# A Policy Framework for Integrated and Differentiated Services in the Internet

**Raju Rajan**
**AT&T Labs**

**Dinesh Verma**
**IBM T. J. Watson Labs**

**Sanjay Kamat**
**Bell Labs**

**Eyal Felstaine**
**Allot Communications***

**Shai Herzog**
**IP Highway**

## Abstract

One of the key issues in deployment of QoS is determining the set of applications or users, which are allowed to have a preferential access to network resources. The administrative criteria for regulating access to resources constitute the QoS policies. A policy could determine which of the reservation requests in the network be honored during the processing of a signaling protocol such as RSVP, or it could determine the class of applications or users which are to be placed in a specific DiffServ class of service.

In this paper, we look at the issues that arise in the definition, deployment and management of policies related to QoS in an IP network. The paper provides an overview of the requirements for QoS policies, the alternate policy architectures that can be deployed in a network, the different protocols that can be used to exchange policy information, and the exchange of policy information among different administrative domains. We provide a coverage of the current issues being examined in IETF and other standard bodies, as well as issues explored in policy related research ongoing at different universities and research labs.

## 1. Introduction

The Internet is transitioning from a best-effort service model where all transmissions are considered equal and no delivery guarantees are made, to one that can provide predictable and different service levels for specific Quality of Service requirements. This transition is driven as much by qualitatively diverse requirements of emerging applications, as by the market push for service differentiation. Various mechanisms and protocols proposed for integrated and differentiated services seek to provide inter-operable, customizable solutions that may be deployed throughout a network. These solutions would involve the participation of different types of network equipment: end-hosts, network access routers,

* and the Computer Science Department, Technion, Haifa, Israel.

backbone routers and switches. Introduction of these services implies that network devices now need to discriminate between different packets in contrast to existing best-effort networks, which treat all packets equally. The vigorous interest in quality of service (QoS) issues within the Internet community is evidenced by the rapid development of two IP standards in the last few years -- the RSVP signaling (IntServ/RSVP) approach [6], and the differentiated services (DiffServ) approach [8,9].

Integrated services with RSVP signaling approach attempts to provide per-flow QoS assurances with dynamic resource reservation. A flow is defined by the 5-tuple consisting of source IP address, destination IP address, transport protocol, source port, and destination port. In this context, there is a need to provide policy control of individual flows, and regulate their ability to reserve network resources. (See [1] for a discussion of policy based admission control framework and sample policies).

Differentiated services (DiffServ), on the other hand, are aimed at traffic aggregates that may not correspond to fine-grained flows. In a differentiated service environment, network devices may take on responsibilities for classifying packets into aggregates of desired granularity, policing traffic, and allocating static or dynamic resources to satisfy QoS requirements. The DiffServ approach relies on administrative control of bandwidth, delay or dropping preferences, rather than per-flow signaling to communicate the service level information to network elements. For such services we wish to enable flexible definition of class-based packet handling behaviors and class-based policy control.

While these new services standardize mechanisms for delivering QoS, there is an equally important regulatory infrastructure that needs to be developed. Namely, network administrators need means to regulate which users, applications or hosts should have access to what resources/services and under what conditions. Large-scale deployment of such services is critically dependent on the presence of a network-wide policy infrastructure that allows Internet Service Providers (ISPs) and corporate administrators to regulate the network rather than configure individual devices.

For instance, an e-tailer (on-line retailer) may wish to provide preferential treatment to real-time transaction oriented web-traffic; or an ISP may seek to ensure that voice-over-IP is assigned to a low-loss, low-delay class of service, while limiting the number of simultaneously supported voice calls. Or consider a network administrator of an RSVP capable intranet who wishes to restrict individual Controlled Load reservations from certain sources during the day to a certain token rate and also limit the total bandwidth of such reserved flows. In these and other examples, the utility of a QoS service depends heavily on administrative mechanisms to regulate access to network resources based on categories such as users, hosts, applications, accounts, etc. The *policy infrastructure* is the set of protocols, information models and services that allow administrative intentions to be translated into differential packet treatment of network packet flows. In this article, we provide an overview of several components of such an infrastructure, and discuss how these fit into a scalable, cross-service policy solution.

