DMM Working Group   M. Liebsch
Internet-Draft   NEC
Intended status: Standards Track   S. Matsushima
Expires: September 22, 2016   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-02.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.

Copyright Notice

Table of Contents

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).

```
                    +-------------------------+
                    |     Mobility Control    |
                    +-------------------------+
                      |           |        |
                      |           |        |
                      |           |        |
              \ /     V           V        V
        +--+     -o-  +---+       +---+    +---+   +--+
        |MN| ---- |---|DPN|<=======|DPN|<----|DPN|<--|CN|
        +--+     |   +---+       +---+    +---+   +--+
                     Rules:      Rules:    Rules:
                     Encap,Decap, Encap,Decap Policy-Route,
                     Forward,QoS  Forward,QoS Forward
```

      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.

```
                +-------------------------+
                | Mobility Control-Plane  |
                |                         |
                |+--------[API]----------+|
                ||   FPC Client Function ||
                |+----------^------------+|
                +-----------|-------------+
                            |
                            | DMM FPC protocol
                            |
                +-----------|-------------+
                |+----------v------------+|
                ||   FPC Agent Function  ||
                |+----------------------+|
                |                         |
                |   DPN Configuration API |
                +-------------------------+
```
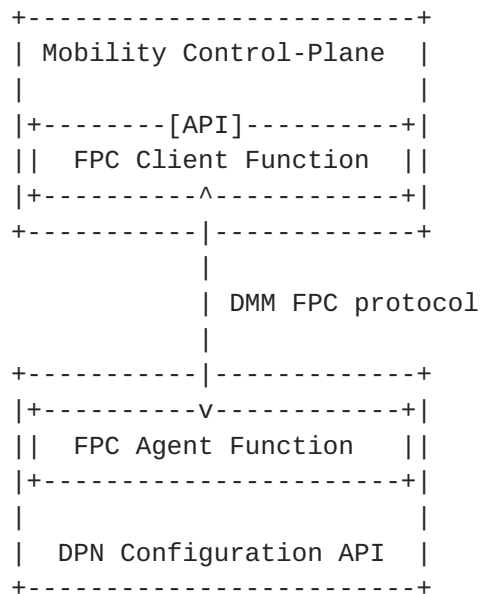
Figure 2: Functional reference architecture for DMM Forwarding Policy
                     Configuration (FPC)

