INTERNET-DRAFT
<draft-isoyama-policy-mpls-info-model-00.txt>
Expires January 2001

K. Isoyama
M. Yoshida
NEC Corporation
M. Brunner
A. Kind
J. Quittek
NEC Europe Ltd.
12 July 2000

Policy Framework QoS Information Model for MPLS
<draft-isoyama-policy-mpls-info-model-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document presents an information model for representing QoS policies for Multi-Protocol Label Switching (MPLS). The model is based on the Policy Framework QoS Information Model (QPIM) and the Policy Core Information Model (PCIM). These models are extended with new classes for controlling and managing the life-cycle of Label Switched Paths (LSPs) and for mapping of traffic onto existing LSPs. The control and management of LSP life-cycles includes mainly the creation, update, and deletion of LSPs and the assignment of QoS properties to LSPs.

Even if this document is primarily intended to cover the QoS related MPLS areas, it also covers traffic engineering related policies.

Table of Contents

1. Introduction 2
1.1 Goals 3
2. Information Model Hierarchy 4
2.1 Relation with PCIM 4
2.2 Relation with QPIM 5
2.3 Class Hierarchy 5
3. MPLS and Label Switched Paths 8
3.1 QoS Properties 8
3.2 QoS routing for LSP 8
3.3 Signaling LSPs 9
3.4 Controlling the Signaling 9
3.5 Forward Equivalence Classes (FEC) 9
4. Constructing MPLS Policy Rule 9
4.1 Actions related to MPLS signaling 10
4.2 MPLS classifier actions 10
4.3 DiffServ with MPLS policy rules 11
4.4 MPLS tunneling 11
4.5 MPLS Policy Variables 12
5. Class Definitions 12
5.1 Class mplsPolicyLSP 12
5.2 Class mplsPolicyCRLSP 13
5.3 Class mplsPolicyLSPResvAction 15
5.4 Class mplsPolicyCRLSPResvAction 16
5.5 Class mplsPolicyCRLDPSignalCtrlAction 18
5.6 Class mplsPolicyMPLSPRAction 21
5.7 Class mplsPolicyFEC 21
5.8 Class mplsPolicyFECIP 21
5.9 Class mplsPolicyFECIPDS 22
5.10 Class mplsPolicyCRLSPTrfcProf 23
5.11 Class mplsResourceClass 24
5.12 Class mplsCRLDPResourceClass 25
6. Examples 26
7. Security Considerations 28
8. References 28
9. Author's Addresses 30

1. Introduction

This document presents an information model for representing QoS policies for Multi-Protocol Label Switching (MPLS). The model is based on the Policy Framework QoS Information Model (QPIM) and the Policy Core Information Model (PCIM). These models are extended with new classes for controlling and managing the life-cycle of Label Switched Paths (LSPs) and for mapping of traffic onto existing LSPs. The control and management of LSP life-cycles includes mainly the

creation, update, and deletion of LSPs and the assignment of QoS properties to LSPs.

The informational model described in this draft is independent of a binding to a specific type of repository. A future draft may provide a mapping to a schema for a repository that can be accessed, for instance, via the Lightweight Directory Access Protocol (LDAP).

1.1 Goals

The ultimate goal of policy-based networking is to support the trend away from individual device management, toward managing and controlling a network as a whole [POLICY-FW]. Policy-based networking allows that network elements from different vendors, equipped with different capabilities can be consistently configured according to network policies. Network policies may be, for instance, aligned to the business goals of a company.

MPLS is a combination of switched forwarding with network layer routing. The added value of MPLS is provided by a better price/performance ratio of network layer routing, improved scalability in the network layer, and greater flexibility in the delivery of routing services [MPLS-FW]. These advantages are achieved by label switching: A packet is assigned to a Forwarding Equivalence Class (FEC) when it enters the MPLS network. The FEC is encoded as a label in the packet so that it can then be used at subsequent hops between ingress and egress nodes to determine the forwarding treatment by indexing into a table. All packets belonging to a particular FEC travel the same path through the network.

Typically, information about bindings of labels to FECs is distributed by a label distribution protocol (e.g. LDP, CR-LDP, BGP, RSVP-TE). The use of policies in the context of MPLS is motivated by the need to provide high-level support for the operation of MPLS networks beyond a standard way of label distribution with LDP or other label distribution protocols. An important aim is to provide high-level means for mapping traffic that matches a specific traffic filter onto an LSP with specific QoS characteristics. Such high-level policies could be used, for instance, with DiffServ over MPLS [DS-MPLS].

Wright et al. state requirements for policy-enabled MPLS networks in [REQPMPLS]. These include

- a higher-level MPLS abstraction for specifying network-wide MPLS policies,
- controllability of the LSP Life Cycle,

- consistency with other techniques, such as DiffServ,
- flexibility in LSP admission control, and
- integration with network service objectives.

The model introduced below complies with these requirements. It is also suited to specify MPLS load balancing policies as described in [PBLBMPLS]. The classes introduced in this document focus on the representation of MPLS-related policy actions, including the representation of LSPs, FECs, and CR-LSP traffic profiles. These classes can be instantiated in order to specify the MPLS traffic profile for different kind of traffic entering the MPLS domain. Currently, the classes only reflect traffic profiles according to CR-LSP. However, the class hierarchy is designed such that other means for describing traffic profiles in MPLS (e.g. RSVP) can be added later. Whether LSPs that have been set up already are able to accommodate traffic according to the specified profile, a new LSP has to be created, or the resources in the MPLS domain do not suffice to accommodate the traffic, is transparent with this model.

