Internet Engineering Task Force
INTERNET DRAFT

Authors:
R. Rajan/S. Kamat/J. C. Martin
AT&T/IBM/Sun Microsystems
M. See/ R. Chaudhury
IBM/Xylan/ Telstra
D. Verma/ G. Powers/ R. Yavatkar
IBM/ Packeteer/ Intel
5/ April/1999

Policy Action Classes for Differentiated Services and Integrated Services
draft-rajan-policy-qosschema-01.txt

Status of Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 except for the right to produce derivative works.

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.

This draft is in accordance with the policy information model presented in draft-ietf-policy-core-schema-02.txt, and elaborates on the action class for specifying quality of service (QoS) related policies. This draft replaces draft-rajan-policy-qosschema-00.txt by modeling networking QoS policy actions as subclass of policyAction.

Discussion of this draft will be carried on the policy mailing list (policy@raleigh.ibm.com). Suggestions for improvement may also be sent to rajan@research.att.com.

Abstract

This document describes the structure of a directory schema to enable and support administration of QoS policies in networks through regulation or configuration of nodes that support Differentiated Services and/or Integrated Services with RSVP signaling. This draft is consistent with the core schema described in draft-ietf-policy-core-schema-02.txt and must be used in conjunction with draft-ietf-rajan-policy-conditions-00.txt for definition of complete QoS policies.

1. Purpose and Overview

As protocol aspects of providing quality of service in IP networks begin to get standardized, there is a need for corresponding standards towards enabling policy based administration of these protocols. Supporting QoS in a network consumes network resources and policy based administration seeks to regulate such resource usage to ensure that it is consistent with the network service provider's objectives or Service Level Agreements. The contents or nature of the high level objectives or SLAs are clearly not within IETF scope. However, there is a need to standardize lower level schema definitions that facilitate the administration of QoS protocols to ensure interoperability among network devices from multiple vendors.

Currently, there are two sets of standards for providing QoS in IP networks. Integrated services with RSVP [8] signaling approach allows providing per-flow QoS assurances with dynamic resource reservation. A flow is defined by the 5-tuple (source address, destination address, protocol, source port, destination port). In this context, there is a need to provide policy control of individual flows, and to regulate their ability to reserve network resources. (See [9] for a discussion of policy based admission control framework and sample policies). Differentiated services [4], on the other hand, are aimed at traffic aggregates that are identified by the DS code point (part of TOS field) in IP headers of packets. This approach primarily relies on administrative control of bandwidth, delay or dropping preferences through simple policies and configuration rather than per-flow signaling protocols to communicate

the service level information to network elements [3]. For such services we wish to enable flexible definition of class-based packet handling behaviors and class-based policy control. (See [7] for a discussion of DiffServ framework and sample behavior/service descriptions).

In either of these environments, network administrators need the ability to specify different classes of packets and appropriate bounds on the usage of network resources by packets from these classes. For example, with the signaled QoS approach of IntServ with RSVP, a policy may specify an upper limit on the total number of active controlled load reservations from receivers in a particular subnet or the size of individual controlled load or guaranteed service reservations in terms of the request parameters such as token buket rates, delay, etc. In the DiffServ scenario, a policy may specify that an ingress router classify packets from a specific set of applications as requiring DiffServ Expedited Forwarding (EF) [6] treatment, and specify the token bucket parameters for conditioning (policing and shaping) the traffic. In order to provide the desired level of service through consistent per hop behaviours at all nodes for this traffic, the DiffServ policy may also specify the resource allocation information at all network nodes to ensure that the aggregate EF traffic has the well defined minimum departure rate at Similarly, for the Assured Forwarding (AF) [5] class, policies for ingress routers may specify that packets from a certain set of sources be classified into a particular AF class, and further, assign different drop precedence values to packets based on additional criteria. Furthermore, for ensuring the desired level of service, the DiffServ policy may specify traffic conditioning actions at the ingress to ensure that the aggregate traffic belonging to the two lowest drop precedence values in each AF class is within specified limits and appropriate levels of bandwidth and buffer resoures are allocated for the class at all nodes.

1.1. Objectives and Scope

[1] defines five very general classes for defining policies: policyGroup, policyRule, policyCondition, policyTimePeriodCondition, and policyAction. Policy solutions for specific areas, such as QoS and security are expected to use some of these classes directly while creating their own subclasses derived from policyCondition and policyAction in order to represent their own application-specific needs. A subclass of policyCondition to facilitate definition of packet classifiers applicable for both QoS and security policies is proposed in [2]. This draft defines subclasses of policyAction for QoS related policies and is consistent with the above proposals.

