Support for Real-Time Traffic in the Internet, and QoS Issues

Mansoor Mohsin, Winnie Wong, and Yogesh Bhatt
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75083-0688
{mmohsin, winniew, byogesh}@utdallas.edu

Abstract

Over the last few years the Internet has gone through an incredible increase in size and traffic. This evolution is driven by the development of new applications. Despite the increase in the number of users, the availability of high bandwidth, and the growth of commercial of networks, these applications are still unable to fully exploit the capabilities of the network. As more and more business is being conducted over the Internet, this has let to a considerable amount of research within the field of Quality of Service (QoS) and support for real-time traffic. This paper provides an overview of the QoS protocols and architectures available for the Internet. In particular we discuss support for real-time traffic in TCP/IP and ATM networks.

1 Introduction

Traditional Internet provides *best effort* data delivery by default. This is because at the time Internet was designed, there were no real-time and business applications running on it. Although this solution works well with typical Internet applications like email, file transfer, and web applications, other applications with real-time requirements such as those that deliver multimedia cannot adapt to inconsistent service levels with increase in traffic.

Meeting Quality of Service (QoS) is an end-to-end issue and QoS assurances should be applied to the complete flow of real-time traffic, i.e. from the sender, across the network and to the receiver. This generally requires end-to-end admission testing and resource reservation in the first instance, followed by careful co-ordination of disk and thread scheduling in the end-system, packet/cell scheduling and flow control in the network, and finally active monitoring and maintenance of the delivered Quality of Service [1].

1.1 What is Quality of Service?

There is more than one way to characterize Quality of Service (QoS). In the simplest sense, QoS means providing consistent, predictable data delivery. In other words, satisfying customer application requirements.

QoS does not create bandwidth. It isn't possible for the network to give what it does not have, so bandwidth availability is the starting point. QoS only manages bandwidth so that it can be used more effectively to meet the range of application requirements. The goal of QoS is to provide some level of predictability and control beyond the current *best effort* service. This should be done in such a way that best-effort traffic is not starved

after QoS reservations are made, i.e. QoS-enabled (high-priority) applications must not disable the mundane (low-priority) Internet applications.

Any QoS assurances are only as good as the weakest *chain* between sender and receiver. Hence any solution to support real-time traffic should take into consideration the overall QoS architecture that spans the entire network. In other words, just increasing the bandwidth at the backbone is not sufficient to support applications that are running on a network with low bandwidth and high congestion.

1.2 Why Quality of Service is Needed?

The Internet Protocol (IP), and the architecture of the Internet itself, is very simple and based on the concept that datagrams with source and destination addresses can traverse a network of (IP) routers independently, without the help of their sender or receiver. The robustness of Internet is a result of this simplified model, which is keeping it still running despite an enormous growth in it's size and traffic over the past several years.

There is a price to pay for this simplicity, however. The reason IP is simple is because it does not provide many services. IP provides addressing, and that enables the independence of each datagram. IP can fragment datagrams (in routers) and reassemble them (at the receiver), and that allows traversal of different network media. But IP does not provide reliable data delivery. Routers are allowed to drop IP datagrams without notice to sender or receiver. IP relies on upper-level transports (e.g. TCP) to keep track of datagrams, and retransmit as necessary. These "reliability" mechanisms can only assure data delivery; neither IP nor its high-level protocols can ensure timely delivery or provide any guarantees about data throughput. IP provides what is called a *best effort* service. It can make guarantees about when data will arrive, or how much it can deliver.

This limitation has not been a problem for traditional Internet applications like web, email, file transfer, and the like. But the new breed of applications, including audio and video streaming, demand high data throughput capacity (bandwidth) and have low-latency requirements when used in two-way communications (i.e. conferencing and telephony). Public and private IP Networks are also being used increasingly for delivery of mission-critical information that cannot tolerate unpredictable losses.

