

QoS/Policy/Constraint Based Routing

Wei Sun, wsun@cse.ohio-state.edu

Abstract:

This is a survey paper on Quality-of-Service(QoS) based routing. In this paper, we first introduce the concept of Quality-of-Service and its background. Second, we discuss the concepts of QoS-based routing, its objectives and main issues. After that, several types of QoS based routing algorithms are compared, and the advantages and disadvantages of each type discussed. Then, the relations between QoS based routing and some relevant techniques are studied, including traffic engineering, high level admission control, resource reservation protocols(e.g. RSVP), differential services(DiffServ) and MPLS (MultiProtocol Label Switching). And finally, a related topic --- policy-based routing is examined.

See also: [QoS in Data Networks: Products](#)| [QoS in Data Networks: Protocols and Standards](#)| [Quality of Service over IP: References](#)| [Books on Quality of Service over IP](#)
[Other reports on recent advances in networking](#)
[Back to Raj Jain's Home Page](#)

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
<http://www.cse.wustl.edu/~jain/>

Table of Contents:

- 1. [Introduction](#)
 - 1.1 [QoS Concept](#)
 - 1.2 [QoS Metrics](#)
 - 1.3 [QoS in ATM Network and Telephone Network](#)
 - 1.4 [Connection-oriented Nature of the QoS-based Services](#)
- 2. [QoS-based Routing](#)
 - 2.1 [Definitions](#)
 - 2.2 [Objectives of QoS-based Routing](#)
 - 2.3 [Main Issues of QoS-based Routing](#)
 - 2.4 [Intra-domain vs. Inter-domain QoS-based Routing](#)
- 3. [QoS-based Routing Algorithms](#)
 - 3.1 [Requirements for QoS-based Routing Algorithm](#)
 - 3.2 [Three Types of QoS-based Routing Algorithms](#)
 - 3.3 [Comparisons](#)
 - 3.4 [Examples](#)
 - 3.5 [QoS-based Multicast Routing](#)
 - 3.6 [Wireless QoS-based Routing](#)
- 4. [QoS-based Routing and Related Techniques](#)
 - 4.1 [QoS-based Routing and Traffic Engineering](#)
 - 4.2 [QoS-based Routing and Admission Control](#)
 - 4.3 [QoS-based Routing and Resource Reservation](#)
 - 4.4 [QoS-based Routing and DiffServ](#)
 - 4.5 [QoS-based Routing and MPLS](#)

- 5. [Policy-based Routing](#)
 - 5.1 [What is Policy](#)
 - 5.2 [Policy-based Routing](#)
 - 5.3 [Policy Framework and Architecture](#)
 - 5.4 [Policy-based Routing vs. QoS-based Routing](#)
 - 5.5 [Current Status and Future Directions](#)
 - 6. [Summary](#)
 - [References](#)
 - [List of Acronyms](#)
-

1. Introduction

Today's Internet can only provide "best-effort" service, which means it will try its best to forward user traffic, but can provide no guarantees regarding loss rate, bandwidth, delay, delay jitter, etc. For example, packets can be dropped indiscriminately in the event of congestion. While this kind of service works fine for some traditional applications (such as FTP and email), it's intolerable for newly emerged real-time, multimedia applications such as Internet Telephony, Video-conferencing and Video on-Demand, which require high bandwidth, low delay, and low delay jitter. In other words, these new applications require better transmission services than "best-effort". Thus, the study of Quality-of-Service (QoS) is very important nowadays.

1.1 QoS Concept

As defined in [[RFC2386](#)], Quality-of-Service is "a set of service requirements to be met by the network while transporting a flow." Here a flow is "a packet stream from source to a destination (unicast or multicast) with an associated Quality of Service (QoS)" [[RFC2386](#)]. In other words, QoS is a measurable level of service delivered to network users, which can be characterized by packet loss probability, available bandwidth, end-to-end delay, etc. Such QoS can be provided by network service providers in terms of some agreement (Service Level Agreement, or SLA) between network users and service providers. For example, users can require that for some traffic flows, the network should choose a path with minimum 2M bandwidth.

1.2 QoS Metrics

To be implemented, service requirements have to be expressed in some measurable QoS metrics. Well-known metrics include bandwidth, delay, jitter, cost, loss probability, etc. Different metrics may have different features. There are 3 types of metrics: additive, multiplicative, and concave [[CHEN99](#)]. They are defined as follows:

Let $m(n_1, n_2)$ be a metric for link (n_1, n_2) . For any path $P = (n_1, n_2, \dots, n_i, n_j)$, metric m is: (Note here $n_1, n_2, n_3, \dots, n_i, n_j$ represent network nodes)

- **additive**, if $m(P) = m(n_1, n_2) + m(n_2, n_3) + \dots + m(n_i, n_j)$

Examples are delay, jitter, cost and hop-count. For instance, the delay of a path is the sum of the delay of every hop.

- **multiplicative**, if $m(P) = m(n_1, n_2) * m(n_2, n_3) * \dots * m(n_i, n_j)$

Example is reliability, in which case $0 < m(n_i, n_j) < 1$.

- **concave**, if $m(P) = \min\{ m(n_1, n_2), m(n_2, n_3), \dots, m(n_i, n_j) \}$

Example is bandwidth, which means that the bandwidth of a path is determined by the link with the minimum available bandwidth.

Later in section 2.3, we will further discuss the metric issues.

1.3 QoS in ATM Network and Telephone Network

Different from IP network, ATM network naturally supports multiple service types, thus provides different QoS. The service types range from CBR (Constant Bit Rate) which guarantees bandwidth, delay and delay jitter, to UBR(Unspecified Bit Rate) which virtually provides no guarantees (just like today's "best-effort" IP network). Correspondingly, ATM defines different AAL(ATM Adaptation Layer) interfaces to support such services. Table 1 shows the types of services ATM supports.

Table 1: ATM Traffic Services[FH98]

ATM Service types	Typical Uses
Constant Bit Rate(CBR)	Real-time, QoS guarantees
Real-Time Variable Bit Rate(rt-VBR)	Statistical multiplex
Non-Real-Time Variable Bit Rate(nrt-VBR)	Statistical multiplex
Available Bit Rate(ABR)	Resource Exploitation, feedback control
Unspecified Bit Rate (UBR)	Best effort, no guarantees

Telephone network has been around for much longer time than computer network. So it's not surprising that it is well developed and has complex control schemes. Basically telephone network is circuit-switched network. When a call is coming, a path is setup by the switches which select and reserve the path. The path can be selected in such way that it meets the QoS requirements of the call. Then the path is used exclusively by the call. In other words, users have the quality of service once the call is setup. The path is released after the call finishes.

Both ATM network and telephone network have QoS-based routing schemes, such as ATM's PNNI (Private Network-to-Network Interface) and RTNR(Real-Time Network Routing) of AT&T[ACFH92]. The experience and knowledge gained in both networks provide insight of how to provide QoS and design QoS-based routing in the Internet.

1.4 Connection-oriented Nature of the QoS-based Services

Current Internet is connectionless and stateless. IP is a connectionless protocol, which means there is not such a process to setup a connection between source and destination before packet transmission, as in ATM network and telephone network. Stateless means the routers along the path of the traffic flows do not maintain specific information about the state of each flow. The packets in a flow are routed according only to routers' routing table. While this scheme is simple and scalable, and leads to the success of the Internet, it's not enough to provide QoS.

Different from above "best-effort" services, QoS service usually requires resource reservation. A path is pre-determined and associated resources(link bandwidth, buffer space, etc.) along the path are reserved before the actual transmission. In other words, the path or connection between source and destination is setup first. When the transmission finishes, the path and associated resources are released. To reserve the resources of a flow, the routers along the path need to keep track of the state of the flow. Some information is maintained in the routers regarding the state of the flow. In short, to provide QoS, both connection and state information are needed.