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.

```
          TrafficDescriptor_1-+              +--Property_1
                              |              |
          TrafficDescriptor_2-+---<PORT#>---+--Property_2
                  :                |   +--------+|
                  :                :  /Adm Data/ +--Property_3
                  :                | +--------+  :       :
          TrafficDescriptor_M-+   +--------+ +--Property_N
                                  /OP Data/
                                 +-------+
        +-------------------+          +--------------------+
        | Bind 1..M traffic |          |  Bind 1..N traffic |
        |    templates to   | -------> |  treatment actions |
        |        a port     |          |       to a port    |
        +-------------------+          +--------------------+


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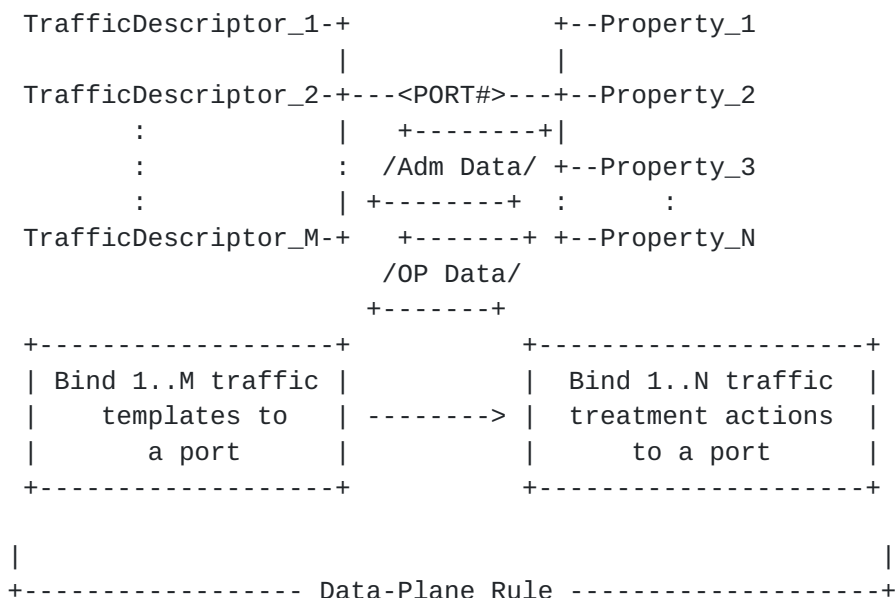
        Figure 3: Structure of rules on Client/Agent defining Data-Plane
                            traffic treatment

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]          | Identifies a logical Port   |
|                | [8,PROP_ID]          | and one of its properties   |
+--------------------------------------------------------------------+
|   PRT_TD_ID    | [32,PRT_ID]          | Identifies a logical Port   |
|                | [8,TD_ID]            | and a traffic descriptor    |
|                |                      | that applies to the port    |
+--------------------------------------------------------------------+
|    CLI_ID      | [16, Carrier ID]     | Identifies an               |
|                | [16, Network ID]     | FPC Client function         |
|                | [32, Client ID]      |                             |
+--------------------------------------------------------------------+
|    AGT_ID      | [16, Carrier ID]     | Identifies an               |
|                | [16, Network ID]     | FPC Agent function          |
|                | [32, Agent ID]       |                             |
+--------------------------------------------------------------------+
|    DPN_ID      | [16, Carrier ID]     | Identifies a Data Plane     |
|                | [16, Network ID]     | Node (DPN)                  |
|                | [32, DPN ID]         |                             |
+--------------------------------------------------------------------+
|  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]   | Property NAT defines    |
|               | [in_dst_ip][out_dst_ip]   | IP address and port     |
|               | [in_src_port][out_src_port]| re-write rules         |
|               | [in_dst_port][out_dst_port]|                        |
+------------------------------------------------------------------+
|   PROP_QOS    | [QoS index type][index]   | Property QoS refers to  |
|               | [DSCP]                    | 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]         | Include NSH             |
|               | [Service Index]           |                         |
+------------------------------------------------------------------+
```

   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-#]   | GRE_encapsulation        |
|                   | [key]                | configuration attribute  |
+--------------------------------------------------------------------+
|   GTP_CONF        | [TEID_local]         | GTP-U encapsulation      |
|                   | [TEID_remote]        | configuration attribute  |
|                   | [seq-#]              |                          |
+--------------------------------------------------------------------+
```

Figure 8: Model I Protocol Attributes: Protocol-specific

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

Figure 9: Protocol Attributes: Traffic Description Container