### 1.1 What is Policy?

Given the wide and varied usage of the term "policy", it is imperative that the word be clearly defined in the networking context. As a starting point, we shall use policy to denote the unified regulation of access to network resources and services based on administrative criteria. The schematic in Figure 1 describes different levels at which regulation may be expressed and exercised. The *network view* of policy is an intuitive,

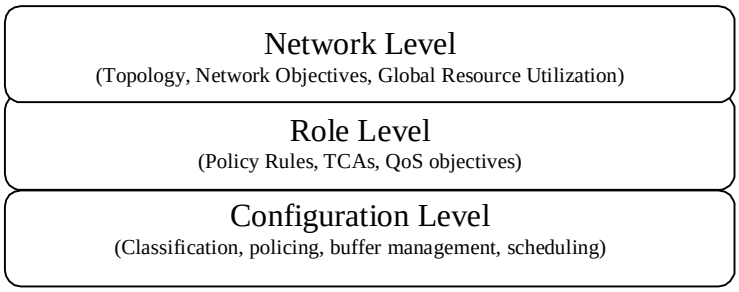


Figure 1: A conceptual policy hierarchy

high-level perspective of topology, connectivity, end-to-end performance objectives and dynamic state of the network. The network view is composed of different *nodal views*, which correspond to the policy objectives and requirements at various network nodes. These, in turn, are composed of policy rules, which may be regarded as atomic injunctions through which various network nodes are controlled. Each network node has vendor-specific resource allocation mechanisms. Hence, packet forwarding paths and nodal views need to be ultimately translated into *device-specific* instructions. For example, the network view corresponds to the intention of an administrator that all http traffic from server A to managers in campus B (see Figure 2) must be allocated an end-to-end reservation of 2 MBps, while all UDP traffic is transmitted as best effort. This is composed of multiple nodal views, each corresponding to a different device that could participate in enforcing the networking policy. For instance, server A may be instructed to mark the ToS (Type of Service) field in the IP headers for web pages accessed by managers in campus B, while an access router assigns these marked packets to a pre-established reservation. In this case, the different roles played by different devices depend on their capabilities (for instance, whether a router is RSVP capable or not) as well as their location in the network. The nodal policy is a totality of rules that are to be enforced at that device. For instance, nodal policy at server A is composed of separate rules for dealing with http and UDP traffic. Finally, the implementation of these rules may depend on the hardware and software architecture and on the particular configuration of server A.

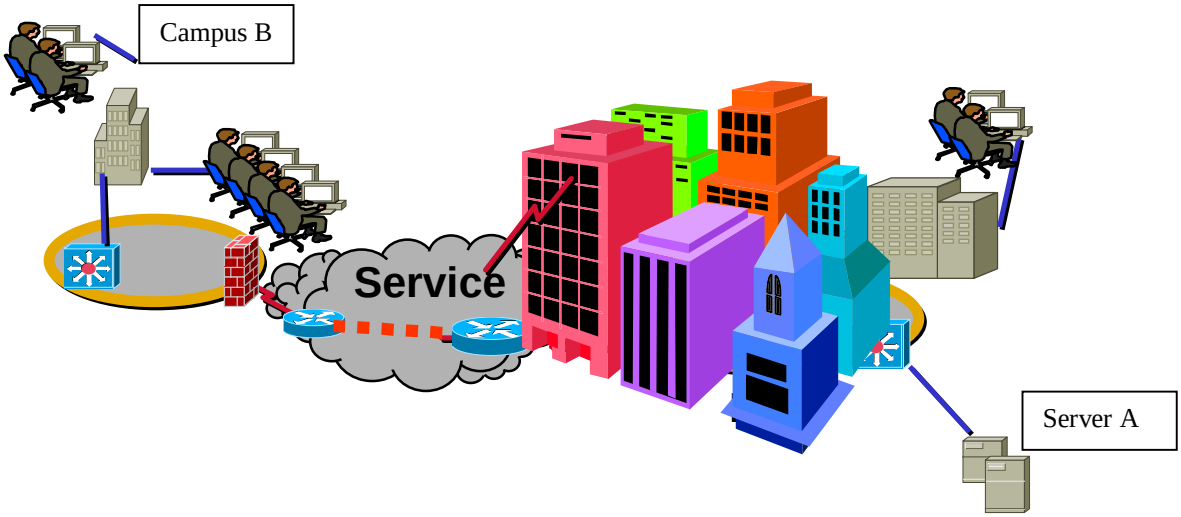


Figure 2: Example of policy views within a network

Within this abstract framework, there is need for standard ways of describing, storing, and communicating policy information so that policy based network management solutions can work in heterogeneous multi-vendor networking environments In the next section we describe the objectives of and requirements for QoS policy. Following this, in Section 3, we outline a functional policy architecture and its relationship to recently developed policy protocols and information models. Section 4 provides an in-depth examination of the challenges of fashioning an end-to-end QoS solution using policy as the glue that binds different intra-domain QoS solutions. Section 5 concludes this article with a summary of key issues and new directions for policy research.