To provide QoS in the Internet, many techniques have been proposed and studied, including Integrated Services(IntServ)[[RFC1633](#)], Differential Services(DiffServ)[[RFC2475](#)], MPLS(MultiProtocol Label Switching)[[MPLS](#)], Traffic Engineering and QoS-based Routing. Specific working groups are also organized under IETF(Internet Engineering Task Force)[[IETFWG](#)]. In this paper, we will focus on QoS-based routing, which is an important component in the whole QoS framework in the Internet.

In the next section, the definition of QoS-based routing is given. Some related concepts(constraint-based routing and policy-based routing) are also introduced. Then the objectives of QoS-based routing and main issues of it are discussed. In section 3, three types of QoS-based routing algorithms are examined and compared. QoS-based routing for multicast and wireless network are also discussed. Relations between QoS-based routing and some relevant QoS techniques are discussed in section 4. In section 5, we discuss policy-based routing, which is similar to QoS-based routing but different. Finally, a summary is given.

[Return to Table of Contents](#)

2. QoS-based Routing

Due to the importance of QoS-based routing, IETF set up a QoS Routing Working Group[[QOSR](#)] to guide the research on QoS-based routing techniques. It defined a framework of QoS-based routing in the Internet[[RFC2386](#)].

2.1 Definitions

QoS-based routing is defined as: "A routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of the flows." [[RFC2386](#)] Or "a dynamic routing protocol that has expanded its path-selection criteria to include QoS parameters such as available bandwidth, link and end-to-end path utilization, node resources consumption, delay and latency, and induced jitter." [[QOSF2](#)] In short, it's a dynamic routing scheme with QoS consideration.

Figure 1 shows a simple example of QoS-based routing. Suppose there is a traffic flow from node A to node C which requires 4M bandwidth. As we can see, although path A-B-C is shorter, it will not be selected because it doesn't have enough bandwidth. Instead, path A-D-E-C is selected.

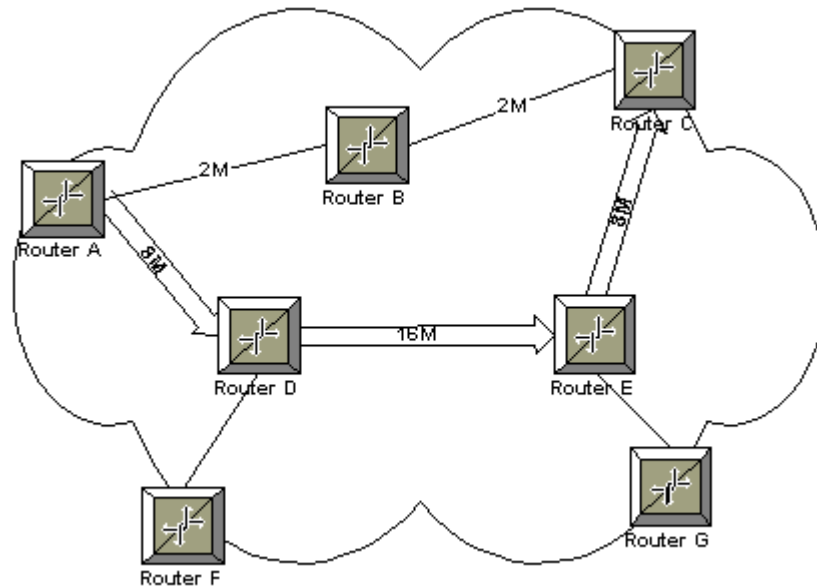


Figure 1: QoS-based routing example

Besides, there are two relevant concepts called *Policy-based Routing* and *Constraint-based Routing*. Policy-based Routing commonly means the routing decision is not based on the knowledge of the network topology and metrics, but on some administrative policies. For instance, a policy may prohibit a traffic flow from using a specific link for security reason, even if the link has enough bandwidth and low delay. Policy-based Routing is usually statically configured.

Constraint-based Routing is a new concept, which is derived from QoS-based routing but has broader sense. It means to compute routes that are subject to multiple constraints, including both QoS constraints (QoS requirements and resource availability) and policy constraints. Both QoS-based routing and policy-based routing can be considered as special cases of constraint-based routing.

In the next few sections, we first focus on QoS-based routing. Policy-based routing will be discussed later in section 5.

2.2 Objectives of QoS-based Routing

Current Internet routing protocols such as OSPF(Open Shortest Path First), RIP(Routing Information Protocol), and BGP(Border Gateway Protocol) are called "best-effort" routing protocols. They use only the "shortest path" to the destination. (Note "shortest path" here doesn't necessarily mean the path with shortest physical distance. It may also mean the path with the least cost or fewest hop counts, for instance.) In other words, they normally use single objective optimization algorithms which consider only one metric(bandwidth, hop count, cost). Thus, all the traffic is routed to the "shortest path". Even if there are some alternate paths exist, they are not used as long as they are not the shortest ones. One drawback of this scheme is that it may lead to the congestion of some links, while some other links are not fully used.

Second, today's "best-effort" routing will shift the traffic from one path to "better" path whenever the "better" path is found. This happens even if the current used path meets the service requirements of the traffic. This kind of shift is undesirable because it will bring routing oscillations when the routing is based on metrics like available bandwidth, which changes rapidly from time to time. The traffic will be routed back and forth between alternate paths. Even worse, this kind of oscillations can increase the variation in the delay and jitter experienced by the end users.

QoS-based routing is supposed to solve or avoid the problems mentioned above. The main objectives of QoS-based routing are:

- First, of course, to meet the QoS requirements of end users. QoS-based routing is supposed to find a path from source to destination which can satisfy user's requirements on bandwidth, end-to-end delay, etc. Besides, this should be done dynamically, instead of being configured statically. In case there are several feasible paths available, the path selection can be based on some policy constraints. For example, we can choose the path which costs less money, or the one via the designated service provider.
- Second, to optimize the network resource usage. This is an objective from service providers' point of view. Every service provider wants to maximize the utilization of their current network facilities, thus to maximize its revenue. Besides, this is also a requirement from network engineering's perspective. QoS-based routing is expected to direct network traffic in an efficient way that can maximize the total network throughput. One common scheme is to always choose the shortest path among the feasible candidates, because longer path means using more network resources.
- Third, gracefully degrade network performance when things like congestion happen. When network is in heavy load, QoS-based routing is expected to give better performance (e.g. better throughput) than best-effort routing, which can degrade the performance dramatically.

2.3 Main Issues of QoS-based Routing

This section discusses the major design issues of QoS-based routing algorithms. As we will see below, QoS-based routing is much more difficult to design and implement than "best-effort" routing. Many tradeoffs have to be made. In most cases the goal is not to find a best solution, but rather to find a viable solution with acceptable cost.

- **Metric and path computation**

Two basic issues of QoS-based routing are: first, how to measure and collect network state information; second, how to compute routes based on the information collected. Metric selection is very important in the sense that, "the metrics must represent the basic network properties of interest." [RFC2386] Thus metrics like available bandwidth, delay, jitter, etc. are commonly used. Also, metrics define the types of QoS guarantees the network can provide. There is no way to support a QoS requirement which can not be mapped onto some combination of existing metrics.

Besides, computation complexity must be considered, which means path computation based on a metric or a combination of metrics must not be too complex. Unfortunately, QoS-based routing usually is under multiple constraints (a simple example is to find a path with 4M bandwidth and 50ms delay limit), and path computation based on certain combinations of metrics is proved to be NP-complete [WC96]. A lot of heuristic algorithms are proposed to solve the problem. A common

method is called "*sequential filtering*", "under which a combination of metrics is ordered in some fashion, reflecting the importance of different metrics(e.g. cost followed by delay, etc.). Paths based on the primary metric are computed first and a subset of them are eliminated based on the secondary metric and so forth until a single path is found."[[RFC2386](#)] This is a tradeoff between performance optimization and computation simplicity.

