

Distributed Mobility Management (DMM) WG

DMM Work Item:
Forwarding Path & Signaling Management
(FPSM)

FPSM work team

IETF92, Dallas

2015-03-26

Starting point and progress

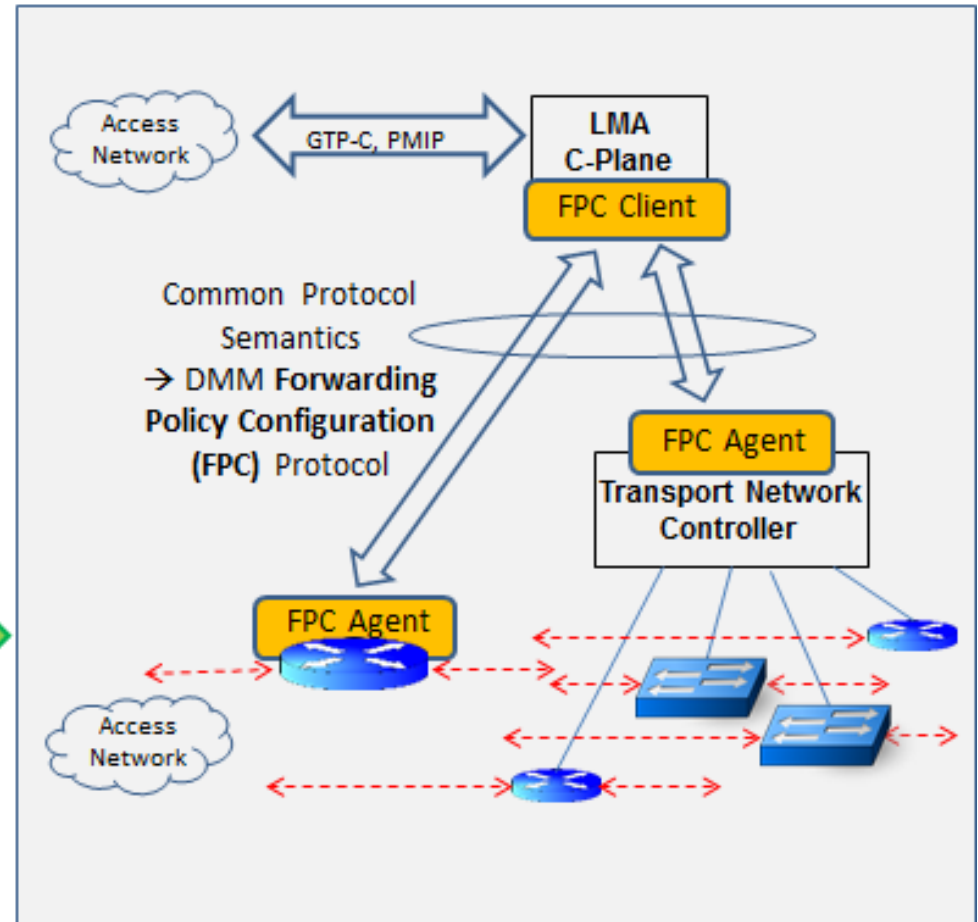
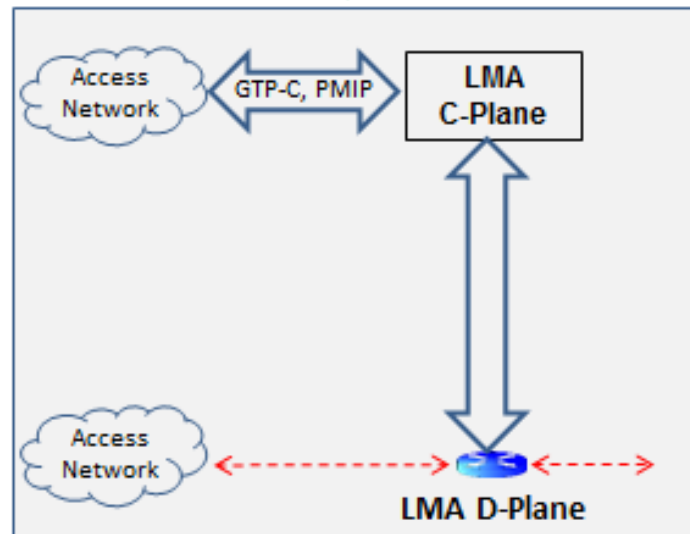
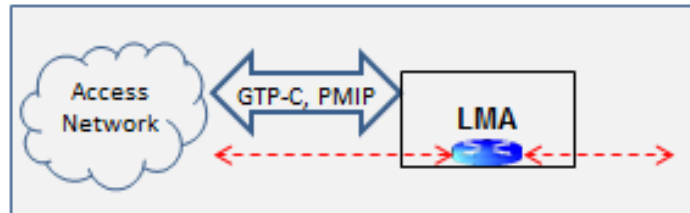
☐ FPSM topic stating point