## 2. Requirements for QoS Policy

Starting with the broad aim that administrators require mechanisms to control access to QoS resources, we are drawn to describe the specific bases or criteria for discrimination in QoS networks. These would include:

1. The end-points of communication: The source and destination IP addresses and subnets may be directly obtained from the IP packet, as long as no intermediate proxies that encapsulate packets are involved. Other information, such as source/destination MAC addresses or other layer 2 end-point information may be used to regulate communication. Finally, abstract information such as DNS name prefix and BGP domain identification can be bound into IP addresses and checked against the end-points of communication.
2. The route or path of communication: While policy may not be used directly to route packets (as its interactions with network routing packets is not well understood), the treatment of packets and the availability of resources depends on the route along which the data packets flow. For instance, packets flowing over a dedicated line may require no particular reservation, while the same packets routed over a backup public network would need special treatment. Simplistic routing-based policies may be

described based on the incoming or outgoing interfaces of devices at which the policy is enforced.

3. Communicating users or groups: The identity and organizational status of people involved in communication plays a large role in determining their access to network resources. Rich and versatile policy solutions are made possible by the availability of this information within the network, for instance, using signaling protocols such as RSVP, or domain user-name repository.
4. Application information: The characteristics of the particular application generating traffic, whether it has real-time requirements, for instance, will determine the quality and nature of resources allocated to the traffic. While some such information may be deduced from port and protocol numbers in IP packets, or though content inspection of the packet flow. A more comprehensive solution is possible when the traffic generating host is involved.
5. Dynamic Network Characteristics: It is often useful to take the availability of resources in the network, or their scarcity, into account while allowing or denying the use of resources by a particular flow or group of flows.
6. Time-of-Day: Policy may need to adapt to administrative changes. These may change at different times of the day, or at special dates. Policies that are not universally active become active according to the validity period associated with them.

While certain information is available easily from data packets themselves, many useful policies may require that information communicated out-of-band from the traffic source (user identities, for instance), or may require co-ordination amongst multiple network nodes (policies enforcing limits on network usage, for instance). Policy deployment may be considerably enhanced through improved mechanisms for communicating information required for policy enforcement.

The criteria listed above for discriminating amongst flows or groups of flows are not peculiar to QoS policy. However, there are specific ends that are sought to be achieved in the case of integrated or differentiated services through control over resource allocation.

**Integrated services secured through the use of RSVP signaling:** RSVP has been devised to explicitly carry and process policy objects together with reservation requests and responses. In its simplest form, policy may be used to control the number, size and the nature of RSVP reservation requests depending on the information in the header of RSVP Path and Resv messages, or the TSpec and RSpec parameters. This 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. However, a number of interesting and important forms of policy may only be expressed using policy objects carried with RSVP signaling messages. These include objects that identify and authenticate users, applications or hosts, define the relative (reservation) priorities, carry accounting and charging information, etc.

**Proxy RSVP:** This refers to the use of policy to control the establishment of RSVP tunnels between routers or other intermediate devices within the network, and the use of

these tunnels by traffic flowing through these intermediate devices. There are three common proxy instances:

- The first where RSVP tunnels best effort traffic,
- RSVP aggregation, i.e., combining multiple RSVP tunnels into one,
- Other QoS protocol to RSVP translation (eg. DiffServ over RSVP tunnels).

**Differentiated services secured through provisioning**: This includes the case of using policy to specify and control DiffServ within a domain, and, in an inter-domain scenario, control bilateral agreements across peer network boundaries. In such cases, policies are used to map bilateral agreements 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. This is explained in detail in Section 4.

**Multiple QoS Protocols Tunneling through DiffServ:** RSVP or other QoS protocols may be used within a domain, and mapped onto differentiated services across domains. In such cases, policies are needed at the domain boundary to translate between signaled and differentiated service semantics, to enforce traffic monitoring and access control to network resources.

