

Terminology for describing network policy and services

Specification

<[draft-weiss-policy-device-qos-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/lid-abstracts.txt>

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

Abstract

The purpose of this draft is to define an information model that describes the attributes common to the forwarding characteristics of Differentiated Services QoS and RSVP. Once this information model is defined, then a separate draft can be written to refine the core information model defined in the Policy Framework working group to model policies that control the configuration of DS-capable network devices. This approach enables the definition of policies that can be used to define Services for DiffServ-capable devices and networks.

The approach taken enables a common set of attributes to be defined that can be used to model Differentiated Services QoS classes at the

INTERNET-DRAFT

Terminology for policy and services

June 25, 1999

behavioral level. Vendors can then map these attributes, either directly or using a proxy agent like a PIB, to their own device-specific implementations.

This draft also concentrates on separating the concepts of state and configuration. Configuration attributes are used to describe the desired state of the device, whereas state attributes reflect the actual state of the device. Both state as well as configuration attributes are necessary in order to model and manage QoS at the device level.

In addition, this draft derives the QoS schema from the core Networks schema defined in the DMTF rather than the core Policy schema. In order to support broader policies that can encompass not only QoS, but also security, address management, network topology management, and routing policies to name a few, it makes more sense to derive the attributes from a schema that is already modeling these devices and services rather than reinventing them under the umbrella of the policy schema.

Finally, this draft considers various aggregation methods for describing groups of devices and groups of interfaces that require a common configuration.

A future iteration of this draft will expand the information model to include modeling and managing the signaling characteristics of RSVP. A separate draft will map the information model presented in this draft to a form suitable for implementation in a directory that uses LDAP as its access protocol.

INTERNET-DRAFT Terminology for policy and services June 25, 1999

Table of Contents

Status of this memo	1
Copyright Notice	1
Abstract	1
Definition of Key Word Usage	2
Table of Contents	2

[1.](#) Introduction

[2.](#) Approach

[2.1](#) Common Needs Of DiffServ and RSVP

[2.2](#) Specific Needs of DiffServ

[2.3](#) Specific Needs of RSVP

[3.](#) Methodology

[3.1](#) Level of Abstraction for Expressing QoS Policies

[3.2](#) Level of Abstraction for Defining QoS Attributes and Classes

[3.3](#) Characterization of QoS Attributes

[3.4](#) Identifying Common Shared Policies

[3.5](#) QoS Schema Derivation

[3.6](#) Attribute Representation

[4.](#) The Class Hierarchy

[4.1](#) Structure of the Class Hierarchy

[4.2](#) Class and Attribute Definitions

[4.2.1](#) ManagedSystemElement

[4.2.2](#) LogicalElement

[4.2.3](#) Service

[4.2.4](#) NetworkService

[4.2.5](#) ForwardingService

[4.2.6](#) DiffServService

[4.2.6.1](#) The Attribute ConsumedBandwidth

[4.2.6.1](#) The Attribute ConsumedPackets

[4.2.6.1](#) The Attribute CurrentBandwidth

[4.2.6.1](#) The Attribute CurrentDelay

- [4.2.6.1](#) The Attribute LostPackets
- [4.2.7](#) AFService
 - [4.2.7.1](#) The Attribute AssignedBandwidth
 - [4.2.7.2](#) The Attribute MaxDelay
 - [4.2.7.3](#) The Attribute SmoothingInterval
 - [4.2.7.4](#) The Relationship AFDropPrecedenceServices
- [4.2.8](#) AFEdgeService
- [4.2.9](#) EFSERVICE
 - [4.2.9.1](#) The Attribute DSCP
 - [4.2.9.2](#) The Attribute MaxAssignedBandwidth
 - [4.2.9.3](#) The Attribute MaxDelay
 - [4.2.9.4](#) The Relationship EFDropPrecedenceServices

- [4.2.10](#) EFEdgeService
- [4.2.11](#) PrecedenceService
 - [4.2.11.1](#) The Attribute ToS
- [4.2.12](#) IntServService
- [4.2.13](#) DropPrecedenceService
 - [4.2.13.1](#) The Attribute DSCP
 - [4.2.13.2](#) The Attribute InitialLossLevelStart
 - [4.2.13.3](#) The Attribute InitialLossLevelStop
 - [4.2.13.4](#) The Attribute FullLossLevelStart
 - [4.2.13.5](#) The Attribute FullLossLevelStop
- [4.2.14](#) Configuration
- [4.2.15](#) BehaviorConfiguration
- [4.2.16](#) AFConfiguration
 - [4.2.16.1](#) The Attribute AssignedBandwidth
 - [4.2.16.2](#) The Attribute MaxDelay
 - [4.2.16.3](#) The Attribute SmoothingInterval
 - [4.2.16.4](#) The Relationship AFConfigDropPrecedenceServices
- [4.2.17](#) AFEdgeConfiguration
- [4.2.18](#) AFRoleConfiguration
- [4.2.19](#) AFVendorConfiguration
 - [4.2.19.1](#) The Multi-valued Attribute Constraint
 - [4.2.19.2](#) The Attribute ConstraintEncoding
- [4.2.20](#) EFConfiguration
 - [4.2.20.1](#) The Attribute DSCP
 - [4.2.20.2](#) The Attribute MaxAssignedBandwidth
 - [4.2.20.3](#) The Attribute MaxDelay
 - [4.2.20.4](#) The Relationship EFConfigDropPrecedenceServices
- [4.2.21](#) EFEdgeConfiguration
- [4.2.22](#) EFRoleConfiguration

- [4.2.23](#) EFVendorConfiguration
- [4.2.23.1](#) The Multi-valued Attribute Constraint
- [4.2.23.2](#) The Attribute ConstraintEncoding
- [4.2.24](#) IntServConfiguration
- [4.2.25](#) DropPrecedenceConfiguration
- [4.2.26](#) PrecedenceConfiguration
- [4.2.26.1](#) The Attribute ToS

[5.](#) Mapping to Example Policies

[6.](#) Security Considerations

[7.](#) Acknowledgments

[8.](#) References

[9.](#) Author's Addresses

[10.](#) Full Copyright Statement

Weiss & Co

Expires in six months

[Page 4]

INTERNET-DRAFT

Terminology for policy and services

June 25, 1999

[1.](#) Introduction

Policy conditions and actions have two principle components: operands and operators. Operands can be constants or variables. You can't construct a policy without specifying what operands you want to examine and what operands you want to change. Operands can be high level like Joe (a user) or Gold (a service). Operands can also be low level like IP Address or a queue's bandwidth allocation. Irrespective of what level the operands are specified at, they still need to be specified and standardized.

