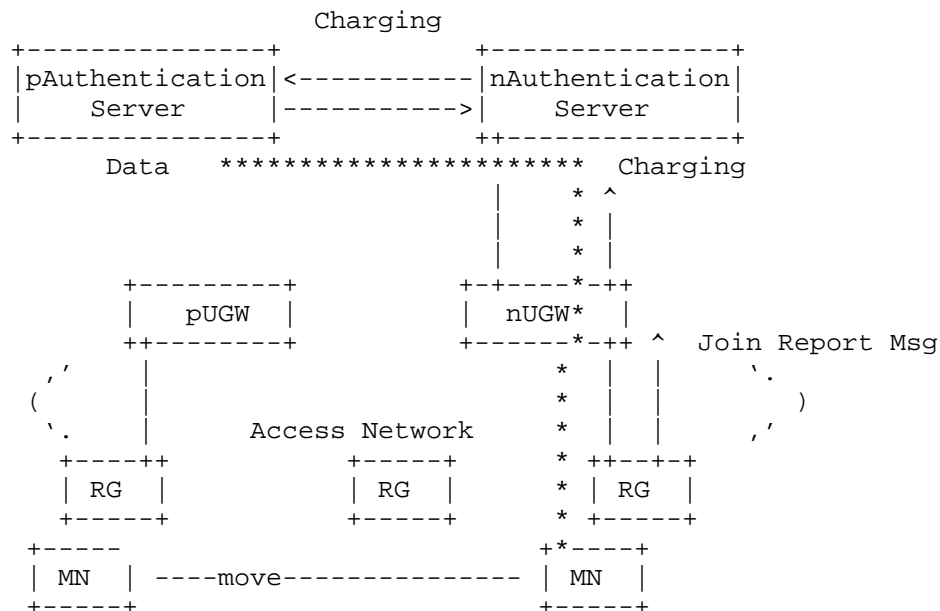
#### 4.4.2. Charging

When MN moves from one UGW into another UGW, the charging needs to be considered. In this document we describe two cases, one operator and two interworking operators.

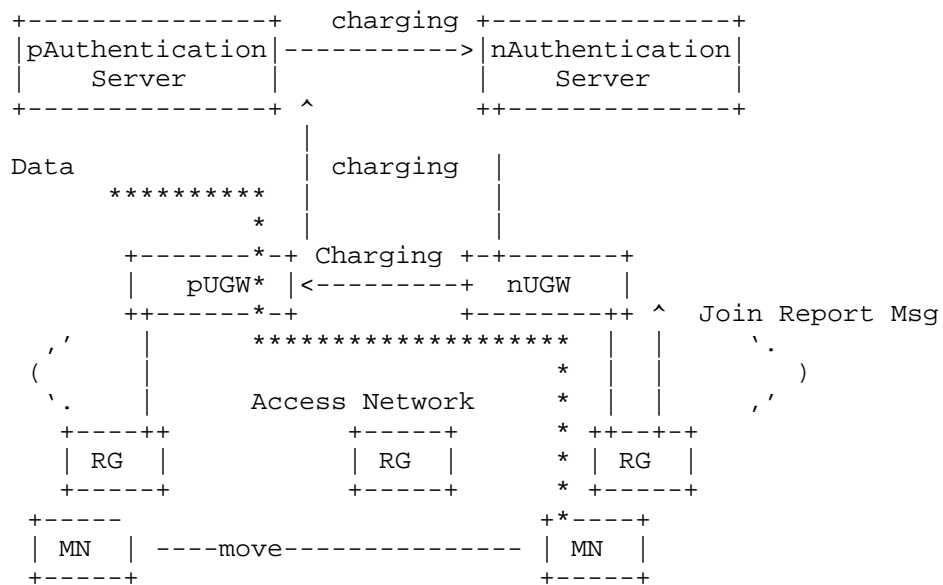
One operator case.

Two operators case. If the pUGW and nUGW are belonging to two different operators.