## 3. Policy architecture

This section describes how policy components may be deployed in a single administrative domain, e.g. a corporate intranet or an individual ISP. The next Section (Section 4) discusses how policy components may communicate across multiple administrative domains.

### The Three Tier Policy Model

A generalized model of the policy architectural framework can be visualized as in Figure 3. The architecture shows the different policy related components that may be operational within a single *policy domain*, i.e., a portion of the network that is administered and managed using a common set of policy definitions.

The figure shows a three-tier model for policy definition and enforcement. The three tiers consist of (a) PEP or policy enforcement point, (b) a PDP or Policy Decision Point and (c) a Policy Repository.

The PEP is the component that actually encounters the packets and is responsible for enforcement and execution of policy actions.. It would typically be co-located with the packet forwarding component of an access router or network server. The PEP is an operational component that can take actions like filtering, packet marking, rate enforcement, shaping, resource management, etc.

The PDP is the component that is responsible for determining what actions are applicable to which packets. The PDP interprets policy rules for one or more PEPs based on

information contained in data or signaling packets, current network conditions, as well as dynamic information such as account balances, dynamically allocated addresses, etc. As an example, the PDP could decide whether a specific reservation request ought to be honored or rejected depending on the identity of the originator of the request. A PEP may query the PDP to make decisions on its behalf on the occurrence of specific events, such as the arrival of a new reservation request or a data packet.

The policy repository is the location where the policies defined for the domain are stored. The repository may be located in a single physical site within the policy domain, or it may be replicated at several devices. The repository could be a database, a flat file, an administrative server or a directory server.

Policies are stored in the repository by means of a policy management tool. The policy management tool can also provide functions that validate that policies stored in the repository are well-formed, mutually consistent and can be satisfied by the network.

**The Two Tier Policy Model**

The two-tier policy model is a simplification of the three tier policy model, and is shown in Figure 4. The two tier model originates from the fact that some network components may be powerful and flexible enough to make their own policy decisions, in addition to enforce them. Examples of such network components would include most network servers and high-end routers which typically have a processor dedicated to control operations. In these network components, the PEP and PDP are combined into a single entity, shown as the policy client in the two-tier model.

In a network, it is possible to have a combination of the two and the three tier policy model. Resource-constrained routers may access an intermediate PDP for their policy decisions, while servers and high-end routers may implement the PEP and PDP functionality in a single box.

**Policy Protocols:**

In either the three-tier or the two-tier policy models, there is a need for standard protocols to exchange information between the different tiers. A set of standard protocols needs to be used to enhance inter-operability of vendor products and to ensure an open solution to the policy problem.

For communication between the PEP and PDP, COPS [1] is the standard protocol for exchanging policy information and decisions between the PEP and the PDP. Unlike legacy control protocols such as Command Line Interpreter (CLI) and SNMP, COPS was designed to operate with minimal overhead, reliably and in real-time, to provide a dedicated QoS controller for the PEP. COPS was originally designed to provide policy control for the RSVP protocol [6], however, its design is generic enough to be extended for other policy decisions. For example, a proposal for extending COPS for Differentiated Services policy has been submitted to the IETF [10]. The communication between the

PDP and the policy repository may be performed via several protocols, depending on the nature of the policy repository. When the policy repository is a network directory, LDAP [3] is the protocol of choice for most vendors. When the policy repository is a database, standard SQL queries can be used to access the policies stored at a repository.

The Diameter protocol [4] was designed for authentication of remote users dialing into an office or a server. Proposals have been made to extend it for supporting Quality of Service policies [5] as well.

Current industry momentum appears to evolve towards the use of COPS for exchange of RSVP policies, the use of LDAP for repository access, and the use of Diameter for user authentication and accounting.

**Policy Schema Representation**

The policy is represented in a repository as a set of rules. The syntax and semantics of the policy representation at the repository need to be standardized to permit inter-operation among different vendors. When the policy repository is an LDAP directory, the policy description is done by means of an LDAP schema. The schema defines the types of entries that can be stored in the repository, as well as the number of attributes in each type of entry, and the relationships between the different types of entries.