1.3 What are the benefits?

As business is increasingly conducted over the web, it becomes more important that IT managers ensure that these networks deliver appropriate levels of quality. QoS technologies provide tools for IT managers to deliver mission critical business over the public network.

With the passage of time, applications are getting more demanding. Mission-critical applications deployed over IP networks increasingly require quality, reliability, and timeliness assurances. In particular, applications that use voice, video streams, or multimedia must be carefully managed within an IP network to preserve their integrity.

1.4 QoS Parameters

Following are a list of parameters that give a measure of Quality of Service [2]:

- **Delay** is the elapsed time between a node sending a message and another node receiving that message. It is a measure of the amount of data held in *transit* in the network. The greater the delay between sender

and receiver, the more insensitive the feedback loop becomes and therefore the end-to-end protocols become more insensitive to short term dynamic changes in network load. For interactive voice and video applications, the introduction of delay causes the system to appear unresponsive, e.g. in an Internet telephony application there is sometimes a delay of upto several seconds before the receiver can hear what the sender is saying. This greatly hinders interactive communication between the two parties.

- **Jitter** is the variation in end-to-end transit delay. It is an aberration that occurs when video or voice is transmitted over a network, and packets do not arrive at its destination in consecutive order or on a timely basis, i.e. they vary in latency. High levels of jitter in applications are unacceptable in situations where the application is real-time based, such as an audio or video signal. In such cases, jitter causes the signal to be distorted which is particularly damaging to multimedia traffic. For example, the playback of audio or video data may have a jittery or shaky quality.

- **Bandwidth** is a measure of data transmission capacity. It is the maximal data transfer rate that can be sustained between two end points. By increasing bandwidth we can transfer more data. Network bandwidth can be visualized as a pipe that transfers data. The larger the pipe, the more data can be sent through it. By increasing bandwidth, we can always achieve QoS. This is a brute force solution and not applicable because bandwidth is not cheap. Hence the issue here is to obtain certain level of QoS by using the minimum bandwidth required.

- **Reliability** is a property of the transmission medium and can be thought of as the average error rate of the medium. An unreliable or error-prone network is a result of faulty channels that not only drop packets in transit but also alter their order. Unreliability causes induced distortion in the original signal at the receiver's end.

2 QoS Protocols and Architectures

The simplest solution to support real-time traffic during peak periods is to increase network bandwidth, i.e. to provide surplus bandwidth capacity in anticipation of these peak data rates during high-demand periods. However, this is not enough to avoid jitter during traffic bursts, since it is not possible to predict bursty traffic and the network region on which it might occur. Moreover, it is obvious that this solution is not scalable and economically viable.

To provide service guarantees for real-time traffic, i.e. some level of quantifiable reliability, the network must be supplemented with the ability to differentiate traffic with strict timing requirements from those that can tolerate delay, jitter and loss. A number of QoS protocols have evolved to satisfy the variety of application needs. Some applications are more stringent about their QoS requirements than others, and for this reason we have two basic types of QoS available [3]:

- **Resource Reservation (Integrated Services)** Network resources are allotted according to an application's QoS request, and subject to bandwidth management policy. RSVP provides mechanisms to do this, as a part of the IntServ architecture.

- **Prioritization (Differentiated Services)** Network traffic is classified and allotted network resources according to bandwidth management policy criteria. To enable QoS, network elements give preferential treatment to traffic identified as having more demanding requirements.

We will discuss these types of QoS in detail when we talk about TCP/IP and ATM networks in sections 3 and 4.

These types of QoS can be applied to individual application flows or to flow aggregates. Applications, network topology and policy dictate which type of QoS is most appropriate for individual flows or aggregates. Next we discuss a number of different QoS protocols and algorithms to accommodate the need for these different types of QoS.

2.1 ReSerVation protocol (RSVP)

The RSVP protocol is defined in [4]. RSVP is used by a host to request specific qualities of service from the network for particular application data streams or flows. RSVP is also used by routers to deliver QoS requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. RSVP requests will generally result in resources being reserved in each node along the data path.