Path computation is also closely related to resource reservation, which means once a feasible path is chosen, the corresponding resources(bandwidth, buffer space in routers etc.) must be reserved for the traffic flow thus are not available to other flows. Consequently, the amount of available resources(such as bandwidth) after the reservation must be recalculated and such information be propagated to other routers. This way, all the routers can make right decision for other flows.

- **Knowledge propagation and maintenance**

One important issue is how often the routing information is exchanged between the routers. QoS-based routing needs to exchange more information than "best-effort" routing. First, besides the routing information needed by "best-effort" routing(like connection topology information), QoS information such as available bandwidth needs to be exchanged, too. Second, the metrics used by QoS-based routing could be changing very quickly. Again, available bandwidth is a typical example. If the routing information is exchanged every time the values of metrics change, it will cause a great burden for the network links and routers --- consuming network bandwidth and routers' CPU cycles.

One common way is to set a threshold to distinguish significant changes from minor changes. The information is exchanged only when a significant change occurs. By doing so, it can also bring stability of the QoS routes. Again, this is a tradeoff --- tradeoff between routing information accuracy and efficiency. Another method is to consider only the available resources after reservation, instead of the actual available resources. Take bandwidth as example, suppose a network link has 4M bandwidth, and 3M has been reserved by some flows. So the available bandwidth is 1M. As long as no new flows reserve the available bandwidth and no flows release current reserved bandwidth, the available bandwidth is considered 1M. In other words, we don't consider the bandwidth which is reserved but not used, even though the actual used bandwidth could be fluctuating from time to time(which could be 1.5M at one moment and 2.5M at another). These two methods can be used together.

A related problem is how to maintain the information collected. If we maintain the information for every flow in routers, the size of routing table will increase very rapidly. One suggestion is to keep only the routing table for best-effort traffic, and compute the paths for QoS flows on demand. This is to trade computation time for storage space. Flow aggregation is another possible method. Instead of storing information about individual flows, we can aggregate the flows and maintain only the information about aggregated flows, which is much fewer in number.

- **Scaling by hierarchical aggregation**

This issue is related to the path computation and information propagation/maintenance issue mentioned above. Considering the size of the Internet, QoS-based routing is expected to be scalable --- with the number of nodes and links in the network increasing, the complexity of path computation and the amount of information need to be exchanged and maintained won't be out of control. One way is to use hierarchical aggregation, as in PNNI and OSPF. However, such aggregation can bring inaccuracy in regard of routing information. And such inaccuracy may

eventually lead to accepting a flow which is unacceptable or rejecting a flow which is indeed acceptable. Thus, we must be careful on how to aggregate the information.

- **Imprecise State Information Model**

A trend in QoS-based routing algorithm design is that more and more people realize the imprecise nature of QoS routing. *Imprecision* means the routing state information based on which the routing decision is made is not accurate or precise.

In [[LO98](#)] four sources of inaccuracy are discussed:

- Network dynamics: some parameters or metrics (particularly available bandwidth, delay) associated with network links and nodes vary from time to time. It's very hard (if not impossible) to keep the accurate information.
- Aggregation of routing information: As we discussed earlier, routing aggregation is encouraged to decrease the routing update overload and routing storage overload, especially for large network.
- Hidden information: For security or other reasons, some routing information is hidden and thus unknown.
- Approximate calculation: no values of network parameters or metrics can be truly accurate. They are just approximations of real values. In fact, study in [[LMJ98](#)] shows that 99% of the routing information in the current Internet is not accurate.

Several QoS-based routing algorithms are proposed, based on this imprecision assumption. [[AGKT99](#)] gives a safety-based routing algorithm, where "safety" is based on probability. [[CHEN99](#)] suggests another method which use a range rather an "exact" value to represent the metrics. A range is indicated by both a lower bound and an upper bound.

- **Administrative Control**

There are also some control issues regarding QoS.

- Flow priorities and preemption

Different flows in the network have different QoS requirements, thus should have different priorities. Critical flows can be assigned higher priority than other flows. When the resources (such as bandwidth) are not enough, such flows can preempt the resources from flows with lower priority. For instance, a voice or video flow (which has strict delay and bandwidth requirements) can be assigned higher priority and is allowed to preempt bandwidth or buffers from FTP flows.

- Resource control

In the network framework which has multiple service classes of traffic (DiffServ, for instance), the resources should be allocated fairly among all the classes. Thus starvation of lower priority classes can be avoid. This allocation can be done in a dynamic fashion.

Such control schemes should be included in QoS-based routing.

- **Integrate QoS-based routing and Best-effort routing**

For compatibility, QoS-based routing must be able to support best-effort routing, too. This means these two routing schemes must be able to coexist. One question here is how to allocation network resources between them. Intuitively QoS-based routing should have higher priority. However, we should have overall control so that QoS-based routing doesn't use too much of the resources. Otherwise best-effort traffic would have virtually no resources to use. We need the control methods mentioned above. [[CHEN99](#)] proposed a routing algorithm to fairly share the resources between these two.

2.4 Intra-domain vs. Inter-domain QoS-based routing

Many different QoS-based routing have been proposed in recent years(for both unicast and multicast routing), with most of them concentrating on one particular problem. These algorithms have different assumptions of the network condition and thus can not work together. Then it is realized that a common framework is needed, which can accommodate different kinds of algorithms. [[RFC2386](#)] provides such a framework, which has two levels --- *intra-domain*QoS-based routing and *inter-domain*QoS-based routing. This hierarchical model is compatible with the routing hierarchy of today's Internet(which has the concept of Autonomous System --- AS). Figure 2 shows such a two-level routing structure. The routing among nodes A, B and C belongs to intra-domain routing, while that between node B and E or F belongs to inter-domain level.

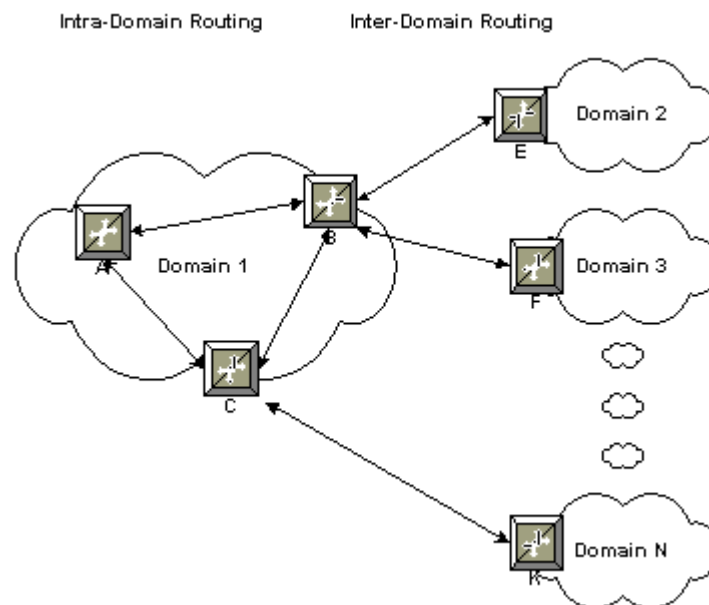


Figure 2: intra-domain vs. inter-domain routing

For intra-domain QoS-based routing, it is intended to accommodate many different algorithm schemes in one domain. The network manager should have the freedom to use whatever QoS-based routing inside the domain, which is independent of the QoS-based routing used in other domains. Diversity is encouraged at this level. QoS-based routing services can range from dynamic path computation based on current state information, to statically provisioned paths supporting a few service classes.

However, some common features are required for intra-domain QoS-based routing. They are listed as follows:

- the routing scheme should find path which can meet the QoS requirement of the flow, if such a

- path exists. Otherwise it should indicate that the flow cannot be admitted.
- to optimize resource usage
- the routing scheme must indicate path disruption whenever the path is affected by topological changes of the network.
- the routing scheme should support best-effort flows, which doesn't have any resource reservation requirements. In other words, current best effort application and protocols need not be changed in a QoS-based routing domain.
- the routing scheme is expected to support multicast QoS-based routing, with receiver heterogeneity and shared reservation styles
- the routing scheme should have higher level admission control(as mentioned earlier) to limit the overall resource utilization by individual flows.