- ☐ Many high-level ideas, slides, opinions as per previous IETF meetings
- ☐ No Internet draft..

☐ Work team progress

- ☐ 5 WebEx calls; eMail discussion
 - ☐ Participants: Satoru, Danny, Sri, Marco, Pierrick, Alper, Carlos, Georgios
- ☐ Categorization of required protocol function between Mobility Management Control-/Data-Plane
- ☐ Design objectives derived from deployment cases
 - ☐ Functional Architecture
 - ☐ Level of abstraction (from DPN particularities, configuration-specifics)
- ☐ First draft compiled, circulated, discussed, formed
- ☐ Published prior to IETF92: draft-wt-dmm-fpc-cpdp-00

Functional decomposition and FPSM scope



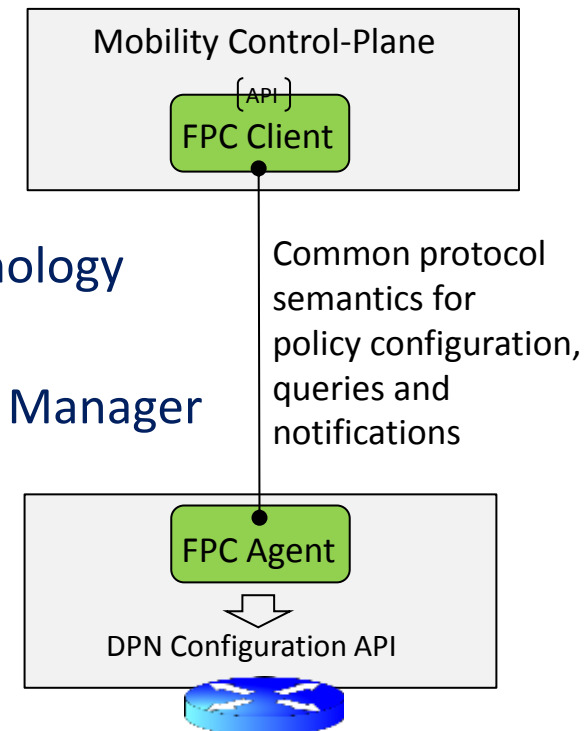
Functional Architecture

❑ Client Function – Associated with Mobility Control-Plane

- ❑ Mobility Control-Plane can utilize Client to configure policies and forwarding rules on one or multiple selected DPN(s)
- ❑ Mobility Control-Plane can integrate and use Client through API
- ❑ Client uses common protocol semantics and policy/forwarding description (applicable to any DPN technology) to communicate with Agent

❑ Agent Function

- ❑ Agent can be installed on Router, Switch or Network Controller
- ❑ Applies common protocol semantics to DPN-technology specific configuration API, e.g.
 - ❑ Router configuration, e.g. associated with RIB Manager
 - ❑ Switch, e.g. using local configuration API
 - ❑ Network Controller