The second component of policy conditions and actions is a set of operators. Operators can express relationships (greater then, in the set, boolean OR, etc.) or assignments. Together, operators and operands can express a variety of conditions and actions:

If Bob is an Engineer...

If the Src IP address is in the Marketing Subnet...

Set Joe's ip address to 2.3.4.5

Limit the bandwidth of application x to 10 Mb

We recognize that the definition of operator semantics are critical to the definition of policies. However, the definition of these operators is beyond the scope of this document. Rather, this draft takes the first steps in identifying and standardizing a set of operands for use in policies.

[2.](#) Approach

QoS activities in the IETF have mainly focused in two areas: Integrated Services and Differentiated Services. This draft initially focuses on the specification of QoS attributes and classes for the policy management of Differentiated Services capabilities.

[2.1.](#) Common Needs Of DiffServ and RSVP

First we consider the common needs for RSVP and DiffServ. RSVP has two principle components. One component is embedded in the forwarding engine of the networking device. Its functions include the classification and policing of individual flows and scheduling admitted packets for the outbound link. The other component of RSVP

is the management of the signaling protocol (e.g., PATH and RESV messages). This component processes reservation requests, manages bandwidth, outsources decision making to policy servers, and interacts with Routing Table manager. We will take RSVP into consideration when defining the schema structure. As this draft initially focuses on the forwarding engine, elements of RSVP applicable to the forwarding engine will be considered in the structure of the schema.

This draft focuses on a small subset of the QoS policy problem in hopes of constructing a methodology that can be adapted for other aspects of QoS and policy construction in general. Hence, RSVP will not be considered in any detail in this iteration of the draft.

DiffServ operates exclusively in the forwarding engine. It has all the same components of the RSVP forwarding engine with two major differences. First, DiffServ performs classification based solely on the DSCP field while RSVP examines a subset of a standard flow's addressing 5-tuple. There are special cases in DiffServ where the addressing 5-tuple can be examined. Most notably, this can occur at the boundary between a DS domain and a non-DS domain. However, most routers in a DiffServ domain will only need to classify based on the DSCP field.

The second difference between RSVP and DiffServ is that the effects of RSVP on the forwarding engine are more dynamic because each newly admitted RSVP reservation requires a reconfiguration of the forwarding engine. In contrast, DiffServ requires far fewer changes to the forwarding engine after the PHBs have been configured.

The approach advocated by the authors for the creation of policies is to first identify the attributes with which policies are to be constructed. These attributes are the parameters to expressions that

are necessary to construct policies. There is also a parallel desire to define the operators, relations, precedence constructs necessary to construct the conditions and actions proposed by the core policy schema. These efforts are beyond the scope of this document. However, this aspect of the policy schema will be addressed in a subsequent document.

[2.1.](#) Specific Needs Of DiffServ

DiffServ-specific rules can focus on two particular areas: the core of the network and the edges of the network. The attributes used to manipulate QoS capabilities in the core of the network primarily address the behavioral characteristics of each supported DiffServ class (or queue). At the edges of the DiffServ network, the additional complexities of flow classification, policing, RSVP mappings, remarkings, and billing have to be considered. In addition, the standards for edges of the DiffServ network need more detail before the edges can be incorporated in the policy model. Therefore, this draft will initially focus on the core of the DiffServ network. However, it is expected that as the DiffServ standards evolve to better define the semantics of edge devices, these attributes will also be added to the QoS schema.

[2.1.](#) Specific Needs Of RSVP

This first iteration of this document will focus exclusively on the forwarding aspects of network QoS. Therefore, while the forwarding aspects of RSVP will be considered, the management of the RSVP signaling protocol will not be considered. This will be considered in a future iteration of this draft.

[3.](#) Methodology

As there is a clear need to define QoS attributes from which to construct policies, there are some very basic issues that need to be considered. Considering these issues should help to construct the proper schema for QoS attributes as well as a general Policy schema.

[3.1](#) Level of Abstraction for Expressing QoS Policies

The first issue requiring consideration is the level of abstraction at which QoS policies should be expressed. If we consider policies as rules used to react to events and manipulate attributes or generate new events, this concept can be applied as broadly as a business goal and as specifically as an interaction within a device. An example of business level policy might be: from 1:00 pm PST to 7:00 am EST sell off 40% of capacity on the open market. In contrast, a device specific policy might be: if the queue depth grows at a geometric rate over a specified duration, trigger a potential link failure event.

As policies are a function of parameters (attributes) and operators (boolean, arithmetic, relational, etc.), both need to be defined in order to construct policies that are broadly implementable. If the parameters to the rules are specified too narrowly, they will reflect the individual implementations of QoS in each device. As there is currently little consensus in the industry on what the correct implementation model for QoS is, most defined attributes would only be applicable to the unique characteristics of a few individual devices. Lastly, standardizing all of these potential implementation alternatives would be a never-ending task as new implementations appear on the market.

In contrast, if you start with a business policy like "Bob gets Gold Service," that is not enough. We also have to define the service semantics if we want to have uniform application of the policy in the network devices. Any service definition would have to be grounded in semantics like Delay, Jitter, Bandwidth, Reliability, Loss, Billing, and over-subscription rules, just to name a few. Getting a written agreement to the service semantics of Gold Service would not only be extremely painful given the broad number of possible combinations, but it would also limit the number of business policies available to network administrators.

[3.2](#) Level of Abstraction for Defining QoS Attributes and Classes

We therefore propose that the Policy Framework Working Group first focus on those policies that define the Services for DiffServ capable devices and networks. This means that we will need to standardize the

attributes that are needed to support policies at the services level. For example, to preserve the Delay characteristics committed to the end-user, a network administrator may wish to create policies that monitor the queue depths in a device and adjust resource allocations when delay budgets are at risk (perhaps as a result of a network topology change). Another example could be the criteria for allowing over-subscription of a service and the consequences (notify billing agent or terminate session if network characteristics change). This has the benefit of maximizing the flexibility that network administrators have in defining new services. In addition, once the policies for the services are defined, it is relatively easy for network administrators to construct policies that define business goals on top of the policies that define the service goals.

As mentioned above, the one obstacle in creating service-oriented policies is the issue of supporting diverse implementations of QoS. The solution is to define the QoS attributes at the behavioral level. For DiffServ, this means that we must define the attributes that represent the characteristics of the DiffServ PHBs. Just as it is up to vendors to map PHBs to individual implementations, it is up to these same vendors to map the attributes necessary to monitor and manage the PHB to the specific vendor's implementation of the behavior. This mapping can either be accommodated by a proxy agent like a PIB, or it can be supported directly in the device.

