

DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 13, 2019

S. Matsushima
SoftBank
L. Bertz
Sprint
M. Liebsch
NEC
S. Gundavelli
Cisco
D. Moses
Intel Corporation
C. Perkins
Futurewei
April 11, 2019

YANG for Protocol for Forwarding Policy Configuration (FPC)
draft-ietf-dmm-fpc-yang-00

Abstract

This document provides YANG modules to exhibit a data model for the information models defining Forwarding Policy Configuration (FPC) to manage the separation of data-plane and control-plane. FPC defines a flexible mobility management system using FPC agent and FPC client functions. A FPC agent provides an abstract interface to the data-plane. The FPC client configures data-plane nodes by using the functions and abstractions provided by the FPC agent for the data-plane nodes.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on October 13, 2019.

Internet-Draft

YANG for DMM FPC Protocol

April 2019

Copyright Notice

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

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

Table of Contents

1.	Introduction	2
2.	Terminology	2
3.	YANG Data Model for the FPC protocol	3
3.1.	FPC YANG Model	5
3.2.	FPC YANG Settings and Extensions Model	27
3.3.	PMIP QoS Model	39
3.4.	Traffic Selectors YANG Model	46
3.5.	RFC 5777 Classifier YANG Model	54
4.	FPC YANG Tree Structure	62
5.	Work Team Participants	80
6.	References	80
6.1.	Normative References	80
6.2.	Informative References	81
	Authors' Addresses	81

[1.](#) Introduction

This document provides YANG modules to exhibit a data model for the information models defining Forwarding Policy Configuration (FPC) [[I-D.ietf-dmm-fpc-cpdp](#)] to manage the separation of data-plane and control-plane. FPC defines a flexible mobility management system using FPC agent and FPC client functions. A FPC agent provides an abstract interface to the data-plane. The FPC client configures data-plane nodes by using the functions and abstractions provided by the FPC agent for the data-plane nodes.

2. Terminology

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

following FPC-specific terms used in this document are defined in [[I-D.ietf-dmm-fpc-cpdp](#)].

- o Domain
- o DPN
- o FPC Agent
- o FPC Client
- o Mobility Context
- o Monitor
- o Policy
- o Template
- o Tenant
- o Topology

3. YANG Data Model for the FPC protocol

This section provides a type mapping for FPC structures in YANG. When being mapped to a specific information such as YANG the data type MAY change.

Action and Descriptor Templates are mapped as choices. This was done to ensure no duplication of Types and avoid use of identityref for typing.

Policy Expressions are provided as default values. NOTE that a static value CANNOT be supported in YANG.

Mapping of templates to YANG are performed as follows:

Value is defined as a choice statement for extensibility and therefore a type value is not necessary to discriminated types.

Generic attributes are distinguished by the "Settings" type and holds ANY value. It is in a data node under configurations.

The CONFIGURE and CONFIGURE-RESULT-NOTIFICATION use the yang-patch-

status which is a container for edits. This was done to maximize YANG reuse.

In the configure rpc, operation-id is mapped to patch-id and in an edit the edit-type is mapped to operation.

The Result-Status attribute is mapped to the 'ok' (empty leaf) or errors structure.

The Policy-Status is mapped to entity-state to reduce YANG size.

Five modules are defined:

- o ietf-dmm-fpc (fpc) - Defines the base model and messages for FPC that are meant to be static in FPC.
- o ietf-dmm-fpc-settingsext - A FPC module that defines the information model elements that are likely to be extended in FPC.
- o ietf-pmip-qos (pmip-qos) - Defines proxy mobile IPv6 QoS parameters per [\[RFC7222\]](#)
- o ietf-trafficselectors-types (traffic-selectors) - Defines Traffic Selectors per [\[RFC6088\]](#)
- o ietf-diam-trafficclassifier (diamclassifier) - Defines the Classifier per [\[RFC5777\]](#)

All modules defined in this specification make use of (import) ietf-inet-types as defined in [\[RFC6991\]](#).

ietf-dmm-fpc-settingsext and ietf-diam-trafficclassifier make use of (imports) ietf-yang-types as defined in [\[RFC6991\]](#).

ietf-dmm-fpc imports the restconf (ietf-restconf) [\[RFC8040\]](#) and yang patch (ietf-yang-patch) [\[RFC8072\]](#) modules.

ietf-pmip-qos and ietf-dmm-fpc-settings import the trafficselector from the ietf-traffic-selector-types module.

ietf-dmm-fpc-settings also imports the qosattribute (ietf-pmip-qos) and classifier (ietf-diam-trafficclassifier).

ietf-dmm-fpc-settingsext groups various settings, actions and descriptors and is used by the fpc module (ietf-dmm-fpc).

The following groupings are intended for reuse (import) by other modules.

- o qosoption (ietf-qos-pmip module)
- o qosattribute (ietf-qos-pmip module)
- o qosoption (ietf-qos-pmip module)
- o Allocation-Retention-Priority-Value (ietf-qos-pmip module)
- o trafficselector (ietf-traffic-selector-types)
- o classifier (ietf-diam-trafficclassifier)
- o packet-filter (ietf-dmm-fpc-settingsext)
- o instructions (ietf-dmm-fpc-settingsext)
- o fpc-descriptor-value (ietf-dmm-fpc-settingsext)
- o fpc-action-value (ietf-dmm-fpc-settingsext)

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) defined in [[RFC8342](#)].

A DPN conformant to NMDA MAY only have policies, installed policies, topology, domains and mobility session information that has been assigned to it in its intended and operational datastores.

ServiceGroups are not expected to appear in operational datastores of DPNs as they remain in and are used by FPC Agents and Clients. They MAY be operationally present in DNS when using the Dynamic Delegation and Discovery System (DDDS) as defined in [[RFC3958](#)] or the operational datastore of systems that provide equivalent functionality.

[3.1.](#) FPC YANG Model

This module defines the information model and protocol elements specified in this document.

This module references [[RFC6991](#)], [[RFC8040](#)] and the fpc-settingsext module defined in this document.

```
<CODE BEGINS> file "ietf-dmm-fpc@2018-05-17.yang"
module ietf-dmm-fpc {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpc";
  prefix fpc;

  import ietf-inet-types { prefix inet;
    revision-date 2013-07-15; }
  import ietf-dmm-fpc-settingsext { prefix fpcbase;
    revision-date 2018-05-17; }
  import ietf-diam-trafficclassifier { prefix rfc5777;
    revision-date 2018-05-17; }
  import ietf-restconf { prefix rc;
    revision-date 2017-01-26; }
  import ietf-yang-patch { prefix ypatch;
    revision-date 2017-02-22; }

  organization "IETF Distributed Mobility Management (DMM)
    Working Group";

  contact
    "WG Web: <http://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    WG Chair: Dapeng Liu
    <mailto:maxpassion@gmail.com>
```

WG Chair: Jouni Korhonen
<mailto:jouni.nospam@gmail.com>

Editor: Satoru Matsushima
<mailto:satoru.matsushima@g.softbank.co.jp>

Editor: Lyle Bertz
<mailto:lylebe551144@gmail.com>;

description

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

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

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

```
revision 2018-05-17 {  
  description "Initial Revision."  
  reference "draft-ietf-dmm-fpc-cpdp-10";  
}
```

```
//General Structures  
grouping templatedef {  
  leaf extensible {  
    type boolean;  
    description "Indicates if the template is extensible";  
  }  
  leaf-list static-attributes {  
    type string;  
    description "Attribute (Name) whose value cannot  
      change";  
  }  
  leaf-list mandatory-attributes {  
    type string;  
    description "Attribute (Name) of optional attributes  
      that MUST be present in instances of this template."  
  }  
  leaf entity-state {
```

```
  type enumeration {  
    enum initial {  
      description "Initial Configuration";  
    }  
    enum partially-configured {  
      description "Partial Configuration";  
    }  
    enum configured {
```

```

        description "Configured";
    }
    enum active {
        description "Active";
    }
}
default initial;
description "Entity State";
}
leaf version {
    type uint32;
    description "Template Version";
}
description "Teampale Definition";
}
typedef fpc-identity {
    type union {
        type uint32;
        type instance-identifier;
        type string;
    }
    description "FPC Identity";
}
grouping index {
    leaf index {
        type uint16;
        description "Index";
    }
    description "Index Value";
}

// Policy Structures
grouping descriptor-template-key {
    leaf descriptor-template-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Descriptor Key";
    }
    description "Descriptor-Template Key";
}

```

```

grouping action-template-key {

```



```

        leaf action-template-key {
            type fpc:fpc-identity;
            mandatory true;
            description "Action Key";
        }
        description "Action-Template Key";
    }
    grouping rule-template-key {
        leaf rule-template-key {
            type fpc:fpc-identity;
            mandatory true;
            description "Rule Identifier";
        }
        description "Rule Key";
    }
    grouping policy-template-key {
        leaf policy-template-key {
            type fpc:fpc-identity;
            mandatory true;
            description "Rule Identifier";
        }
        description "Rule Key";
    }
}

grouping fpc-setting-value {
    anydata setting;
    description "FPC Setting Value";
}

// Configuration / Settings
grouping policy-configuration-choice {
    choice policy-configuration-value {
        case descriptor-value {
            uses fpcbase:fpc-descriptor-value;
            description "Descriptor Value";
        }
        case action-value {
            uses fpcbase:fpc-action-value;
            description "Action Value";
        }
        case setting-value {
            uses fpc:fpc-setting-value;
            description "Setting";
        }
    }
    description "Policy Attributes";
}
description "Policy Configuration Value Choice";
}

```

```
grouping policy-configuration {
  list policy-configuration {
    key index;
    uses fpc:index;
    uses fpc:policy-configuration-choice;
    description "Policy Configuration";
  }
  description "Policy Configuration Value";
}
grouping ref-configuration {
  uses fpc:policy-template-key;
  uses fpc:policy-configuration;
  uses fpc:templatedef;
  description "Policy-Configuration Entry";
}

// FPC Policy
grouping policy-information-model {
  list action-template {
    key action-template-key;
    uses fpc:action-template-key;
    uses fpcbase:fpc-action-value;
    uses fpc:templatedef;
    description "Action Template";
  }
  list descriptor-template {
    key descriptor-template-key;
    uses fpc:descriptor-template-key;
    uses fpcbase:fpc-descriptor-value;
    uses fpc:templatedef;
    description "Descriptor Template";
  }
  list rule-template {
    key rule-template-key;
    uses fpc:rule-template-key;
    leaf descriptor-match-type {
      type enumeration {
        enum or {
          value 0;
          description "OR logic";
        }
        enum and {
          value 1;
          description "AND logic";
        }
      }
    }
  }
}
```

mandatory true;
description "Type of Match (OR or AND) applied

```
        to the descriptor-configurations";
    }
    list descriptor-configuration {
        key "descriptor-template-key";
        uses fpc:descriptor-template-key;
        leaf direction {
            type rfc5777:direction-type;
            description "Direction";
        }
        list attribute-expression {
            key index;
            uses fpc:index;
            uses fpcbase:fpc-descriptor-value;
            description "Descriptor Attributes";
        }
        uses fpc:fpc-setting-value;
        description "A set of Descriptor references";
    }
    list action-configuration {
        key "action-order";
        leaf action-order {
            type uint32;
            mandatory true;
            description "Action Execution Order";
        }
        uses fpc:action-template-key;
        list attribute-expression {
            key index;
            uses fpc:index;
            uses fpcbase:fpc-action-value;
            description "Action Attributes";
        }
        uses fpc:fpc-setting-value;
        description "A set of Action references";
    }
    uses fpc:templatedef;
    list rule-configuration {
        key index;
        uses fpc:index;
```

```

        uses fpc:policy-configuration-choice;
        description "Rule Configuration";
    }
    description "Rule Template";
}
list policy-template {
    key policy-template-key;
    uses fpc:policy-template-key;
    list rule-template {

```

```

        key "precedence";
        unique "rule-template-key";
        leaf precedence {
            type uint32;
            mandatory true;
            description "Rule Precedence";
        }
        uses fpc:rule-template-key;
        description "Rule Entry";
    }
    uses fpc:templatedef;
    uses fpc:policy-configuration;
    description "Policy Template";
}
description "FPC Policy Structures";
}

// Topology Information Model
identity role {
    description "Role";
}
grouping dpn-key {
    leaf dpn-key {
        type fpc:fpc-identity;
        description "DPN Key";
    }
    description "DPN Key";
}
grouping role-key {
    leaf role-key {
        type identityref {
            base "fpc:role";