```
+--------------------------------------------------------------------+
|    Attribute      | Format Clarification |      Description         |
+====================================================================+
|                        Traffic Descriptors                         |
+--------------------------------------------------------------------+
|  TD_DST_IP        | [IP address]         | Aggregated or per-host dst |
|                   | [Prefix Len]         | IP address/prefix rule   |
+--------------------------------------------------------------------+
|  TD_SRC_IP        | [IP address]         | Aggregated or per-host src |
|                   | [Prefix Len]         | 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 | A Monitor                |
|                    | [REPORT CONFIG]      |                          |
+----------------------------------------------------------------------+
| REPORT_CONFIG      | [8, REPORT-TYPE]     | The type of report and   |
|                    | [TYPE_SPECIFIC_INFO] | type-specific            |
|                    |                      | configurations           |
+----------------------------------------------------------------------+
| PERIODIC_CONFIG    | [32, period]         | REPORT-TYPE is PERIODIC,  |
|                    |                      | period specifies the      |
|                    |                      | report interval (ms)      |
+----------------------------------------------------------------------+
| THRESHOLD_CONFIG   | [32, low]            | REPORT-TYPE is THRESHOLD, |
|                    | [32, hi]             | 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]        | Monitors to deregister,   |
|                    |  [boolean]           | Boolean (optional)        |
|                    |                      | indicates if a successful |
|                    |                      | DEREG triggers a NOTIFY   |
|                    |                      | with final data           |
+----------------------------------------------------------------------+
| NOTIFY_INFO        | [32, Notification-Id]|ID used for Client ordering|
|                    | [MONITOR-ID]         |Monitor-ID of the NOTIFY,  |
|                    | [32, TRIGGER]        |TRIGGER for the NOTIFY,    |
|                    | [32, timestamp]      |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 the Client about the status of a        |
|    NOTIFY          | 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 themself 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
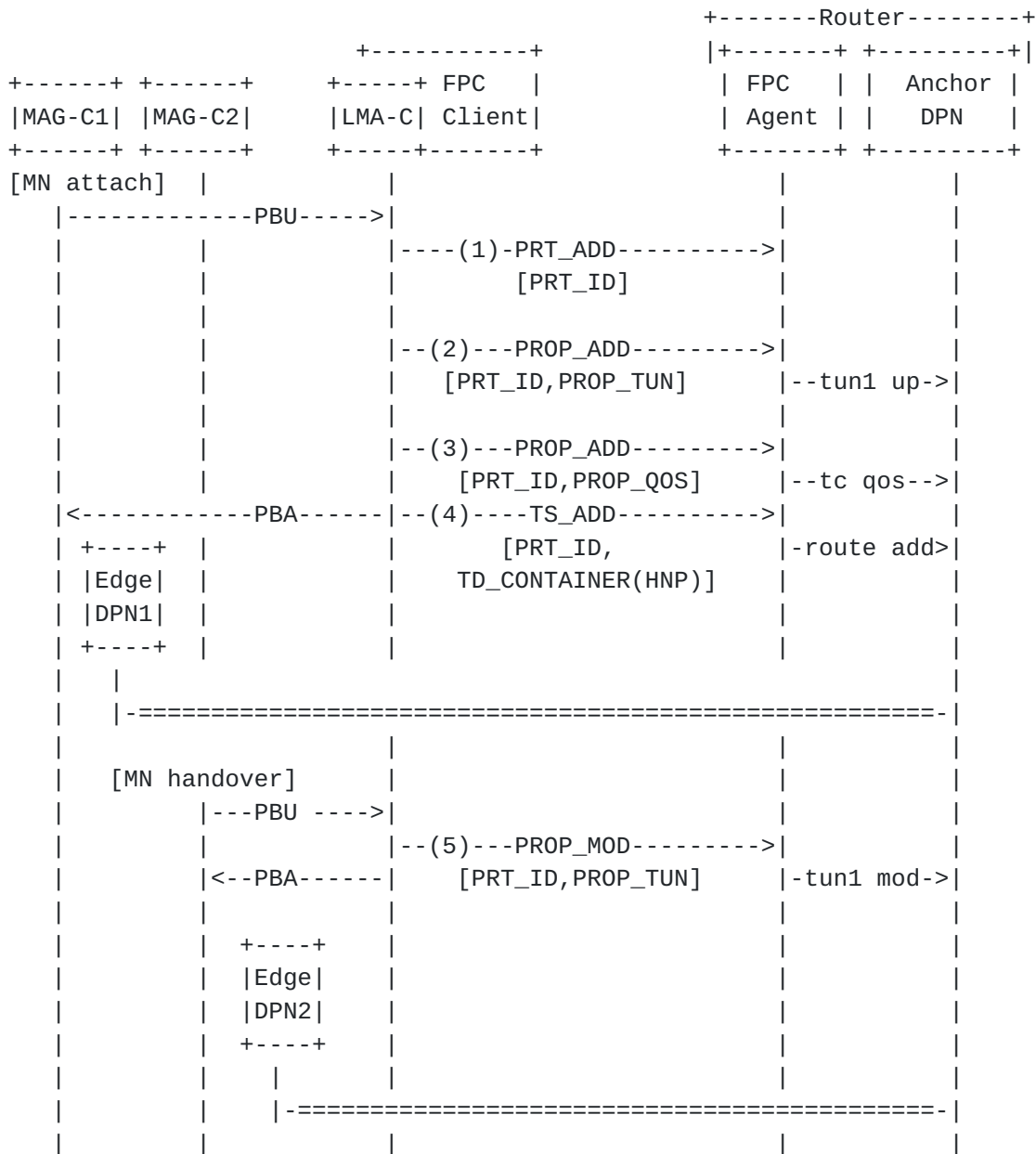with the Anchor-DPN, e.g. a Router.

```
                                            +-------Router--------+
                          +-----------+     |+-------+ +---------+|
  +------+ +------+     +-----+ FPC   |     | FPC   | | Anchor  |
  |MAG-C1| |MAG-C2|     |LMA-C| Client|     | Agent | |  DPN    |
  +------+ +------+     +-----+-------+     +-------+ +---------+
  [MN attach]  |            |                  |          |
    |------------PBU----->|                   |          |
    |           |          |----(1)-PRT_ADD---------->|          |
    |           |          |        [PRT_ID]          |          |
    |           |          |                          |          |
    |           |          |--(2)---PROP_ADD--------->|          |
    |           |          |    [PRT_ID,PROP_TUN]     |--tun1 up->|
    |           |          |                          |          |
    |           |          |--(3)---PROP_ADD--------->|          |
    |           |          |      [PRT_ID,PROP_QOS]   |--tc qos-->|
    |<------------PBA------|--(4)----TS_ADD---------->|          |
    | +----+ |            |         [PRT_ID,         |-route add>|
    | |Edge| |            |      TD_CONTAINER(HNP)]  |          |
    | |DPN1| |            |                          |          |
    | +----+ |            |                          |          |
    |    |                                                      |
    |    |-=====================================================-|
    |           |          |                  |          |
    |     [MN handover]    |                  |          |
    |          |---PBU ---->|                 |          |
    |          |          |--(5)---PROP_MOD--------->|          |
    |          |<--PBA------|     [PRT_ID,PROP_TUN]   |-tun1 mod->|
    |          |          |                          |          |
    |          | +----+   |                          |          |
    |          | |Edge|   |                          |          |
    |          | |DPN2|   |                          |          |
    |          | +----+   |                          |          |
    |          |    |     |                          |          |
    |          |    |-=========================================-|
    |          |          |                  |          |
```