Given below are the main attributes of RSVP:

- RSVP requests resources in only one direction, i.e. it treats a sender separately from a receiver, although the same application might be running at both the sender and the receiver.
- RSVP makes resource reservation for both unicast and multi-cast applications.
- RSVP is receiver-oriented, i.e. the receiver of a data flow initiates and maintains the resource reservation used for that flow.
- RSVP itself is not a routing protocol, but it is designed to work with the existing routing protocols.
- RSVP supports both IPv4 and IPv6.

To make a resource reservation at a node [5], the RSVP daemon communicates with two local decision modules, *admission control* and *policy control*. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has administrative permission to make the reservation. If either check fails, the RSVP program returns an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a *packet classifier* and *packet scheduler* to obtain the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream.

Although RSVP provides the highest level of QoS available, but it comes at the price of complexity and overhead. Simpler, less fine-tuned methods are needed, and this is what DiffServ provides.

2.2 Differentiated Services (DiffServ)

Differentiated Services provides a coarse and simple way to categorize and prioritize network traffic (flow) aggregates. The goal of DiffServ framework is to provide a means of offering a spectrum of services in the Internet without the need for per-flow state and signaling in every router.

The Differentiated Services architecture contains two main components [6]. One is the fairly well-understood behavior in the forwarding path and the other is the more complex background policy and allocation component

that configures parameters used in the forwarding path. The forwarding path behaviors include the differential treatment an individual packet receives. Based on network policies, different kinds of traffic can be marked for different kinds of forwarding. Resources can then be allocated according to the marking and the policies. The process of delivering a DiffServ service consists of the following three steps:

- The IP header contains a Differentiated Services Code Point (DSCP), indicating the level of service desired.
- The DSCP maps the packet to a particular mapping behavior (PHB) for processing by DS-compliant router.
- The PHB provides a particular service level (bandwidth, queueing, and dropping decisions) in accordance with network policy.

The Differentiated Services Code Point (DSCP) is defined as the first six bits of the IPv4 TOS octet and the IPv6 Traffic Class octet. DSCP is used to select the PHB a packet experiences at each node. A two-bit currently unused (CU) field is reserved.

Per-hop behaviors (PHBs) are applied to the traffic at a network *ingress* point (network border entry) according to pre-determined policy criteria [3]. The traffic may be marked at this point, and routed according to the marking, then unmarked at the network *egress* (network border exit). DiffServ assumes the existence of a service level agreement (SLA) between networks that share a border. The SLA establishes the policy criteria, and defines the traffic profile. It is expected that traffic will be policed and smoothes at egress points according to the SLA. The policy criteria used can include time of day, source and destination addresses, transport, and/or port numbers.

For example, mission-critical packets could be encoded with a DSCP that indicated a high bandwidth, 0-frame-loss routing path. Interactive video conferencing data and all data from the CEO's computer may carry the same requirement and be aggregated with mission-critical packets. E-mail and web browsing data could be coded with a DSCP indicating routine traffic handling with minimal packet drops. The DS-compliant boundary router would then make route selections and forward the packets accordingly as defined by network policy and the PHBs the network supports. The highest class traffic would get preferential treatment in queuing and bandwidth while the lower class packets would be relegated to slower service.

2.3 Multiprotocol Labeling Switching (MPLS)

Multiprotocol Label Switching (MPLS) is similar to DiffServ in some respect [3], as it also marks traffic at ingress boundaries in a network, and unmarks at egress points. But unlike DiffServ, which uses the marking to determine priority within a router, MPLS markings are primarily designed to determine the next router hop.