In contrast, inter-domain routing is expected to be as simple as possible. Stability and Scalability are the most important issues at this level. Thus the routing cannot be based on highly dynamic network state information. Rather, The QoS information exchange between different routing domains should be relatively static.

The inter-domain routing scheme must have the following basic functions:

- determination of whether a destination is reachable
- avoid routing loop
- support address aggregation
- determination of whether the QoS requirements can be supported on the path to a destination.
- determination of multiple paths to a given destination, based on service classes(this is optional).
- mapping routing policies(e.g. monetary cost, usage and administrative factors) to flow metrics.

Most of the QoS-based routing algorithms proposed belong to the intra-domain level.

[Return to Table of Contents](#)

3. QoS-based Routing Algorithms

So far, many QoS-based routing algorithms have been proposed. Most of them start from extending the ability of current "best-effort" routing algorithms. The current Internet routing protocols are based on two routing algorithms --- Distance vector algorithm and Link-State algorithm. In Distance vector algorithm, neighboring routers exchange routing information periodically. Thus every router can learn the routing information from others. Based on that information, the shortest path to every destination can be computed. This is also called Bellman-Ford algorithm.

While in Link-State algorithm, every router advertises its link state information to the whole network, thus every router can receive the link-state information. Such information is maintained in a local database in every router, from which the routing table is calculated using Dijkstra's shortest path algorithm. The advertising is triggered by events, and it also happens periodically.

3.1 Requirements for QoS-based Routing Algorithms

Below are some basic requirements for QoS-based routing algorithms:

- efficient and scalable to large network
- complexity not far greater than the currently used routing algorithm
- suitable to current Internet architecture

Note that some of them are conflicting with each other, so that compromise has to be made. On one hand, we need efficient algorithms. And the algorithms should be scalable enough that they can be used in the Internet; On the other hand, these algorithms should not be too complicated. We have to make some tradeoff between efficiency and complexity.

QoS-based routing algorithms are expected to be employed in current Internet. They must be easy to implement. As discussed in section 2.3, they must be compatible with the current "best-effort" routing. In other words, they don't require the modification of current Internet applications and routing protocols.

3.2 Three Types of QoS-based Routing Algorithms

Basically, QoS-based routing algorithms can be classified into 3 categories --- hop-by-hop routing(also called distributed routing), source-based routing, and hierarchical routing algorithm. "They are classified according to the way how the state information is maintained and how the search of feasible paths is carried out." [[CHEN99](#)]

In source routing, every router has global state information about the network, and the path is locally selected based on the state information. After the path is determined, the source router notifies the other router along that path how to forward the traffic flow. Then the flow will be routed to the destination accordingly.

In hop-by-hop routing, each router just knows the next hop towards the destination. Thus when a packet comes, the router just forwards it to the next-hop router. Step by step, the packet gets to the destination. Most current Internet routing protocols(such as RIP) use this method.

Hierarchical routing is most suitable for large network. The routing structure consists of multiple levels. The bottom level contains the actual routers. These routers are organized into some logical groups, which in turn form the next level. Each group is a logical nodes in the next level. The groups can be further organized into some higher level groups. This process can continue recursively, so that each level doesn't have too many nodes(routers). The routing information is integrated at the border nodes of each groups. Every node contains the detailed information about its group and integrated information about other groups. PNNI is a typical example of hierarchical routing. We will see its details later.

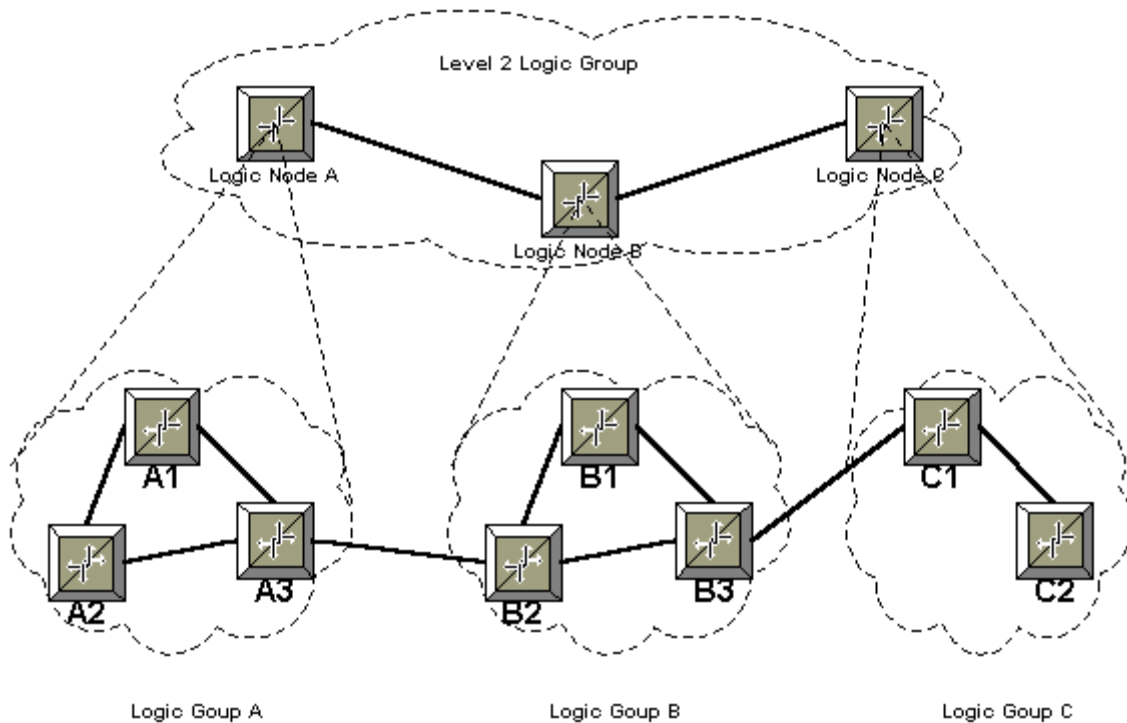


Figure 3: Hierarchical routing structure

Figure 3 shows an example of hierarchical routing. As we can see, A1, A2 and A3 form a logical group, which is represented by logical node A. Similarly, B1, B2, and B3 form a group, C1 and C2 form another group, which are represented by logical nodes B and C, respectively. Logical nodes A, B, and C form a group in the next level.

Figure 4 shows the routing table of A1 in Figure 3's structure. It has the detailed information about its own group(group A), while the information about group B and C is aggregated.

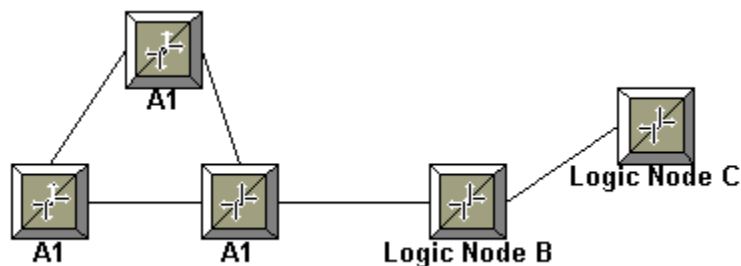


Figure 4: Hierarchical routing table example(A1's routing table)

3.3 Comparisons

In this section we will compare the advantages and disadvantages of the three types of routing schemes.

Source routing is simpler in the sense that it's decided solely by the source. Other routers along the path just need to follow the pre-determined path. And it will not cause routing loop. However, it has drawbacks. First of all, it requires that each router has complete state information of the network, which is hard to maintain, especially for large network. It will cause lots of state information updates, thus

bring much traffic overhead to the network. Second, if we aggregate the state information updates to decrease the traffic burden, the accuracy of the information may be affected. Thus it may not be able to find an existing feasible path. Third, although it's easy for the other routers to forward the traffic, the computation overhead at the source routers is very high. In short, source routing algorithm has the scalability problem, thus is not good for large network.