Even if this document is primarily intended to cover the QoS related MPLS areas, it also covers traffic engineering related policies.

2. Information Model Hierarchy

The model is based on the Policy Framework QoS Information Model (QPIM) [QPIM]. QPIM defines classes for representing QoS policy rules, including QoS policy rule conditions and QoS policy rule actions. The classes specified in QPIM address QoS provisioning using Differentiated Services (DiffServ) as well as QoS control via RSVP for Integrated Services (IntServ). QPIM itself refines the Core Information Model (CIM) [PCIM] which includes generic classes for policy-based networking.

2.1 Relation with PCIM

The object-oriented information model described in PCIM provides generic classes for representing policy information. The model allows to specify network policies in terms of policy rules. A policy rule is an object that stands for a "IF conditions THEN actions" statement. A policy rule is most importantly defined by a list of policy conditions and a list of policy actions.

The QoS model for MPLS refines the notion of policy actions as described in PCIM. New classes are introduced in order to model the following MPLS-related policy actions:

- Generic reservation of a LSP
- Reservation of a LSP with QoS properties or using an CR-LDP Label Request Message
- Decision for CR-LDP Messages
- Mapping of traffic into a LSP

Furthermore, new policy classes are specified directly under the PCIM class Policy. These classes are used because their instances may be reused and shared between several MPLS policies. The new classes are related to:

- CR-LSP traffic profiles
- Representation of LSPs
- Representation of FECs
- Mapping of LSPs to specific network resources

2.2 Relation with QPIM

QPIM defines a condition class to model individual conditions in terms of variable, operator and value [QPIM]. The variable specifies a specific traffic attribute that can be compared with the value using the operation. QPIM lists a set of variables that are typically required on layer 2 and 3. This draft extends the list of predefined variables to access the top label entry on the label stack of an MPLS packet [MPLS-ENC].

2.3 Class Hierarchy

This section presents with Figure 1 the inheritance class hierarchy for the MPLS information model. The classes introduced with PCIM and QPIM are indicated, so that the relation of these classes to the classes defined later in this document is visible.


[unrooted]
Policy (abstract, defined in PCIM)
PolicyGroup (PCIM)
qosPolicyDomain (QPIM)
qosNamedPolicyContainer (QPIM)
PolicyRule (PCIM)
PolicyCondition (PCIM)
PolicyTimePeriodCondition (PCIM)
VendorPolicyCondition (PCIM)
qosPolicySimpleCondition (QPIM)
PolicyAction (PCIM)
VendorPolicyAction (PCIM)
qosPolicyPRAction (QPIM)
qosPolicyRSVPAction (QPIM)
qosPolicyRSVPSignalCtrlAction (QPIM)
qosPolicyRSVPInstallAction (QPIM)
mplsPolicyLSPResvAction (this document)
mplsPolicyCRLSPResvAction (this document)
mplsPolicyCRLDPSignalCtrlAction (this document)
mplsPolicyMPLSPRAction (this document)
qosPolicyPRTrfcProf (QPIM)
qosPolicyRSVPTrfcProf (QPIM)
mplsPolicyCRLSPTrfcProf (this document)
qosPolicyVariable (QPIM)

Figure 1: Inheritance Class Hierarchy

3. MPLS and Label Switched Paths

A general discussion of issues related to MPLS is presented in the framework [MPLS-FW] and architecture [MPLS-ARCH] documents. A brief summary of salient features is provided below to provide background information for the later sections.

MPLS has the concept of a Label Switched Path (LSP) with or without QoS guarantees. A path needs to be setup using some sort of a signaling protocol. Currently under discussion are [LDP], [CRLDP], and [RSVP-TE]. CR-LDP and RSVP-TE support setting up LSPs with QoS guarantees. Furthermore, LSPs can be specified in a explicit or implicit manner. Explicit routed LSPs specify the path the LSP is taking explicitly. Implicit routed LSPs require a routing mechanisms within the network, which determines the path. A mixture of these are loosely routed LSPs, where some nodes are specified others need to be chosen by the signaling protocol from a given set of nodes.

LSPs are the key objects to be modeled in an MPLS information model. The properties of the LSP depend on what kind of LSPs are taken into account (explicit vs implicit, with or without QoS etc.). The explicit route of an explicitly routed LSP may, for instance, be a property of an LSP. Note however, that the labels an LSP gets on links is not a property of the LSP, needed on this level of abstraction.

3.1 QoS Properties

MPLS in its original fashion has no notion of QoS. However, using MPLS for appropriate traffic engineering already enables an ISP to provide better quality to its customers than without MPLS.

On the other hand, an LSP can have QoS properties, which need to be enforced by QoS mechanisms in LSRs. In a recent proposal to support DiffServ over MPLS [DS-MPLS] MPLS is used in conjunction with DiffServ for QoS enforcement. This approach allows a Label Switched Router (LSR) to apply a specified Per-Hop Behavior (PHB) to packets of an LSP.

3.2 QoS routing for LSP

In order to maintain the quality, the LSP setup process is constraint to QoS requirements. However, different possibilities of QoS routing can be applied. First, a central instance can calculate the best route with QoS state information of the whole network and signal the route with the explicit route feature of CR-LSP or any other

configuration mechanism. Second, the route can be calculated in a distributed fashion, during the signaling process. The MPLS model of this document is transparent to the different routing issues.

3.3 Signaling LSPs

Setting up a new LSP requires signaling or configuration mechanisms. Fundamentally two types are possible (1) using a hop-by-hop signaling protocol like LDP, CR-LDP, or RSVP-TE, or (2) configuring the LSP over SNMP or COPS.