[3.3](#) Characterization of QoS Attributes

The QoS attributes and container classes will be described in more detail in [section 4](#). However, we should consider the basic characteristics of the attributes to understand the methodology for representing them.

There are essentially two types of attributes, state and configuration. Configuration attributes describe the desired state of the device. State attributes describe the actual state of the device. Configuration attributes include desired or proposed thresholds, bandwidth allocations, and classification characteristics (more rules...). State attributes include the current actual value of these configuration attributes in devices as well as attributes that represent dynamic state (queue depths, excess capacity consumption, loss rates, and so forth).

One should note that state attributes tend to be device-specific. In contrast, a configuration attribute can be represented and applied to a set of devices (e.g., all devices in the same domain that share the same AS number, or all core devices that share the same delay bound

for a specific service). This suggests that the schema for configuration attributes will not be the same as the schema that

supports state attributes.

[3.4](#) Identifying Common Shared Policies

Another issue that arises is how to distinguish policies for individual devices or even components of a device from policies that apply to a set of devices. These contextual issues depend on more than just the policy schema in order for them to function properly. Hence, ongoing efforts in the DMTF to define devices, services, users, groups, and collections of devices are all relevant to the proper construction of a policy model. Context is a key aspect of policies that still requires a great deal more work. The next iteration of this draft will include some preliminary results from these modeling efforts.

[3.5](#) QoS Schema Derivation

The question of contexts also begs the question of what core model QoS attributes should be derived from. Current thinking is that QoS is part of the policy model. However, QoS is fundamentally a set of characteristics of a networking device. It is supported with schedulers, classifiers, policers, and buffer managers. Similarly, security is also a characteristic of devices, as authentication and encryption capabilities represent services that networked devices perform (irrespective of interactions with policy servers). As such, we argue that QoS attributes should be part of a networking device schema rather than a policy schema. Although a policy schema may use QoS attributes to define policies, the policy schema really needs to focus on the semantics of representing the policy itself (conditions, actions, operators, etc.). However, this does not preclude the Policy Framework working group from standardizing the QoS schema. While this iteration of this draft concentrates on defining an information model that can represent DiffServ QoS, the ultimate goal is to be able to apply policies that control these features in network devices.

[3.6](#) Attribute Representation

The last issue to be considered is the question of how attributes are represented. If QoS attributes are represented as absolute numbers

(e.g., Class AF2 gets 2 Mbs of bandwidth), it is more difficult to make them uniform across multiple ports in a device or multiple devices because of the broad variation in link capacities. However, expressing attributes in relative or proportional terms (e.g., Class AF2 gets 5% of the total link bandwidth) makes it more difficult to express certain types of conditions and actions, such as:

(If ConsumedBandwidth = AssignedBandwidth Then ...).

There are really three alternatives to address this problem. First, multiple attributes can be defined to express the same value in various forms. This idea has been rejected because of the likelihood of inconsistencies between the attributes. The second alternative is to create multi-modal attributes that express the same value but in different terms based on the access or assignment mode. This option was rejected because it significantly complicates the model and is impossible to express in current directory access protocols (e.g., LDAP). The last option is to express all attributes in absolutes, but make the operators in the policy schema more sophisticated. Thus, to represent a percentage, division and multiplication operators are required (e.g., Class AF2 gets $.05 \times$ the total link bandwidth). This is the approach that has been taken in this draft.

[4.](#) The Class Hierarchy

The following sections describe the class hierarchy of the information model for modeling QoS at the device level. [Section 4.1](#) defines the structure of the class hierarchy, and [section 4.2](#) defines the classes and their attributes that make up this class hierarchy.

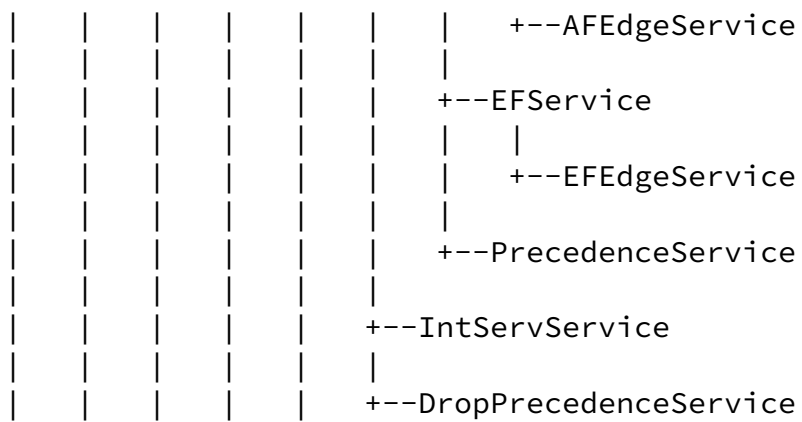
[4.1](#) Structure of the Class Hierarchy

This draft defines two different class hierarchies that are not necessarily rooted from the same point in the overall schema. However, it is intended that these two hierarchies be used together to control and administer devices that are running Differentiated and Integrated Services. Therefore, we propose one class hierarchy to manage the state of these services, and a different class hierarchy to manage the configuration of these services. Both of these hierarchies are derived from the Common Information Model, and are compatible with the Directory Enabled Networks (DEN) effort.

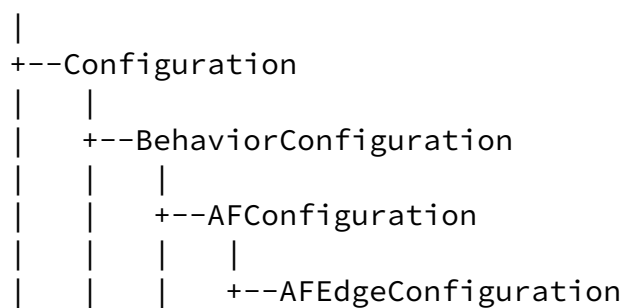
INTERNET-DRAFT Terminology for policy and services June 25, 1999

The structure of the Class Hierarchy for managing the state of Differentiated and Integrated Services is as follows:

```
|
|--ManagedSystemElement
|  |--LogicalElement
|  |  |--Service
|  |  |  |--NetworkService
|  |  |  |  |--ForwardingService
|  |  |  |  |  |--DiffServService
|  |  |  |  |  |  |--AFService
|  |  |  |  |  |  |
```

The structure of the Class Hierarchy for managing the configuration of Differentiated and Integrated Services is as follows:




```

| | | | |
| | | | +---AFRoleConfiguration
| | | | |
| | | | +---AFVendorConfiguration
| | | | |
| | | +---EFConfiguration
| | | | |
| | | | +---EFEEdgeConfiguration
| | | | |
| | | | +---EFRoleConfiguration
| | | | |
| | | | +---EFVendorConfiguration
| | | | |
| | | +---IntServConfiguration
| | | | |
| | | +---DropPrecedenceConfiguration
| | | | |
| | | +---PrecedenceConfiguration

```

4.2 Class and Attribute Definitions

4.2.1 ManagedSystemElement

This is an abstract class that is defined in the Core Model of CIM. This is the base class for the System Element hierarchy. Any distinguishable component of a System is a candidate for inclusion in this class, including physical components (e.g., chips and cards) and logical components (e.g., software components, services, and other objects). Please refer to [CIM] for the full definition of this class.