As a packet of a connectionless network layer protocol travels from one router to the next, each router makes an independent forwarding decision for that packet. Packet headers contain considerably more information than is needed simply to choose the next hop. MPLS simplifies the routing process while it also increases flexibility with a layer of indirection. Given below are the steps used by MPLS-enabled routers, called a *Label Switching Router* (LSR) [7]:

- At the first hop router in he MPLS network, the router makes a forwarding decision based on the destination address, then determines the appropriate label value, attaches the label to the packet and forwards it to the next hop.

- At the next hop, the router uses the label value as an index into a table that specifies the next hop and a new label. The LSR attaches the new label, then forwards the packet to the next hop.

In MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "labeled" before they are forwarded. At subsequent hops, there is no further analysis of the packet's network layer header. Rather, the label is used as an index into a table which specifies the next hop, and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop. The route taken by an MPLS-labeled packet is called the *Label Switched Path* (LSP).

The MPLS technology can operate over various link-level technologies, which includes packet-over-Sonet, frame relay, ATM, Ethernet, and token ring. MPLS combines layer 2 switching technology with layer 3 network layer services while reducing the complexity and operational costs.

2.4 Subnet Bandwidth Manager (SBM)

Subnet Bandwidth Management (SBM) is a signaling method and a protocol for RSVP-based admission control over IEEE 802-style LANs [8]. SBM provides a method for mapping an Internet-level setup protocol such as RSVP onto IEEE 802-style networks. In particular, it describes the operation of RSVP-enabled hosts/routers and link layer devices (switches, bridges) to support reservation of LAN resources for RSVP-enabled data flows.

The SBM and its use for admission control and bandwidth management is based on the following architectural goals:

- SBM manages bandwidth over both legacy and newer LAN topologies and takes advantage of the additional functionality (such as an explicit support for different traffic classes or integrated services classes) as it becomes available in the new generation of switches, hubs, or bridges.
- SBM specifies only a signaling method and protocol for LAN-based admission control over RSVP flows. It does not define any traffic control mechanisms for the link layer.
- In the absence of any link-layer traffic control or priority queuing mechanisms in the underlying LAN (such as a shared LAN segment), the SBM-based admission control mechanism only limits the total amount of traffic load imposed by RSVP-enabled flows on a shared LAN.

The SBM server contains the limitations of available resources that SBM clients or RSVP-aware clients can reserve on the managed subnet. SBM limits only traffic imposed by RSVP-based flows. Therefore, SBM admission control focuses solely on the QoS traffic. Reservations are made and kept track of. Because TCP/IP best-effort traffic is rate-adaptive, it uses the Slow Start algorithm during periods of congestion and adapts to stay within the LAN's bandwidth. If the client is not QoS-aware (RSVP), the client should use the SBM client software for admission control, and then keep the traffic control defined by IEEE 802 (802.1p). If the client is RSVP enabled, SBM expects traffic control to stay within the boundaries previously implemented in the RSVP reservation.

A fundamental requirement in the SBM framework is that all traffic must pass through at least one SBM-enabled switch. The primary (logical) components of the SBM system are [3]:

- **Bandwidth Allocator (BA)** maintains state about allocation of resources on the subnet and performs admission control according to the resources available and other administrator-defined policy criteria.
- **Request Module (RM)** resides in every end-station and *not* in any switches. The RM maps between Layer 2 priority levels and the higher-level QoS protocol parameters according to the administrator-defined policy.

The SBM protocol provides an RM-to-BA or BA-to-BA signaling mechanism for initiating reservations, querying a BA about available resources, and changing or deleting reservations. The SBM protocol is also used between the QoS-enabled application and the RM.

3 QoS and TCP/IP

Real time communication applications are highly time sensitive in a sense that communication is only useful if message reaches the destination in a given time. Time deadlines are end-to-end in nature with an upper bound on maximum delay and delay variation (known as jitter). Messages received after deadline expiration, are of little use or of no use at all. On the other hand early arrival may also be expensive since it requires buffering at the destination to meet constant end-to-end delay-jitter requirement. For packet data applications queuing delay is significant delay since packetization, switching and propagation delays are fixed and deterministic.