Policy and Forwarding Rules

❑ Objectives

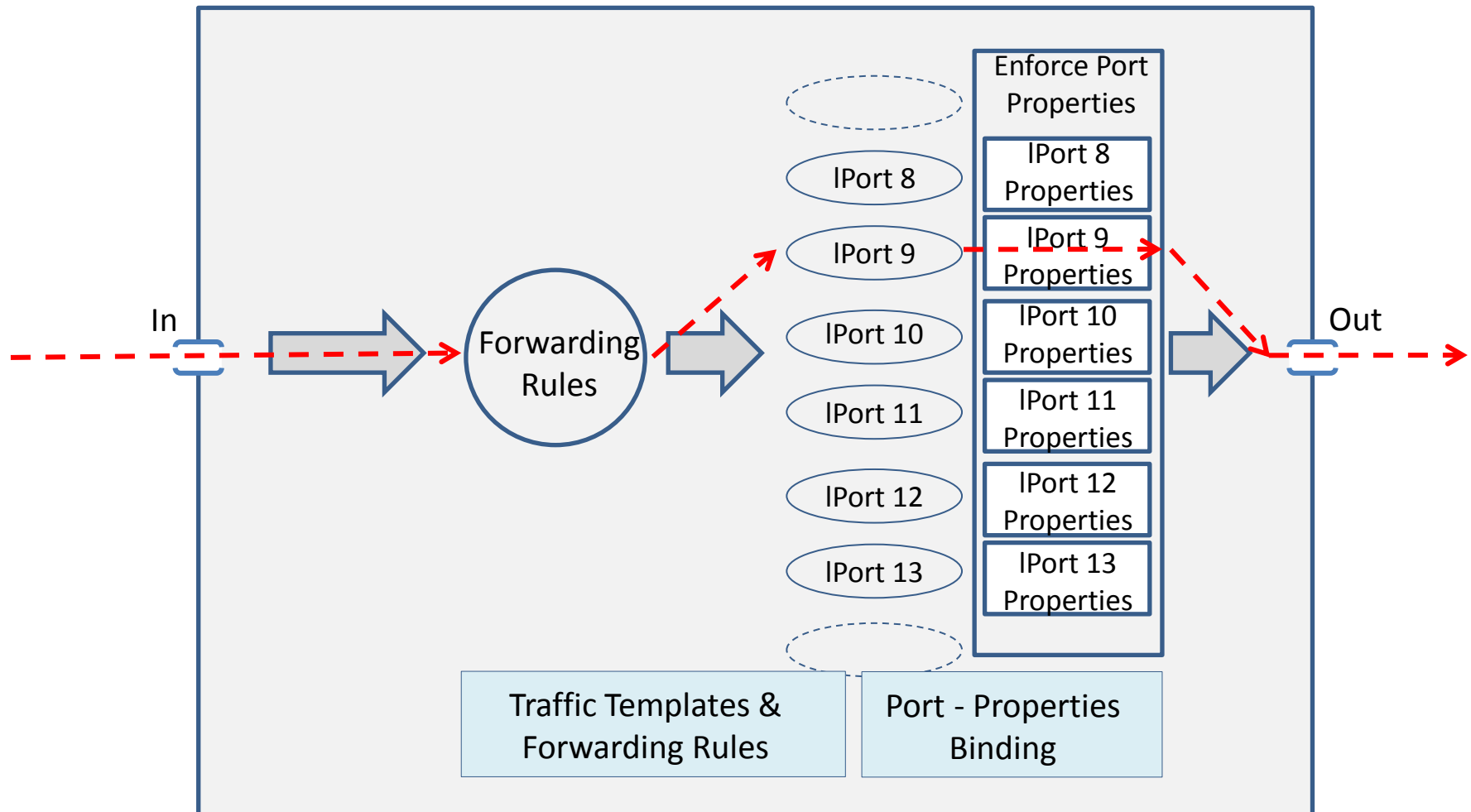
- ❑ Define traffic forwarding rules in a transport network technology agnostic manner
- ❑ Define common semantics for traffic treatment rules and policies (QoS differentiation, encapsulation, address re-writing, ..) without the need to be specific to a Data-Plane Node's configuration (Router, Switch, ..)