Hop-by-hop routing is used by most current "best-effort" routing protocols(such as RIP). So that it's more natural to design and more compatible with existing routing protocols. The routing computation burden is distributed among all the routers along the path, from source to destination. However, it has the routing loop problem when the routing state information in different routers is not consistent. Besides, it also has the scalability problem.

The biggest advantage of hierarchical routing is its scalability. It's thus suitable for large networks. The routing state information can be aggregated to decrease the burden of the routing updates and storage. However, aggregation also decreases the accuracy of the routing state information, thus has impact on the performance of the QoS-based routing.

Table 2 compares the performance and features of many different algorithms.

Table 2: Unicast Algorithms Comparison [CHEN99]

Algorithm	Problem solving	Routing strategy	Time complexity	Comm. complexity	Comm. Complexity
				maintaining state	routing
Wang-Crowcroft	Bandwidth-delay-constrained routing	source	$O(v \log v + e)$	global	zero
Ma-Steenkiste	multi-constrained routing	source	$O(kve)$	global	zero
	Bandwidth-constrained routing	source	$O(v \log v + e)$	global	zero
Guerin-Orda	Bandwidth-constrained routing	source	$O(v \log v + e)$	Imprecise global	zero
	Delay-constrained	source	polynomial	Imprecise global	zero

	routing				
Chen-Nahrstedt	Bandwidth-cost-constrained routing	source	$O(xve)$	global	zero
Wang-Crowcroft	Bandwidth-optimization routing	distributed	$O(v^2)$	global	$O(v)$
Salama et al	Delay-constrained least-cost routing	distributed	$O(v^3)$	global	$O(v^3)$
Sun-Landendorfer	Delay-constrained least-cost routing	distributed	$O(v)$	global	$O(v)$
Cidon et al	Generic routing	distributed	$O(e)$	global	$O(e)$
Shin-Chou	Delay-constrained routing	distributed	$O(e)$	local	$O(e)$
Chen-Nahrstedt	Generic routing	distributed	$O(e)$	local	$O(e)$
PNNI	Generic routing	hierarchical	polynomial	aggregated	$O(v)$

Note: v is the number of nodes;

e is the number of edges;

x is a constant in the algorithm;

k in the time complexity is the number of all possible residual bandwidth that a link may have.

3.4 Examples

In this section we give two examples of QoS-based routing --- PNNI and QOSPF.

- PNNI

PNNI(Private Network-Network Interface) is a hierarchical, dynamic routing protocol for ATM network, which supports QoS naturally.

This protocol is based on link-state algorithm. Topology information(including information about nodes, links, addresses) is flooded through the network. Network resources are defined by metrics and attributes (delay, available bandwidth, jitter, etc.), which are grouped by supported traffic class. Since some of the metrics used will change frequently(e.g., available bandwidth), threshold algorithms are used to determine if the change in a metric or attribute is significant enough to require propagation of updated information.

Since it's hierarchical, it has the concepts of levels and logical nodes. It also supports aggregation and summarization of topology and reachability information.

PNNI has other features such as auto configuration of the routing hierarchy, connection admission control(as part of path calculation).

However, PNNI routing protocol does not support multicast routing, policy routing and control of alternate routing. It also inherits the common problem with link state QoS-based routing --- how to efficient broadcast state information, especially for dynamic metrics.

- QOSPF

QOSPF(QoS routing extensions to OSPF)is an extension to OSPF, into which QoS functions are added. OSPF is also based on link-state algorithm, and is hierarchical protocol. The design goal of QOSPF is to "define an approach which, while achieving the target of improving performance for QoS flows(likelihood to be routed on a path capable of providing the requested QoS), does so with the least possible impact on the existing OSPF protocol."[[RFC2676](#)] It's supposed to be working in an environment in which both QoS-based routing and best-effort routing are needed.

For simplicity, link bandwidth and propagation delay are the only metrics extension added to Link State Advertisements(LSAs). Similar to PNNI, in order to decrease protocol overhead, LSAs are triggered only when there is a significant change in the value of the metrics since the last advertisement. There are two options for QoS path computation --- pre-computation and on-demand computation. Again, for simplicity, pre-computation is chosen. For every possible destination, the algorithm pre-compute a "widest-shortest path", which is a minimum hop count path with maximum bandwidth. Simulations[[RFC2676](#)] show that the complexity of the algorithm is tolerable and the performance is acceptable. The computational complexity is comparable to that of Bellman-Ford shortest-path algorithm.

3.5 QoS-based Multicast Routing

So far, we only consider the QoS-based routing for unicast. In this section we discuss QoS-based multicast routing. Multicast is more suitable than unicast for multimedia, real-time applications, such as video-conferencing and video-on-demand. Thus it's very important to study QoS-based routing for multicast flows. Different from unicast QoS-based routing, the goal here is not to find a path which meets some specific QoS requirements(as in QoS-based unicast routing), but to find such a *tree*, in

which the source is the root of the tree, and the destinations are the leaves.

Some typical problems of QoS-based multicast routing are:

- scalability to large groups. First, the algorithm must be able to cover all the members of a multicast group, even when the group is large. Second, the overheads associated with QoS-based multicast path computation should not be too high.
- support for dynamic membership. The multicast group could be dynamic, with some new members joining and some current members leaving, from time to time. The multicast routing algorithm must be able to extend the tree to cover the new members in a timely manner, and to prune the branch to which has no members attached. This issue is addressed in some proposals such as CarlBerg-Crowcroft algorithm[[CC97](#)].
- support for receiver-initiated, heterogeneous reservations. In the same multicast group, different members may have different QoS requirements. For example, one receiver may need 1M bandwidth, while another may need 2M bandwidth.

QoS-based multicast routing can be implemented in two ways. First, multicast source routing "is one under which multicast trees are computed by the first-hop router from the source, based on sender traffic advertisements." [[RFC2386](#)] The advantage of this scheme is that it is compatible with the current RSVP signaling model. When receiver reservations are homogeneous, this scheme works well . The disadvantages of this scheme is that it is difficult to support heterogeneous reservations.

Another method is receiver-oriented multicast routing. Under this scheme, sender first multicast its advertisements over a best-effort tree which can be different from the QoS tree for sending data. Then, each receiver gets the advertisements and independently computes a QoS path from the source, based on the receiver reservation. In other words, multicast path computation is broken up into multiple concurrent unicast path computations. This scheme supports heterogeneous reservations well.

Although potentially there are many possible problems, most existing research on QoS-based multicast routing focuses on several problems: bandwidth-constrained multicast routing; delay-constrained multicast routing; delay-constrained least-cost multicast routing; and delay and delay-jitter constrained multicast routing. The proposed algorithms can be divided into two types: source routing and distributed (hop-by-hop) routing.

Table 3 summarizes and compares the performance of many QoS-based multicast routing algorithms.

Table 3: QoS-based Multicast Routing Algorithms Comparisons[[CHEN99](#)]

Algorithm	Problem solving	Routing strategy	Time complexity	Comm. complexity	Comm. Complexity
				maintaining state	routing
MOSPF	lest-delay routing	source	$O(v \log v)$	global	zero
Kou et al	least-cost routing	source	$O(gv^2)$	global	zero

Takahashi-Matsuyama	least-cost routing	source	$O(gv^2)$	global	zero
Kompella et al	Delay-constrained least-cost routing	source	$O(v^3 d)$	global	zero
Sun-Landgendorfer	Delay-constrained least-cost routing	source	$O(v \log v + e)$	global	zero
Widyono	Delay-constrained least-cost routing	source	exponential	global	zero
Zhu et al	Delay-constrained least-cost routing	source	$O(kv^3 \log v)$	global	zero
Rouskas-Baldine	Delay-constrained least-cost routing	source	$O(klgv^4)$	global	zero
Kompella et al	Delay-constrained least-cost routing	distributed	$O(v^3)$	global	$O(v^3)$
Chen-Nahrstedt	Generic routing	distributed	$O(ge)$	local	$O(ge)$

Note: v is the number of nodes;

e is the number of edges;

g is the number of destination;

d is the delay requirement;

k and l are constants in the algorithm. A large k (or l) results in a higher probability of finding a feasible tree and a higher overhead.