```

```

        }
        mandatory true;
        description "Access Technology Role";
    }
    description "Access Technology Role key";
}
grouping interface-key {
    leaf interface-key{
        type fpc:fpc-identity;
        mandatory true;
        description "interface identifier";
    }
    description "Interface Identifier key";
}
identity interface-protocols {
    description "Protocol supported by the interface";
}

```

```

    }
    identity features {
        description "Protocol features";
    }
}

// Mobility Context
grouping mobility-context {
    leaf mobility-context-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Mobility Context Key";
    }
    leaf-list delegating-ip-prefix {
        type inet:ip-prefix;
        description "IP Prefix";
    }
    leaf parent-context {
        type fpc:fpc-identity;
        description "Parent Mobility Context";
    }
    leaf-list child-context {
        type fpc:fpc-identity;
        description "Child Mobility Context";
    }
}
container mobile-node {

```

```

leaf-list ip-address {
    type inet:ip-address;
    description "IP Address";
}
leaf imsi {
    type fpcbase:imsi-type;
    description "IMSI";
}
list mn-policy-configuration {
    key policy-template-key;
    uses fpc:ref-configuration;
    description "MN Policy Configuration";
}
description "Mobile Node";
}
container domain {
    leaf domain-key {
        type fpc:fpc-identity;
        description "Domain Key";
    }
    list domain-policy-settings {
        key policy-template-key;
        uses fpc:ref-configuration;
    }
}

```

```

        description "MN Policy Configuration";
    }
    description "Domain";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    list dpn-policy-configuration {
        key policy-template-key;
        uses fpc:ref-configuration;
        description "DPN Policy Configuration";
    }
    leaf role {
        type identityref {
            base "fpc:role";
        }
        description "Role";
    }
}

```

```

    list service-data-flow {
        key identifier;
        leaf identifier {
            type uint32;
            description "Generic Identifier";
        }
        leaf service-group-key {
            type fpc:fpc-identity;
            description "Service Group Key";
        }
        list interface {
            key interface-key;
            uses fpc:interface-key;
            description "interface assigned";
        }
        list service-data-flow-policy-configuration {
            key policy-template-key;
            uses fpc:ref-configuration;
            description "Flow Policy Configuration";
        }
        description "Service Dataflow";
    }
    description "DPN";
}
description "Mobility Context";
}

// Events, Probes & Notifications
identity event-type {
    description "Base Event Type";
}

```

```

}
typedef event-type-id {
    type uint32;
    description "Event ID Type";
}
grouping monitor-key {
    leaf monitor-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Monitor Key";
    }
}

```

```

    description "Monitor Id";
}
grouping monitor-config {
  uses fpc:templatedef;
  uses fpc:monitor-key;
  leaf target {
    type string;
    description "target";
  }
  leaf deferrable {
    type boolean;
    description "Indicates reports related to this
      config can be delayed.";
  }
  choice configuration {
    mandatory true;
    leaf period {
      type uint32;
      description "Period";
    }
    case threshold-config {
      leaf low {
        type uint32;
        description "low threshold";
      }
      leaf hi {
        type uint32;
        description "high threshold";
      }
    }
    description "Threshold Config Case";
  }
  leaf schedule {
    type uint32;
    description "Reporting Time";
  }
  leaf-list event-identities {
    type identityref {

```

```

    base "fpc:event-type";
  }
  description "Event Identities";
}

```



```

    leaf-list event-ids {
        type uint32;
        description "Event IDs";
    }
    description "Event Config Value";
}
description "Monitor Configuration";
}

// Top Level Structures
list tenant {
    key "tenant-key";
    leaf tenant-key {
        type fpc:fpc-identity;
        description "Tenant Key";
    }
}
container topology-information-model {
    config false;
    list service-group {
        key "service-group-key role-key";
        leaf service-group-key {
            type fpc:fpc-identity;
            mandatory true;
            description "Service Group Key";
        }
        leaf service-group-name {
            type string;
            description "Service Group Name";
        }
    }
    uses fpc:role-key;
    leaf role-name {
        type string;
        mandatory true;
        description "Role Name";
    }
}
leaf-list protocol {
    type identityref {
        base "interface-protocols";
    }
    min-elements 1;
    description "Supported protocols";
}
leaf-list feature {
    type identityref {

```

```
        base "interface-protocols";
    }
    description "Supported features";
}
list service-group-configuration {
    key index;
    uses fpc:index;
    uses fpc:policy-configuration-choice;
    description "Settings";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    min-elements 1;
    list referenced-interface {
        key interface-key;
        uses fpc:interface-key;
        leaf-list peer-service-group-key {
            type fpc:fpc-identity;
            description "Peer Service Group";
        }
        description "Referenced Interface";
    }
    description "DPN";
}
description "Service Group";
}
list dpn {
    key dpn-key;
    uses fpc:dpn-key;
    leaf dpn-name {
        type string;
        description "DPN name";
    }
    leaf dpn-resource-mapping-reference {
        type string;
        description "Reference to underlying DPN resource(s)";
    }
    leaf domain-key {
        type fpc:fpc-identity;
        description "Domains";
    }
    leaf-list service-group-key {
        type fpc:fpc-identity;
        description "Service Group";
    }
}
list interface {
```

key "interface-key";

```
    uses fpc:interface-key;
    leaf interface-name {
        type string;
        description "Service Endpoint Interface Name";
    }
    leaf role {
        type identityref {
            base "fpc:role";
        }
        description "Roles supported";
    }
    leaf-list protocol {
        type identityref {
            base "interface-protocols";
        }
        description "Supported protocols";
    }
    list interface-configuration {
        key index;
        uses fpc:index;
        uses fpc:policy-configuration-choice;
        description "Interface settings";
    }
    description "DPN interfaces";
}
list dpn-policy-configuration {
    key policy-template-key;
    uses fpc:ref-configuration;
    description "DPN Policy Configuration";
}
description "Set of DPNs";
}
list domain {
    key domain-key;
    leaf domain-key {
        type fpc:fpc-identity;
        mandatory true;
        description "Domain Key";
    }
    leaf domain-name {
```

```
    type string;
    description "Domain displayname";
}
list domain-policy-configuration {
    key policy-template-key;
    uses fpc:ref-configuration;
    description "Domain Configuration";
}
```

```
    description "List of Domains";
}
container dpn-checkpoint {
    uses fpc:basename-info;
    description "DPN Checkpoint information";
}
container service-group-checkpoint {
    uses fpc:basename-info;
    description "Service Group Checkpoint information";
}
container domain-checkpoint {
    uses fpc:basename-info;
    description "Domain Checkpoint information";
}
description "FPC Topology grouping";
}
container policy-information-model {
    config false;
    uses fpc:policy-information-model;
    uses fpc:basename-info;
    description "Policy";
}
list mobility-context {
    key "mobility-context-key";
    config false;
    uses fpc:mobility-context;
    description "Mobility Context";
}
list monitor {
    key monitor-key;
    config false;
    uses fpc:monitor-config;
    description "Monitor";
}
```

```

}
description "Tenant";
}

typedef agent-identifier {
    type fpc:fpc-identity;
    description "Agent Identifier";
}
typedef client-identifier {
    type fpc:fpc-identity;
    description "Client Identifier";
}
grouping basename-info {
    leaf basename {
        type fpc:fpc-identity;

```

```

        description "Rules Basename";
    }
    leaf base-checkpoint {
        type string;
        description "Checkpoint";
    }
    description "Basename Information";
}

// RPCs
grouping client-id {
    leaf client-id {
        type fpc:client-identifier;
        mandatory true;
        description "Client Id";
    }
    description "Client Identifier";
}
grouping execution-delay {
    leaf execution-delay {
        type uint32;
        description "Execution Delay (ms)";
    }
    description "Execution Delay";
}
typedef ref-scope {

```

```

type enumeration {
  enum none {
    value 0;
    description "no references";
  }
  enum op {
    value 1;
    description "All references are intra-operation";
  }
  enum bundle {
    value 2;
    description "All references in exist in bundle";
  }
  enum storage {
    value 3;
    description "One or more references exist in storage.";
  }
  enum unknown {
    value 4;
    description "The location of the references are unknown.";
  }
}

```

```

    description "Search scope for references in the operation.";
  }
  rpc configure {
    description "Configure RPC";
    input {
      uses client-id;
      uses execution-delay;
      uses ypatch:yang-patch;
    }
    output {
      uses ypatch:yang-patch-status;
    }
  }
  augment "/configure/input/yang-patch/edit" {
    leaf reference-scope {
      type fpc:ref-scope;
      description "Reference Scope";
    }
  }
  uses fpcbase:instructions;

```

```

        description "yang-patch edit augments for configure rpc";
    }
    grouping subsequent-edits {
        list subsequent-edit {
            key edit-id;
            ordered-by user;

            description "Edit list";

            leaf edit-id {
                type string;
                description "Arbitrary string index for the edit.";
            }

            leaf operation {
                type enumeration {
                    enum create {
                        description "Create";
                    }
                    enum delete {
                        description "Delete";
                    }
                    enum insert {
                        description "Insert";
                    }
                    enum merge {
                        description "Merge";
                    }
                    enum move {

```

```

        description "Move";
    }
    enum replace {
        description "Replace";
    }
    enum remove {
        description
            "Delete the target node if it currently exists.";
    }
}
mandatory true;
description

```

```

        "The datastore operation requested";
    }

    leaf target {
        type ypatch:target-resource-offset;
        mandatory true;
        description
            "Identifies the target data node";
    }

    leaf point {
        when "(../operation = 'insert' or ../operation = 'move')"
            + "and (../where = 'before' or ../where = 'after')" {
            description
                "This leaf only applies for 'insert' or 'move'
                operations, before or after an existing entry.";
        }
        type ypatch:target-resource-offset;
        description
            "The absolute URL path for the data node";
    }

    leaf where {
        when "../operation = 'insert' or ../operation = 'move'" {
            description
                "This leaf only applies for 'insert' or 'move'
                operations.";
        }
        type enumeration {
            enum before {
                description
                    "Insert or move a data node before.";
            }
            enum after {
                description
                    "Insert or move a data node after.";
            }
        }
    }

```

```

    }
    enum first {
        description
            "Insert or move a data node so it becomes ordered
            as the first entry.";
    }

```



```

    }
    enum last {
        description
            "Insert or move a data node so it becomes ordered
            as the last entry.";
    }
}
default last;
description
    "Identifies where a data resource will be inserted
    or moved.";
}

anydata value {
    when "../operation = 'create' "
        + "or ../operation = 'merge' "
        + "or ../operation = 'replace' "
        + "or ../operation = 'insert'" {
        description
            "The anydata 'value' is only used for 'create',
            'merge', 'replace', and 'insert' operations.";
    }
    description
        "Value used for this edit operation.";
}
}
description "Subsequent Edits";
}
augment "/configure/output/yang-patch-status/edit-status/edit/"
+ "edit-status-choice/ok" {
    leaf notify-follows {
        type boolean;
        description "Notify Follows Indication";
    }
    uses fpc:subsequent-edits;
    description "Configure output augments";
}

grouping op-header {
    uses client-id;
    uses execution-delay;
    leaf operation-id {
        type uint64;
    }
}

```

```

        mandatory true;
        description "Operation Identifier";
    }
    description "Common Operation header";
}
grouping monitor-response {
    leaf operation-id {
        type uint64;
        mandatory true;
        description "Operation Identifier";
    }
    choice edit-status-choice {
        description
            "A choice between different types of status
            responses for each 'edit' entry.";
        leaf ok {
            type empty;
            description
                "This 'edit' entry was invoked without any
                errors detected by the server associated
                with this edit.";
        }
        case errors {
            uses rc:errors;
            description
                "The server detected errors associated with the
                edit identified by the same 'edit-id' value.";
        }
    }
    description "Monitor Response";
}

// Common RPCs
rpc register_monitor {
    description "Used to register monitoring of parameters/events";
    input {
        uses fpc:op-header;
        list monitor {
            key monitor-key;
            uses fpc:monitor-config;
            description "Monitor Configuration";
        }
    }
    output {
        uses fpc:monitor-response;
    }
}
rpc deregister_monitor {

