

**Protocol for Forwarding Policy Configuration (FPC)  
in DMM  
draft-ietf-dmm-fpc-cpdp-12**

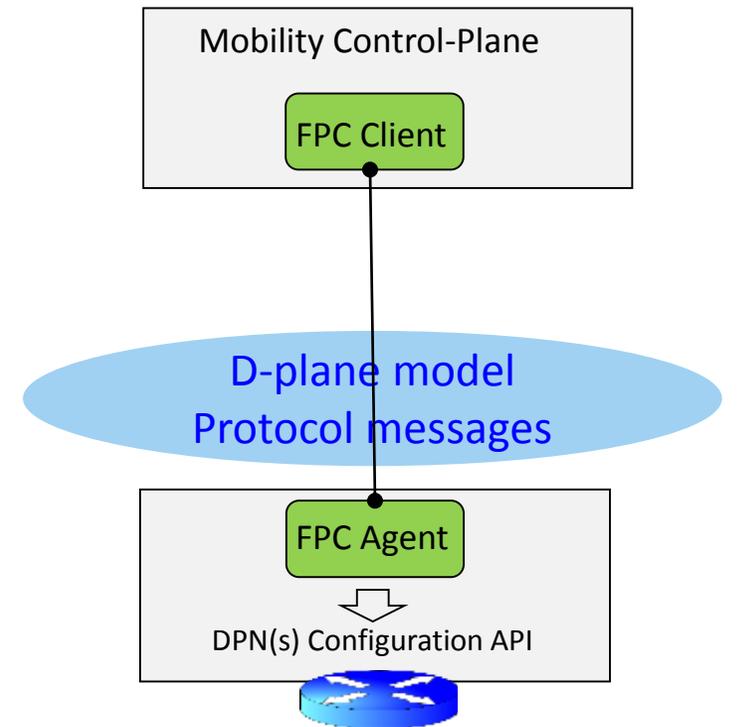
L. Bertz, S. Matsushima, M. Liebsch, S. Gundavelli, D. Moses, C. Perkins

IETF 102, Montreal

July 2018

# What is this work about..?

- Enable the separation of a mobility network's Control-Plane function from its Data-Plane function
- Enable distributed deployment of Control- and Data-Plane functions by abstracted Data-plane model and protocol messages
- Support multi-tenancy on a single real deployed D-plane network and multiple domains within a tenant



# From IETF 101

- Determine if Service-Endpoints are eliminated and modify Service-Groups
  - Resolved during IETF 101 and update was sent out to ML (<https://mailarchive.ietf.org/arch/msg/dmm/nJzgamhzHO0ULFKf4sjgRNEDsDY>)
- Review YANG for NMDA compliance – Complete
  - Added language in Appendix
  - Changed configure statements
- More examples
  - Use of Service Groups during DPN selection
  - Policy Creation
  - DPN Policy Installation
  - Monitor lifecycle
- Editing passes
  - Service-Group resides under the Topology-Information-Mode
  - Domain now has a checkpoint and the Topology Information Model checkpoint was removed

# Service-Groups

- Information model used to select an DPN interface(s) that will meet the needs for a FPC Client
- Considers
  - Protocol, e.g pmip
  - Function, e.g. lma
  - Protocol message (sub)set, e.g. gtp-s5, gtp-s8
  - Features (and Settings) that MUST be known in order to successfully connect, e.g. a key feature MUST be turned OFF
- This is an information model – FPC does NOT decide who (Client or Agent) selects or how

```
|
+--[Service-Group] <G-Key>, <Name> (O) <Set>
|   +--[Extensible: FALSE]
|   +--[Role] <U-Key>
|   +--[Protocol] <Set>
|   +--[Feature] <Set> (O)
|   +--[Service-Group-Configuration] <Set> (O)
|   +--[DPN-Key] <Set>
|   |   +--[Referenced-Interface] <Set>
|   |   |   +--[Interface-Key] <L-Key>
|   |   |   +--[Peer-Service-Group-Key] <Set> (O)
```

# Templates

- Simplify development and maintenance of the needed policies and other objects
- A Template Attribute Notation (Attribute Expression) is provided

```
+-[Template] <U-Key, Name> (M) <Set>
|
| +-[Attributes] <Set> (M)
|
| +-[Extensible ~ FALSE]
|
| +-[Entity-State ~ Initial]
|
| +-[Version]
```