Real-time communication applications are classified based on packet loss tolerance because the networking mechanisms to guarantee desired packet loss rate vary significantly. Applications, which tolerate some amount of loss, are classified as soft real-time while applications with zero loss tolerance are classified as hard real-time. Since soft real-time applications require less stringent service, interleaving other non-real time traffic can maximize network utilization. On the contrary, in hard real time applications, deterministic predictability of network delays precedes network utilization considerations. Another important performance metric for real-time traffic is delay jitter, often defined as the maximum variation in delay experienced by messages in a single session. For example, if the minimum end-to-end delay seen by any packet in a call is 1 ms and the maximum delay is 6 ms, then the delay jitter of the call is 5 ms. Many real-time applications, particularly those which are interactive, require a bound on jitter, in addition to a bound on the delay. Some methods of real-time communication specifically manage the jitter, while others do not. Also, certain applications such as non-interactive television and audio broadcasting require bounds on jitter but not delay while other applications such as multimedia video and audio may tolerate some packet loss but can not tolerate delay and jitter.

3.1 Goals of Real-Time Communication Techniques

Any approach to real-time communication can be evaluated based how it provides real-time message delivery with either low or zero loss rates (soft or hard real-time, respectively) and low jitter. Comparing mechanisms against the goals of Real-time communications [11] include:

- **Delay-Jitter Performance** What are the values of the minimum and/or maximum latency, which can be guaranteed, i.e., what kind of real-time performance levels can a mechanism provide?
- **Optimizing Network Utilization** Can the mechanism accommodate non-real-time traffic as well? Can such traffic be accommodated at other than the lowest priority?

- **Adaptability to Network Dynamics** How robust and adaptable is a mechanism in the face of dynamically changing network configuration and traffic requirements?
- **Scalability and Traffic Aggregation** Does the mechanism scale for the large number of connections likely to be carried by a backbone network? Are there possibilities for aggregation of similar traffic streams?
- **Buffer Space Requirement and Null Buffering Related Packet Loss** How much buffer space is required in the end nodes and within the network? Can the buffer requirement be bounded to rule out packet loss due to buffer overflow?
- **Bandwidth Allocation for Real-Time Traffic** What fraction of the bandwidth can be utilized by real-time traffic?
- **Information Overhead** How much information needs to be carried within each packet?
- **Computational Overhead** What is the per-connection and per-packet computational load at the edges and within the network? For a 1Gbit/second ATM link, less than half a microsecond is available to process each packet?

3.2 Characteristics of Real-Time Traffic

Real-time communication traffic in general doesn't impose any inherent traffic characteristics and speed range varies widely but many of the sources have distinct characteristics, which set them apart from traditional data traffic. For example, multimedia conferencing, remote medical diagnosis, command and control systems, distributed interactive simulation and games. Some general properties of data rate, packet size and loss tolerance assist the network in determining the resources to be allocated, so characterization this characterization must be a best compromise among "unambiguity", "ease of specification", "enforceability" and "ability to map to resource reservation". This characterization serves two purposes: protects from misbehaving clients and distinguishes between negotiated traffic and excess traffic.

3.3 Considerations for Supporting Real-Time Traffic

Following are the design considerations affected by requirements to support real-time traffic at various levels of network architecture:

3.3.1 Resource Aware Connection Setup

Before initiating a new connection to the network, the session layer protocols negotiate between the sender and receiver to establish an agreed upon grade of service [11].