The policy schema proposed in IETF [7] consists of three main types of entries: policies, conditions, and actions. A policy entry specifies rules with the general semantics of "if *condition*, then *action*". Policy conditions are clauses used to identify a sub-stream of packets to which the policy rule applies while policy actions specify the treatment that these packets must receive. Policy conditions are expressed in terms of administrative categories such as hosts, users, applications, etc. Simple policy conditions are based on IP header fields such as source and destination addresses, protocol and TCP or UDP header fields such as source and destination port numbers. Actions are service specific, for example, an RSVP action would be to allow or disallow a reservation request and a DiffServ action would specify the outgoing TOS field value.
Other aspects stored in the policy repository include information about the time of day and days when the policy is valid, as well as specifying priority information so that situations where more than one conflicting policies are applicable can be resolved by the PDP.

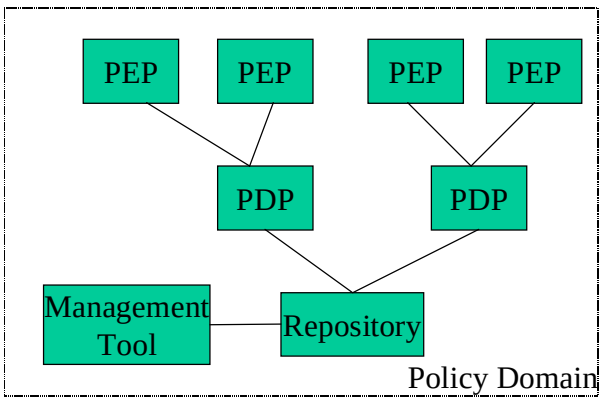


Figure 3. Three Tier Policy Model

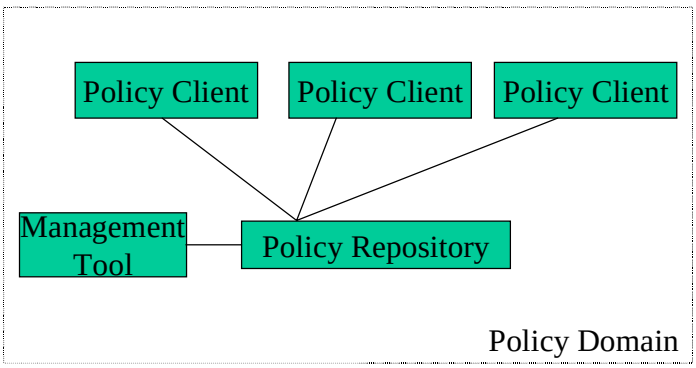


Figure 4: Two Tier Policy Model

## 4. Inter-Domain Policy architecture.

Previous sections describe the use of policy within a single domain. In a multi-domain Internet, it is unrealistic and unscalable to assume a single point of administrative control for the entire network. A scalable alternative direction is to emulate the model of BGP inter-domain routing in relying on bilateral communication for exchanging policy

information between independent domains. This section describes the enforcement of inter-domain QoS policies using bilateral mechanisms called bandwidth brokers (BB) [2], and the related network architecture based on policy aware edge-devices [9].

Each domain is placed under the administrative control of a bandwidth broker. Adjacent domains negotiate in order to determine the nature and extent of traffic that will traverse across their common boundaries. As part of this process, each domain describes its requested level of service to its neighbor's bandwidth broker. The latter provides an admission decision based on its resource availability, bilateral financial arrangements as well as the set of administrative policies in effect. The decision is enforced by monitoring incoming flows into each domain. Consider for example a Service Level Agreement (SLA) between two neighboring domains that specifies a *premium service* that guarantees packet delivery for a given traffic profile by reserving the required resources for this traffic. Thus, to support such a premium service, the BBs in these domains reach an agreement called Traffic Control Agreement (TCA) regarding resource reservation for the specific traffic that is represented by a set of traffic control parameters.

In the framework of a conceptual hierarchy model depicted in Figure 1, an SLA is a Network Level element whereas a TCA is a Role level element. In the inter-domain context, edge routers that are at the boundary between two domains play a special role in the enforcement of the SLA between the two domains. The TCA is essentially the portion of the SLA parameters that relates to the role of a specific edge router as the network "front end".

A bandwidth broker (BB) is conceptually a **policy management** entity located in a policy-enabled domain. It negotiates SLAs with neighboring domains and assures compliance of the administered domain with the SLAs contracted. The BB performs three distinct tasks:

1. Negotiation of SLAs with BBs of neighboring domains.
2. Translation of SLAs into one or several TCAs for edge devices.
3. Delivery of the TCAs to the edge routers of the administered domain, using one of many proposed protocols.