How the LSP is setup is also transparent to this draft, even so, we decided to use the CR-LDP traffic profile. The action to be taken from a policy servers point of view is initiating the setup process.

3.4 Controlling the Signaling

The signaling process for setting up a new LSP may be under different policy or resource constraints. Therefore, we introduce the mplsPolicyCRLDPSignallingAction as a mean to control this process with policies. In some network operation environments this will not be needed, if the policy control is applied before the signaling process in the domain.

In others, it is very convenient for the policy-based management system to define policies for the signaling process.

3.5 Forward Equivalence Classes (FEC)

The concept of a FEC specifies, what traffic is mapped into a specific LSP. The specification of FECs can range from very simple, e.g., just the destination IP address, to very complex, e.g., a set of filters including IP addresses, ports, DSCPs etc. In our model of a FEC, we regard a FEC as a property of an LSP. The binding of FECs to LSPs may be performed at or after the LSP setup.

4. Constructing MPLS Policy Rule

Policies in the MPLS domain mainly focus on

- life-cycle management of LSPs, including the signaling, resource control, and admission control etc., and
- mapping of traffic to LSPs, including mapping of DiffServ traffic and tunneling of MPLS traffic over a different LSP for the purpose of traffic engineering.

The QoS information model is extended mainly with new objects to be stored in a directory or database, such as an LSP, a FEC etc, and with policy actions controlling the signaling process, initiating the LSP setup, and mapping traffic into LSPs.

This section discusses the policy actions, objects, and variables that are necessary to construct MPLS Policy Rules.

4.1 Actions related to MPLS signaling

Signaling in MPLS comprises setting up, releasing and updating an LSP along a number of LSRs.

It should be possible to control the initiation and the execution of these processes by policies. For some signaling protocols, e.g. the Label Distribution Protocol (LDP), the signaling is initiated by a LSR. However, even in this case the start of the signaling process may be triggered by a policy.

Therefore, the draft specifies policy actions to initiate the setup, update, or release of an LSP (mplsPolicyLSPResvAction, mplsPolicyCRLSPResvAction), as well as a policy action controlling the LSP setup process (mplsPolicyCRLDPSignalingAction).

4.2 MPLS classifier actions

The mapping of traffic to LSPs is performed at the edge LSR of an MPLS domain. The MPLS framework document describes different examples of traffic granularities, that can be mapped to LSPs. In the MPLS context, the traffic mapped is often referred to as Forward Equivalence Class (FEC), which forwards a group of packets in the same manner. This draft defines the mplsPolicyPRAction class for specifying the mapping.

Since there are so many possibilities, the definition of all kinds of FECs is for further study. In this document, we propose two kinds of FECs, one specified through the IP destination address (mplsPolicyFECIP) and the other through the IP destination address together with the DSCP value in DiffServ (mplsPolicyFECIPDS). However, note that the FEC class is mainly used to store a FEC together with an LSP.

The definition of the mapping between FECs and LSPs, is specified in the policy rule condition. In the policy condition any kind of filters can be specified. See QPIM for a deeper discussion of the specification of filters.

4.3 DiffServ with MPLS policy rules

MPLS with DiffServ capable Label Switched Routers extends also the policy rule set in order to perform the mapping from DiffServ Ordered Aggregates to MPLS paths. This mainly includes

- the mapping of DiffServ specific traffic into LSPs on the edge LSRs, and
- the mapping of LSRs to Per-Hop Behaviors on the core LSRs

4.3.1 Mapping of DiffServ specific traffic into LSPs

The mapping consists of a specification of a DiffServ-MPLS specific FEC and a reference to an LSP object. The FEC includes the specification of the DSCP value and the destination IP address of a packet. Together with the mplsPolicyPRAction this allows a policy manager to specify the mapping of DiffServ classes to LSPs.

4.3.2 Assignment of LSP to PHB

At core LSRs, arriving packets are forwarded according to the MPLS label. As soon as DiffServ-enabled LSRs are deployed, packets get a pre-defined per-hop behavior. Most likely the assignment of a packet with MPLS label X to behavior class Y has been configured already at the LSP setup time. However, the mapping could be policy controlled.

4.4 MPLS tunneling

In MPLS, a mechanism for label stacking is defined, and the label stacking is mainly used for tunneling/aggregating MPLS traffic for traffic engineering purposes. Since the LSP is defined very open in this draft, no differentiation between an LSP used as an MPLS tunnel and an LSP used as an IP path is needed. However, the mapping of traffic into an LSP is different. In the first case, the mapping is in the MPLS domain, meaning MPLS packets with specified labels are mapped to an MPLS tunnel. In the IP case, the mapping from IP traffic to an LSP needs to be defined.

This document does not specify special class for MPLS tunneling. However, tunneling can be achieved by specifying mplsPolicyPRAction for tunneling points. In order to specify the mapping of an LSP into another LSP, we only define the MPLS label variable in the following section.

4.5 MPLS Policy Variables

The QoS policy information model specifies a set of pre-defined variables to support a set of fundamental QoS terms that are commonly used in policy conditions [QPIM]. In order to specify meaningful policy rules in the MPLS domain, we need to extend the set of pre-defined variables with MPLS specific variables.

The following table defines the pre-defined variables for MPLS and its local binding.

Variable name	Logical binding
MPLSLabel	The MPLS label of the flow. Compared to the MPLS label in the MPLS shim header field or the combined VPI/VCI in ATM.
MPLSExp	The MPLS Exp field of the flow. Compared to the MPLS Experimental field in the MPLS shim header.

Table 1: Pre-defined Variable Names and their Bindings

Both variables can be of class type qosPolicyIntegerValue or qosPolicyBitStringValue.