```

```
description "Used to de-register monitoring of
  parameters/events";
input {
  uses fpc:op-header;
  list monitor {
    key monitor-key;
    uses fpc:monitor-key;
    min-elements 1;
    leaf send_data {
      type boolean;
      description "Indicates if NOTIFY with final data
        is desired upon deregistration";
    }
    description "Monitor Identifier";
  }
}
output {
  uses fpc:monitor-response;
}
}
rpc probe {
  description "Probe the status of a registered monitor";
  input {
    uses fpc:op-header;
    list monitor {
      key monitor-key;
      uses fpc:monitor-key;
      min-elements 1;
      description "Monitor";
    }
  }
  output {
    uses fpc:monitor-response;
  }
}

// Notification Messages & Structures
notification config-result-notification {
  uses ypatch:yang-patch-status;
  description "Configuration Result Notification";
}
augment "/config-result-notification" {
```

```
        uses fpc:subsequent-edits;
        description "config-result-notificatio augment";
    }
```

```
identity notification-cause {
    description "Notification Cause";
}
```

```
    }
    identity subscribed-event-occurred {
        base "notification-cause";
        description "Subscribed Event Occurence";
    }
    identity low-threshold-crossed {
        base "notification-cause";
        description "Subscribed Event Occurence";
    }
    identity high-threshold-crossed {
        base "notification-cause";
        description "Subscribed Event Occurence";
    }
    identity periodic-report {
        base "notification-cause";
        description "Periodic Report";
    }
    identity scheduled-report {
        base "notification-cause";
        description "Scheduled Report";
    }
    identity probe {
        base "notification-cause";
        description "Probe";
    }
    identity deregistration-final-value {
        base "notification-cause";
        description "Probe";
    }
    identity monitoring-suspension {
        base "notification-cause";
        description "Indicates monitoring suspension";
    }
    identity monitoring-resumption {
        base "notification-cause";
    }
}
```

```

    description "Indicates that monitoring has resumed";
  }
  identity dpn-available {
    base "notification-cause";
    description "DPN Candidate Available";
  }
  identity dpn-unavailable {
    base "notification-cause";
    description "DPN Unavailable";
  }
  notification notify {
    leaf notification-id {
      type uint32;
    }
  }

```

```

    description "Notification Identifier";
  }
  leaf timestamp {
    type uint32;
    description "timestamp";
  }
  list report {
    key monitor-key;
    uses fpc:monitor-key;
    min-elements 1;
    leaf trigger {
      type identityref {
        base "notification-cause";
      }
      description "Notification Cause";
    }
  }
  choice value {
    case dpn-candidate-available {
      leaf node-id {
        type inet:uri;
        description "Topology URI";
      }
      list supported-interface-list {
        key role-key;
        uses fpc:role-key;
        description "Support Intefaces";
      }
    }
  }
  description "DPN Candidate Information";

```

```

    }
    case dpn-unavailable {
        leaf dpn-id {
            type fpc:fpc-identity;
            description "DPN Identifier for DPN Unavailable";
        }
        description "DPN Unavailable";
    }
    anydata report-value {
        description "Any non integer report";
    }
    description "Report Value";
}
description "Report";
}
description "Notify Message";
}
}
<CODE ENDS>

```

[3.2.](#) FPC YANG Settings and Extensions Model

This module defines the base data elements in FPC that are likely to be extended.

This module references [\[RFC6991\]](#), `ietf-trafficselector-types` and `ietf-pmip-qos` modules.

```

<CODE BEGINS> file "ietf-dmm-fpc-settingsext@2018-05-17.yang"
module ietf-dmm-fpc-settingsext {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-dmm-fpc-settingsext";
    prefix fpcbase;

    import ietf-inet-types { prefix inet;
        revision-date 2013-07-15; }
    import ietf-trafficselector-types { prefix traffic-selectors;
        revision-date 2018-05-17; }
    import ietf-yang-types { prefix ytypes;
        revision-date 2013-07-15; }
    import ietf-pmip-qos { prefix pmipqos;

```

```
    revision-date 2018-05-17; }
import ietf-diam-trafficclassifier { prefix rfc5777;
    revision-date 2018-05-17; }

organization "IETF Distributed Mobility Management (DMM)
    Working Group";

contact
    "WG Web:    <http://tools.ietf.org/wg/netmod/>
    WG List:   <mailto:netmod@ietf.org>

    WG Chair:  Dapeng Liu
               <mailto:maxpassion@gmail.com>

    WG Chair:  Sri Gundavelli
               <mailto:sgundave@cisco.com>

    Editor:    Satoru Matsushima
               <mailto:satoru.matsushima@g.softbank.co.jp>

    Editor:    Lyle Bertz
               <mailto:lylebe551144@gmail.com>";

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

Matsushima, et al. Expires October 13, 2019 [Page 27]

Internet-Draft YANG for DMM FPC Protocol April 2019

It contains Settings definitions as well as Descriptor and Action extensions.

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

This document is subject to [BCP 78](http://trustee.ietf.org/bcp78) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](http://trustee.ietf.org/trust-legal-provisions) and are provided

without warranty as described in the Simplified BSD License.";

```
revision 2018-05-17 {  
  description "Initial Revision."  
  reference "draft-ietf-dmm-fpc-cpdp-10";  
}
```

```
//Tunnel Information  
identity tunnel-type {  
  description "Tunnel Type";  
}  
identity grev1 {  
  base "fpcbase:tunnel-type";  
  description "GRE v1";  
}  
identity grev2 {  
  base "fpcbase:tunnel-type";  
  description "GRE v2";  
}  
identity ipinip {  
  base "fpcbase:tunnel-type";  
  description "IP in IP";  
}  
identity gtpv1 {  
  base "fpcbase:tunnel-type";  
  description "GTP version 1 Tunnel";  
}  
identity gtpv2 {  
  base "fpcbase:tunnel-type";  
  description "GTP version 2 Tunnel";  
}  
  
grouping tunnel-value {  
  container tunnel-info {
```

```
  leaf tunnel-local-address {  
    type inet:ip-address;  
    description "local tunnel address";  
  }  
  leaf tunnel-remote-address {  
    type inet:ip-address;  
    description "remote tunnel address";
```



```

}
leaf mtu-size {
    type uint32;
    description "MTU size";
}
leaf tunnel {
    type identityref {
        base "fpcbase:tunnel-type";
    }
}
description "tunnel type";
}
leaf payload-type {
    type enumeration {
        enum ipv4 {
            value 0;
            description "IPv4";
        }
        enum ipv6 {
            value 1;
            description "IPv6";
        }
        enum dual {
            value 2;
            description "IPv4 and IPv6";
        }
    }
    description "Payload Type";
}
leaf gre-key {
    type uint32;
    description "GRE_KEY";
}
container gtp-tunnel-info {
    leaf local-tunnel-identifier {
        type uint32;
        description "Tunnel Endpoint Identifier (TEID)";
    }
    leaf remote-tunnel-identifier {
        type uint32;
        description "Tunnel Endpoint Identifier (TEID)";
    }
}

```

```

        leaf sequence-numbers-enabled {
            type boolean;
            description "Sequence No. Enabled";
        }
        description "GTP Tunnel Information";
    }
    leaf ebi {
        type fpcbase:ebi-type;
        description "EPS Bearier Identifier";
    }
    leaf lbi {
        type fpcbase:ebi-type;
        description "Linked Bearier Identifier";
    }
    description "Tunnel Information";
}
description "Tunnel Value";
}

```

```

////////////////////////////////////
// DESCRIPTOR DEFINITIONS

```

```

// From 3GPP TS 24.008 version 13.5.0 Release 13
typedef packet-filter-direction {
    type enumeration {
        enum preRel7Tft {
            value 0;
            description "Pre-Release 7 TFT";
        }
        enum uplink {
            value 1;
            description "uplink";
        }
        enum downlink {
            value 2;
            description "downlink";
        }
        enum bidirectional {
            value 3;
            description "bi-direcitonal";
        }
    }
    description "Packet Filter Direction";
}
typedef component-type-id {
    type uint8 {
        range "16 | 17 | 32 | 33 | 35 | 48 | 64 | 65 |"
        + " 80 | 81 | 96 | 112 | 128";
    }
}

```

```
    }
    description "Specifies the Component Type";
}
grouping packet-filter {
  leaf direction {
    type fpcbase:packet-filter-direction;
    description "Filter Direction";
  }
  leaf identifier {
    type uint8 {
      range "1..15";
    }
    description "Filter Identifier";
  }
  leaf evaluation-precedence {
    type uint8;
    description "Evaluation Precedence";
  }
  list contents {
    key component-type-identifier;
    description "Filter Contents";
    leaf component-type-identifier {
      type fpcbase:component-type-id;
      description "Component Type";
    }
  }
  choice value {
    leaf ipv4-local {
      type inet:ipv4-address;
      description "IPv4 Local Address";
    }
    leaf ipv6-prefix-local {
      type inet:ipv6-prefix;
      description "IPv6 Local Prefix";
    }
    leaf ipv4-ipv6-remote {
      type inet:ip-address;
      description "Ipv4 Ipv6 remote address";
    }
    leaf ipv6-prefix-remote {
      type inet:ipv6-prefix;
      description "IPv6 Remote Prefix";
    }
  }
  leaf next-header {
```

```
        type uint8;
        description "Next Header";
    }
    leaf local-port {
        type inet:port-number;
    }
```

```
        description "Local Port";
    }
    case local-port-range {
        leaf local-port-lo {
            type inet:port-number;
            description "Local Port Min Value";
        }
        leaf local-port-hi {
            type inet:port-number;
            description "Local Port Max Value";
        }
    }
    leaf remote-port {
        type inet:port-number;
        description "Remote Port";
    }
    case remote-port-range {
        leaf remote-port-lo {
            type inet:port-number;
            description "Remote Por Min Value";
        }
        leaf remote-port-hi {
            type inet:port-number;
            description "Remote Port Max Value";
        }
    }
    leaf ipsec-index {
        type traffic-selectors:ipsec-spi;
        description "IPSec Index";
    }
    leaf traffic-class {
        type inet:dscp;
        description "Traffic Class";
    }
    case traffic-class-range {
        leaf traffic-class-lo {
```

```

        type inet:dscp;
        description "Traffic Class Min Value";
    }
    leaf traffic-class-hi {
        type inet:dscp;
        description "Traffic Class Max Value";
    }
}
leaf-list flow-label {
    type inet:ipv6-flow-label;
    description "Flow Label";
}

```

```

        description "Component Value";
    }
}
description "Packet Filter";
}

grouping prefix-descriptor {
    leaf destination-ip {
        type inet:ip-prefix;
        description "Rule of destination IP";
    }
    leaf source-ip {
        type inet:ip-prefix;
        description "Rule of source IP";
    }
    description "Traffic descriptor based upon source/
        destination as IP prefixes";
}

grouping fpc-descriptor-value {
    choice descriptor-value {
        mandatory true;
        leaf all-traffic {
            type empty;
            description "admit any";
        }
        leaf no-traffic {
            type empty;
            description "deny any";
        }
    }
}

```

```

}
case prefix-descriptor {
    uses fpcbase:prefix-descriptor;
    description "IP Prefix descriptor";
}
case pmip-selector {
    uses traffic-selectors:traffic-selector;
    description "PMIP Selector";
}
container rfc5777-classifier-template {
    uses rfc5777:classifier;
    description "RFC 5777 Classifier";
}
container packet-filter {
    uses fpcbase:packet-filter;
    description "Packet Filter";
}
case tunnel-info {
    uses fpcbase:tunnel-value;

```

```

        description "Tunnel Descriptor (only
            considers source info)";
    }
    description "Descriptor Value";
}
description "FPC Descriptor Values";
}

// Next Hop Structures
typedef fpc-service-path-id {
    type uint32 {
        range "0..33554431";
    }
    description "SERVICE_PATH_ID";
}
typedef fpc-mpls-label {
    type uint32 {
        range "0..1048575";
    }
    description "MPLS label";
}
typedef segment-id {

```

```

    type string {
        length "16";
    }
    description "SR Segment Identifier";
}
grouping fpc-next-hop {
    choice next-hop-value {
        leaf ip-address {
            type inet:ip-address;
            description "IP Value";
        }
        leaf mac-address {
            type ytypes:mac-address;
            description "MAC Address Value";
        }
        leaf service-path {
            type fpcbase:fpc-service-path-id;
            description "Service Path Value";
        }
        leaf mpls-path {
            type fpcbase:fpc-mpls-label;
            description "MPLS Value";
        }
        leaf nsh {
            type string {
                length "16";
            }
        }
    }
}

```

```

    }
    description "Network Service Header";
}
leaf interface {
    type uint16;
    description "If (interface) Value";
}
leaf segment-identifier {
    type fpcbase:segment-id;
    description "Segment Id";
}
leaf-list mpls-label-stack {
    type fpcbase:fpc-mpls-label;
    description "MPLS Stack";
}
}

```

```

leaf-list mpls-sr-stack {
    type fpcbase:fpc-mpls-label;
    description "MPLS SR Stack";
}
leaf-list srv6-stack {
    type fpcbase:segment-id;
    description "Segment Id";
}
case tunnel-info {
    uses fpcbase:tunnel-value;
    description "Tunnel Descriptor (only
    considers source info)";
}
description "Value";
}
description "Nexthop Value";
}

```