This document aims to support QoS policies for differentiated and integrated services networks that fall within the following three broad scenarios:

- Integrated services secured through the use of RSVP signaling, within or across domains. The use of policy in such an environment allows enterprises to be able to police QoS requests on a per-flow, per-user or per-application basis. Directory schema are meant to be used in conjunction with the use of the policy elements in RSVP signaling messages, to enable routers to identify users and applications to which policy must be applied.

- Differentiated services secured through provisioning within a domain, and, in an inter-domain scenario, bilateral agreements across peer network boundaries. In such cases, policies are used to map across the two domain specific semantics, and enforce access control restrictions, such as ensuring that the amount of in-profile traffic is within the specified contractual limits. More specifically, policies for DiffServ facilitate specification of the following:

  1. packet classification filters to be installed into DS-compliant nodes allowing wide granularity for edge nodes and simple DSCP based aggregate traffic filters for DS interior nodes;

  2. appropriate traffic conditioning actions to be performed at different nodes, specifically allowing for marking, metering, policing, shaping and dropping;

  3. bandwidth and buffer resources to be allocated at DS-compliant nodes for different traffic aggregates identified by DS code points.

- Integrated services secured within a domain, being mapped onto differentiated services across domains. In such cases, policies are needed at the domain boundary to translate between integrated and differentiated service semantics, to enforce traffic monitoring and to provide access control to network resources.

Two important scenarios, not explicitly supported by the schema in this draft, but which may be covered by extensions to it are:

- RSVP aggregation, i.e., the mapping of several RSVP flows into pre-configured RSVP tunnels,

- Support of differentiated services using RSVP tunnels.

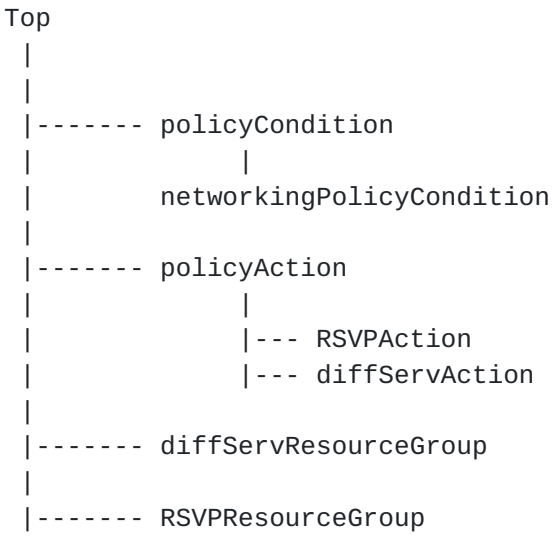
We have the following objectives in defining this schema.

1. We want to cover a broad range of PEP clients that enforce QoS policies. These include: 1) an edge device that marks/drops/buffers/schedules certain packets to enforce a service differentiation policy, 2) an RSVP capable router that accepts/denies resource reservation requests based on allowed policies, and 3) hosts capable of packet marking and traffic conditioning. We would like the schema definition to be generic enough to support a wide range of resource control environments including the clients mentioned above.

2. We want to describe QoS actions in terms of observable or measurable behaviors rather than implementation specific configuration parameters. We believe that adminstrators specifying QoS policies in a heterogeneous multi-vendor network would prefer such descriptive semantics instead of detailed prescription of configuration parameters for specific scheduling or buffer management mechanisms used in network nodes. We assume that translation of QoS action directives to implementation specific actions will be carried out in the policy clients (within PDP/PEP/proxies).

1.2. Class Hierarchy

The new classes and subclasses added to support QoS policy actions are depicted in the class hierarchy below:



2. The Subclass diffServAction

The class policyRule specifies a sequence of actions to be employed on a sub-stream of packets identified through a simple or complex set of conditions. Assuming that a packet substream has been identified (for details, see drafts [1] and [2]), the diffServAction class describes a component per-hop behaviour (PHB) in force at a network device. The diffServAction class does not seek to describe in detail resource reservations, packet treatment and configuration of every possible network device. Instead, it provides a high-level description of QoS services through a simple model that uses the following three descriptors -- traffic profile descriptors, packet marking descriptors and resource group descriptors. Of these, traffic profile descriptors and packet marking descriptors constitute attributes of diffServAction itself, while resource group descriptors are used through reference to another class diffServResourceGroup.

The traffic profile descriptor is used to describe precisely, the portion of the packets of the identified sub-stream that are to be regarded as in-profile, with the understanding that the remaining packets are to be regarded as out-of-profile (or excess traffic). To this end we employ a leaky bucket model with three attributes -- a mean rate, a peak rate and a bucket size. Of course, the sorting of packets into in-profile and out-of-profile packets is possible only under the assumption that a policing device is present at the policy enforcement point; otherwise the attributes may be omitted or will be ignored (i.e., all packets will be regarded as in-profile).