3.6 Wireless QoS-based Routing

Wireless network such as Ad-Hoc network also needs QoS guarantees. However, the dynamic nature of such network makes it more difficult to provide QoS, because it's hard to keep routing state information up-to-date. One unique problem of wireless network is the mobility of the nodes, which could lead to the breaking of existing paths and the adding of new links. In the case of the Infrastructure wireless network, this may lead to the Handoff problem, which happens when a node in the network moves from one cell to another cell.

One way to handle this problem is to divide the network links into two types: stationary links and transient links. Stationary links are those between the stationary nodes or slowly moving nodes, which are likely to exist for a long time. While transient links are those between nodes moving very fast. To reduce the probability of path breaking, stationary links are always preferred when choosing a QoS path. When an existing path is broken, the traffic flow is rerouted to another feasible path. During the period after the old path is broken and before the new path is set up, best-effort routing is used to route the traffic flow. For this reason, wireless network normally can only provide *soft QoS*, which means the required QoS is not guaranteed for some transient time periods, when the routing path is broken or the network is partitioned due to the moving of network nodes.

Many QoS-based routing algorithms are proposed, which are based on either local states or imprecise global states[[CHEN99](#)].

[Return to Table of Contents](#)

4. QoS-based Routing and Related Techniques

QoS-based routing is only one component to provide QoS in the Internet. It needs to work together with many other techniques. Below we will discuss the relations between QoS-based routing and some such techniques.

4.1 QoS-based Routing and Traffic Engineering

"Traffic engineering is the process of arranging how traffic flows through the network so that congestion caused by uneven network utilization can be avoided."[[XN98](#)] It includes a lot of different mechanisms, static control or dynamic control. QoS-based routing is an important part of traffic engineering, which can help make the traffic engineering process automatic.

4.2 QoS-based Routing and Admission Control

As we mentioned earlier, one goal of QoS-based routing is to maximum the network utilization and improve the total throughput of the network. In this sense, simply route a flow to a path which can meet the flow's QoS requirements is not good enough. We have to take into account the total resource allocation for a flow along a path, in relation to available resources. If this flow need too much resources, we may reject it even if the network has the capability to accept it. By doing so, the resources can be used by other flows which cost less. This is called "higher level admission control".

Another related problem is fairness. Larger flows tend to need more resource while small flows need less. Thus small flows always have a better chance to be accepted. To be fair, we need to guarantee that larger flows can get a certain level of acceptance rate.

These kinds of mechanisms need to be incorporated in QoS-based routing.

4.3 QoS-based Routing and Resource Reservation

First of all, QoS-based routing and resource reservation are closely connected. To provide QoS guarantees to user flows, there are two tasks. The first is to find a feasible path from source to destination, which can meet the QoS requirements; The second is to reserve the resources along the path. The first task is done by QoS-based routing, while the second one is done by resource reservation protocols (such as RSVP). However, it's worth to note that QoS-based routing and resource reservation are two different techniques. In short, QoS-based routing itself can not reserve resources, and resource reservation protocols are not supposed to find the feasible path.

"Consequently, QoS-based routing is usually used in conjunction with some form of resource reservation or resource allocation mechanism... Combining a resource reservation protocol with QoS-based routing allows fine control over the route and resources at the cost of additional state and setup time. For example, a protocol such as RSVP may be used to trigger QoS-based routing calculations to meet the needs of a specific flow." [[RFC2386](#)]

4.4 QoS-based Routing and DiffServ

DiffServ is proposed to provide QoS on the Internet, while solve the scalability problem with IntServ. In DiffServ framework, the routers supporting DiffServ form a DiffServ domain. Packets entering the domain are marked differently at the edge routers to create several packet classes. Then inside the domain, packets in different classes are treated differently according to their marks, thus receive different services.

The advantage of DiffServ is that it doesn't require the routers to maintain state information for each flow, which is a huge burden for the routers. However, it also has problems. One is that since the packets are marked just at the edge routers, it can not solve the congestion inside the domain. For example, a lot of flows in the same class can be routed through the same link, thus cause congestion there.

QoS-based routing will be very helpful in this case. It can route the flows to the paths which have the capacity to accept the flows, or simply reject a flow if there is not resource for it.

4.5 QoS-based Routing and MPLS

MPLS was first proposed as a fast forwarding scheme, but later found also useful for QoS. Similar to DiffServ domain, MPLS also has the concept of MPLS domain, which consists of the MPLS-capable routers. Packets are assigned labels at the ingress routers of the MPLS domain. Then inside the domain, classification, forwarding, and services for the packets are based on the labels. The labels will be removed when the packets leave the domain.

QoS-based routing and MPLS can work together, too. QoS-based routing can select the path, and MPLS will do the packet forwarding along the path. MPLS can also provide more precise routing information for QoS-based routing, which may help QoS-based routing to select better paths.

[Return to Table of Contents](#)

5. Policy-based Routing

One result of enabling QoS on the Internet is that some users will get better network service than others. The users who pay more for service quality should get better service guarantees. However, it's inevitable that some illegal users may want to get such service without paying. To prevent such service "steal", and to give legal users the service they pay for, we need some policy rules and ways to enforce these rules.

5.1 What is policy?

By definition, a policy is a set of rules which associate with some services. The rules define the criteria for obtaining those services. A rule consists of one or more conditions and actions. It describes the conditions that must be met before some specified actions can be taken, as showed in figure 4. Note that policy itself has broader sense than just QoS policy. A policy could be about issues such as security or administrative regulations.

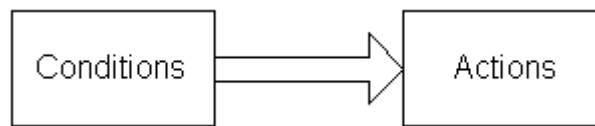


Figure 5: policy rule

A simple policy example is: In computer department, the traffic flows from faculty's offices should get higher priority than those from student labs. Here the condition is: "if the traffic flows are from faculty's offices"; while the action is: "assign higher priority to those flows".

5.2 Policy-based Routing

As we said earlier in section 2.1, policy-based routing means to implement routing based on the policies defined by the network administrator. It provides a routing scheme which is beyond the ability of traditional "best-effort" routing protocols. For instance, it allows the routers to route traffic from different users through different Internet connections. In other words, it allows routing based on the source information of the packets, instead of destination information used by traditional routing protocols. It can also support QoS. For example, it can support DiffServ by using packet marking, which differentiate traffic by setting the DS(Differential Service) or TOS(Type of Service) fields in the IP header at the edge routers of the network. Then the routers in the core of the network can treat the packets differently based on the marking.

Although some routing protocol such as BGP supports exchanging policy information via routing updates, normally the policy information is set up manually in terms of network configuration, by using tools such as Telnet. The whole process consists of iterating through the network devices(routers or switches) one by one. Not only is this process tedious, it can not prevent the conflicts between different policy rules. Thus there is a need for automatic policy management system.

5.3 Policy Framework and Architecture

The IETF Resource Allocation Protocol(RAP) working group proposed a policy framework which intends to "establish a scalable policy control model for RSVP." [[RAP](#)] However, it's recognized that this is a general model, which is applicable to many technologies that need policy support. It can be applied to other QoS technologies(such as DiffServ) or non-QoS technologies(such as network security),

for instance. Figure 6 shows its general architecture and main functional components.