This determines the local resource requirements at end hosts. For example, during the setup of a video-conference between two workstations, the session layer protocols may reserve a certain percentage of workstation processing capabilities to sustain packet video reception, decoding and display for the duration of the conference. The next task is to find a suitable route, which meets these service requirements. Many techniques for routing in packet switched networks have been proposed and used [11]. These techniques generally optimize either the path length (i.e., minimum number of hops), or the link utilization (i.e., minimize the maximum utilization). For real-time packet switched networks, resource reservation (whether for exclusive use by a session

or in a statistically shared manner) results in a routing problem that has many of the characteristics similar to the circuit switched (telephone) call setup. The route chosen must satisfy the requirements of the new call, without invalidating the guarantees made to existing connections. Ideally, the overall network utilization should be maximized; noting that calls with the same bandwidth-consumption may have different value to the network provider.

3.3.2 Errors and FLow Control

The transport layer deals with end-to-end issues, such as flow and error control. For most applications, the end-to-end flow control normally necessary to keep the sender from over-running the receiver is built in so that receiver keeps up with the sender's data rate. Transport layer protocols like TCP has mechanism to adjust flow in reaction to network congestion. This mechanism's interference with end-to-end delay guarantee mechanism is an open issue yet to be resolved (TCP retransmission may delay packet beyond real-time delay expiration). Protocols for data transmission perform end-to-end error detection and recovery by embedding an error detection code and sequence numbering of packets. Successful transmission is acknowledged, with a lack of acknowledgment or a negative acknowledgment indicating the need for retransmission. For real-time traffic, retransmission is often unworkable as the delays incurred by timeouts and retransmission is likely to exceed the deadline for the missing packet. Particularly if reordering is possible, the receiver is only going to know that the packet was lost after the expiration of the network's delay guarantee. Thus, the application would have to tolerate a total delay and possibly also the minimally acceptable of three times the delay guarantee provided by the network in order to make use of retransmission [11].

3.3.3 Policing and Packet Scheduling

During the session, devices at the edge of the network need to police the packet stream to enforce compliance with the negotiated traffic parameters and behavior in order to protect other calls against malicious or malfunctioning end systems and maintain service guarantees for other connections. The source itself may be able to adjust its transmission rate to changing network or receiver conditions.

The delay and loss experienced by the packet between entering and leaving the network is cumulative result of actions taken at each of the nodes along its path. For each output link there are one or more queues of packets waiting to be transmitted. There must thus be a policy for selecting the queue in which to insert an outgoing packet, and position in the queue where the packet is to be inserted. Hence, the network design should provision different queues for different negotiated service traffic. The queuing discipline should also specify the order in which queues are serviced. As the standard FC-FS discipline is usually unsatisfactory for real-time services, other, e.g., deadline based policies are called for [11].

3.3.4 Isochronous Applications

The audio and video applications previously discussed form an important subclass of real-time applications: the receiver is expected to deliver data a fixed amount of time after its generation to the final consumer (for example, the video frame buffer), reconstructing the timing pattern at the sender exactly [11]. In telephony parlance, these applications are referred to as isochronous. Unless packets traversing the network experience deterministic delays, isochronous applications have to delay packets that arrive before their deadline (play out time) to compensate for the stochastic network delay (delay jitter). The synchronization method depends on

whether the isochronous data stream can be broken up into smaller units that can be shifted slightly with respect to each other (synchronization units). Voice data streams can be broken up by silence detectors into talks-purts.

3.4 Unsatisfactory Approaches to Real-Time Communications

There are several mechanisms for supporting real time communications, which are considered to be unsatisfactory. For instance circuit switching can meet real time delivery guarantees easily because it sets aside fixed portion of the network bandwidth as per the estimated peak bandwidth requirement of individual application. Since real time traffic is often very bursty it leads to lower effective bandwidth utilization, unless idle time is filled by non real time traffic. In addition, the vastly varying bandwidth granularity of wide range of services results in inefficient allocation.

So, packet switched networks should be designed aptly to meet real time requirements and optimize the resource utilization. The rest of this section is divided in to following subsections: section 3.4.1 describes major network design requirements for the support of real-time traffic. Section 3.5 illustrates causes of QoS related problems. Section 3.6 briefly defines the term "IP QoS Network" and subsequent sections details the QoS based routing schemes and various approaches proposed to meet the requirements.