❑ Logical Port and bound properties

- ❑ A logical port represents a virtual next hop for Data-Plane traffic and binds configuration for traffic treatment (QoS, tunnel endpoint,..)
- ❑ Configuration determined by one or multiple properties, which are bound to a logical port
- ❑ Forwarding rules direct traffic to one logical port
- ❑ All traffic being directed to a logical port will experience the same treatment through the enforcement of rules/policies according to bound properties

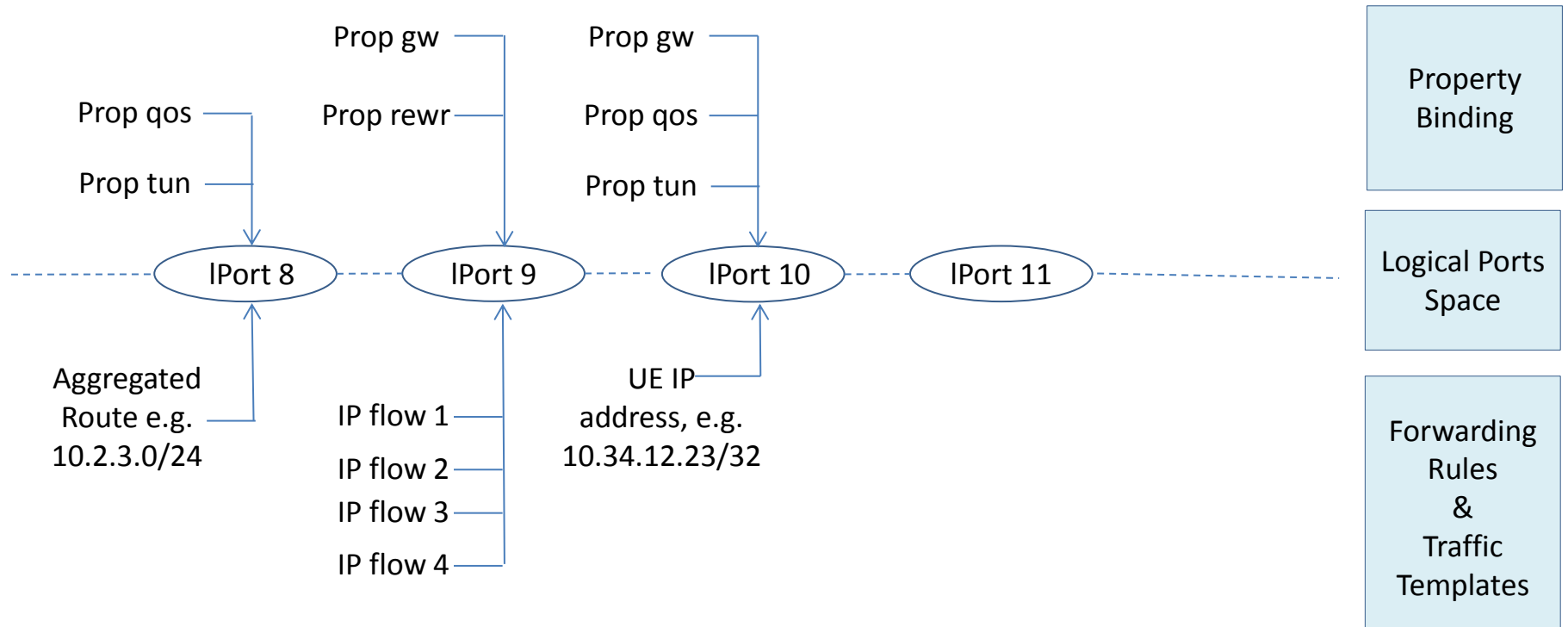
Abstraction Model

DPN Traffic treatment and forwarding



Abstraction Model

Traffic treatment and forwarding – exemplary illustration



Messages

Message	Description	Direction
PRT_ADD	Add a logical port	Client → Agent
PRT_DEL	Delete a logical port	Client → Agent
PROP_ADD	Add a property to a logical port	Client → Agent
PROP_MOD	Modify a property	Client → Agent
PROP_DEL	Delete a property	Client → Agent
RULE_ADD	Add forwarding rule (bind traffic template to logical port)	Client → Agent
RULE_MOD	Modify an existing forwarding rule	Client → Agent