The packet marking descriptor provides an edge device with the ability to mark the DS byte for enabling simpler QoS classification at downstream network devices. We allow for differential marking of in and out of profile traffic (which makes sense if policing is present); as well as for the enforcement point to mark or remark the DS code point portion of the DS byte in the packet header. The latter is facilitated by allowing masking certain bits of the DS byte while modifying others.

The resource group descriptor describes the treatment expected by packets within a common service group (identified by the forwarding class). The descriptor does NOT aim to describe how the service is to be provided, i.e., scheduling, buffer management, and other resource control details cannot be dictated by policy. The resource group descriptor instead encapsulates the (qualitative/quantitative) nature of the service to be accorded to identified packets. There are two aspects of such a description:
(a) The treatment of in-profile traffic: The principal service descriptors are rate, delay and loss. These may be defined deterministically, stochastically or qualitiatively. For instance,

a rate may be defined deterministically through the transmission of a certain number of bits over time period, stochastically through an on-off Markov process, or qualititatively as Class 1 traffic. We allow for deterministic and qualitative descriptions; stochastic descriptors may be introduced as future extension.
(b) Treatment of out-of-profile traffic: Excess traffic may be tolerated, reshaped, provided a different class of service or dropped.

It is important to note that trafficProfileDescriptor and resourceGroupDescriptor objects refer to states created in the network device. To understand this better, consider two policy rules, as follows.

```
Rule1:    If PolicyCondition1 then PolicyDSAction1

Rule2:    If PolicyCondition2  then PolicyDSAction1

PolicyDSAction1:  TrafficProfileDescriptor1
                  PacketMarkingDescriptor1
                  ResourceGroupDescriptor1
```

Now, all packets described either by PolicyCondition1 or by PolicyCondition2 will be policed together, and share the same rate resource reservations. This is different from the case where we have

```
Rule1: If PolicyCondition1 then PolicyDSAction1

Rule2: If PolicyCondition2 then PolicyDSAction2

PolicyDSAction1:  TrafficProfileDescriptor1
                  PacketMarkingDescriptor1
                  ResourceGroupDescriptor1

PolicyDSAction2:  TrafficProfileDescriptor2
                  PacketMarkingDescriptor2
                  ResourceGroupDescriptor2
```

Even if the two traffic profile descriptors were numerically identical, the two streams will not be policed together by the same policer -- they will be policed by identical policers. Similarly, they will not share the same buffer or bandwidth resources. In fact, the resource requirements for the second example will be twice those of the first (ignoring multiplexing gains).

The class description of diffServAction is as follows:

```
NAME              diffServAction
```

```
TYPE               Structural
DERIVED FROM       policyAction
AUXILIARY CLASSES  NONE
MUST
      CommonName
      diffServPermission
MAY
      diffServInProfileRate,
      diffServInProfilePeakRate,
      diffServInProfileTokenBucket,
      diffServInProfileTransmittedTOSByte,
      diffServOutProfileTransmittedTOSByte,
      diffServResourceGroupRef,
      diffServActionName
```

The first three MAY attributes describe the traffic profile, the next two specify the marking, and next is a reference to the resource group.

The following set of attributes are currently defined for the DiffServ policy clients, namely hosts, edge devices, and routers that do traffic conditioning (packet marking, dropping, shaping, etc).

```
NAME     diffServPermission
DESC     Allow/drop data packets
SYNTAX   IA5String
EQUALITY caseExactIA5Match
SINGLE-VALUED
FORMAT   The currently defined values for this attribute are:
            Accept
            Deny
```

With the permission attribute set to ``Accept'', and no other attribute present, the packets matching the PolicyCondition are given the ``default'' service.

```
NAME      diffServActionName
DESC      The user friendly name of this entry.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
SINGLE-VALUED
DEFAULT   No name

NAME      diffServInProfileRate
DESC      Specifies the token rate for the in-profile traffic descriptor in
kbps
SYNTAX    INTEGER
EQUALITY  integerMatch
```