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
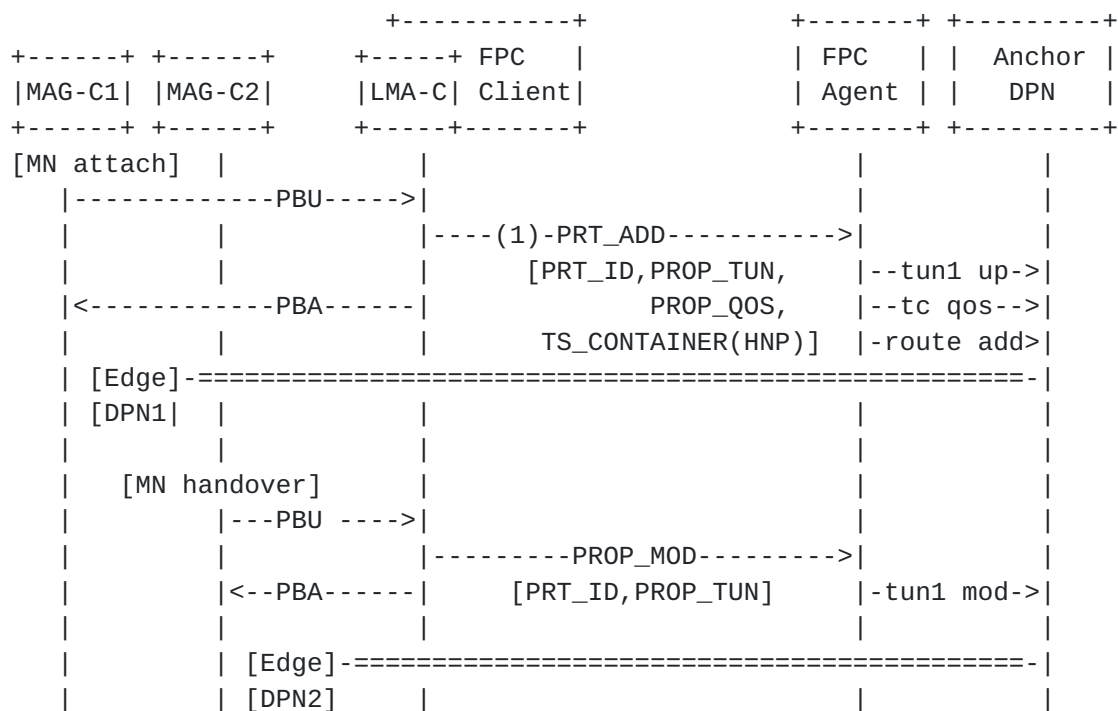Figure 15.

```
                          +-----------+
 +------+ +------+        +-----+ FPC  |         +-------+ +---------+
 |MAG-C1| |MAG-C2|        |LMA-C| Client|         | FPC   | | Anchor  |
 +------+ +------+        +-----+-------+         | Agent | |  DPN    |
                                                  +-------+ +---------+
 [MN attach]  |                |                    |          |
    |-------------PBU----->|                         |          |
    |         |              |----(1)-PRT_ADD----------->|       |
    |         |              |       [PRT_ID,PROP_TUN,  |--tun1 up->|
    |<------------PBA------|           PROP_QOS,        |--tc qos-->|
    |         |              |          TS_CONTAINER(HNP)] |-route add>|
    | [Edge]-=============================================-|
    | [DPN1|  |              |                          |          |
    |         |              |                          |          |
    |   [MN handover]        |                          |          |
    |         |---PBU ---->|                             |          |
    |         |              |---------PROP_MOD--------->|          |
    |         |<--PBA------|      [PRT_ID,PROP_TUN]     |-tun1 mod->|
    |         |              |                          |          |
    |         | [Edge]-=======================================-|
    |         | [DPN2]     |                            |          |
```