```

////////////////////////////////////
// PMIP Integration //
typedef pmip-commandset {
    type bits {
        bit assign-ip {
            position 0;
            description "Assign IP";
        }
        bit assign-dpn {
            position 1;
            description "Assign DPN";
        }
        bit session {
            position 2;
            description "Session Level";
        }
    }
}

```

```

}
bit uplink {
    position 3;
    description "Uplink";
}
bit downlink {
    position 4;
    description "Downlink";
}

```



```

    }
  }
  description "PMIP Instructions";
}
}
////////////////////////////////////
// 3GPP Integration          //

// Type Defs
typedef fpc-qos-class-identifier {
  type uint8 {
    range "1..9";
  }
  description "QoS Class Identifier (QCI)";
}
typedef ebi-type {
  type uint8 {
    range "0..15";
  }
  description "EUTRAN Bearere Identifier (EBI) Type";
}
typedef imsi-type {
  type uint64;
  description
    "International Mobile Subscriber Identity (IMSI)
    Value Type";
}
// Instructions
typedef threegpp-instr {
  type bits {
    bit assign-ip {
      position 0;
      description "Assign IP Address/Prefix";
    }
    bit assign-fteid-ip {
      position 1;
      description "Assign FTEID-IP";
    }
    bit assign-fteid-teid {
      position 2;
      description "Assign FTEID-TEID";
    }
  }
}

```

}

```

bit session {
    position 3;
    description "Commands apply to the Session Level";
}
bit uplink {
    position 4;
    description "Commands apply to the Uplink";
}
bit downlink {
    position 5;
    description "Commands apply to the Downlink";
}
bit assign-dpn {
    position 6;
    description "Assign DPN";
}
}
description "Instruction Set for 3GPP R11";
}

```

```

////////////////////////////////////
// ACTION VALUE AUGMENTS
grouping fpc-action-value {
    choice action-value {
        mandatory true;
        leaf drop {
            type empty;
            description "Drop Traffic";
        }
        container rewrite {
            choice rewrite-value {
                case prefix-descriptor {
                    uses fpcbase:prefix-descriptor;
                    description "IP Prefix descriptor";
                }
                case pmip-selector {
                    uses traffic-selectors:traffic-selector;
                    description "PMIP Selector";
                }
                container rfc5777-classifier-template {
                    uses rfc5777:classifier;
                    description "RFC 5777 Classifier";
                }
            }
            description "Rewrite Choice";
        }
    }
    description "Rewrite/NAT value";
}
}

```

```
    container copy-forward-nextthop {
        uses fpcbase:fpc-nextthop;
        description "Copy Forward Value";
    }
    container nextthop {
        uses fpcbase:fpc-nextthop;
        description "NextHop Value";
    }
    case qos {
        leaf trafficclass {
            type inet:dscp;
            description "Traffic Class";
        }
        uses pmipqos:qosattribute;
        leaf qci {
            type fpcbase:fpc-qos-class-identifier;
            description "QCI";
        }
        leaf ue-aggr-max-bitrate {
            type uint32;
            description "UE Aggregate Max Bitrate";
        }
        leaf apn-ambr {
            type uint32;
            description
                "Access Point Name Aggregate Max Bit Rate";
        }
        description "QoS Attributes";
    }
    description "Action Value";
}
description "FPC Action Value";
}

// Instructions
grouping instructions {
    container command-set {
        choice instr-type {
            leaf instr-3gpp-mob {
                type fpcbase:threegpp-instr;
                description "3GPP GTP Mobility Instructions";
            }
            leaf instr-pmip {
                type pmip-commandset;
                description "PMIP Instructions";
            }
        }
    }
}
```

```
    description "Instruction Value Choice";
}
```

```
    description "Instructions";
  }
  description "Instructions Value";
}
}
<CODE ENDS>
```

[3.3.](#) PMIP QoS Model

This module defines the base protocol elements specified in this document.

This module references [\[RFC6991\]](#).

```
<CODE BEGINS> file "ietf-pmip-qos@2018-05-17.yang"
module ietf-pmip-qos {
  yang-version 1.1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-pmip-qos";

  prefix "qos-pmip";

  import ietf-inet-types {
    prefix inet;
    revision-date 2013-07-15;
  }
  import ietf-trafficselector-types { prefix traffic-selectors;
    revision-date 2018-05-17; }

  organization "IETF Distributed Mobility Management (DMM)
    Working Group";

  contact
    "WG Web: <http://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    WG Chair: Dapeng Liu
    <mailto:maxpassion@gmail.com>
```

WG Chair: Sri Gundavelli
<mailto:sgundave@cisco.com>

Editor: Satoru Matsushima
<mailto:satoru.matsushima@g.softbank.co.jp>

Editor: Lyle Bertz
<mailto:lylebe551144@gmail.com>;

Matsushima, et al.

Expires October 13, 2019

[Page 39]

Internet-Draft

YANG for DMM FPC Protocol

April 2019

description

"This module contains a collection of YANG definitions for quality of service parameters used in Proxy Mobile IPv6.

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

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

```
revision 2018-05-17 {  
  description "Initial Revision."  
  reference "RFC 6088: Traffic Selectors for Flow Bindings";  
}
```

```
// Type Definitions
```

```
// QoS Option Field Type Definitions  
typedef sr-id {  
  type uint8;  
  description  
    "An 8-bit unsigned integer used for identifying the QoS  
    Service Request."  
}
```

```

typedef traffic-class {
    type inet:dscp;
    description
        "Traffic Class consists of a 6-bit DSCP field followed by a
        2-bit reserved field.";
    reference
        "RFC 3289: Management Information Base for the
        Differentiated Services Architecture
        RFC 2474: Definition of the Differentiated Services Field
        (DS Field) in the IPv4 and IPv6 Headers
        RFC 2780: IANA Allocation Guidelines For Values In
        the Internet Protocol and Related Headers";
}

typedef operational-code {
    type enumeration {

```

```

enum RESPONSE {
    value 0;
    description "Response to a QoS request";
}
enum ALLOCATE {
    value 1;
    description "Request to allocate QoS resources";
}
enum DE-ALLOCATE {
    value 2;
    description "Request to de-allocate QoS resources";
}
enum MODIFY {
    value 3;
    description "Request to modify QoS parameters for a
    previously negotiated QoS Service Request";
}
enum QUERY {
    value 4;
    description "Query to list the previously negotiated QoS
    Service Requests that are still active";
}
enum NEGOTIATE {
    value 5;
    description "Response to a QoS Service Request with a

```

```

        counter QoS proposal";
    }
}
description
    "The type of QoS request. Reserved values: (6) to (255)
    Currently not used. Receiver MUST ignore the option
    received with any value in this range.";
}

//Value definitions
typedef Per-MN-Agg-Max-DL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for all the mobile node's IP flows.
        The measurement units are bits per second.";
}

typedef Per-MN-Agg-Max-UL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum uplink bit rate that is
        requested/allocated for the mobile node's IP flows. The

```

```

        measurement units are bits per second.";
    }

// Generic Structure for the uplink and downlink
grouping Per-Session-Agg-Max-Bit-Rate-Value {
    leaf max-rate {
        type uint32;
        mandatory true;
        description
            "The aggregate maximum bit rate that is requested/allocated
            for all the IP flows associated with that mobility session.
            The measurement units are bits per second.";
    }
    leaf service-flag {
        type boolean;
        mandatory true;
        description
            "This flag is used for extending the scope of the

```

```

        target flows for Per-Session-Agg-Max-UL/DL-Bit-Rate
        from(UL)/to(DL) the mobile node's other mobility sessions
        sharing the same Service Identifier.";
reference
    "RFC 5149 - Service Selection mobility option";
}
leaf exclude-flag {
    type boolean;
    mandatory true;
    description
        "This flag is used to request that the uplink/downlink
        flows for which the network is providing
        Guaranteed-Bit-Rate service be excluded from the
        target IP flows for which
        Per-Session-Agg-Max-UL/DL-Bit-Rate is measured.";
}
description "Per-Session-Agg-Max-Bit-Rate Value";
}

grouping Allocation-Retention-Priority-Value {
    leaf priority-level {
        type uint8 {
            range "0..15";
        }
        mandatory true;
        description
            "This is a 4-bit unsigned integer value. It is used to decide
            whether a mobility session establishment or modification
            request can be accepted; this is typically used for
            admission control of Guaranteed Bit Rate traffic in case of

```

```

        resource limitations.";
}
leaf preemption-capability {
    type enumeration {
        enum enabled {
            value 0;
            description "enabled";
        }
        enum disabled {
            value 1;
            description "disabled";

```



```

    }
    enum reserved1 {
        value 2;
        description "reserved1";
    }
    enum reserved2 {
        value 3;
        description "reserved2";
    }
}
mandatory true;
description
"This is a 2-bit unsigned integer value. It defines whether a
service data flow can get resources tha were already
assigned to another service data flow with a lower priority
level.";
}
leaf preemption-vulnerability {
    type enumeration {
        enum enabled {
            value 0;
            description "enabled";
        }
        enum disabled {
            value 1;
            description "disabled";
        }
        enum reserved1 {
            value 2;
            description "reserved1";
        }
        enum reserved2 {
            value 3;
            description "reserved2";
        }
    }
}
mandatory true;

```

description

"This is a 2-bit unsigned integer value. It defines whether a service data flow can lose the resources assigned to it in order to admit a service data flow with a higher priority

```

        level.";
    }
    description "Allocation-Retention-Priority Value";
}

typedef Aggregate-Max-DL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for downlink IP flows. The measurement
        units are bits per second.";
}

typedef Aggregate-Max-UL-Bit-Rate-Value {
    type uint32;
    description
        "The aggregate maximum downlink bit rate that is
        requested/allocated for downlink IP flows. The measurement
        units are bits per second.";
}

typedef Guaranteed-DL-Bit-Rate-Value {
    type uint32;
    description
        "The guaranteed bandwidth in bits per second for downlink
        IP flows. The measurement units are bits per second.";
}

typedef Guaranteed-UL-Bit-Rate-Value {
    type uint32;
    description
        "The guaranteed bandwidth in bits per second for uplink
        IP flows. The measurement units are bits per second.";
}

grouping QoS-Vendor-Specific-Attribute-Value-Base {
    leaf vendorid {
        type uint32;
        mandatory true;
        description
            "The Vendor ID is the SMI (Structure of Management
            Information) Network Management Private Enterprise Code of
            the IANA-maintained 'Private Enterprise Numbers'
            registry.";
    }
}

```

```
reference
  "'PRIVATE ENTERPRISE NUMBERS', SMI Network Management
  Private Enterprise Codes, April 2014,
  <http://www.iana.org/assignments/enterprise-numbers>";
}
leaf subtype {
  type uint8;
  mandatory true;
  description
    "An 8-bit field indicating the type of vendor-specific
    information carried in the option. The namespace for this
    sub-type is managed by the vendor identified by the
    Vendor ID field.";
}
description
  "QoS Vendor-Specific Attribute.";
}

//Primary Structures (groupings)
grouping qosattribute {
  leaf per-mn-agg-max-dl {
    type qos-pmip:Per-MN-Agg-Max-DL-Bit-Rate-Value;
    description "Per-MN-Agg-Max-DL-Bit-Rate Value";
  }
  leaf per-mn-agg-max-ul {
    type qos-pmip:Per-MN-Agg-Max-UL-Bit-Rate-Value;
    description "Per-MN-Agg-Max-UL-Bit-Rate Value";
  }
  container per-session-agg-max-dl {
    uses qos-pmip:Per-Session-Agg-Max-Bit-Rate-Value;
    description "Per-Session-Agg-Max-Bit-Rate Value";
  }
  container per-session-agg-max-ul {
    uses qos-pmip:Per-Session-Agg-Max-Bit-Rate-Value;
    description "Per-Session-Agg-Max-Bit-Rate Value";
  }
  uses qos-pmip:Allocation-Retention-Priority-Value;
  leaf agg-max-dl {
    type qos-pmip:Aggregate-Max-DL-Bit-Rate-Value;
    description "Aggregate-Max-DL-Bit-Rate Value";
  }
  leaf agg-max-ul {
    type qos-pmip:Aggregate-Max-UL-Bit-Rate-Value;
    description "Aggregate-Max-UL-Bit-Rate Value";
  }
  leaf gbr-dl {
    type qos-pmip:Guaranteed-DL-Bit-Rate-Value;
```

```
description "Guaranteed-DL-Bit-Rate Value";
```

```
    }
    leaf gbr-ul {
        type qos-pmip:Guaranteed-UL-Bit-Rate-Value;
        description "Guaranteed-UL-Bit-Rate Value";
    }
    description "PMIP QoS Attributes. Note Vendor option
is not a part of this grouping";
}

grouping qosoption {
    leaf srid {
        type sr-id;
        mandatory true;
        description "Service Request Identifier";
    }
    leaf trafficclass {
        type traffic-class;
        mandatory true;
        description "Traffic Class";
    }
    leaf operationcode {
        type operational-code;
        mandatory true;
        description "Operation Code";
    }
    uses qos-pmip:qosattribute;
    uses qos-pmip:QoS-Vendor-Specific-Attribute-Value-Base;
    container traffic-selector {
        uses traffic-selectors:traffic-selector;
        description "traffic selector";
    }
    description "PMIP QoS Option";
}
}
<CODE ENDS>
```

[3.4.](#) Traffic Selectors YANG Model

This module defines traffic selector types commonly used in Proxy Mobile IP (PMIP).

This module references [[RFC6991](#)].