5. Class Definitions

5.1 Class mplsPolicyLSP

This class represents properties of a general LSP. It contains only a minimal set of properties. It includes a unique LSP identifier and a Forward Equivalence Class (FEC). The LSP identifier is composed of the ingress LSR router ID (or any of its own IP addresses) and a locally unique LSP ID. See also the LSPID TLV in [CRLDP]. The identifier is used to refer to an LSP in the whole MPLS network. The FEC property stores the FECs currently mapped to this LSP. The FEC may be empty in the LSP setup phase, and filled with FEC in the process of mapping different kind of FEC to this LSP.

NAME	mplsPolicyLSP
DERIVED FROM	Policy (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mpLocalLSPID, mpIngressLSRRouterID, mpFEC

5.1.1 The Property mpLocalLSPID

This property is an integer that indicate the LSPID which is unique with reference to Ingress LSR.

NAME	mpLocalLSPID
SYNTAX	Integer (must be in the range 0-65535)

5.1.2 The Property mpIngressLSRRouterID

This property represents Router ID of the Ingress LSR of the LSP. Typically the router id is specified by one of its own IP addresses.

NAME	mpIngressLSRRouterID
SYNTAX	String
FORMAT	IPv4address | hostname | number

5.1.3 The Property mpFEC

This property specifies the Forward Equivalence Class (FEC) of the LSP. The property contains a reference to an object of the mplsPolicyFEC class. The FEC can be specified in many different ways depending on the area, MPLS is applied. For example, in the case of DiffServ with MPLS, the FEC consist of an IP destination address (prefix) and a set of DSCPs.

NAME	mpFEC
SYNTAX	Reference to a mplsPolicyFEC object

5.2 Class mplsPolicyCRLSP

This class represents properties of a CR-LSP, which is typically established using CR-LDP [CRLDP]. The explicit route property specifies the path an LSP takes in the setup phase, or stores the path an LSP setup process has chosen. It refers to the Explicit Route TLV of CR-LDP. Furthermore, the CR-LSP contains a traffic profile description modeled after the CR-LDP traffic model. As soon as resources are associated with a path, it is very convenient from the management perspective to have different resource classes in a network. So the resource class property specifies from which resource class the CR-LSP draws its resources. The route pinning property defines whether the part of the route, which is loosely specified, is pinned after the setup or if the network is allowed to re-route the loosely routed part of the LSP. The holding priority is the counter part to the set priority specified in the mplsPolicyCRLSPResvAction.

Setting up a new CR-LSP with a higher set priority than the hold priority of existing LSPs, results in releasing the existing CR-LSP. However, decisions in case of resource shortage, can also be made by the policy server through the possibility of the mplsPolicyCRLDPSignalCtrlAction. Finally, the LSP type is mainly introduced to easier distinguish explicit routed from implicit routed LSPs and LSPs with or without QoS.

NAME	mplsPolicyCRLSP
DERIVED FROM	mplsPolicyLSP
ABSTRACT	False
PROPERTIES	mpExplicitRoute, mpCRLSPTrfcProf, mpResourceClass, mpCRLSPRoutePinning, mpCRLSPHoldPrio, mpLSPType

5.2.1 The Property mpExplicitRoute

This property represents the explicit routing path of the CR-LSP. The explicit path is specified through a list of routers (IP address or hostname) an LSP is traversing in the given order.

NAME	mpExplicitRoute
SYNTAX	String
FORMAT	ordered list of IPv4address | IPv6address | hostname

5.2.2 The Property mpCRLSPTrfcProf

This property is a reference to a mplsPolicyCRLSPTrfcProf object, which defines a CR-LSP traffic parameter according to the Traffic Parameters TLV of [CRLDP].

NAME	mpCRLSPTrfcProf
SYNTAX	Reference to a mplsPolicyCRLSPTrfcProf object

5.2.3 The Property mpResourceClass

This property represents the resource class of the LSP as a reference to a mplsResourceClass object. We propose to specify the property as a reference of an abstract resource class object, because different types, aggregation schemas etc. of resource classes are possible. So this design enables an implementation to extend the resource classes to application specific types without changing the CRLSP class.

NAME	mpResourceClass
SYNTAX	Reference to an mplsResourceClass object.

5.2.4 The Property mpCRLSPRoutePinning

This property indicates whether route pinning of the CR-LSP is requested or not.

```
NAME          mpCRLSPPRoutePinning
SYNTAX Integer (ENUM) - {"NotRequested"=0; "Requested"=1}
```

5.2.5 The Property mpCRLSPHoldPrio

This property represents the holding priority of the CR-LSP which is already set up. A newly setup LSP is only allowed to take the resources of this LSP if its set priority is higher than the hold priority.

```
NAME          mpCRLSPHoldPrio
SYNTAX        Integer (must be in the range 0-255)
```

5.2.6 The property mpLSPType

An LSP can be defined in different ways. It is possible to have explicit or implicit routing with or without QoS traffic profile. This property defines which sort of LSP is defined out of the four combinations. A implicit routed LSP has an empty mpExplicitRoute property. Where as an LSP without QoS has an empty mpCRLSPTrfcProf property.

```
NAME          mpLSPType
SYNTAX        Integer (ENUM) - {
                    "implicit/no QoS"=1;
                    "implicit/QoS"=2;
                    "explicit/no QoS"=3;
                    "explicit/QoS"=4;
              }
```

5.3 Class mplsPolicyLSPResvAction

This class defines a generic LSP reservation action. The action initiates the setup, update, or release of an LSP given in the mpLSP property. The FEC specification in the LSP may determine the route of the LSP. The route is implicitly chosen according to the IP routing tables. No constraints can be applied to this action. See the next section for constraint-based routing of LSPs. The class definition is as follows:

NAME	mplsPolicyLSPResvAction
DERIVED FROM	PolicyAction (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mpLSP, mpMode

5.3.1 The Property mpLSP

This property is a reference to a mplsPolicyLSP object, which defines an LSP. The attribute is defined as follows:

NAME	mpLSP
SYNTAX	Reference to a mplsPolicyLSP object

5.3.2 The Property mpMode

The mpMode property specifies whether the action is a setup, release, or update action of an LSP.

```
NAME          mpMode
SYNTAX        Integer (ENUM) - {
                     "LSPSetup"=0;
                     "LSPUpdate"=1;
                     "LSPRelease"=2;
              }
```

5.4 Class mplsPolicyCRLSPResvAction

This class defines a CR-LSP reservation action using CR-LDP Label Request Messages. The LSP to be setup is specified in the mpCRLSP property, which contains all properties associated with the CR-LSP. The set priority property specifies whether an existing LSP may be preempted in order to setup the new more important LSP. The property is mapped to the SetPrio field of the Preemption TLV [CRLDP]. All negotiable properties specify whether the traffic parameter is negotiable or not. The properties are mapped to the flags field of the Traffic Parameter TLV of [CRLDP]. The class definition is as follows:

NAME	mplsPolicyCRLSPResvAction
DERIVED FROM	PolicyAction (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mpCRLSP, mpCRLSPsetPrio, mpPDRNegotiable, mpPBSNegotiable, mpCDRNegotiable, mpCBSNegotiable, mpEBSNegotiable, mpWeightNegotiable

5.4.1 The Property mpCRLSP

This property is a reference to a mplsPolicyCRLSP object, which defines the CR-LSP to be setup. The property is defined as follows:

NAME	mpCRLSP
SYNTAX	Reference to a mplsPolicyCRLSP object

5.4.2 The Property mpCRLSPSetPrio

This property represents the setting priority of the CR-LSP. The property is defined as follows:

NAME	mpCRLSPSetPrio
SYNTAX	Integer (must be in the range 0-255)

5.4.3 The Property mpPDRNegotiable

This property indicates whether Peak Data Rate of the CR-LSP is negotiable or not. The attribute is defined as follows:

NAME	mpPDRNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.4.4 The Property mpPBSNegotiable

This property indicates whether Peak Burst Size of the CR-LSP is negotiable or not. The attribute is defined as follows:

NAME	mpPBSNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.4.5 The Property mpCDRNegotiable

This property indicates whether Committed Data Rate of the CR-LSP is negotiable or not. The attribute is defined as follows:

NAME	mpCDRNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.4.6 The Property mpCBSNegotiable

This property indicates whether Committed Burst Size of the CR-LSP is

negotiable or not. The attribute is defined as follows:

NAME	mpCBSNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.4.7 The Property mpEBSNegotiable

This property indicates whether Excess Burst Size of the CR-LSP is negotiable or not. The attribute is defined as follows:

NAME	mpEBSNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.4.8 The Property mpWeightNegotiable

This property indicates whether Weight of the CR-LSP is negotiable or not. The attribute is defined as follows:

NAME	mpWeightNegotiable
SYNTAX	Integer (ENUM) - {"NotNegotiable"=0; "Negotiable"=1}

5.5 Class mplsPolicyCRLDPSignalCtrlAction

This class defines a policy action to be applied to CR-LDP messages. The message type property defines which messages are concerned. The mpSendNotification property specifies whether a notification should be sent, in case the notification is sent, the mpErrCode property specifies which one. Furthermore, the replace properties specify whether the traffic profile of the CR-LDP message is changed.

NAME	mplsPolicyCRLDPSignalCtrlAction
DERIVED FROM	PolicyAction (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mpCRLDPMessageType, mpSendNotification, mpSendRelease, mpErrCode mpReplacePDR, mpReplacePBS, mpReplaceCDR, mpReplaceCBS, mpReplaceEBS, mpReplaceWeight

5.5.1 The Property CRLDPMessageType

This property is an enumerated integer and defines different values that limit the scope of the action to be enforced to specific types of CR-LDP Messages. The attribute is defined as follows:

```
NAME          CRLDPMessageType
SYNTAX        Integer (ENUM) - {
                     "LabelMapping"=0;
                     "LabelRequest"=1;
                     "LabelWithdraw"=2;
                     "LabelRelease"=3;
                     "LabelAbortRequest"=4;
                     "Notification"=5;
                 }
```

5.5.2 The Property mpSendNotification

This property is an enumerated integer and controls the generation of CR-LDP Notification Message. The attribute is defined as follows:

```
NAME          mpSendNotification
SYNTAX        Integer (ENUM) - {No=0, Yes=1}
```

5.5.3 The Property mpSendRelease

This property is an enumerated integer and controls the generation of CR-LDP Release Message. The attribute is defined as follows:

```
NAME          mpSendRelease
SYNTAX        Integer (ENUM) - {No=0, Yes=1}
```

5.5.4 The Property mpErrCode

This property specifies the error code in case the mpSendNotification property has the value 1=yes. In the case, the sent notification property is no, this property is undefined. The error codes are the ones specified in LDP and CR-LDP.

```
NAME          mpErrCode
SYNTAX        Integer
```

5.5.5 The Property mpReplacePDR

This property is a non-negative integer that replaces the Peak Data Rate value in a CR-LDP message. The attribute is defined as follows:

```
NAME          mpReplacePDR
SYNTAX        Integer (must be non-negative)
```