RULE_DEL	Delete a rule	Client → Agent
EVENT_REG	Register an event at an Agent, which is to be monitored by the Agent, and event kind (periodic / event trigger / probed)	Client → Agent
PROBE	Probe the status of a registered event	Client → Agent
NOTIFY	Notify a Client about the status of a monitored attribute at any event kind (periodic / event trigger / probed)	Agent → Client
QUERY	Query a Client about missing states/rules	Agent → Client

Attributes – Identifiers

Attribute	Description
PRT_ID	Port Identifier
PRT_PROP_ID	Identifies a property and the associated logical port
CLI_ID	Identifies an FPC Client function
AGT_ID	Identifies an FPC Agent function
DPN_ID	Identifies a Data-Plane Node
EVENT_ID	Identifies a registered event

Attributes – Properties

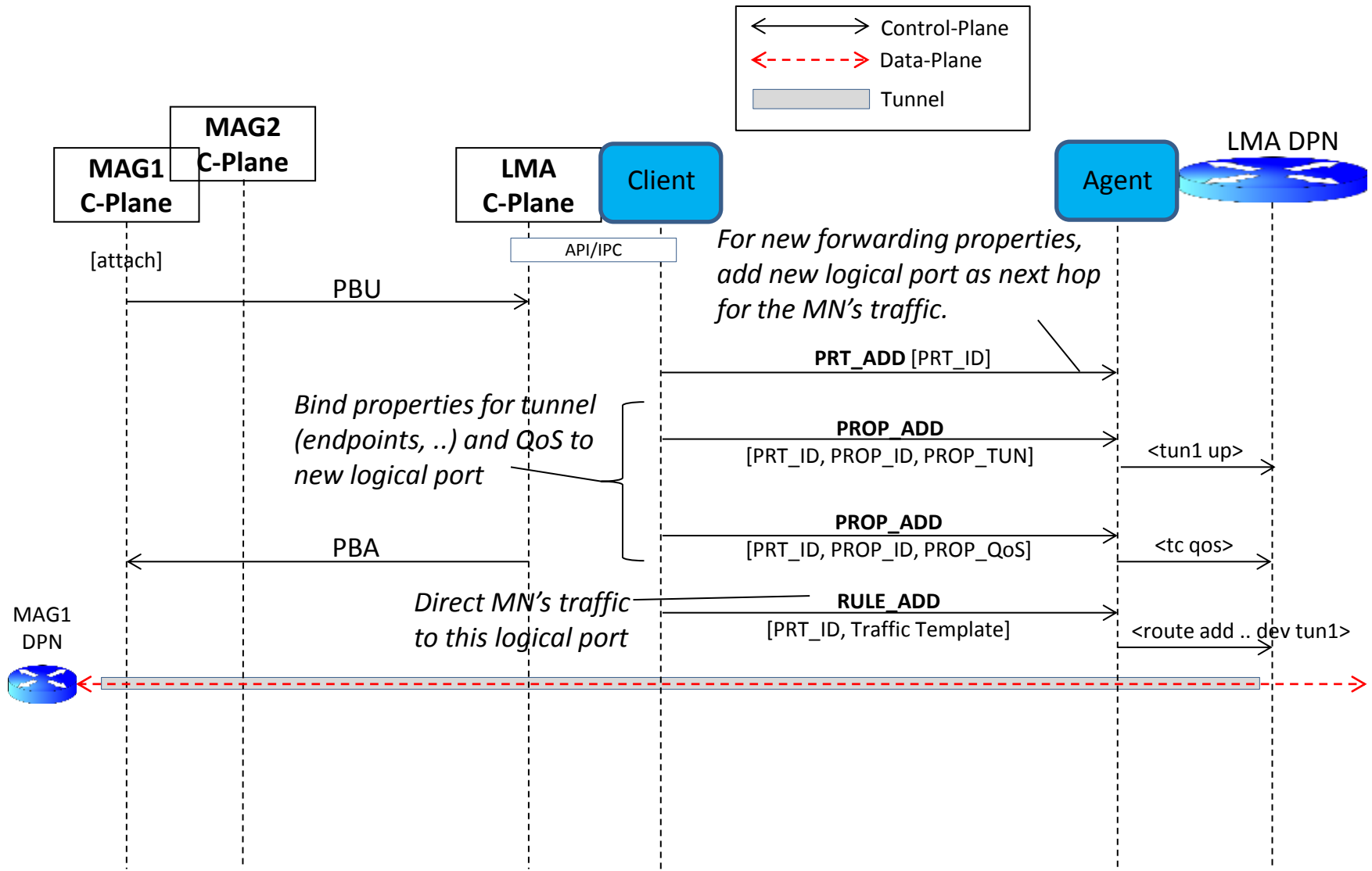
Attribute	Format Clarification	Description
PROP_TUN	[type] [src] [dest]	Property encapsulation; [type] indicates type GRE, GTP-U, IPIP
PROP_REWR	[in_src][in_dst][in_prt] [out_src][out_dst][out_prt]	Property NAT
PROP_QOS		Property QoS
PROP_GW		Property Next Hop