SINGLE-VALUED

```
SEMANTICS All packets in the behavior aggregate are measured against
          a leaky bucket with this token rate. Traffic that passes the leaky
          bucket check is considered in-profile.
DEFAULT   All packets considered in-profile, i.e., infinity


NAME      diffServInProfilePeakRate
DESC      Specifies the peak rate for the in-profile traffic descriptor in kbps
SYNTAX    INTEGER
EQUALITY  integerMatch
SINGLE-VALUED
SEMANTICS All packets in the behaviour aggregate are measured
          against a leaky bucket with this peak rate.
DEFAULT   Same value as diffServInProfileRate


NAME      diffServInProfileTokenBucket
DESC      Specifies the token bucket size for in-profile traffic
          descriptor in kilobits
SYNTAX    INTEGER
EQUALITY  integerMatch
SINGLE-VALUED
SEMANTICS All packets in the behaviour aggregate are measured
          against a leaky bucket with this  token bucket size.
DEFAULT   Defaults to the maximum IP packet size.

NAME      diffServInProfileTransmittedTOSByte
DESC      Specifies the outgoing TOS byte for in profile packet
          marking descriptor
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    String(s) of the form xxxxxxxx:xxxxxxxx, where each of the `x's
          is either 0 or 1.
SINGLE-VALUED
SEMANTICS Each of the two substrings is treated as specifying an 8-bit
          field. The left substring is termed Mask and the right
          substring Modify. 0's in the Mask specify the bit locations
          in the TOS byte that must not be changed and 1's specify
          those that must be changed to match the corresponding ones
          in the Modify field.  The operation involved is: newTOSByte
          = (Mask' & oldTOSbyte) | (Mask & Modify), where Mask' is the
          bitwise complement of Mask and '&' and '|' denote the
          bit-wise AND and OR operations respectively.
EXAMPLE   Consider a policy rule that specifies 11100000:11001010 as
          the value for this attribute. The Mask of 11100000 means
          that when this rule is applied, the 5 least significant bits
          in the TOS byte must be left unchanged but the 3 most
          significant bits must be changed to make them identical to
```

```
          the corresponding ones in the Modify field. Thus, if this
          rule were to be applicable to a packet whose TOS byte is
          10101010, then the TOS byte will be changed to 11001010
          before transmission.
DEFAULT   Don't modify any bit, i.e.,
          Mask   00000000
          Modify 00000000


NAME      diffServOutProfileTransmittedTOSByte
DESC      Specifies the outgoing TOS byte for out-of-profile packet
          marking descriptor
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    same as `DiffServInProfileTransmittedTOSByte' attribute
SINGLE-VALUED
SEMANTICS same as `diffServInProfileTransmittedTOSByte' attribute
DEFAULT   same as `diffServInProfileTransmittedTOSByte' attribute

NAME      diffServResourceGroupRef
DESC      Absolute distinguished name of LDAP entry, from the objectclass
          diffServResourceGroup, that identifies the service category and
          resource group descriptors that apply to the traffic.
SYNTAX    DN
EQUALITY  distinguishedNameMatch
SINGLE-VALUED
DEFAULT   Best Effort Service
```

2.0.1. The class diffServResourceGroup

Objects of this class fully specify the treatment accorded to in-profile and out-of-profile traffic, in terms of their access to QoS resources. The class description of diffServResourceGroup is as follows:

```
NAME               diffServResourceGroup
TYPE               Structural
DERIVED FROM       Top
AUXILIARY CLASSES  NONE
MUST               CommonName
MAY
                   diffServLossParameter,
                   diffServDelayParameter,
                   diffServBandwidthShare,
                   diffServExcessTrafficTreatment
                   diffServAutoStart
```

diffServResourceGroupName

The attributes are described below.

```
NAME      diffServResourceGroupName
DESC      The user friendly name of this entry.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
SINGLE-VALUED
DEFAULT   No name

NAME      diffServLossParameter
DESC      Packet loss paremeters  for in-profile traffic for this
          class of service
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    Colon seperate numeric strings, A:B, where at most A packets
          may be dropped for every B packets received.
SINGLE-VALUED
EXAMPLE   2:1000 implies a maximum loss rate of .002.
DEFAULT   Best Effort

NAME      diffServDelayParameters
DESC      Packet delay paremeters  for out-of-profile traffic for this
          class of service
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    Colon seperated integer:string pair. Currently defined
          1:<absolute delay>  where the absolute delay is expressed in msec
          2:<relative delay > where the relative delay is expressed as
            a priority (1 means best effort)
SINGLE-VALUED
DEFAULT   Best Effort

NAME      diffServBandwidthShare
DESC      Bandwidth share for this class of service
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    Colon seperated integer:string pair. Currently defined
          1:<absolute bandwidth>  where the absolute bw is expressed
            in kilobits/sec
          2:<bandwidth percent> where the bandwidth percent is a number
            between 0 and 100 expressing the share of the bandwidth
            allowed to this class
SINGLE-VALUED
SEMANTICS The intervals over which bandwidths are delivered are PEP-specific.
          Percentages are provided for instances where the policy rule is
unaware
```