            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,| Encapsulation Type         |
|                     | ENCAP_IP]            |                            |
+-------------------------------------------------------------------------+
|TUN_TYPE_UDP         |[SRC_PRT, DST_PRT]    | UDP Direction - Source     |
|                     |                      | or Destination             |
+-------------------------------------------------------------------------+
|TUN_TYPE_GRE         |[UPLINK_GRE_KEY,      | GRE Tunnel Type            |
|                     | DOWNLINK_GRE_KEY]    |                            |
```

```
+-------------------------------------------------------------------+
|TUN_IF_MTU             |[MTU]                  | Tunnel Interface MTU   |
|                       |                       |                        |
+-------------------------------------------------------------------+
|TUN_PAYLOAD_TYPE       |[PAYLOAD_IPV4,         | Tunnel Payload Type    |
|                       | PAYLOAD_IPV6,         |                        |
|                       | PAYLOAD_DUAL]         |                        |
+-------------------------------------------------------------------+
|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      |Aggregate Maximum       |
|                       | (32 bit)]             |Bitrate                 |
+-------------------------------------------------------------------+
|GBR                    |[Unsigned Integer      |Guaranteed Bitrate      |
|                       | (32 bit)]             |                        |
+-------------------------------------------------------------------+
|TCLASS                 |[Unsigned Integer      |Traffic Class           |
|                       | (32 bit)]             |                        |
+-------------------------------------------------------------------+
|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.

   o 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.

   o 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.

   o 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.

   o QUERY_TUNNEL_IF -

   o CREATE_POLICY_ROUTE - Message can include INPUT_IF, OUTPUT_IF,
      NEXT_HOP_IP_GW_ADDR, VRF_ID, PBR_ID, CTRL_PLANE_ID, CONTEXT_ID.