Attributes – Property-specific

Attribute	Format Clarification	Description
IPIP_CONF		IP encapsulation configuration attribute
GRE_CONF	[protocol-type][seq-#][key]	GRE encapsulation configuration attribute
GTP_CONF	[TEID local] [TEID remote] [seq-#]	GTP-U encapsulation configuration attribute

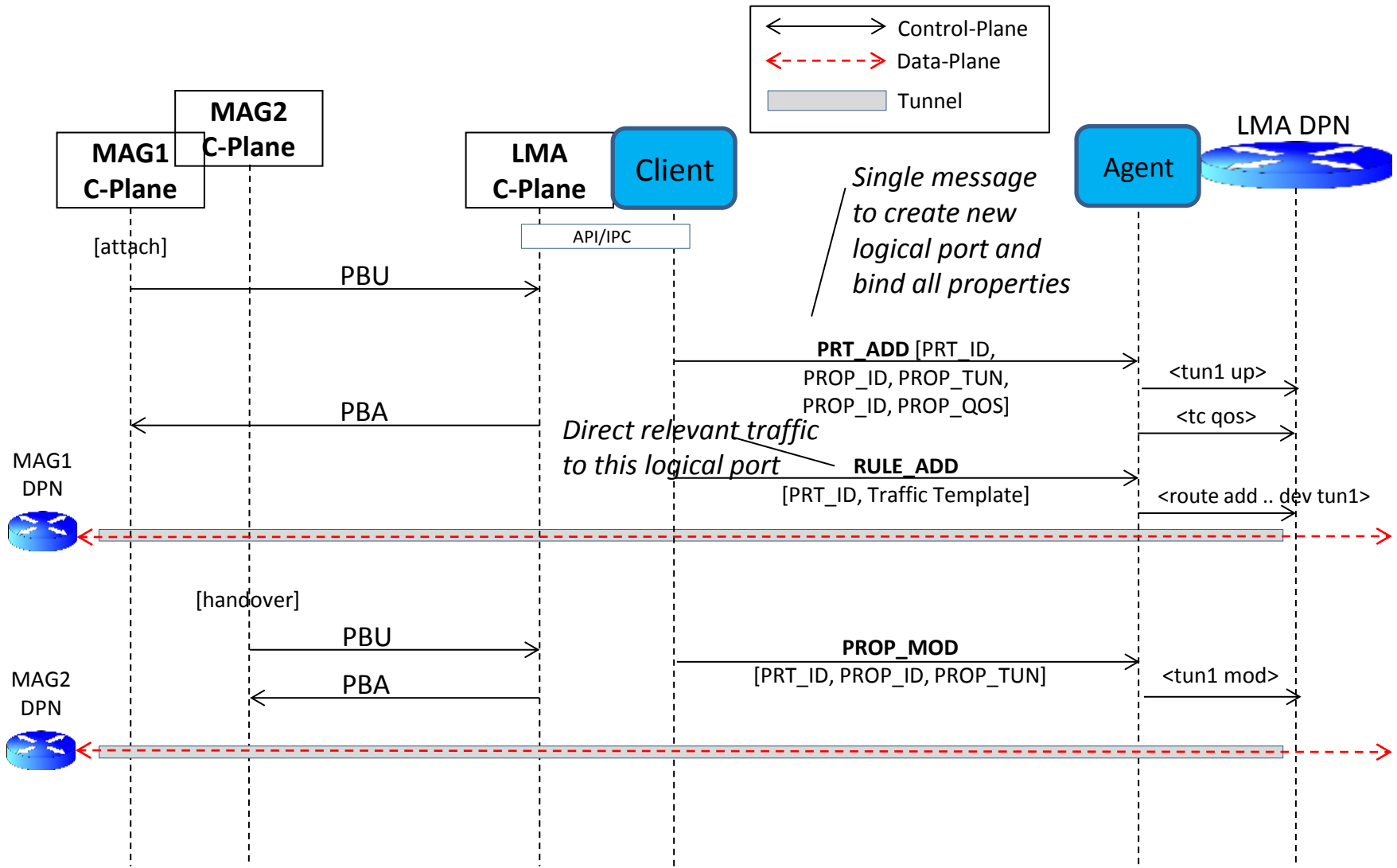
Exemplary Message Sequence (PMIPv6)