## Attribute Expression Notation

- '[Att-Name: ]' Mandatory Attribute is defined, but template does not provide any configured value.
- '[Att-Name: Att-Value]' Mandatory Attribute is defined, and has a statically configured value.
- '[Att-Name: ~ Att-Value]' Mandatory Attribute is defined, and has a default value.
- '[Att-Name]' Non-mandatory Attribute may be included but template does not provide any configured value.
- '[Att-Name = Att-Value]' Non-mandatory Attribute may be included and has a statically configured value.
- '[Att-Name ~ Att-Value]' Non-mandatory Attribute may be included and has a default value.

# Entity Configuration and Domain

## Entity Configuration

- References a Template (Entity)
- Provides Attribute Expressions that further define/refine the Template

[Entity Configuration Block]

| +-[Entity-Key] (M)

| +-[Attribute-Expression] <Set> (M)

## Domain

- A group of heterogeneous Topology resources typically sharing a common administrative authority.
- Other models, outside of the scope of this specification, provide the details for the Domain.

|

+-[Domain] <G-Key>, <Name> (O) <Set>

|

+-[Domain-Policy-Configuration] (O) <Set>

|

# Policy

- Policy contains Rules (by reference) ordered by Precedence (search order)
  - Has a Configuration
- Rules
  - Contain Descriptors (by Reference) and Direction applied
  - Contains Actions (by Reference) executed by Order
  - Uses Descriptor-Match-Type (AND/OR) to note how the list of Descriptors is logically applied, an AND or OR list
  - Has a Configuration
- Descriptor – Template that describes traffic
- Action – Template that defines action to take on wrt a packet

## +-[Policy Information Model]

	+-[Extensible:]
	+-[Policy-Template] <G-Key> (M) <Set>
	+-[Policy-Configuration] <Set> (O)
	+-[Rule-Template-Key] <List> (M)
	+-[Precedence] (M)
	+-[Rule-Template] <L-Key> (M) <Set>
	+-[Descriptor-Match-Type] (M)
	+-[Descriptor-Configuration] <Set> (M)
	+-[Direction] (O)
	+-[Action-Configuration] <Set> (M)
	+-[Action-Order] (M)
	+-[Rule-Configuration] (O)
	+-[Descriptor-Template] <L-Key> (M) <Set>
	+-[Descriptor-Type] (O)
	+-[Attribute-Expression] <Set> (M)
	+-[Action-Template] <L-Key> (M) <Set>
	+-[Action-Type] (O)
	+-[Attribute-Expression] <Set> (M)

## Usage

Descriptors & Actions are created first, then Rules, then Policies.

Rule Configurations can refine/extend Actions/Descriptors.

Policy Configurations can refine/extend Rules/Actions/Descriptors.

# DPNs

- Has interfaces categorized by Role, Protocols and Configurations
- Domain Reference
- Service-Groups it is part of
- DPN-Policy-Configuration
  - Policies installed on the DPN
- DPN-Resource-Mapping-Reference
  - A reference to the underlying implementation, e.g. physical node, software module, etc. that supports this DPN.

Policy  
Usage

Policies MUST be created prior to installing them on a DPN in DPN-Policy-Configuration or DPN Interface (Interface-Configuration).

A DPN-Policy-Configuration or Interface-Configuration can refine/extend Policies/Rules/Actions/Descriptors.

```
+-[DPN] <G-Key>, <Name> (O) <Set>
|   +-[Extensible: FALSE]
|   +-[Interface] <L-Key> <Set>
|   |   +-[Role] <U-Key>
|   |   +-[Protocol] <Set>
|   |   +-[Interface-Configuration] <Set> (O)
|   +-[Domain-Key]
|   +-[Service-Group-Key] <Set> (O)
|   +-[DPN-Policy-Configuration] <List> (M)
|   +-[DPN-Resource-Mapping-Reference] (O)
```