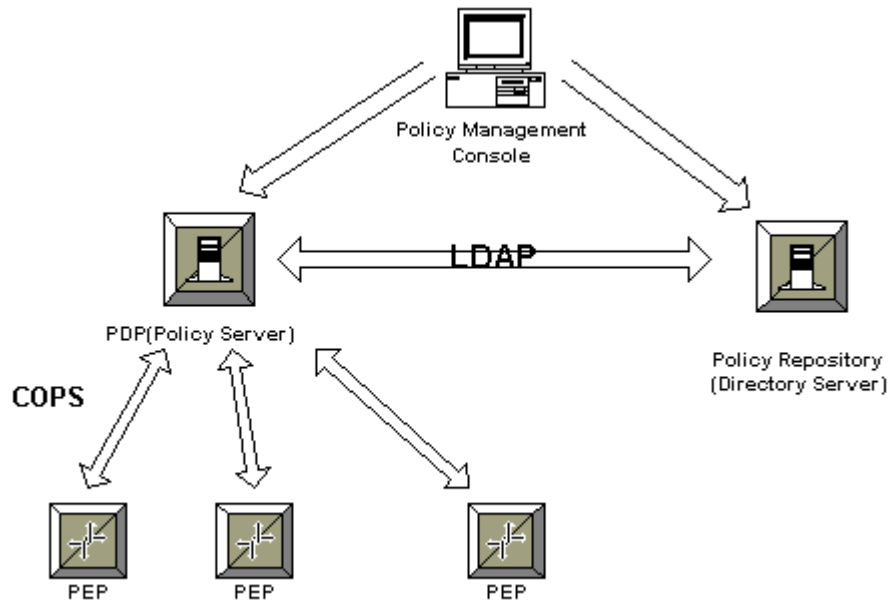


Figure 6: Policy Architecture

As we can see, the architecture includes four major components:

- Policy Management Console. Policy management console is the central user console, which is in charge of editing, viewing policies and entering policy rules into the policy repository. It also has other functions such as checking the conflicts between policy rules. Usually the console has user friendly GUI(Graphical User Interface).
- Policy Repository. Policy repository is a dedicated server, which is in charge of policy storage and retrieval. As described earlier, policy rules consists of conditions and actions. Policies can also be defined recursively, which means a policy can contain other policies. A policy group is aggregation of policy rules or aggregation of policy groups.

"To represent, manage, share, and reuse policies and policy information in a vendor-independent, interoperable, and scalable manner"[[POLICY](#)], the IETF Policy Framework Working Group [[POLICY](#)] defines a core information model and schema[[MES99](#)]. The policy core schema is an LDAP schema[[SEMM99](#)] which consists of a total of 13 classes, including *policyGroup*, *policyRule*, *policyCondition*and *policyAction*.

The immediate goal of the group is to support QoS. Based on the core information model and schema, the group also defines the QoS information model and schema to address the needs of QoS traffic management[[SRS99](#)]. However, the policy model defined "is generic in nature and is applicable to Quality of Service (QoS), to non-QoS networking applications (e.g., DHCP and IPSEC), and to non- networking applications (e.g., backup policies, auditing access, etc.)."
"[[MES99](#)]

- PDP is also called Policy Server, which makes decisions based on the policy rules it retrieves from the policy repository. PDP can also get information from other entities such as authentication server or SNMP agents. Optionally, PDP can also translate the rules from vendor- and device-

independent rules into the low-level, vendor- and device-specific rules.

- PEP is the actual execution point of the policies. It is the network device such as router or switch which can execute the policies. The separation of PDP and PEP is a logic one. They can actually be implemented on the same network device. One PDP can manage multiple PEPs in the same administrative domain.

Besides the four major components mentioned above, there are two communication protocols used in this architecture: COPS(Common Open Policy Service) and LDAP(Lightweight Directory Access Protocol). PDP and PEP exchange information by using COPS[[COPS](#)], which is a query and response protocol between policy server(PDP) and client(PEP). COPS is reliable. It uses TCP as its transport protocol. Port number 3288 is used on server side. The PEP is responsible for initiating a TCP connection to a PDP. Then PEP uses this TCP connection to send policy requests to and receive policy decisions from the remote PDP. In other words, it employs the client/server model. COPS is also extensible. It can be used for general administration, configuration, and policy enforcement.

LDAP is the other protocol, which is "a standardized TCP/IP protocol for access to a central X.500-based directory that is shared by many different services."[[QOSF2](#)] PDP uses LDAP to retrieve policy information from the policy repository.

The feature of this architecture is that: first, it's a general purpose architecture, which can be used to manage many different policy information. Second, the policy information stored in the policy repository is vendor- and device-independent. Users just need to enter high level, user-friendly policy rules, which in turn can be translated automatically into low-level, vendor- and device-specific rules or commands by PDP or PEP before they are actually executed. Third, it's also much easier to check the consistency of the policy rules and detect the potential conflicts. In short, it makes policy management much easier.

5.4 Policy-based Routing vs. QoS-based Routing

Policy-based routing is related to QoS-based routing. Although we can have all kinds of policy, and the policy architecture mentioned in the previous section is a general model, QoS policy is the immediate interest. QoS requirements can be further divided into two types: *provisioned QoS* and *signaled QoS* [[HP99](#)]. Provisioned QoS is usually provided by statically configuring the network resources, which in turn produces some QoS treatment for certain traffic. It is natural to provide provisioned QoS using policy rules. While signaled QoS is provided in a dynamic, on-demand fashion by using some signal protocol(RSVP, for example). The signal contains some specific QoS requirement information. Thus QoS-based routing is more suitable for providing signaled QoS. These two schemes are expected to be implemented together in the same network environment.

5.5 Current status and future directions

By now, the standards about policy management are still underway. The Policy Framework work group and RAP work group work together to define them. As mentioned earlier, the policy framework, the core information model and schema, and COPS are proposed and will be standardized.

Besides, there are also some other standard bodies such as DEN(Directory Enabled Networking) Ad Hoc Working Group and Distributed Management Task Force(DMTF)[[DMTF](#)]. DEN "defines a directory as a centralized repository that coordinates information storage and retrieval, enabling other data- and application-specific repositories to be united. Applications include QoS policy information."

The DEN standards have been incorporated into DMTF's Common Information Model(CIM).

Another IETF working group related to routing policy is the Routing Policy System Working Group [[RPS](#)], which defines a language called Routing Policy Specification Language(RPSL) to describe routing policy constraints[[RFC2622](#)]. Besides, a distributed registry model for publishing routing policy constraints is also defined.

By now, policy related network products are already available on the market. And customers are starting to use such products. For example, Cisco IOS(Internet Operating System) supports policy-based routing since version 11.0[[CISCO1](#)]. Vendors such as Cisco, Nortel, HP and 3Com already have policy management products.

Since the standards are not finished yet, the products today are not quite compatible with each other. Although some vendors made some efforts in that direction(several vendors support Cisco IOS in their products), the compatibility is rather limited. Besides, today's products focus primarily on enterprise network. Among the products from 12 vendors tested by Data Communications Magazine[[DATA](#)], all but one support only Ethernet environment, while the other one support ATM. The range of management is single domain. The policy management across multiple administrative domain is the future direction.

[Return to Table of Contents](#)

6. Summary

To provide QoS guarantee in the Internet, QoS-based routing is an important component. In this paper we introduce the concepts of QoS and QoS-based routing, examine different QoS-based routing algorithms and its relations to some other QoS techniques. QoS-based routing for multicast and wireless network are also included. Due to the complexity of the problem, by now, QoS-based routing is still in the research stage.

Another related technique is policy-based routing, which means making routing decision based on administrative policies. The management of policy is the current research interest. Policy related products are already available.

Both QoS-based routing and policy-based routing belong to constraint-based routing, which is the routing scheme subject to multiple constraints(including QoS constraints or policy constraints).

[Return to Table of Contents](#)

References

The following references are organized approximately in the order of their usefulness and relevance.