Agent co-located with Router – sequential configuration



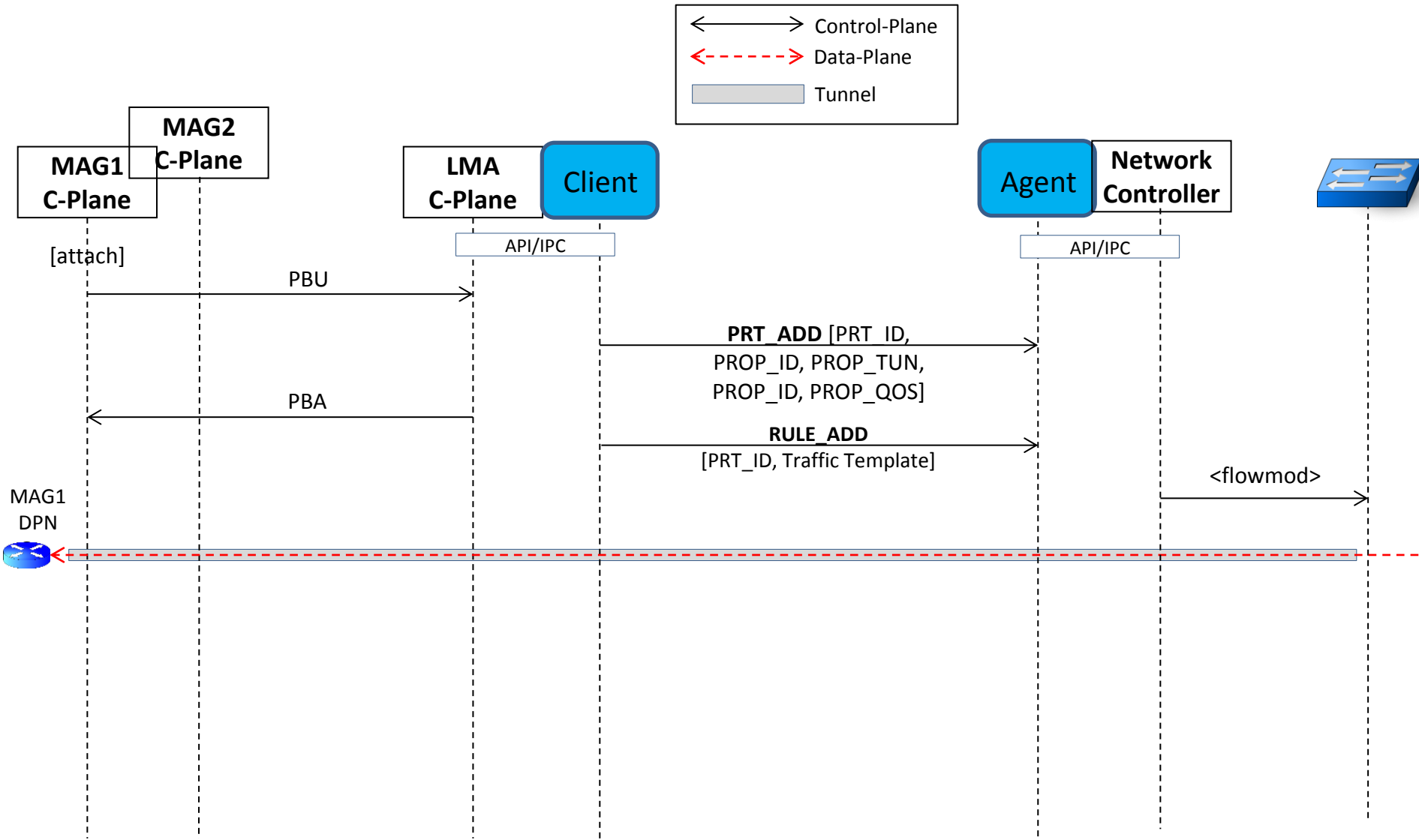
Exemplary Message Sequence (PMIPv6)

Agent co-located with Router – aggregated configuration



Exemplary Message Sequence (PMIPv6)

Agent co-located with Network Controller



Status & Open Items

- ❑ Draft represents WT's proposal
- ❑ Deep-dive to complete some components
 - ❑ Some attributes missing, e.g. for event monitoring, QoS
 - ❑ Details about autonomous operation of Client/Agent and CDSs
 - ❑ Interworking with dependent functions, e.g. discovery of DPNs and meta-data (identifiers, supported functions, load, ..)
- ❑ Adoption of suitable information/data model
 - ❑ Appendix provides experimental Yang model
 - ❑ If we choose information model, we should ensure interoperability above Client-level, e.g. for exposed identifiers, event description, QoS class representation
- ❑ Scope of draft may be tuned (to be discussed)
 - ❑ Focus on mobile traffic forwarding and event reporting (e.g. failures)
 - ❑ Omit QoS for now

Next Steps

- ☐ Converge on open items and progress the draft
- ☐ This is the Work Team's proposal to the WG
- ☐ Does this draft go into the right direction?
- ☐ If yes, adopt as WG document?

ADDITIONAL SLIDES

General deployment of the protocol

- ❑ Protocol for operation between Mobility Control-Plane and Data-Plane
- ❑ Control-Plane comprises instances of Mobility Control functions
- ❑ Data-Plane comprises network components of different kind (switches, routers)
 - ❑ Provisioning of network functions (encapsulation, IP address/port rewrite, QoS, host routing/switching, flow switching/routing, event monitoring)
- ❑ Mobility Control-Plane enforces policies for relevant functions in the Data-Plane
- ❑ Data-Plane exposes the status of scheduled events towards the Control-Plane