4.2.2 LogicalElement

This is an abstract class that is defined in the Core Model of CIM. It is a subclass of the ManagedSystemElement class. This is the base class for all components of a managed System that represent abstract system components, such as Files, Processes, or system capabilities

in the form of Logical Devices and Services. Please refer to [CIM] for the full definition of this class.

4.2.3 Service

This is an abstract class that is defined in the Core Model of CIM. It is a subclass of the LogicalElement class. This is the base class for all objects that contain the information necessary to represent and manage the functionality provided by a Device and/or SoftwareFeature. Note that a Service is a general-purpose object that is used to configure and manage the implementation of functionality. It is not the functionality itself. Please refer to [CIM] for the full definition of this class.

[4.2.4](#) NetworkService

This is an abstract class that is defined in the Network Common Model of CIM. It is a subclass of the Service class. This is the base class that serves as the root of the network service hierarchy. Network services represent generic functions that are available from the network that configure and/or modify the traffic being sent. Please refer to [CIM] for the full definition of this class.

[4.2.5](#) ForwardingService

This is a concrete class that is defined in the Network Common Model of CIM. It is a subclass of the NetworkService class. This class represents the forwarding of network traffic by receiving data from one or more communication sources and sending that data via other communication sources. Please refer to [CIM] for the full definition of this class.

4.2.6 DiffServService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the ForwardingService class. This class represents a specialization of the general concept of forwarding network traffic by adding specific semantics that characterize the operation of Differentiated Services.

The attributes defined below all have the semantics of a counter that is being reset every second. The class definition is as follows:

NAME	DiffServService
DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding.
DERIVED FROM	ForwardingService
TYPE	Structural
PROPERTIES	ConsumedBandwidth ConsumedPackets CurrentBandwidth CurrentDelay LostPackets

[4.2.6.1](#) The Attribute ConsumedBandwidth

The ConsumedBandwidth attribute defines the total bandwidth that has been consumed during the past second, including lost packets.

NAME	ConsumedBandwidth
DESCRIPTION	Total bandwidth consumed during the last second, in Kbits/second
SYNTAX	Integer

[4.2.6.2](#) The Attribute ConsumedPackets

The ConsumedPackets attribute defines the total number of packets that have been consumed during the past second, including lost packets.

NAME	ConsumedPackets
DESCRIPTION	Total number of packets consumed during the last second, in packets/second
SYNTAX	Integer

[4.2.6.3](#) The Attribute CurrentBandwidth

The CurrentBandwidth attribute defines the instantaneous value of the current bandwidth available for this link.

NAME	CurrentBandwidth
DESCRIPTION	Instantaneous bandwidth currently available in this link, in Kbits/Second
SYNTAX	Integer

[4.2.6.4](#) The Attribute CurrentDelay

The CurrentDelay attribute defines the instantaneous value of the current delay for this link.

INTERNET-DRAFT Terminology for policy and services June 25, 1999

NAME	CurrentDelay
DESCRIPTION	Instantaneous delay in this link, in Kbits/sec
SYNTAX	Integer

[4.2.6.5](#) The Attribute LostPackets

The LostPackets attribute defines the total number of packets that have been lost during the last second on this link.

NAME	LostPackets
DESCRIPTION	Total number of packets lost during the last second
SYNTAX	Integer

[4.2.7](#) AFSERVICE

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the DiffServService class. This class represents a specialization to the general concept of forwarding network traffic by adding specific semantics that characterize the operation of the Assured Forwarding Service defined in Differentiated Services [AF].

The first two attributes defined below have the semantics of a counter that is being reset every second. The class definition is as follows:

NAME	AFService
DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding using the AF PHB Group.
DERIVED FROM	DiffServService
TYPE	Structural
PROPERTIES	AssignedBandwidth MaxDelay SmoothingInterval
RELATIONSHIPS	AFDropPrecedenceServices

[4.2.7.1](#) The Attribute AssignedBandwidth

The AssignedBandwidth attribute defines the bandwidth that has been assigned to this link through device-specific configuration commands.

NAME	AssignedBandwidth
DESCRIPTION	Assigned bandwidth to this link in Kbits/second
SYNTAX	Integer

[4.2.7.2](#) The Attribute MaxDelay

The MaxDelay attribute defines the total delay through this link.

NAME	MaxDelay
DESCRIPTION	Total delay in this link, in microseconds
SYNTAX	Integer

[4.2.7.3](#) The Attribute SmoothingInterval

The SmoothingInterval is a constant used to calculate the level of congestion in the link, so that instantaneous bursts can be properly filtered over time.

NAME	SmoothingInterval
DESCRIPTION	Constant used to calculate the level of congestion in the link
SYNTAX	Integer

[4.2.7.4](#) The Relationship AFDropPrecedenceServices

The AFDropPrecedenceServices relationship is an aggregation that makes explicit the dependency between an instance of an AF service class and the instances of the drop precedence classes that it uses. For example, [AF] defines four service classes, each with three drop precedences. However, the semantics are that the AF class can not be completely specified until the drop precedences are specified. This is an example of a "whole-part" relationship, where the antecedent (the AFService instance) is not "complete" until its constituent parts (the instances of the DropPrecedence class) are defined and "attached" to it.

The multiplicity of this relationship is one-or-more (on the

AService side) to zero-or-more (on the DropPrecedence side). This means that a particular AService instance can use zero or more DropPrecedence instances, and that multiple AService instances can use the same DropPrecedence instance.

[4.2.8](#) AEdgeService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the AService class. This class represents a specialization to the general concept of forwarding network traffic by adding specific semantics that characterize the operation of the Assured Forwarding Service defined in Differentiated Services [AF] that are specific to edge devices (as opposed to distribution and core devices).

The class definition is as follows:

NAME	AEdgeService
DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding using the AF PHB Group, specifically for edge devices.
DERIVED FROM	AService
TYPE	Structural
PROPERTIES	<to be defined>

[4.2.9](#) EService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the DiffServService class. This class represents a specialization to the general concept of forwarding network traffic by adding specific semantics that characterize the operation of the Expedited Forwarding Service defined in Differentiated Services [EF].