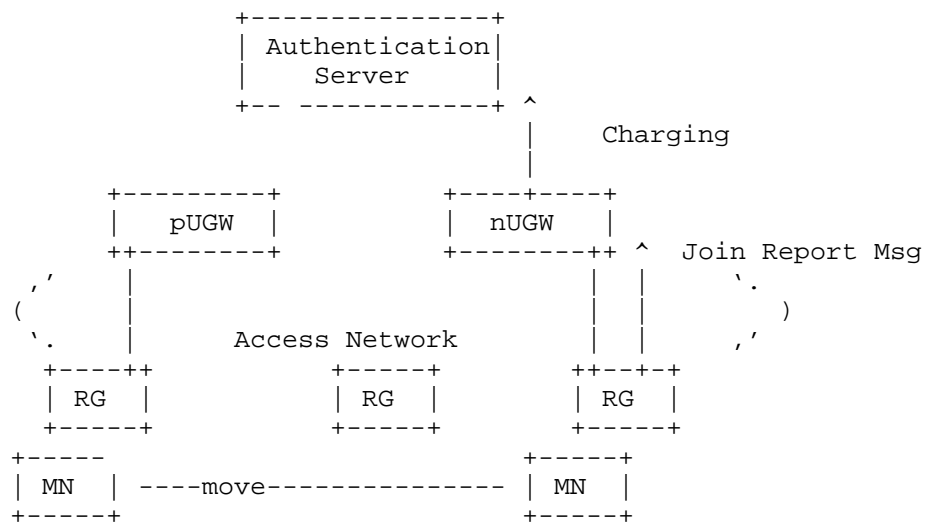
There are two possibilities. The traffic is directed to the visited network, and traffic routed back to home.



Two operators interworking - Traffic offload in visited network



Two operators interworking - Traffic routed back to home



Charging in one operator

## 5. Multicast Support

Multicast communication to the mobile nodes can be supported with an Multicast Listener Discovery (MLD) Proxy at the Unified Gateway [RFC4605]. Downstream protocol operations between the UGW and the mobile nodes, is the MLD protocol [RFC3810]. Both any source and source specific multicast are supported.

The mobile nodes send MLD Report message when joining a multicast group [RFC3590]. UGW or MLD Proxy sends an aggregated join message upstream. MN and UGW interface works as described in [RFC6224]. After MN joins the group it starts to receive multicast data.

After a handover the mobile node moves to the next UGW, the next UGW needs to get membership or listening state of this MN containing group address and source list. For this purpose, Active Multicast Subscription mobility option (Type 57 for IPv6) [RFC7161] can be used to transfer mobile node's multicast context or subscription information from the previous UGW to the next UGW, as explained below.

In case of predictive handover, pUGW and nUGW follow the sequence of steps shown in Figure 2. In case MN has multicast context established before handover pUGW MUST transfer MN's multicast context to nUGW. pUGW MUST add Active Multicast Subscription mobility option to HI message.

For reactive handover pUGW and nUGW follow the sequence of steps shown in Figure 3. In case MN has multicast context established before handover pUGW MUST transfer MN's multicast context to nUGW. pUGW MUST add Active Multicast Subscription mobility option to HAcK message.

After receiving the multicast context, nUGW upstream joins any new multicast groups on behalf of MN. Downstream, nUGW maps downstream point-to-point link to a proxy instance.

### 5.1. IPv4 Support for Multicast

For MNs with IPv4 addresses, multicast communication to MNs can be supported similar to the way explained above in Section 5. Multicast group management is done using IGMP with IGMP Proxy at the UGW.

In case of handover, the Active Multicast Subscription option compatible with IGMP-based format which transports the multicast membership context of the mobile node is used in handover messaging. Active Multicast Subscription option has type value of 56 for IPv4 [RFC7161].



## 6. IPv4 Support

IPv4 can be supported similarly as in vEPC [I-D.matsushima-stateless-uplane-vepc]. UGW stays as IPv6 node receiving from all RGs IPv6 packets and forwarding them upstream.

IPv4 MN is supported at the RG. RG has B4 functionality of DS-Lite [RFC6333], CLAT entity for 4G4XLAT [RFC6877], Lightweight B4 [RFC7596] or MAP Customer Edge [RFC7597]. RG encapsulates IPv4 packets using these protocols into IPv6 packets making sure that UGW stays IPv6 only.

## 7. Security Considerations

This document introduces no extra new security threat. Security considerations stated in [I-D.ietf-i2rs-architecture] apply.

## 8. IANA Considerations

TBD.

## 9. Acknowledgements

We would like to thank Ladislav Lhotka, Satoru Matsushima for valuable advice.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3590] Haberman, B., "Source Address Selection for the Multicast Listener Discovery (MLD) Protocol", RFC 3590, DOI 10.17487/RFC3590, September 2003, <<http://www.rfc-editor.org/info/rfc3590>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.

- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <<http://www.rfc-editor.org/info/rfc3748>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<http://www.rfc-editor.org/info/rfc3810>>.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, DOI 10.17487/RFC4605, August 2006, <<http://www.rfc-editor.org/info/rfc4605>>.
- [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, <<http://www.rfc-editor.org/info/rfc4861>>.
- [RFC5949] Yokota, H., Chowdhury, K., Koodli, R., Patil, B., and F. Xia, "Fast Handovers for Proxy Mobile IPv6", RFC 5949, DOI 10.17487/RFC5949, September 2010, <<http://www.rfc-editor.org/info/rfc5949>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, DOI 10.17487/RFC6121, March 2011, <<http://www.rfc-editor.org/info/rfc6121>>.
- [RFC6224] Schmidt, T., Waehlisch, M., and S. Krishnan, "Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains", RFC 6224, DOI 10.17487/RFC6224, April 2011, <<http://www.rfc-editor.org/info/rfc6224>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.

- [RFC6244] Shafer, P., "An Architecture for Network Management Using NETCONF and YANG", RFC 6244, DOI 10.17487/RFC6244, June 2011, <<http://www.rfc-editor.org/info/rfc6244>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<http://www.rfc-editor.org/info/rfc6333>>.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <<http://www.rfc-editor.org/info/rfc6877>>.
- [RFC7161] Contreras, LM., Bernardos, CJ., and I. Soto, "Proxy Mobile IPv6 (PMIPv6) Multicast Handover Optimization by the Subscription Information Acquisition through the LMA (SIAL)", RFC 7161, DOI 10.17487/RFC7161, March 2014, <<http://www.rfc-editor.org/info/rfc7161>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<http://www.rfc-editor.org/info/rfc7217>>.
- [I-D.ietf-netmod-routing-cfg]  
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-20 (work in progress), October 2015.
- [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <<http://www.rfc-editor.org/info/rfc7596>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<http://www.rfc-editor.org/info/rfc7597>>.

[IEEE-802.11i]

"Institute of Electrical and Electronics Engineers,  
"Unapproved Draft Supplement to Standard for  
Telecommunications and Information Exchange Between  
Systems-LAN/MAN Specific Requirements -Part 11: Wireless  
LAN Medium Access Control (MAC) and Physical Layer (PHY)  
Specifications: Specification for Enhanced Security" "",  
September 2004.

[IEEE-802.11-2007]

"Institute of Electrical and Electronics Engineers,  
"Telecommunications and information exchange between  
systems-Local and metropolitan area networks specific  
requirements -Part 11: Wireless LAN Medium Access Control  
(MAC) and Physical Layer (PHY) Specifications"", March  
2007.

[ONFv1.5] "Open Networking Foundation, "OpenFlow Switch  
Specification Version 1.5.0 ( Protocol version 0x06)""",  
January 2015.

## 10.2. Informative references

[I-D.ietf-dmm-requirements]

Chan, A., Liu, D., Seite, P., Yokota, H., and J. Korhonen,  
"Requirements for Distributed Mobility Management", draft-  
ietf-dmm-requirements-17 (work in progress), June 2014.

[I-D.matsushima-stateless-uplane-vepc]

Matsushima, S. and R. Wakikawa, "Stateless user-plane  
architecture for virtualized EPC (vEPC)", draft-  
matsushima-stateless-uplane-vepc-05 (work in progress),  
September 2015.

[I-D.ietf-i2rs-architecture]

Atlas, A., Halpern, J., Hares, S., Ward, D., and T.  
Nadeau, "An Architecture for the Interface to the Routing  
System", draft-ietf-i2rs-architecture-13 (work in  
progress), February 2016.

[UTD-Paper]

Jyotirmoy Banik, et al., "IEEE 24th International  
Conference on Computer Communication and Network 2015,  
"Enabling Distributed Mobility Management: A Unified  
Wireless Network Architecture Based on Virtualized Core  
Network", DOI: 10.1109/ICCCN.2015.7288404", August 2015.

## Appendix A. YANG and RPC Programs

In this annex, we present our YANG and RPC solutions.