```
<CODE BEGINS> file "ietf-trafficselector-types@2018-05-17.yang"
module ietf-trafficselector-types {
  yang-version 1.1;

  namespace
```

Matsushima, et al.

Expires October 13, 2019

[Page 46]

Internet-Draft

YANG for DMM FPC Protocol

April 2019

```
"urn:ietf:params:xml:ns:yang:ietf-trafficselector-types";
```

```
prefix "traffic-selectors";
```

```
import ietf-inet-types {
  prefix inet;
  revision-date 2013-07-15;
}
```

```
organization "IETF Distributed Mobility Management (DMM)
Working Group";
```

```
contact
```

```
"WG Web: <http://tools.ietf.org/wg/netmod/>
```

```
WG List: <mailto:netmod@ietf.org>
```

```
WG Chair: Dapeng Liu
```

```
<mailto:maxpassion@gmail.com>
```

```
WG Chair: Sri Gundavelli
```

```
<mailto:sgundave@cisco.com>
```

```
Editor: Satoru Matsushima
```

```
<mailto:satoru.matsushima@g.softbank.co.jp>
```

```
Editor: Lyle Bertz
```

```
<mailto:lylebe551144@gmail.com>";
```

```
description
```

```
"This module contains a collection of YANG definitions for
traffic selectors for flow bindings.
```

```
Copyright (c) 2016 IETF Trust and the persons identified as the
```

document authors. All rights reserved.

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

```
revision 2018-05-17 {  
  description  
    "Initial Revision.";
```

Matsushima, et al. Expires October 13, 2019 [Page 47]

Internet-Draft YANG for DMM FPC Protocol April 2019

```
  reference  
    "RFC 6088: Traffic Selectors for Flow Bindings";  
}  
  
// Identities  
identity traffic-selector-format {  
  description  
    "The base type for Traffic-Selector Formats";  
}  
  
identity ipv4-binary-selector-format {  
  base traffic-selector-format;  
  description  
    "IPv4 Binary Traffic Selector Format";  
}  
  
identity ipv6-binary-selector-format {  
  base traffic-selector-format;  
  description  
    "IPv6 Binary Traffic Selector Format";  
}  
  
// Type definitions and groupings  
typedef ipsec-spi {  
  type uint32;  
  description
```

```

    "The first 32-bit IPsec Security Parameter Index (SPI)
    value on data. This field is defined in [RFC4303].";
    reference
    "RFC 4303: IP Encapsulating Security
    Payload (ESP)";
}

grouping traffic-selector-base {
    description "A grouping of the common leaves between the
    v4 and v6 Traffic Selectors";
    container ipsec-spi-range {
        presence "Enables setting ipsec spi range";
        description
        "Inclusive range representing IPsec Security Parameter
        Indices to be used. When only start-spi is present, it
        represents a single spi.";
    leaf start-spi {
        type ipsec-spi;
        mandatory true;
        description
        "The first 32-bit IPsec SPI value on data.";
    }
}

```

```

leaf end-spi {
    type ipsec-spi;
    must ". >= ../start-spi" {
        error-message
        "The end-spi must be greater than or equal
        to start-spi";
    }
    description
    "If more than one contiguous SPI value needs to be matched,
    then this field indicates the end value of a range.";
}

container source-port-range {
    presence "Enables setting source port range";
    description
    "Inclusive range representing source ports to be used.
    When only start-port is present, it represents a single
    port. These value(s) are from the range of port numbers
    defined by IANA (http://www.iana.org).";
}

```

```

leaf start-port {
    type inet:port-number;
    mandatory true;
    description
        "The first 16-bit source port number to be matched";
}
leaf end-port {
    type inet:port-number;
    must ". >= ../start-port" {
        error-message
            "The end-port must be greater than or equal to start-port";
    }
    description
        "The last 16-bit source port number to be matched";
}
}
container destination-port-range {
    presence "Enables setting destination port range";
    description
        "Inclusive range representing destination ports to be used.
        When only start-port is present, it represents a single
        port.";
    leaf start-port {
        type inet:port-number;
        mandatory true;
        description
            "The first 16-bit destination port number to be matched";
    }
    leaf end-port {

```

```

        type inet:port-number;
        must ". >= ../start-port" {
            error-message
                "The end-port must be greater than or equal to
                start-port";
        }
        description
            "The last 16-bit destination port number to be matched";
    }
}
}
}

```



```

grouping ipv4-binary-traffic-selector {
  container source-address-range-v4 {
    presence "Enables setting source IPv4 address range";
    description
      "Inclusive range representing IPv4 addresses to be used. When
      only start-address is present, it represents a single
      address.";
    leaf start-address {
      type inet:ipv4-address;
      mandatory true;
      description
        "The first source address to be matched";
    }
    leaf end-address {
      type inet:ipv4-address;
      description
        "The last source address to be matched";
    }
  }
  container destination-address-range-v4 {
    presence "Enables setting destination IPv4 address range";
    description
      "Inclusive range representing IPv4 addresses to be used.
      When only start-address is present, it represents a
      single address.";
    leaf start-address {
      type inet:ipv4-address;
      mandatory true;
      description
        "The first destination address to be matched";
    }
    leaf end-address {
      type inet:ipv4-address;
      description
        "The last destination address to be matched";
    }
  }
}

```

```

}
container ds-range {
  presence "Enables setting dscp range";
  description
    "Inclusive range representing DiffServ Codepoints to be used.

```

```

    When only start-ds is present, it represents a single
    Codepoint.";
leaf start-ds {
    type inet:dscp;
    mandatory true;
    description
        "The first differential service value to be matched";
}
leaf end-ds {
    type inet:dscp;
    must ". >= ../start-ds" {
        error-message
            "The end-ds must be greater than or equal to start-ds";
    }
    description
        "The last differential service value to be matched";
}
}
container protocol-range {
    presence "Enables setting protocol range";
    description
        "Inclusive range representing IP protocol(s) to be used. When
        only start-protocol is present, it represents a single
        protocol.";
    leaf start-protocol {
        type uint8;
        mandatory true;
        description
            "The first 8-bit protocol value to be matched.";
    }
    leaf end-protocol {
        type uint8;
        must ". >= ../start-protocol" {
            error-message
                "The end-protocol must be greater than or equal to
                start-protocol";
        }
        description
            "The last 8-bit protocol value to be matched.";
    }
}
description "ipv4 binary traffic selector";
}

```

```

grouping ipv6-binary-traffic-selector {
  container source-address-range-v6 {
    presence "Enables setting source IPv6 address range";
    description
      "Inclusive range representing IPv6 addresses to be used.
      When only start-address is present, it represents a
      single address.";
    leaf start-address {
      type inet:ipv6-address;
      mandatory true;
      description
        "The first source address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
    leaf end-address {
      type inet:ipv6-address;
      description
        "The last source address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
  }
  container destination-address-range-v6 {
    presence "Enables setting destination IPv6 address range";
    description
      "Inclusive range representing IPv6 addresses to be used.
      When only start-address is present, it represents a
      single address.";
    leaf start-address {
      type inet:ipv6-address;
      mandatory true;
      description
        "The first destination address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
    leaf end-address {
      type inet:ipv6-address;
      description
        "The last destination address, from the
        range of 128-bit IPv6 addresses to be matched";
    }
  }
}
container flow-label-range {
  presence "Enables setting Flow Label range";
  description
    "Inclusive range representing IPv4 addresses to be used. When
    only start-flow-label is present, it represents a single
    flow label.";
  leaf start-flow-label {

```

```
    type inet:ipv6-flow-label;
    description
      "The first flow label value to be matched";
  }
  leaf end-flow-label {
    type inet:ipv6-flow-label;
    must ". >= ../start-flow-label" {
      error-message
        "The end-flow-label must be greater than or equal to
        start-flow-label";
    }
    description
      "The first flow label value to be matched";
  }
}
container traffic-class-range {
  presence "Enables setting the traffic class range";
  description
    "Inclusive range representing IPv4 addresses to be used. When
    only start-traffic-class is present, it represents a single
    traffic class.";
  leaf start-traffic-class {
    type inet:dscp;
    description
      "The first traffic class value to be matched";
    reference
      "RFC 3260: New Terminology and Clarifications for Diffserv
      RFC 3168: The Addition of Explicit Congestion Notification
      (ECN) to IP";
  }
  leaf end-traffic-class {
    type inet:dscp;
    must ". >= ../start-traffic-class" {
      error-message
        "The end-traffic-class must be greater than or equal to
        start-traffic-class";
    }
    description
      "The last traffic class value to be matched";
  }
}
container next-header-range {
```

```
presence "Enables setting Next Header range";
description
  "Inclusive range representing Next Headers to be used. When
   only start-next-header is present, it represents a
   single Next Header.";
leaf start-next-header {
```

```
    type uint8;
    description
      "The first 8-bit next header value to be matched.";
  }
  leaf end-next-header {
    type uint8;
    must ". >= ../start-next-header" {
      error-message
        "The end-next-header must be greater than or equal to
         start-next-header";
    }
    description
      "The last 8-bit next header value to be matched.";
  }
}
description "ipv6 binary traffic selector";
}

grouping traffic-selector {
  leaf ts-format {
    type identityref {
      base traffic-selector-format;
    }
    description "Traffic Selector Format";
  }
  uses traffic-selectors:traffic-selector-base;
  uses traffic-selectors:ipv4-binary-traffic-selector;
  uses traffic-selectors:ipv6-binary-traffic-selector;
  description
    "The traffic selector includes the parameters used to match
     packets for a specific flow binding.";
  reference
    "RFC 6089: Flow Bindings in Mobile IPv6 and Network
     Mobility (NEMO) Basic Support";
}
```

```
}  
<CODE ENDS>
```

3.5. [RFC 5777](#) Classifier YANG Model

This module defines the [RFC 5777](#) Classifier.

This module references [[RFC5777](#)].

```
<CODE BEGINS> file "ietf-diam-trafficclassifier@2018-05-17.yang"  
module ietf-diam-trafficclassifier {  
  yang-version 1.1;
```

Matsushima, et al. Expires October 13, 2019 [Page 54]

Internet-Draft YANG for DMM FPC Protocol April 2019

```
namespace  
"urn:ietf:params:xml:ns:yang:ietf-diam-trafficclassifier";  
  
prefix "diamclassifier";  
  
import ietf-inet-types {  
  prefix inet;  
  revision-date 2013-07-15;  
}  
import ietf-yang-types { prefix yang-types; }  
  
organization "IETF Distributed Mobility Management (DMM)  
Working Group";  
  
contact  
"WG Web: <http://tools.ietf.org/wg/netmod/>  
WG List: <mailto:netmod@ietf.org>  
  
WG Chair: Dapeng Liu  
<mailto:maxpassion@gmail.com>  
  
WG Chair: Sri Gundavelli  
<mailto:sgundave@cisco.com>  
  
Editor: Satoru Matsushima  
<mailto:satoru.matsushima@g.softbank.co.jp>  
  
Editor: Lyle Bertz
```

<mailto:lylebe551144@gmail.com>;

description

"This module contains a collection of YANG definitions for traffic classification and QoS Attributes for Diameter.

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

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

revision 2018-05-17 {

Matsushima, et al.

Expires October 13, 2019

[Page 55]

Internet-Draft

YANG for DMM FPC Protocol

April 2019

description

"Initial";

reference

"[RFC 5777](#): Traffic Classification and Quality of Service (QoS) Attributes for Diameter";

}

typedef eui64-address-type {

type string {

length "6";

}

description

"specifies a single layer 2 address in EUI-64 format. The value is an 8-octet encoding of the address as it would appear in the frame header.";

}

typedef direction-type {

type enumeration {

enum IN {

value 0;

description

```

        "Applies to flows from the managed terminal.";
    }
    enum OUT {
        value 1;
        description
            "Applies to flows to the managed terminal.";
    }
    enum BOTH {
        value 2;
        description
            "Applies to flows both to and from the managed
            terminal.";
    }
}
description
    "Specifies in which direction to apply the classifier.";
}
typedef negated-flag-type {
    type enumeration {
        enum False { value 0;
            description "false"; }
        enum True { value 1;
            description "True"; }
    }
}
description
    "When set to True, the meaning of the match is
    inverted and the classifier will match addresses

```

```

    other than those specified by the From-Spec or
    To-Spec AVP.