The first two attributes defined below have the semantics of a counter that is being reset every second. The class definition is as follows:

NAME	EService
------	----------

DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding using the EF PHB Group.
DERIVED FROM	DiffServService
TYPE	Structural
PROPERTIES	DSCP MaxAssignedBandwidth MaxDelay
RELATIONSHIPS	EFDropPrecedenceServices

[4.2.9.1](#) The Attribute DSCP

The DSCP attribute defines the Differentiated Services Code Point that this link uses to represent the EF service through device-specific configuration commands.

NAME	DSCP
DESCRIPTION	DiffServ Code Point used to select the EF service
SYNTAX	String

[4.2.9.2](#) The Attribute MaxAssignedBandwidth

The MaxAssignedBandwidth attribute defines the maximum bandwidth that has been assigned to this link through device-specific configuration commands.

NAME	MaxAssignedBandwidth
DESCRIPTION	Maximum assigned bandwidth to this link in Kbits/second
SYNTAX	Integer

[4.2.9.3](#) The Attribute MaxDelay

The MaxDelay attribute defines the maximum delay that this link has.

NAME	MaxAssignedBandwidth
DESCRIPTION	Maximum delay of this link in microseconds
SYNTAX	Integer

[4.2.9.4](#) The Relationship EFDropPrecedenceServices

The EFDropPrecedenceServices relationship is an aggregation that makes explicit the dependency between an instance of an EF service class and the instances of the drop precedence classes that it uses. For example, [EF] defines a service class. However, one could define additional EF service classes as well as assign a drop precedence to each. If a drop precedence is assigned to an EF service class, then the semantics are that the EF class can not be completely specified until the drop precedence(s) are specified. This is an example of a "whole-part" relationship, where the antecedent (the EFService instance) is not "complete" until its constituent parts (the instances of the DropPrecedence class) are defined and "attached" to it.

The multiplicity of this relationship is one-or-more (on the EFService side) to zero-or-more (on the DropPrecedence side). This means that a particular EFService instance can use zero or more DropPrecedence instances, and that multiple EFService instances can use the same DropPrecedence instance. sp

[4.2.10](#) EFEdgeService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the EFService class. This class represents a specialization to the general concept of

forwarding network traffic by adding specific semantics that characterize the operation of the Expedited Forwarding Service defined in Differentiated Services [EF] that are specific to edge devices (as opposed to distribution and core devices).

The class definition is as follows:

NAME	EFEdgeService
DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding using the EF PHB Group, specifically for

	edge devices.
DERIVED FROM	EFService
TYPE	Structural
PROPERTIES	<to be defined>

[4.2.11](#) PrecedenceService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the DiffServService class. This class represents a specialization to the general concept of forwarding network traffic by adding specific semantics that characterize the operation of the Expedited Forwarding Service defined in Differentiated Services [EF] that are specific to edge devices (as opposed to distribution and core devices).

The class definition is as follows:

NAME	PrecedenceService
DESCRIPTION	A concrete class for describing the common characteristics of differentiated services that are used to affect traffic forwarding. This class defines the concept of traffic precedence, so that precedence may be used in implementing differentiated services such as those specified in [AF] and [EF].
DERIVED FROM	DiffServService
TYPE	Structural
PROPERTIES	ToS

[4.2.11.1](#) The Attribute ToS

The Type of Service, or ToS, attribute, defines the notion of precedence for different types of traffic. See [DIFFSERV] for more information on the definition, backward compatibility with the ToS byte of IPv4 traffic, and use of this attribute.

NAME	ToS
DESCRIPTION	Type of Service setting, used to provide different Priorities for different types of traffic.
SYNTAX	String

[4.2.12](#) IntServService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the ForwardingService class. This class represents a specialization to the general concept of forwarding network traffic as applied to the Integrated Services model. An example of a protocol using this model is RSVP.

The class definition is as follows:

NAME	IntServService
DESCRIPTION	A concrete class for describing the common characteristics of integrated services that are used to affect traffic forwarding when using signaling mechanisms (e.g., RSVP).
DERIVED FROM	ForwardingFService
TYPE	Structural
PROPERTIES	<to be defined>

[4.2.13](#) DropPrecedenceService

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the ForwardingService class. This class represents a specialization to the general concept of forwarding network traffic that meets a particular specification, and dropping traffic that doesn't.

The class definition is as follows:

NAME	DropPrecedenceService
DESCRIPTION	A concrete class for describing the common characteristics of network forwarding services that examine traffic and either forward it or drop it. The dropping is done through an active queue management mechanism (e.g., RED [RED]).
DERIVED FROM	EFService
TYPE	Structural
PROPERTIES	DSCP InitialLossLevelStart InitialLossLevelStop FullLossLevelStart FullLossLevelStop

[4.2.13.1](#) The Attribute DSCP

The DSCP attribute defines the Differentiated Services Code Point that this link uses to represent the EF service through device-specific configuration commands.

NAME	DSCP
DESCRIPTION	DiffServ Code Point used to select the Drop Precedence service
SYNTAX	String

[4.2.13.2](#) The Attribute InitialLossLevelStart

The InitialLossLevelStart attribute defines the value of the average queue depth at which point whatever active queue management algorithm is being used should start dropping packets. The rate of packet drop will increase as a function of the increase of the average queue size, until the average queue size reaches the full loss level. This attribute is used in conjunction with the InitialLossLevelStop attribute to define a rate at which the drop probability should increase until the average queue size reaches the maximum threshold.

NAME	InitialLossLevelStart
DESCRIPTION	Initial value at which packets will be dropped
SYNTAX	String

[4.2.13.3](#) The Attribute InitialLossLevelStop

The InitialLossLevelStop attribute defines the value of the average queue depth at which whatever active queue management algorithm is being used should start dropping packets. It, in conjunction with the InitialLossLevelStart attribute, define the slope of the packet dropping function.

NAME	InitialLossLevelStop
DESCRIPTION	Initial value at which packets will be dropped
SYNTAX	String

[4.2.13.4](#) The Attribute FullLossLevelStart

The FullLossLevelStart attribute defines the value of the average queue depth at which point whatever active queue management algorithm is being used should start dropping all packets.

NAME	FullLossLevelStart
DESCRIPTION	Value at which all packets will be dropped
SYNTAX	String

INTERNET-DRAFT Terminology for policy and services June 25, 1999

[4.2.13.5](#) The Attribute FullLossLevelStop

The FullLossLevelStop attribute defines the value of the average queue depth at which point whatever active queue management algorithm is being used should start dropping all packets.