   o DELETE_POLICY_ROUTE - Message can include PBR_ID, CTRL_PLANE_ID,
      CONTEXT_ID.

   o ADD_TRAFFIC_SELECTOR - Message can include TRAFFIC_SELECTOR_ACL,
      PBR_ID, CTRL_PLANE_ID, CONTEXT_ID.

   o DELETE_TRAFFIC_SELECTOR - Message can include
      TRAFFIC_SELECTOR_ACL, PBR_ID, CTRL_PLANE_ID, CONTEXT_ID.

   o QUERY_POLICY_ROUTE -

   o CREATE_IP_ROUTE - Message can include DST_IP_SUBNET,
      DST_IP_SUBNET_MASK, OUTPUT_IF, VRF_ID, ROUTE_ID, CTRL_PLANE_ID,
      CONTEXT_ID.

   o DELETE_IP_ROUTE - Message can include ROUTE_ID, CTRL_PLANE_ID,
      CONTEXT_ID.

   o QUERY_IP_ROUTE -

   o 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. 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.

6.1. FPC Base

6.1.1. 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

6.1.2. 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-discriptor-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-discriptor-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-discriptor-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

6.2. FPC PMIP

6.2.1. 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'
|
                     ../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-
carriers/carrier/qosprofile.
    augment "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-profile/
fpcp-base:index-type" {
        case pmip {
            when "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-
profile/fpcp-base:qos-index-type = 'fpcp-qos-index-pmip'";
            uses qosattribute-pmip;
        }
    }
```

```
    // Configuration choice augmentation in the fpcp-base under the fpcp-
carriers/carrier/traffic-descriptor.
    augment "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:traffic-
descriptor/fpcp-base:descriptor-type" {
        case traffic-selector-mip6 {
            when "/fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:traffic-
descriptor/fpcp-base:traffic-descriptor-type = 'traffic-selector-mip6'";
```

```
            uses traffic-selectors:traffic-selector;
        }
    }

    // Operational choice augmentation in the fpcp-base under the port-entries/
port-entry/property.
    augment "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:property/
fpcp-base:qos/fpcp-base:index-type" {
        case pmip {
            when "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-
base:property/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/
port-entry/forwarding-rule.
    augment "/fpcp-base:port-entries/fpcp-base:port-entry/fpcp-base:forwarding-
rule/fpcp-base:descriptor-type" {
        case traffic-selector-mip6 {
            when "/fpcp-base:port-entries/fpcp-base:port-entry/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_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-
property/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-index-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:modifying-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/
fpcp-base:forwarding-rule/fpcp-base:descriptor-type" {
        case traffic-selector-mip6 {
            when "/fpcp-base:rule_modify/fpcp-base:input/fpcp-base:modifying-
rule/fpcp-base:forwarding-rule/fpcp-base:traffic-descriptor-type = 'traffic-
selector-mip6'";
            uses traffic-selectors:traffic-selector;
        }
    }

}
```

Figure 21: caption1

6.2.2. **FPC PMIP tree**

```
module: ietf-dmm-fpcp-pmip
augment /fpcp-base:fpcp-carriers/fpcp-base:carrier/fpcp-base:qos-profile/fpcp-
base:index-type:
   +--:(pmip)
      +--rw dscp?                  inet:dscp
      +--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
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/
fpcp-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: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_add/fpcp-base:input/fpcp-base:adding-port/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
augment /fpcp-base:port_property_add/fpcp-base:input/fpcp-base:adding-property/
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:port_property_modify/fpcp-base:input/fpcp-base:modifying-
property/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: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
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

7. 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.

8. IANA Considerations

This document provides a data model and protocol operation for DMM
Forwarding Policy Configuration. Yang models are included to specify
the data model as well as to enable an implementation of this
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.

9. 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.

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>.

   [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>.

10.2. 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>.

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

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

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

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

   Email: sgundave@cisco.com

   Danny Moses

   Email: danny.moses@intel.com

   Lyle Bertz
   6220 Sprint Parkway
   Overland Park KS, 66251
   USA

   Email: lyleb551144@gmail.com