of the link capacity at the enforcement entity. Percentages for all

```
          classes must add to less than 100.
EXAMPLES  1:200  -- This class is allocated a 200kbps
          2:40   -- This class is allocated 40% of the link bandwidth
DEFAULT   0, i.e., No reserved share

NAME      diffServExcessTrafficTreatment
DESC      Describes how excess traffic is to be treated.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    The following values are defined
              `Drop'  -- Allow no excess traffic
              `Best Effort' -- Treat excess traffic as best effort
              `Reshape'--Reshape excess traffic
SINGLE-VALUED
SEMANTICS This field specifes the actions that must be taken if an
          incoming packet cannot be placed within the reserved buffer
          allocation of the stream.
DEFAULT   Best Effort

NAME      diffServAutoStart
DESC      Indicates if the resource allocation for this service should be
          done at time of enforcement entity startup, or should be packet
          driven. This attribute is for guidance only, and its
          interpretation is implementation specific.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT    The following strings are defined
             `AutoStart' --- Allocate resources at device startup
             `NoAutoStart' --- Allocate resources when packets for
                             flow are seen.
SINGLE-VALUED
DEFAULT   Autostart
```

2.1. The Class RSVPAction

The class RSVPAction contains policies to be applied to RSVP signalling packets, i.e., PATH messages and RESV messages, satisfying the conditions specified in the policy conditions. In this draft of the schema we allow for simple forms of policy based control, where administrative restrictions may be placed on the amount of resources that a single RSVP flow, or group of flows, may consume. We also allow for an RSVP reservation to be supported through a mapping into a DiffServ service category. Support for additional rules based on richer classes of policy information such as user ids for signalled QoS will be supported in future extensions to this schema.

Currently, there are two kinds of restrictions that we may place on resource usage by RSVP flows -- individual restrictions and group restrictions.

The class description of RSVPAction is as follows:

```
NAME               RSVPAction
TYPE               Structural
DERIVED FROM       policyAction
AUXILIARY CLASSES  NONE
MUST               CommonName
                   RSVPFlowServiceType,
                   RSVPPermission
MAY                RSVPActionName
                   RSVPMaxRatePerFlow,
                   RSVPMaxTokenBucketPerFlow,
                   RSVPMinDelay,
                   RSVPMaxFlowDuration,
                   RSVPResourceGroupRef,
                   diffServActionRef
```

The syntax and semantics of the attributes of an `RSVPAction' entry are described below.

```
NAME      RSVPFlowServiceType
DESC      IntServ service type that a flow can request
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT:   String with allowed values
           ControlledLoad
           GuaranteedService
MULTI-VALUED

Name      RSVPPermission
DESC      Allow/Dissallow RSVP flows
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
FORMAT:   String with allowed values
            Accept
            Deny
SEMANTICS Accept or deny RSVP sessions of the specified Service Type(s).
          The remaining attributes make sense only in the case of `Accept'
SINGLE-VALUED

NAME      RSVPActionName
DESC      The user friendly name of this entry.
```

```
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
SINGLE-VALUED
DEFAULT   No Name

NAME      RSVPMaxRatePerFlow
DESC      The maximum token rate for any individual flow in kilobits per second
SYNTAX    INTEGER
EQUALITY  integerMatch
SINGLE-VALUED
SEMANTICS  Reservation requests for higher per-flow bandwidth are denied.
DEFAULT    No limit

NAME       RSVPMaxPeakRatePerFlow
DESC       The maximum peak rate for any individual flow in kilobits per second
SYNTAX     INTEGER
EQUALITY   integerMatch
SINGLE-VALUED
SEMANTICS  Reservation requests for higher per-flow peak rate are denied.
DEFAULT    Assigned the same value as RSVPMaxRatePerFlow.

NAME       RSVPMaxTokenBucketPerFlow
DESC       The maximum token bucket size for any individual flow in kilobits
SYNTAX     INTEGER
EQUALITY   integerMatch
SINGLE-VALUED
SEMANTICS: Reservation requests for higher per-flow token bucket size
           are denied.
DEFAULT    Implementation Specific.

NAME       RSVPMinDelay
DESC       The minimum delay value an individual flow may request in millisec
SYNTAX     INTEGER
EQUALITY   integerMatch
SINGLE-VALUED
DEFAULT    No limit

NAME       RSVPMaxFlowDuration
DESC       Maximum time (in seconds) an RSVP flow matching the profile may last
SYNTAX     INTEGER
EQUALITY   integerMatch
SINGLE-VALUED
DEFAULT    No limit


NAME       RSVPUserAuthPolicy
DESC       Manner of authentication to be performed to authenticate user
SYNTAX     IA5String
```

```
EQUALITY   caseExactIA5Match
SINGLE-VALUED
FORMAT: String, currently defined values are
           Plain       (Plain text password)
           Kerberos    (Kerberos ticket authentication)
           Public-Key  (Public Key authentication)
           None        (for no authentication)
DEFAULT    None
```