NAME	FullLossLevelStop
DESCRIPTION	Value at which all packets will be dropped
SYNTAX	String

[4.2.14](#) Configuration

This is a concrete class that is defined in the Core Model of CIM. It is a top-level class that enables the grouping of sets of parameters (defined in, for example, Setting objects) and dependencies for one or more ManagedSystemElements. The Configuration object represents a certain behavior, or a desired functional state, for the ManagedSystemElements. The desired functional state is typically driven by external requirements such as time or location. Please refer to [CIM] for the full definition of this class.

[4.2.15](#) BehaviorConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the Configuration class. This class represents specialized configuration needs that can be used to configure the behavior of network services, such as Assured Forwarding and Expedited Forwarding.

The class definition is as follows:

NAME	BehaviorConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of network services, such as Assured Forwarding and Expedited Forwarding.
DERIVED FROM	Configuration
TYPE	Structural
PROPERTIES	<tbid>

[4.2.16](#) AFConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the BehaviorConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Assured Forwarding service.

The class definition is as follows:

INTERNET-DRAFT Terminology for policy and services June 25, 1999

NAME	AFConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Assured Forwarding service.
DERIVED FROM	BehaviorConfiguration
TYPE	Structural
PROPERTIES	AssignedBandwidth MaxDelay SmoothingInterval
RELATIONSHIPS	AFConfigDropPrecedenceServices

[4.2.16.1](#) The Attribute AssignedBandwidth

The AssignedBandwidth attribute defines the desired bandwidth (e.g., the bandwidth that will be configured for this device) to this link through device-specific configuration commands.

NAME	AssignedBandwidth
DESCRIPTION	Assigned bandwidth to this link in Kbits/second
SYNTAX	Integer

[4.2.16.2](#) The Attribute MaxDelay

The MaxDelay attribute defines the desired total delay (e.g., the delay resulting from the configuration of this device) through this link.

NAME	MaxDelay
DESCRIPTION	Total delay in this link, in microseconds
SYNTAX	Integer

[4.2.16.3](#) The Attribute SmoothingInterval

The SmoothingInterval is a constant used to calculate the level of congestion in the link, so that instantaneous bursts can be properly

filtered over time. This attribute represents the value of this constant that will be used when the device is configured.

NAME	SmoothingInterval
DESCRIPTION	Constant used to calculate the level of congestion in the link
SYNTAX	Integer

[4.2.16.4](#) The Relationship AFConfigDropPrecedenceServices

The AFConfigDropPrecedenceServices relationship is an aggregation that makes explicit the dependency between an instance of an AFConfiguration class and the instances of the drop precedence

Weiss & Co

Expires in six months

[Page 25]

INTERNET-DRAFT

Terminology for policy and services

June 25, 1999

classes that it uses. For example, [AF] defines four service classes, each with three drop precedences. However, the semantics are that the AF class can not be completely specified until the drop precedences are specified. This is an example of a "whole-part" relationship, where the antecedent (the AFService instance) is not "complete" until its constituent parts (the instances of the DropPrecedence class) are defined and "attached" to it. This association is used to define the initial configuration of these services.

The multiplicity of this relationship is one-or-more (on the AFConfiguration side) to zero-or-more (on the DropPrecedence side). This means that a particular AFConfiguration instance can use zero or more DropPrecedence instances, and that multiple AFConfiguration instances can use the same DropPrecedence instance.

[4.2.17](#) AFEEdgeConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the AFConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Assured Forwarding service for edge devices.

The class definition is as follows:

NAME	AFEEdgeConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Assured

DERIVED FROM	Forwarding service for edge devices.
TYPE	AFConfiguration
PROPERTIES	Structural
	<tb>

[4.2.18](#) AFRoleConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the AFEEdgeConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Assured Forwarding service for edge devices by using roles to identify groups of devices and groups of interfaces that are to be configured. This enforces the semantics of managing the common configuration of a group of interfaces and/or devices.

The class definition is as follows:

NAME	AFRoleConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Assured Forwarding service for edge devices that are identified by roles. Note that roles can also be used to identify the interfaces of a device or a group of devices.
DERIVED FROM	AFEEdgeConfiguration
TYPE	Structural
PROPERTIES	<tb>

[4.2.19](#) AFVendorConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the AFEEdgeConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Assured Forwarding service for edge devices on a per-vendor and per-device basis.

Vendor-specific configuration is, of course, unique to each vendor. Therefore, a standard set of attributes can not be defined in a

specification. Instead, this class provides an array of OctetStrings (implemented as the Constraint attribute) and a single encoding of who the vendor is (represented by the ConstraintEncoding attribute). The Constraint attribute provides a way for the vendor to store, retrieve, manage, and distribute configuration commands that are specific to the vendor identified by the (registered) OID value contained in the ConstraintEncoding attribute.

This class therefore provides a general escape mechanism for representing common configuration policies that are specific to a particular vendor. This enables the vendor to implement a standards-compliant architecture that can be used to configure vendor-specific functions. As its name suggests, this class is intended for vendor-specific extensions. Standardized extensions are not expected to use this class.

The class definition is as follows:

NAME	AFVendorConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Assured Forwarding service for edge devices that are specific to a particular vendor.
DERIVED FROM	AFEEdgeConfiguration
TYPE	Structural
PROPERTIES	Constraint[] ConstraintEncoding

[4.2.19.1](#). The Multi-valued Attribute Constraint

This attribute provides a general escape mechanism for representing configuration commands controlled by policy that have not been modeled with specific attributes. One use of this is to model vendor-specific configuration commands.

The format of the uint8 array is left unspecified in this definition. It is instead determined by the OID value stored in the ConstraintEncoding attribute. Since ConstraintEncoding is single-valued, all of the values of Constraint share the same format and semantics.

A policy decision point can readily determine whether it supports the values stored in an instance of Constraint by checking the OID value from ConstraintEncoding against the set of OIDs it recognizes. The action for the policy decision point to take in case it does not recognize the format of this data could itself be modeled as a policy rule, governing the behavior of the policy decision point.

The attribute definition is as follows:

NAME	Constraint
DESCRIPTION	Escape mechanism for representing constraints that have not been modeled as specific attributes. The format of the values is identified by the OID stored in the ConstraintEncoding attribute.
SYNTAX	OctetString

[4.2.19.2](#). The Attribute ConstraintEncoding

This attribute identifies the encoding and semantics of the Constraint attribute values in this instance. The value of this attribute is a single string, representing an OID which identifies the vendor.