    Note that the negation does not impact the port
    comparisons.";
}
grouping index {
    leaf index {
        type uint16;
        mandatory true;
        description "Identifier used for referencing";
    }
    description "Index Value";
}
}

```



```

grouping to-from-spec-value {
  leaf-list ip-address {
    type inet:ip-address;
    description "IP address";
  }
  list ip-address-range {
    key index;
    uses diamclassifier:index;
    leaf ip-address-start {
      type inet:ip-address;
      description "IP Address Start";
    }
    leaf ip-address-end {
      type inet:ip-address;
      description "IP Address End";
    }
    description "IP Address Range";
  }
  leaf-list ip-address-mask {
    type inet:ip-prefix;
    description "IP Address Mask";
  }
  leaf-list mac-address {
    type yang-types:mac-address;
    description "MAC address";
  }
  list mac-address-mask {
    key mac-address;
    leaf mac-address {
      type yang-types:mac-address;
      mandatory true;
      description "MAC address";
    }
    leaf macaddress-mask-pattern {

```

```

    type yang-types:mac-address;
    mandatory true;
    description
      "The value specifies the bit positions of a
      MAC address that are taken for matching.";
  }
  description "MAC Address Mask";

```

```

}
leaf-list eui64-address {
    type diamclassifier:eui64-address-type;
    description "EUI64 Address";
}
list eui64-address-mask {
    key eui64-address;
    leaf eui64-address {
        type diamclassifier:eui64-address-type;
        mandatory true;
        description "eui64 address";
    }
    leaf eui64-address-mask-pattern {
        type diamclassifier:eui64-address-type;
        mandatory true;
        description
            "The value is 8 octets specifying the bit
            positions of a EUI64 address that are taken
            for matching.";
    }
    description "EUI64 Address Mask";
}
leaf-list port {
    type inet:port-number;
    description "Port Number";
}
list port-range {
    key index;
    uses diamclassifier:index;
    leaf ip-address-start {
        type inet:port-number;
        description "Port Start";
    }
    leaf ip-address-end {
        type inet:port-number;
        description "Port End";
    }
    description "Port Range";
}
leaf negated {
    type diamclassifier:negated-flag-type;

```

```

        description "Negated";
    }
    leaf use-assigned-address {
        type boolean;
        description "Use Assigned Address";
    }
    description
        "Basic traffic description value";
}

grouping option-type-group {
    leaf option-type {
        type uint8;
        mandatory true;
        description "Option Type";
    }
    leaf-list ip-option-value {
        type string;
        description "Option Value";
    }
    leaf negated {
        type diamclassifier:negated-flag-type;
        description "Negated";
    }
    description "Common X Option Pattern";
}

typedef vlan-id {
    type uint32 {
        range "0..4095";
    }
    description "VLAN ID";
}

grouping classifier {
    leaf protocol {
        type uint8;
        description "Protocol";
    }
    leaf direction {
        type diamclassifier:direction-type;
        description "Direction";
    }
    list from-spec {
        key index;
        uses diamclassifier:index;
        uses diamclassifier:to-from-spec-value;
        description "from specification";
    }
}

```

```
list to-spec {
  key index;
  uses diamclassifier:index;
  uses diamclassifier:to-from-spec-value;
  description "to specification";
}
leaf-list disffserv-code-point {
  type inet:dscp;
  description "DSCP";
}
leaf fragmentation-flag {
  type enumeration {
    enum DF {
      value 0;
      description "Don't Fragment";
    }
    enum MF {
      value 1;
      description "More Fragments";
    }
  }
  description "Fragmenttation Flag";
}
list ip-option {
  key option-type;
  uses diamclassifier:option-type-group;
  description "IP Option Value";
}
list tcp-option {
  key option-type;
  uses diamclassifier:option-type-group;
  description "TCP Option Value";
}
list tcp-flag {
  key tcp-flag-type;
  leaf tcp-flag-type {
    type uint32;
    mandatory true;
    description "TCP Flag Type";
  }
  leaf negated {
    type diamclassifier:negated-flag-type;
    description "Negated";
  }
}
```

```
    }
    description "TCP Flags";
}
list icmp-option {
    key option-type;
```

```
    uses diamclassifier:option-type-group;
    description "ICMP Option Value";
}
list eth-option {
    key index;
    uses diamclassifier:index;
    container eth-proto-type {
        leaf-list eth-ether-type {
            type string {
                length "2";
            }
            description "value of ethertype field";
        }
        leaf-list eth-sap {
            type string {
                length "2";
            }
            description "802.2 SAP";
        }
        description "Ether Proto Type";
    }
    list vlan-id-range {
        key index;
        uses diamclassifier:index;
        leaf-list s-vlan-id-start {
            type diamclassifier:vlan-id;
            description "S-VID VLAN ID Start";
        }
        leaf-list s-vlan-id-end {
            type diamclassifier:vlan-id;
            description "S-VID VLAN ID End";
        }
        leaf-list c-vlan-id-start {
            type diamclassifier:vlan-id;
            description "C-VID VLAN ID Start";
        }
    }
}
```

```

        leaf-list c-vlan-id-end {
            type diamclassifier:vlan-id;
            description "C-VID  VLAN ID End";
        }
        description "VLAN ID Range";
    }
    list user-priority-range {
        key index;
        uses diamclassifier:index;
        leaf-list low-user-priority {
            type uint32 {
                range "0..7";
            }
        }
    }
}

```

```

        }
        description "Low User Priority";
    }
    leaf-list high-user-priority {
        type uint32 {
            range "0..7";
        }
        description "High User Priority";
    }
    description "User priority range";
}
description "Ether Option";
}
description "RFC 5777 Classifier";
}
}
<CODE ENDS>

```

4. FPC YANG Tree Structure

This section only shows the structure for FPC YANG model. NOTE, it does NOT show the settings, Action values or Descriptor Value.

```

descriptor_value:
+--rw (descriptor-value)
  +--:(all-traffic)
  |   +--rw all-traffic?           empty
  +--:(no-traffic)
  |   +--rw no-traffic?           empty

```

```

+--:(prefix-descriptor)
|  +--rw destination-ip?          inet:ip-prefix
|  +--rw source-ip?              inet:ip-prefix
+--:(pmip-selector)
|  +--rw ts-format?              identityref
|  +--rw ipsec-spi-range!
|  |  +--rw start-spi            ipsec-spi
|  |  +--rw end-spi?            ipsec-spi
|  +--rw source-port-range!
|  |  +--rw start-port           inet:port-number
|  |  +--rw end-port?           inet:port-number
|  +--rw destination-port-range!
|  |  +--rw start-port           inet:port-number
|  |  +--rw end-port?           inet:port-number
|  +--rw source-address-range-v4!
|  |  +--rw start-address        inet:ipv4-address
|  |  +--rw end-address?        inet:ipv4-address
|  +--rw destination-address-range-v4!
|  |  +--rw start-address        inet:ipv4-address

```

```

|  |  +--rw end-address?        inet:ipv4-address
|  +--rw ds-range!
|  |  +--rw start-ds            inet:dscp
|  |  +--rw end-ds?            inet:dscp
|  +--rw protocol-range!
|  |  +--rw start-protocol      uint8
|  |  +--rw end-protocol?      uint8
|  +--rw source-address-range-v6!
|  |  +--rw start-address        inet:ipv6-address
|  |  +--rw end-address?        inet:ipv6-address
|  +--rw destination-address-range-v6!
|  |  +--rw start-address        inet:ipv6-address
|  |  +--rw end-address?        inet:ipv6-address
|  +--rw flow-label-range!
|  |  +--rw start-flow-label?   inet:ipv6-flow-label
|  |  +--rw end-flow-label?    inet:ipv6-flow-label
|  +--rw traffic-class-range!
|  |  +--rw start-traffic-class? inet:dscp
|  |  +--rw end-traffic-class?  inet:dscp
|  +--rw next-header-range!
|  |  +--rw start-next-header?   uint8
|  |  +--rw end-next-header?    uint8

```

```

+--:(rfc5777-classifier-template)
|  +--rw rfc5777-classifier-template
|    +--rw protocol?                uint8
|    +--rw direction?              diamclassifier:direction-type
|    +--rw from-spec* [index]
|      |  +--rw index                uint16
|      |  +--rw ip-address*         inet:ip-address
|      |  +--rw ip-address-range* [index]
|      |    |  +--rw index            uint16
|      |    |  +--rw ip-address-start? inet:ip-address
|      |    |  +--rw ip-address-end?  inet:ip-address
|      |  +--rw ip-address-mask*     inet:ip-prefix
|      |  +--rw mac-address*         yang-types:mac-address
|      |  +--rw mac-address-mask* [mac-address]
|      |    |  +--rw mac-address      yang-types:mac-address
|      |    |  +--rw macaddress-mask-pattern yang-types:mac-address
|      |  +--rw eui64-address*
|      |    |  diamclassifier:eui64-address-type
|      |  +--rw eui64-address-mask* [eui64-address]
|      |    |  +--rw eui64-address
|      |    |    |  diamclassifier:eui64-address-type
|      |    |  +--rw eui64-address-mask-pattern
|      |    |    |  diamclassifier:eui64-address-type
|      |  +--rw port*               inet:port-number
|      |  +--rw port-range* [index]
|      |    |  +--rw index            uint16

```

```

|      |    |  +--rw ip-address-start?  inet:port-number
|      |    |  +--rw ip-address-end?    inet:port-number
|      |  +--rw negated?
|      |    |  diamclassifier:negated-flag-type
|      |  +--rw use-assigned-address?  boolean
+--rw to-spec* [index]
|  +--rw index                uint16
|  +--rw ip-address*         inet:ip-address
|  +--rw ip-address-range* [index]
|    |  +--rw index            uint16
|    |  +--rw ip-address-start? inet:ip-address
|    |  +--rw ip-address-end?   inet:ip-address
|  +--rw ip-address-mask*     inet:ip-prefix
|  +--rw mac-address*         yang-types:mac-address
|  +--rw mac-address-mask* [mac-address]

```



```

|   |   |   |--rw mac-address                yang-types:mac-address
|   |   |   |--rw macaddress-mask-pattern  yang-types:mac-address
|   |   |--rw eui64-address*
|   |       |   diamclassifier:eui64-address-type
|   |--rw eui64-address-mask* [eui64-address]
|   |   |   |--rw eui64-address
|   |       |   diamclassifier:eui64-address-type
|   |   |--rw eui64-address-mask-pattern
|   |       |   diamclassifier:eui64-address-type
|   |--rw port*                          inet:port-number
|   |--rw port-range* [index]
|   |   |--rw index                        uint16
|   |   |--rw ip-address-start?           inet:port-number
|   |   |--rw ip-address-end?             inet:port-number
|   |--rw negated?
|       |   diamclassifier:negated-flag-type
|   |--rw use-assigned-address?           boolean
|--rw disffserv-code-point*              inet:dscp
|--rw fragmentation-flag?                 enumeration
|--rw ip-option* [option-type]
|   |--rw option-type                      uint8
|   |--rw ip-option-value*                 string
|   |--rw negated?                         diamclassifier:negated-flag-type
|--rw tcp-option* [option-type]
|   |--rw option-type                      uint8
|   |--rw ip-option-value*                 string
|   |--rw negated?                         diamclassifier:negated-flag-type
|--rw tcp-flag* [tcp-flag-type]
|   |--rw tcp-flag-type                    uint32
|   |--rw negated?                         diamclassifier:negated-flag-type
|--rw icmp-option* [option-type]
|   |--rw option-type                      uint8
|   |--rw ip-option-value*                 string

```

```

|   |   |--rw negated?                      diamclassifier:negated-flag-type
|--rw eth-option* [index]
|   |--rw index                            uint16
|   |--rw eth-proto-type
|   |   |--rw eth-ether-type*              string
|   |   |--rw eth-sap*                     string
|--rw vlan-id-range* [index]
|   |--rw index                            uint16

```