3.4.1 Requirements for Supporting Real-Time Traffic

Based on above discussion, supporting real time traffic in Internet requires [11]:

1. **QoS Routing** Requirements of routing algorithms, which provide QOS guarantees to real-time traffic, are more similar to circuit switched routing algorithms than packet switched routing Algorithms. The issue is to design routing algorithms, which have call-establishment characteristics of circuit-switched algorithms like signaling to reserve resources while forwarding characteristic of packet switched network like IP-routing. Furthermore, shall some measure of schedulability (or blocking probability), rather than utilization and average delay be the performance metrics to evaluate these routing algorithms ? [11]
2. **Fault Tolerance** Detection, reconfiguration, and recovery are required for real-time communication in wide area networks [11]. This is because such networks are very much subject to fluctuations in link and node's availability, residual resources and instantaneously available resources. Any proposed method must address the need of real-time applications for continuity and extreme delay and/or jitter bound.
3. **Real-Time Packet Switching** Switches for high-speed networks will have only fractions of microseconds to execute the regulation-demux-queue-mux functions that are required for real-time communication [11].
4. **Complexity of Switches/Routers** The existing routers are already very complex. Since complexity is inversely proportional to fault tolerance, the tradeoff in the implementation complexity and ability to handle the high data rates and fast switching times of high-speed networks should be carefully weighed.

3.5 Causes of QoS-related Problems

QoS related problems could be divided into two categories [11]:

A. Non-network-related problems are caused by

1. Heavily loaded data servers that users are trying to access; In this case, the only way to improve QoS is to upgrade the data servers or increase the number of data servers and use a better load-balancing scheme among the data servers.
2. Human errors; Configuring routers/switches is a complex and error-prone process. For example, duplicate IP addresses can be mistakenly configured and cause routing problems.

B. Network-related problems are:

1. Equipment problems; Today, routers/switches are very complex systems with sophisticated software and hardware that are required to process millions of packets per second. Equipment vendors are compelled to deliver products as early as possible. It is thus no big surprise that routers/switches have hardware and software problems. Therefore, something must be done to increase network availability, especially for important traffic.
2. Lack of access capacity; For economic reasons, there are always customers with slow access links (e.g. dialup modems) and/or over-subscribed uplinks. Technical solution for this kind of problem is clear:
   a) Adding capacity;
   b) Classifying traffics and marking them differently for different subsequent treatment using policing, shaping, etc. However, it should be pointed out that providing QoS might not make economic sense here because users are not willing to pay for it.
   c) Uneven traffic distribution resulting in several links to be congested, while others underutilized.

3.6 QoS IP Network

QoS IP Network shall support packets with QoS guarantees in addition to continuing support for best effort packets. The distribution of network resources between these two classes should be done in way that assures enough resources to packets with QoS guarantee without starving the best effort service.

1. Dedicate some of the resources to satisfy the requirements of QoS packets
2. Identify and advertise resources that remain available to new QoS flows.

Since IP is connection less and real-time QoS requires stringent bound on delay/jitter, the resource reservation is a must before deciding the routes. This can be achieved by having some form of signaling like circuit switched networks. The existing algorithms do not address this requirement. The signaling mechanism reserves the resources on the data path at connection initialization. Hence routers on the data path can provide QoS-level of service to the packets belonging to this flow by means of packet filtering. In the IntServ model, the data-forwarding path can be selected individually, which facilitates unique treatment of QoS in both the viewpoints, i.e. traffic engineering viewpoint and routing perspective. Each QoS capable router in the network hold proper knowledge of the load on network and based on this information, QoS routing module selects a suitable path for a particular session. In spite of the fact that QoS routing module and resource reservation works well, the IntServ network architecture fails due to many reasons. There is inadequate design of the interface between resource reservation and QoS routing.