All the policy attributes hitherto described for RSVPAction refer to an individual flow for which a reservation is sought to be made. Often an adminstrator might wish to place group restrictions on flows described as an aggregation of multiple policy condition objects. For instance, the administrator might wish to restrict the total number of active RSVP reservations. To facilitate such group restrictions, we allow the reference attribute RSVPResourceGroupRef. The reason for making this a reference is not difficult to see. The group that the adminstrator wishes to control may not be describable through a single profile, or a single profile might belong to different groups in terms of resource control. In such cases, multiple policy rules ``point'' to the same group. Note that policies described through group rules require that the enforcement entity maintain some state; in the example suggested above, the enforcement entity would have to track the number of active flows in order to enforce the policy.

```
NAME       RSVPResourceGroupRef
DESC       Absolute distinguished name(s) of LDAP entry, from the objectclass
           RSVPResourceGroup, which specifies constraints on a group of
           RSVP flows.
SYNTAX     DN
EQUALITY   distinguishedNameMatch
MULTI-VALUED
DEFAULT    No Resource Group
```

The next attribute allows the enforcement entity to map RSVP flows onto DiffServ resource groups.

```
NAME       DiffServActionRef
DESC       Absolute distinguished name of an LDAP entry, from the objectclass
           DiffServAction, which specifies the class of service that the
           RSVP flow must be mapped into.
EQUALITY   distinguishedNameMatch
SINGLE-VALUED
SYNTAX     DN
DEFAULT    No RSVP to DiffServ Translation
```

2.2. The Class RSVPResourceGroup

The class description of RSVPResourceGroup is as follows:

```
NAME               RSVPResourceGroup
TYPE               Structural
DERIVED FROM       Top
AUXILIARY CLASSES  NONE
MUST
                   CommonName
MAY
                   RSVPMaxFlows
                   RSVPMaxAggregateRate
                   RSVPMaxAggregateTokenBucket
                   RSVPResourceGroupName
```

The syntax and semantics of individual attributes are described below.

```
NAME      RSVPResourceGroupName
DESC      The user friendly name of this entry.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
SINGLE-VALUED
DEFAULT   No Name

NAME      RSVPMaxFlows
DESC      The maximum allowed number of reserved flows belonging to the
          group
SYNTAX    INTEGER
EQUALITY  integerMatch
SINGLE-VALUED
DEFAULT   No limit

NAME      RSVPMaxAggregateRate
DESC      The aggregate maximum token rate for all flows in the group
SYNTAX    INTEGER
EQUALITY  integerMatch
SINGLE-VALUED
SEMANTICS Reservation requests that result in a higher aggregate
          bandwidth reservation are denied.
Default   No limit

NAME      RSVPMaxAggregateTokenBucket
DESC      The maximum token bucket size for the aggregate traffic matching
          the profile in kilobits
SYNTAX    INTEGER
EQUALITY  integerMatch
```

SINGLE-VALUED
DEFAULT No limit

3. QoS Schema Usage Examples

In this section we describe some usage scenarios for defining QoS policies for different contexts. We use LDIF notation. These examples broadly adhere to the core Policy schema as defined in [1], and to the policyCondition class as defined in [2]. However, for clarity of exposition we simplify certain attribute syntaxes (policyRuleConditionList, for instance).

3.1. DiffServ PHB

The requirements are:

- All web traffic originating from two server clusters S1 (1.1.1.0 mask 255.255.255.0) and S2 (2.2.2.0 mask 255.255.255.0) traversing a router must be assigned a low delay, low loss service with a shared reservation of 40Mbps.

- This traffic must be marked with the TOS bits set to 10100000.

The following rules achieve the above objective:

```
dn: cn=S1-Web-Rule, o=XYZ, c=US
Objectclass: policyRule
policyRuleConditionList: cn=S1-Web-Condition,o=XYZ, c=US,
policyRuleActionList: cn=DSGoldService, o=XYZ, c=US

dn: cn=S2-Web-Rule, o=XYZ, c=US
Objectclass: policyRule
PolicyRuleConditionList: cn=S2-Web-Condition,o=XYZ, c=US,
PolicyRuleActionList: cn=DSGoldService, o=XYZ, c=US
```