```

|         |   +--rw s-vlan-id-start*   diamclassifier:vlan-id
|         |   +--rw s-vlan-id-end*     diamclassifier:vlan-id
|         |   +--rw c-vlan-id-start* diamclassifier:vlan-id
|         |   +--rw c-vlan-id-end* diamclassifier:vlan-id
|         +--rw user-priority-range* [index]
|           +--rw index          uint16
|           +--rw low-user-priority* uint32
|           +--rw high-user-priority* uint32
+--:(packet-filter)
|   +--rw packet-filter
|     +--rw direction?          fpcbase:packet-filter-direction
|     +--rw identifier?         uint8
|     +--rw evaluation-precedence? uint8
|     +--rw contents* [component-type-identifier]
|       +--rw component-type-identifier fpcbase:component-type-id
|       +--rw (value)?
|         +--:(ipv4-local)
|           | +--rw ipv4-local?          inet:ipv4-address
|         +--:(ipv6-prefix-local)
|           | +--rw ipv6-prefix-local?   inet:ipv6-prefix
|         +--:(ipv4-ipv6-remote)
|           | +--rw ipv4-ipv6-remote?   inet:ip-address
|         +--:(ipv6-prefix-remote)
|           | +--rw ipv6-prefix-remote?  inet:ipv6-prefix
|         +--:(next-header)
|           | +--rw next-header?         uint8
|         +--:(local-port)
|           | +--rw local-port?          inet:port-number
|         +--:(local-port-range)
|           | +--rw local-port-lo?       inet:port-number
|           | +--rw local-port-hi?       inet:port-number
|         +--:(remote-port)
|           | +--rw remote-port?         inet:port-number
|         +--:(remote-port-range)
|           | +--rw remote-port-lo?      inet:port-number
|           | +--rw remote-port-hi?      inet:port-number
|         +--:(ipsec-index)
|           | +--rw ipsec-index?         traffic-selectors:ipsec-spi
|         +--:(traffic-class)
|           | +--rw traffic-class?       inet:dscp

```

```

|         +--:(traffic-class-range)

```

```

|         | +--rw traffic-class-lo?          inet:dscp
|         | +--rw traffic-class-hi?          inet:dscp
|         +--:(flow-label)
|         +--rw flow-label*             inet:ipv6-flow-label
+--:(tunnel-info)
  +--rw tunnel-info
    +--rw tunnel-local-address?         inet:ip-address
    +--rw tunnel-remote-address?        inet:ip-address
    +--rw mtu-size?                     uint32
    +--rw tunnel?                       identityref
    +--rw payload-type?                 enumeration
    +--rw gre-key?                      uint32
    +--rw gtp-tunnel-info
      | +--rw local-tunnel-identifier?   uint32
      | +--rw remote-tunnel-identifier?  uint32
      | +--rw sequence-numbers-enabled?  boolean
    +--rw ebi?                          fpcbase:ebi-type
    +--rw lbi?                          fpcbase:ebi-type

```

action_value:

```

+--:(action-value)
| +--rw (action-value)
|   +--:(drop)
|   | +--rw drop?                       empty
|   +--:(rewrite)
|   | +--rw rewrite
|   |   +--rw (rewrite-value)?
|   |   | +--:(prefix-descriptor)
|   |   | | +--rw destination-ip?       inet:ip-prefix
|   |   | | +--rw source-ip?           inet:ip-prefix
|   |   | +--:(pmip-selector)
|   |   | | +--rw ts-format?            identityref
|   |   | | +--rw ipsec-spi-range!
|   |   | | | +--rw start-spi          ipsec-spi
|   |   | | | +--rw end-spi?          ipsec-spi
|   |   | | +--rw source-port-range!
|   |   | | | +--rw start-port         inet:port-number
|   |   | | | +--rw end-port?         inet:port-number
|   |   | | +--rw destination-port-range!
|   |   | | | +--rw start-port         inet:port-number
|   |   | | | +--rw end-port?         inet:port-number
|   |   | | +--rw source-address-range-v4!
|   |   | | | +--rw start-address      inet:ipv4-address
|   |   | | | +--rw end-address?      inet:ipv4-address
|   |   | | +--rw destination-address-range-v4!
|   |   | | | +--rw start-address      inet:ipv4-address
|   |   | | | +--rw end-address?      inet:ipv4-address

```

```

| | | +--rw ds-range!
| | | | +--rw start-ds      inet:dscp
| | | | +--rw end-ds?      inet:dscp
| | | +--rw protocol-range!
| | | | +--rw start-protocol  uint8
| | | | +--rw end-protocol?  uint8
| | | +--rw source-address-range-v6!
| | | | +--rw start-address  inet:ipv6-address
| | | | +--rw end-address?  inet:ipv6-address
| | | +--rw destination-address-range-v6!
| | | | +--rw start-address  inet:ipv6-address
| | | | +--rw end-address?  inet:ipv6-address
| | | +--rw flow-label-range!
| | | | +--rw start-flow-label?  inet:ipv6-flow-label
| | | | +--rw end-flow-label?   inet:ipv6-flow-label
| | | +--rw traffic-class-range!
| | | | +--rw start-traffic-class?  inet:dscp
| | | | +--rw end-traffic-class?   inet:dscp
| | | +--rw next-header-range!
| | | | +--rw start-next-header?  uint8
| | | | +--rw end-next-header?   uint8
| | | +--:(rfc5777-classifier-template)
| | | +--rw rfc5777-classifier-template
| | | | +--rw protocol?          uint8
| | | | +--rw direction?
| | | | |   diamclassifier:direction-type
| | | | +--rw from-spec* [index]
| | | | |   +--rw index          uint16
| | | | |   +--rw ip-address*    inet:ip-address
| | | | |   +--rw ip-address-range* [index]
| | | | | |   +--rw index          uint16
| | | | | |   +--rw ip-address-start?  inet:ip-address
| | | | | |   +--rw ip-address-end?   inet:ip-address
| | | | | +--rw ip-address-mask*    inet:ip-prefix
| | | | | +--rw mac-address*      yang-types:mac-address
| | | | | +--rw mac-address-mask* [mac-address]
| | | | | |   +--rw mac-address
| | | | | | |   yang-types:mac-address
| | | | | | |   +--rw macaddress-mask-pattern
| | | | | | |   yang-types:mac-address
| | | | | +--rw eui64-address*
| | | | | |   diamclassifier:eui64-address-type
| | | | | +--rw eui64-address-mask* [eui64-address]
| | | | | |   +--rw eui64-address
| | | | | | |   diamclassifier:eui64-address-type
| | | | | |   +--rw eui64-address-mask-pattern

```



```

| | | +--rw ip-option* [option-type]
| | | | +--rw option-type      uint8
| | | | +--rw ip-option-value* string
| | | | +--rw negated?
| | | | | diamclassifier:negated-flag-type
| | | +--rw tcp-option* [option-type]
| | | | +--rw option-type      uint8
| | | | +--rw ip-option-value* string
| | | | +--rw negated?

```

```

| | | | | diamclassifier:negated-flag-type
| | | | +--rw tcp-flag* [tcp-flag-type]
| | | | | +--rw tcp-flag-type  uint32
| | | | | +--rw negated?
| | | | | | diamclassifier:negated-flag-type
| | | | +--rw icmp-option* [option-type]
| | | | | +--rw option-type      uint8
| | | | | +--rw ip-option-value* string
| | | | | +--rw negated?
| | | | | | diamclassifier:negated-flag-type
| | | | +--rw eth-option* [index]
| | | | | +--rw index              uint16
| | | | | +--rw eth-proto-type
| | | | | | +--rw eth-ether-type* string
| | | | | | +--rw eth-sap*        string
| | | | | +--rw vlan-id-range* [index]
| | | | | | +--rw index          uint16
| | | | | | +--rw s-vlan-id-start*
| | | | | | | diamclassifier:vlan-id
| | | | | | +--rw s-vlan-id-end*
| | | | | | | diamclassifier:vlan-id
| | | | | | +--rw c-vlan-id-start*
| | | | | | | diamclassifier:vlan-id
| | | | | | +--rw c-vlan-id-end*
| | | | | | | diamclassifier:vlan-id
| | | | | +--rw user-priority-range* [index]
| | | | | | +--rw index          uint16
| | | | | | +--rw low-user-priority* uint32
| | | | | | +--rw high-user-priority* uint32
| | | +--:(copy-forward-nexthop)
| | | | +--rw copy-forward-nexthop
| | | | | +--rw (next-hop-value)?

```



```

| | | | +---rw max-rate          uint32
| | | | +---rw service-flag      boolean
| | | | +---rw exclude-flag      boolean
| | | | +---rw priority-level          uint8
| | | | +---rw preemption-capability  enumeration
| | | | +---rw preemption-vulnerability enumeration
| | | | +---rw agg-max-dl?
| | | | |   qos-pmip:Aggregate-Max-DL-Bit-Rate-Value
| | | | +---rw agg-max-ul?
| | | | |   qos-pmip:Aggregate-Max-UL-Bit-Rate-Value
| | | | +---rw gbr-dl?
| | | | |   qos-pmip:Guaranteed-DL-Bit-Rate-Value
| | | | +---rw gbr-ul?
| | | | |   qos-pmip:Guaranteed-UL-Bit-Rate-Value
| | | | +---rw qci?
| | | | |   fpcbase:fpc-qos-class-identifier
| | | | +---rw ue-agg-max-bitrate?    uint32
| | | | +---rw apn-ambr?             uint32

```

policy-configuration-value:

```

| | | | +---rw (policy-configuration-value)?
| | | | |   +---:(descriptor-value)
| | | | |   |   ...
| | | | |   +---:(action-value)
| | | | |   |   ...
| | | | |   +---:(setting-value)
| | | | |   +---rw setting? <anydata>

```

policy-configuration:

```

| | | | +---rw policy-configuration* [index]

```

```

| | | | +---rw index          uint16
| | | | +---rw extensible?    boolean
| | | | +---rw static-attributes* string
| | | | +---rw mandatory-attributes* string
| | | | +---rw entity-state?  enumeration
| | | | +---rw version?       uint32
| | | | +---rw (policy-configuration-value)?
| | | | |   ...

```

module: ietf-dmm-fpc

```

+---rw tenant* [tenant-key]

```

```

+--rw tenant-key                    fpc:fpc-identity
+--rw topology-information-model
|  +--rw service-group* [service-group-key role-key]
|  |  +--rw service-group-key      fpc:fpc-identity
|  |  +--rw service-group-name?   string
|  |  +--rw role-key              identityref
|  |  +--rw role-name?           string
|  |  +--rw protocol*            identityref
|  |  +--rw feature*             identityref
|  |  +--rw service-group-configuration* [index]
|  |  |  +--rw index                uint16
|  |  |  +--rw (policy-configuration-value)?
|  |  |  |  ...
|  |  +--rw dpn* [dpn-key]
|  |  |  +--rw dpn-key              fpc:fpc-identity
|  |  |  +--rw referenced-interface* [interface-key]
|  |  |  |  +--rw interface-key      fpc:fpc-identity
|  |  |  |  +--rw peer-service-group-key* fpc:fpc-identity
|  +--rw dpn* [dpn-key]
|  |  +--rw dpn-key                fpc:fpc-identity
|  |  +--rw dpn-name?             string
|  |  +--rw dpn-resource-mapping-reference? string
|  |  +--rw domain-key            fpc:fpc-identity
|  |  +--rw service-group-key*    fpc:fpc-identity
|  |  +--rw interface* [interface-key]
|  |  |  +--rw interface-key        fpc:fpc-identity
|  |  |  +--rw interface-name?     string
|  |  |  +--rw role?              identityref
|  |  |  +--rw protocol*          identityref
|  |  |  +--rw interface-configuration* [index]
|  |  |  |  +--rw (policy-configuration-value)?
|  |  |  |  |  ...
|  |  +--rw dpn-policy-configuration* [policy-template-key]
|  |  |  +--rw policy-template-key  fpc:fpc-identity
|  |  |  +--rw policy-configuration* [index]
|  |  |  |  +--rw index            uint16
|  |  |  |  +--rw (policy-configuration-value)?

```

```

|  |  |  |  |  ...
|  +--rw domain* [domain-key]
|  |  +--rw domain-key      fpc:fpc-identity
|  |  +--rw domain-name?   string

```