Consider the three networks depicted in Figure 5. Assume that Domain 1 represents an intranet, Domain 2 a local ISP, and Domain 3 a large backbone (tier-1) ISP. If we assume that the needs of Domain 1 toward Domain 3 are satisfied by a 64 Kbit/sec flow of premium traffic then the following operations take place. First, BB1 learns (internally) that a 64 Kbit/sec SLA is needed. This can be triggered either by explicit reservation requests from BB1 or by detection of a 64 Kbit/sec flow of premium packets at router R1. Next, BB1 requests the SLA from BB2 (step1). BB2 performs admission control based on whether there are sufficient premium resources available, and whether the commercial agreement allows such requests. If the request is admitted, BB2 sends a TCA derived from the SLA requested to R2 - its administered edge router (step 2) and responds positively to BB1 (step 3). This TCA models the traffic to be transferred from Domain 1

via R2. A similar TCA is sent by BB1 to its administered edge router R1, instructing it to allow the given traffic to flow out to Domain 2.

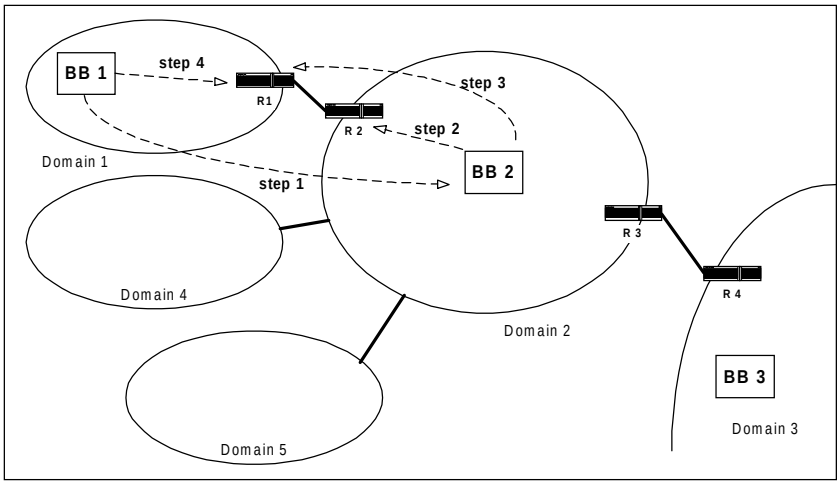


**Figure 5: Domain 1 and Domain 2 negotiating an SLA**

If premium resources are unavailable or if the request isn't supported with the set of applicable commercial agreements and the domain's policies, BB2 may decide to reject the request from BB1. The same applies to the relationship between BB2 and BB3: BB2 may request more premium resources from BB3 to aggregate the new premium traffic demand to the existing SLA between BB2 and BB3. In such a situation, BB2's reply to BB1 is sent only after a positive response from BB3 is received. Such process may be continued several domains further.

The edge routers administered by the BBs (such as R1, R2, R3 and R4 in Figure 5) perform several tasks in order to implement differentiated services. Edge routers may classify packets according to a multi-attribute filter and **mark** each matching outgoing packet with a certain Diff-Serv Code Point. They may need to **shape** traffic according to given traffic profiles to ensure service classes are not over-populated. In our example, edge routers check domain compliance with the TCA, by **metering** the outgoing packets and comparing their flows with the TCA (Flows may be modeled as leaky buckets but this is not a restricting model). Excess packets are either **discarded** or **marked** as low priority in order to comply with the SLA. Similarly, incoming traffic that is not compliant with the agreement is remarked or discarded by the ingress routers to prevent the network from being flooded with high priority traffic coming from a neighboring domain. In the conceptual hierarchy model depicted in Figure 1, the actual parameters for marking, shaping, metering, remarking and discarding are configuration level elements.

Each of the domains is free to use its preferred mechanism (DiffServ, IntServ, etc.) to provide QoS over IP. The role of the BBs is merely to broker between domains but not to supply the intra-domain service. For example, domain 1 may implement IntServ where domains 2 and 3 may implement DiffServ but not IntServ. Hence, edge routers act as

IntServ-DiffServ translation points. In the given example, R1 is the translator and hence it must have both DiffServ and IntServ capabilities.