The conditions and actions referred to in the above rules are:

```
dn: cn=S1-Web-Condition, o=XYZ, c=US
Objectclass: networkingPolicyCondition
Objectclass: hostConditionAuxClass
Objectclass: applicationConditionAuxClass
sourceIPAddressRange: 1:1.1.0:24
SourcePortRange: 8000:8080
protocolNumber: 4                                (TCP)
```

```
dn: cn=S2-Web-Condition, o=XYZ, c=US
objectclass: networkingPolicyCondition
objectclass: hostConditionAuxClass
objectclass: applicationConditionAuxClass
sourceIPAddressRange: 1:2.2.2.0:24
sourcePortRange: 8000:8080
protocolNumber: 4                                (TCP)

dn: cn=DSGoldService, o=XYZ, c=US
Objectclass: DiffServAction
DiffServPermission: Permit
DiffServInProfileTransmittedTOSByte: 11111111:1010000
DiffServResourceGroupRef: cn=S1-S2-WebDSResourceGroup, o=XYZ, c=US

dn: cn=S1-S2-WebDSResourceGroup, o=XYZ, c=US
ObjectClass: DiffServResourceGroup
DiffServQueuePriority: 1                  (implementation specific)
DiffServLossParameter: 1:1000000
DiffServDelayParameter: 1:1
DiffServBandwidthShare: 40000             (kbps)
DiffServAutoStart: AutoStart
```

3.2. DiffServ Policing

Now, consider a policy rule that allows for no more than an aggregate of 5000 kilobits/second of best effort traffic from sources in subnet S3 (range 139.24.2.12 to 139.24.2.255).

```
dn: cn=S3-Policing-Rule, o=XYZ, c=US
objectclass: policyRule
policyRuleConditionList: cn=S3-Condition,o=XYZ, c=US,
policyRuleActionList: cn=S3-DS-Action, o=XYZ, c=US


dn: cn=S3-Condition,o=XYZ, c=US,
objectclass: networkingPolicyCondition
objectclass: hostConditionAuxClass
sourceIPAddressRange:   2:139.24.2.12:139.24.2.255

dn: cn=S3-DS-Action, o=XYZ, c=US
Objectclass: PolicyAction
DiffServInProfileRate: 5000
DiffServInProfileTokenBucket: 1024
DiffServResourceGroupRef: cn=BestEffortPolicing, o=XYZ, c=US

dn: cn=BestEffortPolicing, o=XYZ, c=US
DiffServQueuePriority: 1
```

```
DiffServExcessTrafficTreatment: drop
DiffServAutoStart: NoAutoStart
```

Here, we have allocated a nominal token bucket to take care of the maximum packet size.

3.3. Forbidding RSVP Sessions

Suppose that RSVP traffic from a subnet S1 is to be denied access to the network during the working day (9 am to 5 pm). We have the following entry to express this policy.

```
dn: cn=S1-RSVP-Rule, o=XYZ, c=US
objectclass: Policy
policyRuleConditionList: cn=S1-RSVP-Condition,o=XYZ, c=US,
policyRuleActionList: cn=S1-RSVP-Action, o=XYZ, c=US
policyRuleValidityPeriodList: cn=workinghrs, o=XYZ, c=US

dn: cn=S1-RSVP-Condition,o=XYZ, c=US
objectclass: networkingPolicyCondition
objectclass: hostConditionAuxClass
sourceIPAddressRange: 1:1.1.0:4

dn: cn=workinghrs, o=XYZ, c=US
Objectclass: PolicyValidityPeriod
PolicyValidityTimeOfDayRange: 090000:170000

dn: cn=S1-RSVP-Action, o=XYZ, c=US
Objectclass: RSVPAction
RSVPFlowServiceType: ControlledLoad
RSVPFlowServiceType: GuaranteedService            (multi-valued)
RSVPPermission: Deny
```

3.4. Controlling RSVP Reservations

Consider a policy that specifies that during after hours (5 pm to 9am) each RSVP controlled load reservation for outgoing traffic from subnet S1 have a token rate of no more than 1 Mbps, that there be no more than 100 such reservations active at any time, and that the aggregate reservable amount from that subnet total to no more than 10 Mbps.

```
dn: cn=S1-CL-nw-Rule, o=XYZ, c=US
objectclass: policyRule
policyRuleConditionList: cn=S1-CL-nw-Condition,o=XYZ, c=US,
policyRuleActionList: cn=S1-CL-nw-Action, o=XYZ, c=US
```

```
PolicyValidityPeriodList: cn=non-workinghrs, o=XYZ, c=US

dn: cn=S1-CL-Condition,o=XYZ, c=US
objectclass: networkingPolicyCondition
objectclass: hostConditionAuxClass
sourceIPAddressRange: 1:1.1.0:4

dn: cn=non-workinghrs, o=XYZ, c=US
Objectclass: PolicyValidityPeriod
PolicyValidityTimeOfDayRange: 170000: 090000

dn: cn=S1-RSVP-Action, o=XYZ, c=US
Objectclass: RSVPAction
RSVPFlowServiceType: ControlledLoad
RSVPPermission: Permit
RSVPMaxRatePerFlow: 1000
RSVPResourceGroupReference: cn=S1-RSVPGroup, o=XYZ, c=US

dn: cn=S1-RSVPGroup, o=XYZ, c=US
RSVPMaxFlows: 100
RSVPMaxAggregateRate: 10000
```