```

| | +--rw domain-policy-configuration* [policy-template-key]
| | | +--rw policy-template-key      fpc:fpc-identity
| | | +--rw policy-configuration* [index]
| | | |
| | | | ...
+--rw dpn-checkpoint
| | +--rw basename?                  fpc:fpc-identity
| | +--rw base-checkpoint?          string
+--rw service-group-checkpoint
| | +--rw basename?                  fpc:fpc-identity
| | +--rw base-checkpoint?          string
+--rw dpn-checkpoint
| | +--rw basename?                  fpc:fpc-identity
| | +--rw base-checkpoint?          string
+--rw policy-information-model
| +--rw action-template* [action-template-key]
| | +--rw action-template-key        fpc:fpc-identity
| | +--rw (action-value)
| | |
| | | ...
| | +--rw extensible?                boolean
| | +--rw static-attributes*         string
| | +--rw mandatory-attributes*     string
| | +--rw entity-state?              enumeration
| | +--rw version?                   uint32
+--rw descriptor-template* [descriptor-template-key]
| | +--rw descriptor-template-key     fpc:fpc-identity
| | +--rw (descriptor-value)
| | |
| | | ...
| | +--rw extensible?                boolean
| | +--rw static-attributes*         string
| | +--rw mandatory-attributes*     string
| | +--rw entity-state?              enumeration
| | +--rw version?                   uint32
+--rw rule-template* [rule-template-key]
| | +--rw rule-template-key           fpc:fpc-identity
| | +--rw descriptor-match-type       enumeration
| | +--rw descriptor-configuration* [descriptor-template-key]
| | | +--rw descriptor-template-key   fpc:fpc-identity
| | | +--rw direction?                rfc5777:direction-type
| | | +--rw setting?                  <anydata>
| | | +--rw attribute-expression* [index]
| | | | +--rw index                    uint16
| | | | +--rw (descriptor-value)
| | | | |
| | | | | ...
| | +--rw action-configuration* [action-order]

```

```

| | | +--rw action-order          uint32
| | | +--rw action-template-key   fpc:fpc-identity
| | | +--rw setting?              <anydata>
| | | +--rw attribute-expression* [index]
| | |   +--rw index                uint16
| | |   +--rw (action-value)
| | |   | ...
| | | +--rw extensible?           boolean
| | | +--rw static-attributes*    string
| | | +--rw mandatory-attributes* string
| | | +--rw entity-state?        enumeration
| | | +--rw version?             uint32
| | | +--rw rule-configuration* [index]
| | |   +--rw index              uint16
| | |   +--rw (policy-configuration-value)?
| | |   | ...
+--rw policy-template* [policy-template-key]
| +--rw policy-template-key   fpc:fpc-identity
| +--rw rule-template* [precedence]
| | +--rw precedence          uint32
| | +--rw rule-template-key   fpc:fpc-identity
+--rw extensible?            boolean
+--rw static-attributes*    string
+--rw mandatory-attributes* string
+--rw entity-state?        enumeration
+--rw version?             uint32
+--rw policy-configuration* [index]
| ...
+--rw basename?            fpc:fpc-identity
+--rw base-checkpoint?    string
+--rw mobility-context* [mobility-context-key]
| +--rw mobility-context-key   fpc:fpc-identity
+--rw delegating-ip-prefix* inet:ip-prefix
+--rw parent-context?     fpc:fpc-identity
+--rw child-context*      fpc:fpc-identity
+--rw mobile-node
| +--rw ip-address*          inet:ip-address
| +--rw imsi?                fpcbase:imsi-type
| +--rw mn-policy-configuration* [policy-template-key]
| | +--rw policy-template-key   fpc:fpc-identity
| | +--rw policy-configuration* [index]
| | ...
+--rw domain
| +--rw domain-key?         fpc:fpc-identity
| +--rw domain-policy-configuration* [policy-template-key]
| | +--rw policy-template-key   fpc:fpc-identity
| | +--rw policy-configuration* [index]

```

```

|   +--rw dpn* [dpn-key]
|   |   +--rw dpn-key          fpc:fpc-identity
|   |   +--rw dpn-policy-configuration* [policy-template-key]
|   |   |   +--rw policy-template-key    fpc:fpc-identity
|   |   |   +--rw policy-configuration* [index]
|   |   |   ...
|   |   +--rw role?            identityref
|   |   +--rw service-data-flow* [identifier]
|   |   |   +--rw identifier            uint32
|   |   |   +--rw service-group-key?   fpc:fpc-identity
|   |   |   +--rw interface* [interface-key]
|   |   |   |   +--rw interface-key    fpc:fpc-identity
|   |   |   +--rw service-data-flow-policy-
|   |   |       configuration* [policy-template-key]
|   |   |       +--rw policy-template-key    fpc:fpc-identity
|   |   |       +--rw policy-configuration* [index]
|   |   |       ...
|   +--rw monitor* [monitor-key]
|   |   +--rw extensible?          boolean
|   |   +--rw static-attributes*   string
|   |   +--rw mandatory-attributes* string
|   |   +--rw entity-state?        enumeration
|   |   +--rw version?             uint32
|   |   +--rw monitor-key          fpc:fpc-identity
|   |   +--rw target?              string
|   |   +--rw deferrable?          boolean
|   |   +--rw (configuration)
|   |   |   +--:(period)
|   |   |   |   +--rw period?        uint32
|   |   |   +--:(threshold-config)
|   |   |   |   +--rw low?            uint32
|   |   |   |   +--rw hi?            uint32
|   |   |   +--:(schedule)
|   |   |   |   +--rw schedule?      uint32
|   |   |   +--:(event-identities)
|   |   |   |   +--rw event-identities*    identityref
|   |   |   +--:(event-ids)
|   |   |   |   +--rw event-ids*        uint32

```

rpcs:

```

+---x configure
|   +---w input
|   |   +---w client-id          fpc:client-identifier
|   |   +---w execution-delay?  uint32
|   |   +---w yang-patch
|   |   |   +---w patch-id      string
|   |   |   +---w comment?     string
|   |   |   +---w edit* [edit-id]

```

```

|   |   +---w edit-id          string
|   |   +---w operation       enumeration
|   |   +---w target          target-resource-offset
|   |   +---w point?         target-resource-offset
|   |   +---w where?         enumeration
|   |   +---w value?         <anydata>
|   |   +---w reference-scope? fpc:ref-scope
|   |   +---w command-set
|   |   |   +---w (instr-type)?
|   |   |   |   +---:(instr-3gpp-mob)
|   |   |   |   |   +---w instr-3gpp-mob? fpcbase:threegpp-instr
|   |   |   |   +---:(instr-pmip)
|   |   |   |   |   +---w instr-pmip?      pmip-commandset
+---ro output
+---ro yang-patch-status
+---ro patch-id      string
+---ro (global-status)?
|   +---:(global-errors)
|   |   +---ro errors
|   |   |   +---ro error*
|   |   |   |   +---ro error-type      enumeration
|   |   |   |   +---ro error-tag       string
|   |   |   |   +---ro error-app-tag?  string
|   |   |   |   +---ro error-path?    instance-identifier
|   |   |   |   +---ro error-message?  string
|   |   |   |   +---ro error-info?    <anydata>
|   |   +---:(ok)
|   |   +---ro ok?          empty
+---ro edit-status
+---ro edit* [edit-id]
+---ro edit-id          string
+---ro (edit-status-choice)?
|   +---:(ok)

```

```

|         | +--ro ok?                empty
|         | +--ro notify-follows?   boolean
|         | +--ro subsequent-edit* [edit-id]
|         |         +--ro edit-id      string
|         |         +--ro operation   enumeration
|         |         +--ro target
|         |             ypatch:target-resource-offset
|         | +--ro point?
|         |             ypatch:target-resource-offset
|         | +--ro where?    enumeration
|         | +--ro value?    <anydata>
|         +---:(errors)
|             +--ro errors
|             +--ro error*
|                 +--ro error-type    enumeration

```

```

|         | +--ro error-tag      string
|         | +--ro error-app-tag? string
|         | +--ro error-path?
|         |             instance-identifier
|         | +--ro error-message? string
|         | +--ro error-info?    <anydata>
+---x register_monitor
|   +---w input
|     | +---w client-id      fpc:client-identifier
|     | +---w execution-delay? uint32
|     | +---w operation-id    uint64
|     | +---w monitor* [monitor-key]
|     |     +---w extensible?    boolean
|     |     +---w static-attributes* string
|     |     +---w mandatory-attributes* string
|     |     +---w entity-state?  enumeration
|     |     +---w version?       uint32
|     |     +---w monitor-key    fpc:fpc-identity
|     |     +---w target?        string
|     |     +---w deferrable?    boolean
|     | +---w (configuration)
|     |     +---:(period)
|     |         | +---w period?          uint32
|     |         +---:(threshold-config)
|     |             | +---w low?          uint32
|     |             | +---w hi?          uint32

```



```

|           +---ro error-info?      <anydata>
+---x probe
  +---w input
    | +---w client-id                fpc:client-identifier
    | +---w execution-delay?         uint32
    | +---w operation-id              uint64
    | +---w monitor* [monitor-key]
    |   +---w monitor-key            fpc:fpc-identity
  +---ro output
    +---ro operation-id              uint64
    +---ro (edit-status-choice)?
      +---:(ok)
      | +---ro ok?                    empty
      +---:(errors)
        +---ro errors
          +---ro error*
            +---ro error-type         enumeration
            +---ro error-tag          string
            +---ro error-app-tag?     string
            +---ro error-path?       instance-identifier
            +---ro error-message?     string
            +---ro error-info?       <anydata>

```

notifications:

```

+---n config-result-notification
| +---ro yang-patch-status
| | +---ro patch-id                 string
| | +---ro (global-status)?
| | | +---:(global-errors)

```

```

| | | | +---ro errors
| | | |   +---ro error*
| | | |     +---ro error-type         enumeration
| | | |     +---ro error-tag          string
| | | |     +---ro error-app-tag?     string
| | | |     +---ro error-path?       instance-identifier
| | | |     +---ro error-message?     string
| | | |     +---ro error-info?       <anydata>
| | | | +---:(ok)
| | | |   +---ro ok?                    empty
| | +---ro edit-status
| |   +---ro edit* [edit-id]

```

```

| |      +---ro edit-id      string
| |      +---ro (edit-status-choice)?
| |          +---:(ok)
| |              | +---ro ok?          empty
| |          +---:(errors)
| |              +---ro errors
| |                  +---ro error*
| |                      +---ro error-type      enumeration
| |                      +---ro error-tag       string
| |                      +---ro error-app-tag?  string
| |                      +---ro error-path?
| |                          instance-identifier
| |                      +---ro error-message?  string
| |                      +---ro error-info?    <anydata>
| +---ro subsequent-edit* [edit-id]
|     +---ro edit-id      string
|     +---ro operation    enumeration
|     +---ro target       ypatch:target-resource-offset
|     +---ro point?      ypatch:target-resource-offset
|     +---ro where?      enumeration
|     +---ro value?      <anydata>
+---n notify
    +---ro notification-id?  uint32
    +---ro timestamp?       uint32
    +---ro report* [monitor-key]
        +---ro monitor-key          fpc:fpc-identity
        +---ro trigger?             identityref
        +---ro (value)?
            +---:(dpn-candidate-available)
            | +---ro node-id?          inet:uri
            | +---ro supported-interface-list* [role-key]
            |     +---ro role-key      identityref
            +---:(dpn-unavailable)
            | +---ro dpn-id?          fpc:fpc-identity
            +---:(report-value)
                +---ro report-value? <anydata>

```

Figure 1: YANG FPC Agent Tree

5. 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.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", [RFC 5777](#), DOI 10.17487/RFC5777, February 2010, <<https://www.rfc-editor.org/info/rfc5777>>.
- [RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", [RFC 6088](#), DOI 10.17487/RFC6088, January 2011, <<https://www.rfc-editor.org/info/rfc6088>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6991](#), DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8072] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Patch Media Type", [RFC 8072](#), DOI 10.17487/RFC8072, February 2017, <<https://www.rfc-editor.org/info/rfc8072>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

6.2. Informative References

- [I-D.ietf-dmm-fpc-cpdp]
Matsushima, S., Bertz, L., Liebsch, M., Gundavelli, S., Moses, D., and C. Perkins, "Protocol for Forwarding Policy Configuration (FPC) in DMM", [draft-ietf-dmm-fpc-cpdp-12](#) (work in progress), June 2018.
- [RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", [RFC 3958](#), DOI 10.17487/RFC3958, January 2005, <<https://www.rfc-editor.org/info/rfc3958>>.
- [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, <<https://www.rfc-editor.org/info/rfc7222>>.

Authors' Addresses

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

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

Lyle Bertz
6220 Sprint Parkway
Overland Park KS, 66251
USA

Email: lylebe551144@gmail.com

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

Phone: +49 6221 4342146
Email: liebsch@neclab.eu

Internet-Draft

YANG for DMM FPC Protocol

April 2019

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

Email: sgundave@cisco.com

Danny Moses

Email: danny.moses@intel.com

Charles E. Perkins
Futurewei Inc.
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1-408-330-4586
Email: charliep@computer.org