## 5. Open issues

The concept of bandwidth brokerage gives rise to several interesting questions. For example, how does a BB compute the amount of resources needed for a certain service type given its topology parameters. If a domain admits too many high priority packets, the service level for this class may degrade. Similarly when RSVP link capacities are already allotted to existing connections, newer connections might be rejected (IntServ). If on the other hand, the portion of high priority traffic entering into the network is conservatively restricted, SLAs would be rejected and profits would not be maximized.

The second question relates to the **aggregation** of SLAs and their decomposition to TCAs. These two actions are needed in the BB in order to relay contracted traffic. For example, consider the small local ISP depicted in Figure 5 as domain 2. The ISP grants 3 premium service SLAs to 3 stub organizations to which it sells services, namely to domains1, 4 and 5. The local ISP is likely to relay some portion of the traffic to a "global" ISP that has access to the Internet backbone (domain 3). Hence, the local ISP needs to ask the global ISP to grant it an aggregate SLA, where the global ISP allows the local ISP to relay an aggregate of the traffic injected to the latter by the 3 organizations. Unless such a relay agreement exist, the local ISP compliant with the 3 SLAs is irrelevant, since anyway the incoming traffic will be discarded or remarked in the ingress of the global ISP network. Computation of such aggregates is currently an open research issue. One should note that this computation is significantly affected by the portion of the traffic that needs to be relayed to domain 3 as opposed to the traffic destined to hosts within the local ISP in domain 2.

Another issue relates to multicast flows. In this case, the resources requested by a flow are difficult to assess, due to packet duplication at branches in the multicast tree. As opposed to unicast traffic, one stream of multicast traffic sent between neighboring domains may become huge tree that imposes heavy load on the network (consider a portal multicast to millions of users worldwide).

Several other issues need to be understood and standardized before comprehensive policy based networks that encompass both unicast as well as multi-cast traffic and span intra-domain and inter-domain scenarios are widely deployed. It is not known yet how to account for traffic in the case where the receiver, such as a video client, rather the sender, such as a video server, is willing to pay for the QoS of a certain traffic. Similarly, the actual protocols for inter-BB communications, for BB to edge router communication are not yet specified. COPS, SNMP, LDAP, PFDL, DIAMETER and RADIUS are amongst the offered protocols.

## 6. Summary

In this article, we have examined the issues related to the use and deployment of Quality of Service policies in the network. We have discussed the requirements of QoS policy, defined an architecture for policy deployment within and intranet, and discussed methods which can be used to provide inter-domain policy exchanges. We have also identified several open issues related to policy deployment in the Internet.

While several of the policy architectures and protocols are still being finalized within the IETF, it is very likely that policy will play an increasingly important role in QoS allocation within corporate intranets as well as the public Internet.

## 7. Bibliography


[1] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan and A. Sastry, *The COPS (Common Open Policy Service) Protocol* , Internet Draft draft-ietf-rap-cops-05, January 18, 1999.

[2] K. Nichols, V. Jacobson, and L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet.* URL: ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[3] M. Wahl, T. Howes and S. Kille, *Lightweight Directory Access Protocol V3,* Internet *RFC*, RFC 1997, March 1995.

[4] P. Calhoun and A. Rubens, *Diameter Base Protocol*, Internet Draft draft-calhoun-diameter-07.txt, November 1998.

[5] P. Calhoun, M. Speer and K. Peirce, *DIAMETER QOS Extension*, Internet Draft draft-calhoun-diameter-qos-00.txt, May 1998.

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

[7] R. Rajan, J. C. Martin, S. Kamat, M. See, R. Chaudhury, D. Verma, G. Powers and R. Yavatkar, *Schema for Differentiated Services and Integrated Services in Networks*, Internet Draft *draft-rajan-policy-qosschema-00.txt*, October 1998.

[8] Y. Bernet, J. Binder, S. Blake, M. Carlson, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang and W. Weiss, *A Framework for Differentiated Services.* Internet Draft, October, 1998. URL: http://www.ietf.org/internet-drafts/draft-ietf-diffserv-framework-01.txt.

[9] Y. Bernet, D. Durham and F. Reichmeyer, *Requirements of Diff-serv Boundary Routers.* Internet Draft, November, 1998. URL: http://www.ietf.org/internet-drafts/draft-bernet-diffedge-01.txt.

[10] R. Yavatkar, K. McCloghrie, S. Herzog, F. Reichmeyer, D. Durham, *COPS Usage for Differentiated Services*, Internet Draft, December 1998. URL: http://www.ietf.org/internet-drafts/draft-ietf-rap-cops-pr-00.txt