5.5.6 The Property mpReplacePBS

This property is a non-negative integer that replaces the Peak Burst Size value in a CR-LDP message. The attribute is defined as follows:

NAME	mpReplacePBS
SYNTAX	Integer (must be non-negative)

5.5.7 The Property mpReplaceCDR

This property is a non-negative integer that replaces the Committed Data Rate value in a CR-LDP message. The attribute is defined as follows:

NAME	mpReplaceCDR
SYNTAX	Integer (must be non-negative)

5.5.8 The Property mpReplaceCBS

This property is a non-negative integer that replaces the Committed Burst Size value in CR-LDP message. The attribute is defined as follows:

NAME	mpReplaceCBS
SYNTAX	Integer (must be non-negative)

5.5.9 The Property mpReplaceEBS

This property is a non-negative integer that replaces the Excess Burst Size value a in CR-LDP message. The attribute is defined as follows:

NAME	mpReplaceEBS
SYNTAX	Integer (must be non-negative)

5.5.10 The Property mpReplaceWeight

This property is a non-negative integer that replaces the Weight value in a CR-LDP message. The attribute is defined as follows:

NAME	mpReplaceWeight
SYNTAX	Integer (must be in the range 0-255)

5.6 Class mplsPolicyMPLSPRAction

The mplsPolicyMPLSPRAction class specifies the mapping of traffic into a Label Switched Path (LSP). What kind of traffic is mapped is transparent to the action on this level. The mapping is performed in an edge LSR in the case of IP over MPLS or in the core LSRs in the case of MPLS tunneling. Both cases can be specified with this action. However, the condition of the policy rule may differ. Note that the traffic to be mapped is specified in the condition of the policy rule. The traffic specification may be plain IP style, it may contain DSCP values, or it may be given by MPLS labels. The LSP is given as a reference to an LSP object, which specifies an already setup LSP, the traffic should take. Note that we do not need to specify any label stacking operation on this abstraction level.

The class is defined as follows:

NAME	mplsPolicyMPLSPRAction
DERIVED FROM	PolicyAction (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mplsSetLSP

5.6.1 The property mplsSetLSP

This property contains a reference to an object, which defines an LSP in the network to which the traffic is mapped. The property is defined as follows:

NAME	mplsSetLSP
SYNTAX	Reference to a mplsLSP object

5.7 Class mplsPolicyFEC

The mplsPolicyFEC class is an abstract class specifying the Forward Equivalence Class of an LSP. The class is abstract in order to be open for different kinds of FECs. The FECs may differ depending on the application of MPLS.

NAME	mplsPolicyFEC
DERIVED FROM	Policy (defined in [PCIM])
ABSTRACT	True
PROPERTIES	

5.8 Class mplsPolicyFECIP

This class defines a FEC based on the IP destination address. The class definition is as follows:

NAME	mplsPolicyFECIP
DERIVED FROM	mplsPolicyFEC
ABSTRACT	False
PROPERTIES	mpFECType, mpFECValue

5.8.1 The Property mpFECType

This property is an enumerated integer that defines the type of the FEC. Even if the Wildcard FEC type is used only in the Label Withdraw and Label Release Messages of the LDP specification, we included the type in this property. The wildcard type indicates that the withdraw/release is applied to all FECs associated with the label. The attribute is defined as follows:

NAME	mpFECType
SYNTAX	Integer (ENUM) - {Wildcard=1,Prefix=2,HostAddress=3}

5.8.2 The Property mpFECValue

This property is a set IP addresses that define the FEC of this LSP. The attribute is defined as follows:

NAME	mpFECValue
SYNTAX	IPv4Address | IPv6Address | *

5.9 Class mplsPolicyFECIPDS

The class mplsPolicyFECIPDS is derived from the mplsPolicyFECIP class. It specifies a FEC with IP destination address and DSCP value. This kind of FEC is usually used in the case of mapping from DiffServ traffic specified with IP destination address and DSCP value to an LSP.

NAME	mplsPolicyFECIPDS
DERIVED FROM	mplsPolicyFECIP
ABSTRACT	False
PROPERTIES	mpDSCP

5.9.1 The property mpDSCP

The property specifies a DSCP value or a set of DSCP values.

NAME	mpDSCP
SYNTAX	Integer | Set of Integer (in the range 0..63, inclusive)

5.10 Class mplsPolicyCRLSPTrfcProf

This class represents a CR-LSP traffic parameter as specified in [CRLDP]. The value of CR-LSP traffic profiles corresponds to the Traffic Parameter TLV carried in CR-LDP messages.

NAME	mplsPolicyCRLSPTrfcProf
DERIVED FROM	Policy (defined in [PCIM])
ABSTRACT	False
PROPERTIES	mpCRLSPFrequency, mpCRLSPWeight mpCRLSPPeakDataRate, mpCRLSPPeakBurstSize, mpCRLSPCommittedDataRate, mpCRLSPCommittedBurstSize, mpCRLSPExcessBurstSize

5.10.1 The Property mpCRLSPFrequency

This property specifies at what granularity the CDR allocated to the CR-LSP is made available.

NAME	mpCRLSPFrequency
SYNTAX	Integer(ENUM)-{"Unspecified"=0;"Frequent"=1;"VeryFrequent"=2}

5.10.2 The Property mpCRLSPWeight

This property represents the CR-LSP's relative share of the possible excess bandwidth above its committed rate.