4. Security Considerations

There are two potential security considerations, both of which may be addressed through standards compliant mechanisms. The first is the unauthorized access to read or change policy rules and related objects in the directory repository. The schema in this document SHOULD be used in conjunction with an LDAP access control mechanisms, see for instance [15]. The second exposure for violation of security lies in the communication between policy decision point and the directory repository. Such communication SHOULD be secured, with both ends mutually authenticated using SSL/TLS or IPSec.

Acknowledgments

Thanks to Partha Bhattacharya, Tim Moore, Roch Guerin, Dimitrios Pendarakis and Ellen Stokes for useful discussion and suggestions in this problem space. In addition, we also thank numerous others who have read and commented on this draft in various forms.

References

[1] J. Strassner, E. Ellesson, and B. Moore (editor), Policy Framework Core Information Model, Internet Draft, draft-ietf-policy-core-schema-02.txt, Feb. 1999.

[2] R. Rajan, S. Kamat, and P. Bhattacharya, Networking Policy Condition Information Model, Internet Draft, draft-ietf-policy-conditions-01.txt, Apr. 1999.

[3] K. Nichols, S. Blake, F. Baker and D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC2474, Dec. 1998.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, RFC2475, Dec. 1998.

[5] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, Assured Forwarding PHB Group Internet Draft, draft-ietf-diffserv-af-06.txt, Feb. 1999.

[6] V. Jacobson, K. Nichols, and K. Poduri, An Expedited Forwarding PHB, Internet Draft, draft-ietf-diffserv-phb-ef-02.txt, Feb. 1999.

[7] Y. Bernet, et al, A Framework for Differentiated Services, Internet Draft, draft-ietf-diffserv-framework-02.txt, Feb. 1999.

[8] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification. RFC2205, Sept. 1997.

[9] R. Yavatkar, R. Guerin and D. Pendarakis, A Framework for Policy-based Admission Control Internet Draft, draft-ietf-rap-framework-01.txt, Nov. 1998.

[10] S. Herzog, A. Sastry, R. Rajan, R. Cohen, J. Boyle, and D. Durham, The COPS (Common Open Policy Service) Protocol Internet-Draft, draft-ietf-rap-cops-06.txt, Feb. 1999.

[11] W. Yeong, T. Howes and S. Kille, Lightweight Directory Access Protocol, RFC1777, Mar. 1995.

[12] M. Wahl, A. Coulbeck, T. Howes and S. Kille, Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions Internet Draft draft-ietf-asid-ldapv3-attributes-07.txt, August 1997.

[13] S. Judd and J. Strassner, Directory Enabled Networks - Information Model and Base Schema - Draft v3.0c5 DEN Specifications, Sep. 1998.

[14] Desktop Management Task Force, Common Information Model (CIM) Specification, Version 2.0, Mar. 1998.

[15] E. Stokes, D. Byrne, B. Blakeley and P. Behera, Access Control Requirements for LDAP, Internet Draft, Sep. 1998.

[16] R. Droms, Dynamic Host Configuration Protocol, RFC1541, Oct. 1993.

AUTHORS' ADDRESS

Raju Rajan
AT&T Labs - Research
180 Park Avenue, P.O. Box 971
Florham Park, NJ 07932-0971
email: rajan@research.att.com

Sanjay Kamat
IBM Research
30 Saw Mill River Road
Hawthorne, NY 10532
email: sanjay@watson.ibm.com

Jean-Christophe Martin
Solaris and Network Software Group
Sun Microsystems
France
email:jean-christophe.martin@sun.com

Michael See
email: Michael.See@xylan.com

Rajiv Chaudhury
email: rchaudhu@telstra.com.au

Dinesh Verma
IBM Research
30 Saw Mill River Road
Hawthorne, NY 10532
email: dverma@watson.ibm.com

George Powers
email: george@packeteer.com

Raj Yavatkar
Intel Corporation, JF3-206
2111 NE 25th Avenue,
Hillsboro, OR 97124
email: raj.yavatkar@intel.com

Full Copyright Statement