[RFC2386] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, Aug. 1998, 37 pages, <http://www.ietf.org/rfc/rfc2386.txt>

- [XN98] Xipeng Xiao and Lionel M. Ni, "Internet QoS: the Big Picture", IEEE Network Magazine, pp.8-18, March/April, 1999
- [Aukia] Petri Aukia, "Quality of Service Based Routing", <http://www.nixu.fi/~peba/study/qosrouting>
- [AGKT98] G. Apostolopoulos, R. Guerin, S. Kamat, S. K. Tripathi, "Quality of Service Based Routing: A Performance Perspective", pp.17-28, ACM SIGCOMM'98 Vancouver, B.C., 1998
- [AGKT99] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. "Improving QoS Routing Performance Under Inaccurate Link State Information." Proceedings of the 16th International Teletraffic Congress (ITC'16), Edinburgh, United Kingdom, June 7-11, 1999, <http://www.seas.upenn.edu/~guerin/publications/itc16.ps.gz>
- [MS98] Indu Mahadevan and Krishna M. Sivalingam, "An Architecture for QoS guarantees and Routing in Wireless/Mobile Networks", pp.11-20, ACM WOWMOM'98, Dallas Texas, USA, 1998
- [FBP98] M. Faloutsos, A. Banerjea, R. Pankaj, "QoS MIC: Quality of Service sensitive Multicast Internet protoCol", pp.144-153, ACM SIGCOMM'98 Vancouver, B.C., 1998
- [WC96] Zheng Wang and Jon Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", IEEE JSAC, 14(7): pp. 1288-1234, Sep. 1996
- [RFC2676] G. Apostolopoulos, et. al., "QoS Routing Mechanism and OSPF Extensions", RFC 2676, Aug. 1999, 50 pages, <http://www.ietf.org/rfc/rfc2676.txt>
- [LMJ98] C. Labovitz, G. R. Malan, F. Jahanian, "Internet Routing Instability", pp.515-528, IEEE/ACM Transactions on Networking, Vol. 6, No. 5, Oct. 1998
- [LO98] Dean H. Lorenz, Ariel Orda, "QoS Routing in Networks with Uncertain Parameters", pp.768-778, IEEE/ACM Transactions on Networking, Vol. 6, No. 6, Dec. 1998
- [RFC1633] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, Jun. 1994, 33 pages, <http://www.ietf.org/rfc/rfc1633.txt>
- [CC97] K. Carlberg, J. Crowcroft, "Building Shared Trees Using a One-to-Many Joining Mechanism", pp. 5-11, Computer Communication Review, Jan. 1997
- [Nortel] Nortel Networks, "IP QoS -- A Bold new network", Nortel white paper, 24 pages, http://www.nortelnetworks.com/solutions/collateral/qos_wp.pdf
- [Ma98] Q. Ma, "Quality-of-Service Routing In Integrated Services Networks", Ph.D. thesis, Computer Science Department, Carnegie Mellon Univ., 1998, 153 pages
- [QOSR] IETF Policy Framework(policy) working group, <http://www.ietf.org/html.charters/qosr-charter.html>
- [POLICY] IETF QoS Routing(qosr) working group, <http://www.ietf.org/html.charters/policy-charter.html>

- [RAP] IETF Resource Allocation Protocol working group, <http://www.ietf.org/html.charters/rap-charter.html>
- [RPS] IETF QoS Routing(qosr) working group, <http://www.ietf.org/html.charters/rps-charter.html>
- [CHEN99] Shigang Chen, "Routing Support for Providing Guaranteed End-to-End Quality-of-Service", Ph.D. thesis, UIUC, May 1999, 207 pages, <http://cairo.cs.uiuc.edu/papers/SCthesis.ps>
- [FH98] Paul Ferguson and Geoff Huston, "Quality of Service", Wiley Computer Publishing, 1998, 266 pages
- [IETFWG] Active IETF Working Groups, <http://www.ietf.org/html.charters/wg-dir.html>
- [RFC2475] S.Blake, et. al., "An Architecture for Differentiated Services", RFC 2475, Dec. 1998, 36 pages
<ftp://ftp.isi.edu/in-notes/rfc2475.txt>
- [MPLS] R. Callon, et. al., "A Framework for Multiprotocol Label Switching", draft-ietf-mpls-framework-05.txt, Sep. 1999, 69 pages,
- [STEVENS99] M. Stevens, et. al., "Policy Framework", draft-ietf-policy-framework-00.txt, Sep. 1999, 31 pages,
- [MES99] B. Moore, E. Ellesson, J. Strassner, "Policy Framework Core Information Model", draft-ietf-policy-core-info-model-02.txt, Oct. 1999, 69 pages,
- [SEMM99] J. Strassner, et. al., "Policy Framework LDAP Core Schema", draft-ietf-policy-core-schema-06.txt, Nov. 1999, 46 pages,
- [SRS99] Y. Snir, Y. Ramberg, J. Strassner, "QoS Policy Framework Information Model and Schema", draft-snir-policy-qos-infomodel-00.txt, Jun. 1999, 45 pages,
- [DMTF] Distributed Management Task Force, <http://www.dmtf.org>
- [RFC2622] C. Alaettinoglu, et. al., "Routing Policy Specification Language (RPSL)", RFC 2622, Jun. 1999,
<ftp://ftp.isi.edu/in-notes/rfc2622.txt>
- [HP99] HP, "A Primer on Policy-based Network Management," Sep. 1999, 30 pages,
<http://www.openview.hp.com/docs/429.PDF>
- [QOSF1] "Introduction to QoS Policies", Jul. 1999, 22 pages, http://www.qosforum.com/white-papers/qospol_v11.pdf
- [QOSF2] "Quality of Service - Glossary of Terms", May 1999, 26 pages,
<http://www.qosforum.com/white-papers/qos-glossary-v4.pdf>
- [ACFH92] G. R. Ash, J. S. Chen, A. E. Frey and B. D. Huang, "RealTime Network Routing in a Dynamic Class-of-Service Network", Proceedings of ITC 13, 1992

[YPG99] R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", draft-ietf-rap-framework-03.txt, Apr. 1999, 20 pages.

[COPS] Jim boyle, et. al., "The COPS(Common Open Policy Service) Protocol", draft-ietf-rap-cops-08.txt, Nov. 1999, 35 pages.

[HERZOG] Shai Herzog, "RSVP Extensions for Policy Control", draft-ietf-rap-rsvp-ext-06.txt, Apr. 1999, 12 pages, <http://www.ietf.org/internet-drafts/draft-ietf-rap-rsvp-ext-06.txt>

[DATA] Data Communications, "The Policy Makers", May 1999, <http://www.data.com/issue/990507/policy.html>

[CISCO1] Cisco white paper, "Policy-Based Routing"
http://www.cisco.com/warp/public/cc/cisco/mkt/ios/tech/tch/policy_wp.htm

[CISCO2] Cisco white paper, "Introduction: Quality of Service Overview",
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcintro.htm

[ELOV] Kimmo Elovainio, "Classless and policy-based routing",
http://www.tcm.hut.fi/Opinnot/Tik-110.551/1996/class_rout.html

[Return to Table of Contents](#)

List of Acronyms

AAL - ATM Adaptation Layer

AS - Autonomous System

BGP - Border Gateway Protocol

CBR - Constant Bit Rate

CIM - Common Information Model

COPS - Common Open Policy Service

DiffServ - Differential Services

I-D - Internet Draft

IETF - Internet Engineering Task Force

IntServ - Integrated Services

LDAP - Lightweight Directory Access Protocol

MPLS - MultiProtocol Label Switching

OSPF - Open Shortest Path First Protocol

PDP - Policy Decision Point

PEP - Policy Enforcement Point

PNNI - Private Network-Network Interface

QoS - Quality of Service

RAP - Resource Allocation Protocol

RFC - Request for Comments

RIP - Routing Information Protocol

RPSL - Routing Policy Specification Language

RSVP - Resource reSerVation Protocol

SLA - Service Level Agreement

UBR - Unspecified Bit Rate

[Return to Table of Contents](#)

Date Last Modified: 12/1/1999

Note: This paper is available on-line at http://www.cse.ohio-state.edu/~jain/cis788-99/qos_routing/index.html