### A.1. Host Routing Module

We first obtained host routing YANG module using IPv6 unicast routing module (`ietf-ipv6-unicast-routing`) which is part of `ietf-routing` module. This module defines a list of host routes which contain host address/prefix and corresponding next hop address.

### A.2. Route Establishment RPCs

This program runs on `ietf-ipv6-unicast-host-routing` YANG module which has been obtained from `ietf-ipv6-unicast-routing` module by defining the `hostroute` as a list of host routes. First issue a `get-config` on the configuration data to extract the existing route for the host whose prefix is `destination-prefix` and the next-hop is the next-hop address. Delete the route at `pUGW`. This procedure deletes the route at `pUGW`.

```
<rpc message-id="101" ... >
```

```
get-config(running, filter=(destination-prefix, next-hop-address))
```

```
// check the reply, make sure it is OK, i.e. does not contain <rpc-  
error> element.
```

```
edit-config(running, delete, config)
```

Add a new route for MN at `nUGW`. This route is based on MN's prefix, `destination-prefix` and the upstream router to which MN's traffic should be routed, `next-hop-address`.

```
<rpc message-id="101" ... >
```

```
get-config(running, filter=(destination-prefix, next-hop-address))
```

```
// check the reply, make sure it is an error, i.e. it contains <rpc-  
error> element of type application and tag data-missing i.e. no route  
exists
```

```
edit-config(running, create, config)
```

Add a new host route for MN at `nUGW`. This route is added in case MN's prefix is not topologically correct at `nUGW` and routers above.

```
<rpc message-id="101" ... >
```

```
get-config(running, filter=(destination-prefix, next-hop-address))

// check the reply, make sure it is an error, i.e. it contains <rpc-
error> element of type application and tag data-missing, i.e. no
route exists

edit-config(running, create, config)
```

We next show in Appendix A.3 and Appendix A.4 example RPC procedures for get-config and edit-config. Some arbitrary values for destination prefix and next hop address are used.

#### A.3. get-config RPC procedure for host routes

This RPC procedure shows a get-config filter to find a record in the routing information base for a specific host whose prefix is 2001:db8:1:0::/64 and the next-hop is 2001:db8:0:1::2. It could be used for the get-config's in Appendix A.2. We validated this procedure using the public domain tool pyang.

```

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns:v6ur="urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing"
  xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
  xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"
  xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
  xmlns:rt="urn:ietf:params:xml:ns:yang:ietf-routing">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <t:top xmlns:t="urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-host-rou
ting">
        <t:routing-instance> rtr0 </t:routing-instance>

        <t:rib>
          <t:routes>
            <t:route>
              <t:destination-prefix>
                2001:db8:1:0::/64
              </t:destination-prefix>
              <t:outgoing-interface>eth1</t:outgoing-interface>
              <t:next-hop-address>
                2001:db8:0:1::2
              </t:next-hop-address>
            </t:route>
          </t:routes>
        </t:rib>
      </t:top>
    </filter>
  </get-config>
</rpc>

```

#### A.4. edit-config RPC procedure to create a host route

This RPC procedure shows an edit-config procedure to create a new host route in the routing information base for a specific host whose prefix is 2001:db8:1:0::/64 and the next-hop is 2001:db8:0:1::2. It could be used for the edit-config's in Appendix A.2. We validated this procedure using the public domain tool pyang.

```
<rpc message-id="101"
      xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      xmlns:v6ur="urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-routing"
      xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces"
      xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type"
      xmlns:ip="urn:ietf:params:xml:ns:yang:ietf-ip"
      xmlns:rt="urn:ietf:params:xml:ns:yang:ietf-routing">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="urn:ietf:params:xml:ns:yang:ietf-ipv6-unicast-host-routing
">
    <routing-instance> rtr0 </routing-instance>
    <rib>
    <routes>
      <route xc:operation="create">
        <destination-prefix >
          2001:db8:1:0::/64
        </destination-prefix>
        <outgoing-interface>eth1</outgoing-interface>
        <next-hop-address>
          2001:db8:0:1::2
        </next-hop-address>
      </route>
    </routes>
    </rib>
  </top>
</config>
</edit-config>
</rpc>
```

## Authors' Addresses

Behcet Sarikaya  
Huawei  
5340 Legacy Dr. Building 175  
Plano, TX 75024

Phone: +1 469 277 5839  
Email: sarikaya@ieee.org

Li Xue  
Unaffiliated

Email: 276076389@qq.com