NAME	mpCRLSPWeight
SYNTAX	Integer (must be in the range 0-255)

5.10.3 The Property mpCRLSPPeakDataRate

This property is a non-negative integer that defines the token rate parameter of peak rate token bucket, measured in byte per second. The attribute is defined as follows:

NAME	mpCRLSPPeakDataRate
SYNTAX	Integer (must be non-negative)

5.10.4 The Property mpCRLSPPeakBurstSize

This property is a non-negative integer that defines the token bucket size parameter of peak rate token bucket, measured in byte. The attribute is defined as follows:

NAME	mpCRLSPPeakBurstSize
SYNTAX	Integer (must be non-negative)

5.10.5 The Property mpCRLSPCommittedDataRate

This property is a non-negative integer that defines the token rate parameter of committed data rate token bucket, measured in byte per second. The attribute is defined as follows:

NAME	mpCRLSPCommittedDataRate
SYNTAX	Integer (must be non-negative)

5.10.6 The Property mpCRLSPCommittedBurstSize

This property is a non-negative integer that defines the token bucket size parameter of committed data rate token bucket, measured in byte. The attribute is defined as follows:

NAME	mpCRLSPCommittedBurstSize
SYNTAX	Integer (must be non-negative)

5.10.7 The Property mpCRLSPExcessBurstSize

This property is a non-negative integer that defines the token bucket size parameter of excess burst size, measured in byte. The attribute is defined as follows:

NAME	mpCRLSPExcessBurstSize
SYNTAX	Integer (must be non-negative)

5.11 Class mplsResourceClass

The class defines a resource class, which is defined for the whole MPLS domain. The resource class is specified abstract in order to allow different resource classes to be derived. See the next section for the specification of the CR-LDP specific resource class. The concept of resource classes is regarded a powerful abstraction from a traffic engineering perspective [RFC2702].

NAME	mplsResourceClass
DERIVED FROM	Policy (defined in [PCIM])
ABSTRACT	True

5.12 Class mplsCRLDPResourceClass

MplsCRLDPResourceClass defines a specific resource class, which is defined in an MPLS domain using CR-LDP for signaling purposes. The mpCRLDPRC property contains an integer specifying the resource class according to the resource class TLV in [CRLDP].

NAME	mplsCRLDPResourceClass
DERIVED FROM	mplsResourceClass
ABSTRACT	False
PROPERTY	mpCRLDPRC

5.12.1 The property mpCRLDPRC

The property defines an integer to represent a resource class. The integer can be easily mapped to the resource class TLV in CR-LDP.

NAME	mpCRLDPRC
SYNTAX	Integer (must be non-negative)

6. Examples

This section provides an example that shows how the classes defined above as part of the QoS information model for MPLS may be used in a typical policy scenario. For the example scenario we assume an MPLS network and two hosts (A and B) outside the MPLS domain (see Figure 2).

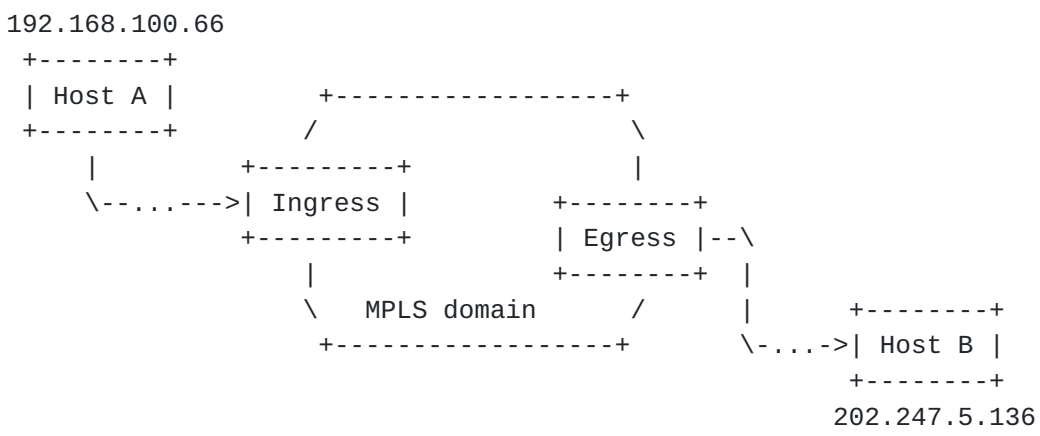

Figure 2: Example MPLS Network

Furthermore, we assume for the scenario that traffic from A to B should be given a committed data rate of 10Mb/s within the MPLS domain. A corresponding policy rule could be given for the MPLS Ingress node:

```
IF SourceIP=192.168.100.66 AND
   DestinationIP=202.247.5.136
THEN
   Provide LSP X with profile.committedDataRate=10Mb/s AND
   Map traffic to be forwarded along LSP X AND
   Shape traffic to 10Mb/s
```

The policy rule as written in a pseudo policy language above can be represented in a policy system as a structured object (see Figure 3).

Figure 3: Representation of the example policy rule

The policy object illustrated in Figure 3 is built from policy classes defined in CIM, QPIM and this document. The root of the policy object is a rule object that is associated with two instances

of class qosPolicySimpleCondition (1a and 1b). The condition objects together specify a traffic filter so that the rule applies only to traffic from source IP address 192.168.100.66 and destination IP addresses 202.247.5.136. The rule object is furthermore associated with a list of instances of class PolicyAction.

The first policy action object (2) is an instance of mplsPolicyCRLSPResvAction and specifies a CR-LSP reservation with a CR-LDP Label Request Message. The path is most importantly defined by a reference to an instance of mplsPolicyCRLSP. Along with other properties, this LSP instance is described by a traffic profile object. The profile object is, in this case, an instance of class mplsPolicyCRLSPTrfcProf that is initialized with a committed data rate of 10Mb/s.