# Mobility Context

- Contains parent/child references
- Mobile Node Address(es) and Configuration
- Domain(s) used
- Service-Group(s) used
- DPNs where each has
  - Role used
  - DPN specific policy configuration (Policies)
  - Service Data Flow(s)
    - Interface used
    - Configuration (Policies)
      - Direction

```

+-[Mobility-Context] <G-Key> <Set>
|
|   +-[Extensible:~ FALSE]
|   +-[Delegating-IP-Prefix:] <Set> (O)
|   +-[Parent-Context] (O)
|   +-[Child-Context] <Set> (O)
|   +-[Service-Group-Key] <Set> (O)
|   +-[Mobile-Node]
|   |   +-[IP-Address] <Set> (O)
|   |   +-[MN-Policy-Configuration] <Set>
|   +-[Domain-Key]
|   |   +-[Domain-Policy-Configuration] <Set>
|   +-[DPN-Key] <Set>
|   |   +-[Role]
|   |   +-[DPN-Policy-Configuration] <Set>
|   |   +-[ServiceDataFlow] <L-Key> <Set> (O)
|   |   |   +-[Service-Group-Key] (O)
|   |   |   +-[Interface-Key] <Set>
|   |   |   +-[ServiceDataFlow-Policy-
|   |   |       Configuration] <Set> (O)
|   |   |   +-[Direction]

```

**Policy Usage**

Policies MUST be created & installed on the DPN in DPN-Policy-Configuration or DPN Interface (Interface-Configuration). All Mobility-Context Policy-Configurations (Mobile, DPN, Domain or ServiceDataFlow can refine/extend Policies/Rules/Actions/Descriptors.

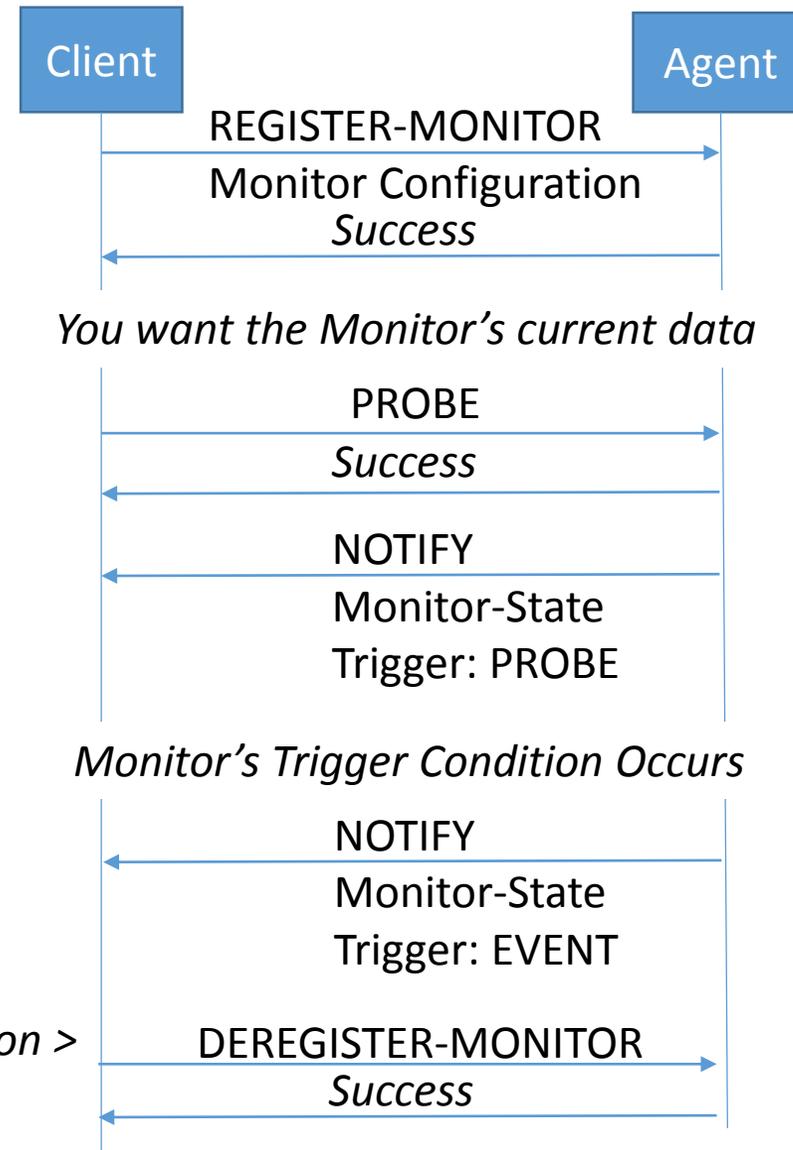
# Monitors

- Target describes what is to be monitored
- Deferrable implies responses can be delayed for updates
- Configuration – Attributes determined by the Monitor sub-type (template). Defaults templates are:
  - Periodic (Report)
  - Event-List (Event Trigger)
  - Scheduled (Report)
  - Threshold (Triggered)

```
+-[Monitor] <G-Key> <List>
|      +-[Extensible:]
|      +-[Target:]
|      +-[Deferrable]
|      +-[Configuration]
```