The attribute definition is as follows:

NAME	ConstraintEncoding
DESCRIPTION	An OID encoded as a string, identifying the format and semantics for this instance's Constraint attribute.
SYNTAX	string

[4.2.20](#) EFConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the BehaviorConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Expedited Forwarding service.

The class definition is as follows:

NAME	EFConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Assured Forwarding service.
DERIVED FROM	BehaviorConfiguration
TYPE	Structural
PROPERTIES	DSCP MaxAssignedBandwidth MaxDelay
RELATIONSHIPS	EFConfigDropPrecedenceServices

[4.2.20.1](#) The Attribute DSCP

The DSCP attribute defines the Differentiated Services Code Point that this link uses to represent the EF service through device-specific configuration commands.

NAME	DSCP
DESCRIPTION	DiffServ Code Point used to select the EF service
SYNTAX	String

[4.2.20.2](#) The Attribute MaxAssignedBandwidth

The MaxAssignedBandwidth attribute defines the desired bandwidth (e.g., the bandwidth that will be configured for this device) to this link through device-specific configuration commands.

NAME	MaxAssignedBandwidth
DESCRIPTION	Maximum assigned bandwidth to this link in Kbits/second
SYNTAX	Integer

[4.2.20.3](#) The Attribute MaxDelay

The MaxDelay attribute defines the desired total delay (e.g., the delay resulting from the configuration of this device) through this link.

NAME	MaxDelay
DESCRIPTION	Total delay in this link, in microseconds
SYNTAX	Integer

[4.2.20.4](#) The Relationship EFConfigDropPrecedenceServices

The EFConfigDropPrecedenceServices relationship is an aggregation that makes explicit the dependency between an instance of an EFConfiguration service class and the instances of the drop precedence classes that it uses. For example, [EF] defines a service class. However, one could define additional EF service classes as well as assign a drop precedence to each. If a drop precedence is assigned to an EF service class, then the semantics are that the EF class can not be completely specified until the drop precedence(s) are specified. This is an example of a "whole-part" relationship, where the antecedent (the EFService instance) is not "complete" until its constituent parts (the instances of the DropPrecedence class) are defined and "attached" to it.

The multiplicity of this relationship is one-or-more (on the EFConfiguration side) to zero-or-more (on the DropPrecedence side). This means that a particular EFConfiguration instance can use zero or more DropPrecedence instances, and that multiple EFConfiguration instances can use the same DropPrecedence instance.

[4.2.21](#) EFEdgeConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the EFConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the Expedited Forwarding service for edge devices.

The class definition is as follows:

NAME	EFEdgeConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Expedited Forwarding service for edge devices.
DERIVED FROM	EFConfiguration
TYPE	Structural
PROPERTIES	<tdb>

[4.2.22](#) EFRoleConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the EFEdgeConfiguration class. This class represents specialized configuration needs that can

INTERNET-DRAFT Terminology for policy and services June 25, 1999

be used to configure the behavior of the Expedited Forwarding service for edge devices by using roles to identify groups of devices and groups of interfaces that are to be configured. This enforces the semantics of managing the common configuration of a group of interfaces and/or devices.

The class definition is as follows:

NAME	EFRoleConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the Expedited Forwarding service for edge devices that are identified by roles. Note that roles can also be used to identify the interfaces of a device or a group of devices.
DERIVED FROM	EFEdgeConfiguration
TYPE	Structural
PROPERTIES	<tdb>

[4.2.23](#) EFVendorConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the EFEdgeConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of the ExpeditedForwarding service for edge devices on a per-vendor and per-device basis.

Vendor-specific configuration is, of course, unique to each vendor. Therefore, a standard set of attributes can not be defined in a specification. Instead, this class provides an array of OctetStrings (implemented as the Constraint attribute) and a single encoding of who the vendor is (represented by the ConstraintEncoding attribute). The Constraint attribute provides a way for the vendor to store, retrieve, manage, and distribute configuration commands that are specific to the vendor identified by the (registered) OID value contained in the ConstraintEncoding attribute.

This class therefore provides a general escape mechanism for representing common configuration policies that are specific to a particular vendor. This enables the vendor to implement a standards-compliant architecture that can be used to configure vendor-specific functions. As its name suggests, this class is intended for vendor-specific extensions. Standardized extensions are not expected to use

this class.

The class definition is as follows:

NAME	EFVendorConfiguration
------	-----------------------

Weiss & Co

Expires in six months

[Page 31]

INTERNET-DRAFT

Terminology for policy and services

June 25, 1999

DESCRIPTION	A concrete class for describing the common configuration characteristics of the Expedited Forwarding service for edge devices that are specific to a particular vendor.
DERIVED FROM	EFEdgeConfiguration
TYPE	Structural
PROPERTIES	Constraint[] ConstraintEncoding

[4.2.23.1](#). The Multi-valued Attribute Constraint

This attribute provides a general escape mechanism for representing configuration commands controlled by policy that have not been modeled with specific attributes. One use of this is to model vendor-specific configuration commands.

The format of the uint8 array is left unspecified in this definition. It is instead determined by the OID value stored in the ConstraintEncoding attribute. Since ConstraintEncoding is single-valued, all of the values of Constraint share the same format and semantics.

A policy decision point can readily determine whether it supports the values stored in an instance of Constraint by checking the OID value from ConstraintEncoding against the set of OIDs it recognizes. The action for the policy decision point to take in case it does not recognize the format of this data could itself be modeled as a policy rule, governing the behavior of the policy decision point.

The attribute definition is as follows:

NAME	Constraint
DESCRIPTION	Escape mechanism for representing constraints that have not been modeled as specific attributes. The format of the values is identified by the OID stored in the

	ConstraintEncoding attribute.
SYNTAX	OctetString

[4.2.23.2](#). The Attribute ConstraintEncoding

This attribute identifies the encoding and semantics of the Constraint attribute values in this instance. The value of this attribute is a single string, representing an OID which identifies the vendor.

The attribute definition is as follows:

Weiss & Co	Expires in six months	[Page 32]
------------	-----------------------	-----------

INTERNET-DRAFT	Terminology for policy and services	June 25, 1999
----------------	-------------------------------------	---------------

NAME	ConstraintEncoding
DESCRIPTION	An OID encoded as a string, identifying the format and semantics for this instance's Constraint attribute.
SYNTAX	string

[4.2.24](#) IntServConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the BehaviorConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of Integrated Services, such as RSVP.

The class definition is as follows:

NAME	IntServConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the signaling protocols, such as RSVP.
DERIVED FROM	BehaviorConfiguration
TYPE	Structural
PROPERTIES	<tbid>

[4.2.25](#) DropPrecedenceConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the BehaviorConfiguration class. This class represents specialized configuration needs that can

be used to configure the behavior of Drop Precedence Services, regardless of which higher-level class is using them.