The second policy action (3) specifies that the traffic characterized by the policy rule conditions should be mapped to the LSP that has been reserved by the previous action. This connection may, for instance, be realized in a DiffServ context with a DSCP value that is used for traffic that matches the filter defined by the policy rule condition objects. The same DSCP value can then be used for mapping the traffic to the reserved LSP.

The third policy action object (4) is an instance of qosPolicyPRAction that represents a DiffServ action to be applied on a (group of) flow(s). This may include marking, dropping, policing and shaping. For the sample policy rule above the traffic profile property of the qosPolicyPRAction instance describes a traffic profile that has to be considered when entering the MPLS domain. The object, thus, represents a shaper with a given rate of 10Mb/s.

The example policy given here illustrates only some aspects of the potential of policy-based networking for MPLS. For the sake of brevity, a fairly simple policy rule is chosen here. Also, not all properties are shown in the object structure representing the example policy rule.

7. Security Considerations

The security considerations for this document are the same as those of the [PCIM] and are not further addressed in this version of the draft.

8. References

[PCIM] B. Moore, E. Ellesson, J. Strassner, "Policy Core Information

Model -- Version 1 Specification", Internet Draft,
work in progress, <draft-ietf-policy-core-info-model-06.txt>,
May 2000

[QPIM] Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, "Policy
Framework QoS Information Model", Internet draft,
work in progress, <draft-ietf-policy-qos-info-model-01.txt>,
April 2000

[LDP] L. Anderson, P. Doolan, N. Feldman, A. Fredette, B. Thomas,
"LSP Specification", Internet Draft, work in progress,
<draft-ietf-mpls-ldp-08.txt>, June 2000.

[CRLDP] B. Jamoussi, "Constraint-Based LSP Setup using LDP",
Internet Draft, work in progress,
<draft-ietf-mpls-cr-ldp-03.txt>, March 2000

[RSVP-TE] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow,
V. Srinivasan, "RSVP-TE: Extensions to RSVP for LSP Tunnels",
work in progress, <draft-ietf-mpls-rsvp-lsp-tunnel-05.txt>,
Februar 2000.

[DS-MPLS] F. LeFaucher, L. Wu, B. Davie, S. Davari, P. Vaanenen,
R. Krishnan, P. Cheval, J. Heinanen, "MPLS Support of
Differentiated Services", Internet Draft, work in progrss,
<draft-ietf-mpls-diff-ext-05.txt>, June 2000

[MPLS-ARCH] E.Rosen, A.Viswanathan, R.Callon, "Multiprotocol Label
Switching Architecture", Internet Draft, work in progress,
<draft-ietf-mpls-arch-06.txt>, August 1999.

[MPLS-FW] R.Callon, P.Doolan, N.Feldman, A.Freddette, G.Swallow,
A.Viswanathan, "A Framework for MPLS", Internet Draft,
work in progress, <draft-ietf-mpls-framework-05.txt>,
September 1999.

[MPLS-ENC] E. Rosen, Y. Rekhter, D. Tappan, D. Farinacci,
G. Fedorkow, T. Li, A. Conta, "MPLS Label Stack Encoding",
work in progress, <draft-ietf-mpls-label-encaps-07.txt>,
September 1999.

[REQPMPLS] S. Wright, S. Herzog, F. Reichmayer, R. Jaeger,
"Requirements for Policy Enabled MPLS", work in progress,
<draft-wright-policy-mpls-00.txt>, March 2000.

[PBLBMPLS] S. Wright, F. Reichmeyer, R. Jaeger, M. Gibson,
"Policy-Based Load Balancing in Traffic-Engineered MPLS
Networks", Internet Draft, work in progress,

  <draft-wright-mpls-te-policy-00.txt>, June 2000.

[POLICY-FW] M. Stevens, W. Weiss, H. Mahon, B. Moore, J. Strassner, G. Waters, A. Westerinen, J. Wheeler, "Policy Framework", work in progress, <draft-ietf-policy-framework-00.txt>, September 1999.

[RFC2702] D. Awduche, J. Malcom, J. Agogbua, M. O'Dell, J. McManus, "Requirements for Traffic Engineerring over MPLS", RFC 2702, September 1999.

9. Authors' Addresses

Kazuhiko Isoyama
NEC Corporation
Development Laboratories
1131, Hinode, Abiko, Chiba, 270-1198, JAPAN
Phone: +81 471-85-6738
Fax: +81 471-85-6841
Email: iso@ptl.abk.nec.co.jp

Makiko Yoshida
NEC Corporation
Development Laboratories
1131, Hinode, Abiko, Chiba, 270-1198, JAPAN
Phone: +81 471-85-6738
Fax: +81 471-85-6841
Email: myoshida@ptl.abk.nec.co.jp

Marcus Brunner
NEC Europe Ltd.
C&C Research Laboratories
Adenauerplatz 6
D-69115 Heidelberg, Germany
Phone: +49 (0)6221 905110
Fax: +49 (0)6221 9051155
Email: brunner@ccrle.nec.de

Andreas Kind
NEC Europe Ltd.
C&C Research Laboratories
Adenauerplatz 6
D-69115 Heidelberg, Germany
Phone: +49 (0)6221 905110
Fax: +49 (0)6221 9051155

Email: ak@ccrle.nec.de

Juergen Quittek
NEC Europe Ltd.
C&C Research Laboratories
Adenauerplatz 6
D-69115 Heidelberg, Germany
Phone: +49 (0)6221 905110
Fax: +49 (0)6221 9051155

Email: quittek@ccrle.nec.de