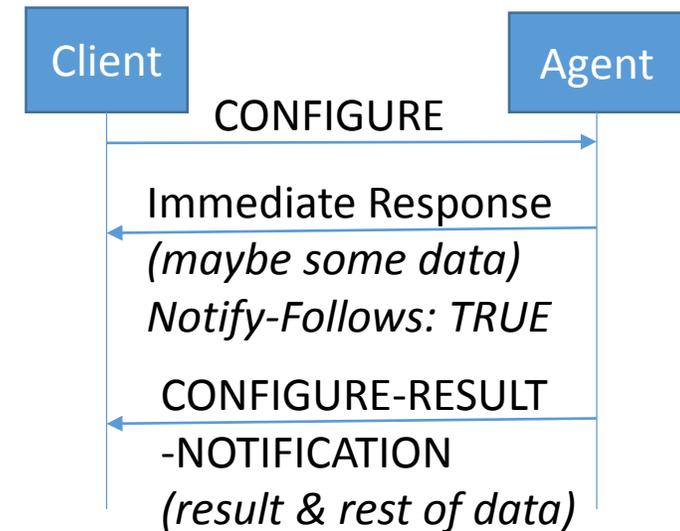
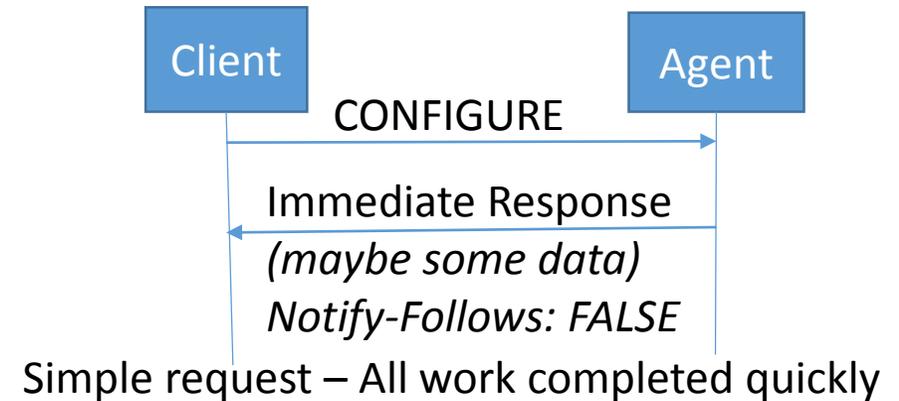
# Monitor Operations

- Register-Monitors (Create)
- Deregister-Monitors (Delete)
- Probe (Force notification of the current Monitor state)
- All data is returned via NOTIFY



# CONFIGURE & CONFIGURE-RESULT-NOTIFICATION

- Follows YANG-PATCH (RFC 8072) style request body & error responses
  - Asks for 1+ 'edits' (changes)
  - Edits can have different operations (next slide)
- In a response the Agent can have *subsequent edits* - those edits required to make the request work
  - Allows Client to only have to fill in part of the information and Agent can add other detail, e.g. Tunnel ID, IP Address, etc.



Complex request – All work cannot be completed quickly.  
Send back priority data and rest later.

# Operations

Operation	Description
create	Creates a new data resource or Entity. If the resource exists an error is returned.
delete	Deletes a resource. If it does not exist an error is returned.
insert	Inserts data in a list or user ordered list.
merge	Merges the edit value with the target data resource; the resource is created if it does not exist.
move	Moves the target data resource.
replace	Replace the target data resource with the edit value.
remove	Removes a data resource if it already exists.
clone	Clones a data resource and places the copy at the new location. If the resource does not exist an error is returned.

FPC information model is a tree & contains lists so operations are designed to take advantage of this.

Clone it NOT a YANG-PATCH operation

# Next Step

- Ask for expert review & feedback

# Appendix

# Service Group

```
| +--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
```