The class definition is as follows:

NAME	DropPrecedenceConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of drop precedence services independent of what other higher-level service uses it.
DERIVED FROM	BehaviorConfiguration
TYPE	Structural
PROPERTIES	<tdb>
RELATIONSHIPS	AFDropPrecedenceServices, EFDropPrecedenceServices, AFConfigDropPrecedenceServices, EFConfigDropPrecedenceServices

Note: AFDropPrecedenceServices and EFDropPrecedenceServices are defined previously in sections [4.2.7.4](#) and [4.2.9.4](#), respectively. The

AFConfigDropPrecedenceServices and EFConfigDropPrecedenceServices are defined previously in sections [4.2.16.4](#) and [4.2.20.4](#), respectively.

[4.2.26](#) PrecedenceConfiguration

This is a concrete class that is a proposed addition to the Network Common Model of CIM. It is a subclass of the BehaviorConfiguration class. This class represents specialized configuration needs that can be used to configure the behavior of Precedence Services, regardless of which higher-level class is using them.

The class definition is as follows:

NAME	PrecedenceConfiguration
DESCRIPTION	A concrete class for describing the common configuration characteristics of the signaling protocols, such as RSVP.
DERIVED FROM	BehaviorConfiguration
TYPE	Structural
PROPERTIES	ToS

[4.2.26.1](#) The Attribute ToS

The Type of Service, or ToS, attribute, defines the notion of precedence for different types of traffic. See [DIFFSERV] for more information on the definition, backward compatibility with the ToS byte of IPv4 traffic, and use of this attribute. The purpose of this attribute is to define the configuration value for the ToS setting of this link.

NAME	ToS
DESCRIPTION	Type of Service setting, used to provide different Priorities for different types of traffic.
SYNTAX	String

[5.](#) Mapping to Example Policies

All that is left is some usage cases that demonstrate how these attributes can be expressed and used to create policies. In order to express these attributes in a meaningful way, they have to be bound to a router or an interface on a router. This requires the introduction of two additional concepts that can flexibly group the set of devices and interfaces. The first concept is a means for binding a particular policy to a set of devices that execute or act on the policy. To support this capability, we propose a new required attribute, called PolicyScope, listing the devices for which this policy is applicable. This attribute should be defined in the PolicyRule object defined in the Core Information Model. The second concept requires a means for binding an attribute to a specific interface or set of interfaces. This binding, called a Role, takes

the form of a qualifier preceding the instance name. Roles are described in [[TERMS](#)].

So let's consider the following condition:

`Customer1.ReliableService.ConsumedBandwidth > 1500`

The qualifier "Customer1" is a Role that refers to a specific interface of a device. The DiffServ class, or queue (e.g., AF3, precedence 6, or EF) is represented in the directory with a unique instance with a unique name. In this example, ReliableService is the instance name for an AFService class using the DSCPs for AF2. ConsumedBandwidth is an attribute containing the current bandwidth rate being sent on specified DiffServ class (queue) of the specified interface. Hence, the condition reads:

If Customer1 is receiving more than 1500 Kb of AF2 traffic, then....

However, the concept of Roles has additional benefits because we can also express a single policy that can be applied to a set of interfaces simultaneously. Let's consider the following sample action:

`CoreInterfaces.LowDelayConfig.MaxDelay = 20`

In this particular case, we are qualifying this action to only apply to the interfaces as specified by the CoreInterfaces role. In this case we are assuming that this is the set of interfaces in a device participating in the core of the DiffServ domain. Further, LowDelayConfig is an instance of the EFConfiguration class. As mentioned earlier, configuration classes are used to request a change in the configuration of device or service while state classes are

used to model the actual state of the device or service. So, this action expresses the changing of the upper bound on the maximum delay that will be tolerated for this service on each sending interface to the core. As a side note, the intent of the attribute MaxDelay is to provide a simple way of expressing an upper bound on the number of packets that can be queued up while avoiding implementation details for where or how packets are queued or scheduled.

[6.](#) Security Considerations

Security and denial of service considerations are not explicitly considered in this memo, as they are appropriate for the underlying policy architecture implementing the distribution and communication of the information described in this draft. However, the information in this draft explicitly makes use of the security measures recommended in the policy architecture defined by the Policy Framework working group. Specifically, any mechanisms used to distribute and communicate the information described in this draft must minimize theft and denial of service threats. Second, it must be ensured that the entities (such as PEPs and PDPs) involved in policy control can verify each other's identity and establish necessary trust before communicating.

The communication tunnel between policy clients and policy servers SHOULD be secured by the use of an IPSEC [IPSEC] channel. It is advisable that this tunnel makes use of both the AH (Authentication Header) and ESP (Encapsulating Security Payload) protocols, in order to provide confidentiality, data origin authentication, integrity and replay prevention.

[7.](#) Acknowledgments

8. References

- [TERMS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", Internet [RFC 2119](#), March 1997.
- [DIFFARCH] D. Black, S. Blake, M. Carlson, E. Davies, Z. Whang, W. Weiss, "An Architecture for Differentiated Services", [RFC2475](#), December 1998
- [DIFFSERV] K. Nichols and S. Blake, "Definition of the Differentiated Services Field (DS Byte) in the IPv4 and IPv6 Headers", [RFC2474](#), December 1998
- [AF] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", [RFC2597](#), June 1999
- [EF] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", [RFC2598](#), June 1999
- [RAPFRAME] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", Internet Draft, [<draft-ietf-rap-framework-01.txt>](#), May 1998
- [CIM] CIM Specification and CIM Schemata are available at: <http://www.dmtf.org/spec/cims.html>
- [IPSEC] R. Atkinson, "Security Architecture for the Internet Protocol", [RFC1825](#), Aug. 1995.
- [RED] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Volume 1, Number 4, August 1993, pp. 397-413.

INTERNET-DRAFT Terminology for policy and services June 25, 1999

[11.](#) Authors' Addresses

Walter Weiss

Lucent Technologies
300 Baker Ave.
Concord, MA. 01742
Phone: +1-978-287-9130
Fax: +1-978-287-9050
E-mail: wweiss@lucent.com

John Strassner

Cisco Systems
Bldg E
190 West Tasman Drive
San Jose, CA 95134
Phone: +1-408-527-1069
Fax: +1-408-527-2477
E-mail: johns@cisco.com

Andrea Westerinen

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
Phone: +1 425-705-2553
Email: andreawe@microsoft.com

INTERNET-DRAFT Terminology for policy and services June 25, 1999

[10.](#) Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Weiss & Co

Expires in six months

[Page 40]

2360