3.7 QoS-based Routing

Selecting an optimum path for the routing of the data packets is very important, since it improves the performance significantly, when coupled with RSVP. Each individual packet is then forced via this route. In case of the multiple paths between the sender and the receiver, the best effort routing will always select the same path. In the case when the "best" path is loaded to its full under a particular QoS, then its always advisable to route the incoming traffic via the longer unused path, which leads to improved performance and overall network utilization. And if the longer paths do not accommodate the increased traffic, then an "Admission Control Error" message is returned to RSVP. This clearly indicates necessity of coordination between the RSVP and routing protocols. Therefore it seems to be remunerative to add QoS sensitivity to traditional routing mechanisms. Combining a RSVP with QoS-based routing allows fine control over the route and resources at the cost of additional state and setup time. Several recent researches are going on for the improvement in the QoS based routing. Increased customer satisfaction implies increased likelihood of development in the field. Despite these benefits, there remains uncertainty regarding the additional costs of QoS routing. Its major components are: computational overhead, due to more sophisticated and frequent routing protocols implementation and a protocol overhead, due to message passing to all the nodes in the path about the QoS requirement.

In this QoS, the delay variation is of more importance than delay [17]. The condition of varying delays is more prominent in the packet switched network than the circuit switched one. So the sole problem reduces to forming a bandwidth constrained "best" path from the source to the destination [see next section for detail]. To reduce contention and the probability of false probes, the router selects one of the paths from the "window" of paths, such that the selected path satisfies the criteria of bandwidth and the QoS requested by the host.

Path computation in itself is a search technique [17], e.g., Shortest Path FIrst (SPF) is a search technique based on dynamic programming. Each link considered by the path computation must be evaluated against the requirements of the flow i.e., the cost of providing the services required by the flow must be estimated with respect to the capabilities of the link. This requires a uniform method of combining features such as delay, bandwidth, priority and other service features. Furthermore, the costs must reflect the lost opportunity of using each link after routing the flow.

3.8 Constraint-Based Routing

It computes paths that are subject to constraints [17] such as bandwidth, QoS requirements such as delay and jitter, or other policy requirements on the paths. These constraints are mapped from QoS requirements to support real-time traffic. Constraint-based routing is an important tool for traffic engineering and for providing QoS.

Because network topology information is needed in constraint-based path selection, constraint-based routing must be used together with a link state IGP, and the IGP must also be enhanced to provide the link attribute information as well.

3.8.1 Issues of Constraint-Based Routing

Since constraint-based routing has to consider many more attributes of network in addition to network topology in computing routes, it may find a longer but lightly loaded path than the heavily loaded shortest path [17]. This avoids congestion and prevents uneven distribution of traffic leading to satisfy QoS requirements of real-time applications.

Important issues of constraint-based routing are discussed below:

1. **Link State Information and Communication Overhead**
   As mentioned above, a router needs link attributes in order to find paths with constraint [17]. Buffer space is usually not considered in constraint-based routing because buffer space can be derived from the input and output rates of the queues and the delay/jitter requirement of the packets. Therefore, link attributes usually refers to the reservable bandwidth of the links and affinity of the links.

   The available bandwidth of a link is equal to the maximum reservable bandwidth of a link (set by network administrators) minus the total bandwidth reserved. It does not depend on the actual amount of available bandwidth on that link. For example, if the maximum reservable bandwidth of a link is 155 Mbps, and the total bandwidth reserved is 100 Mbps, then the reservable bandwidth of the link is 55 Mbps, regardless of whether the link is actually carrying 100 Mbps of traffic or more or less buffer space is usually not considered in constraint-based routing [9].

   In other words, constraint-based routing does not compute paths based on instantaneous residual bandwidth of links. This reduces the probability of routing instability.

   Because link available bandwidth is frequently changing, a tradeoff must be made between the need for accurate information and the need to avoid frequent propagation of LSAs. To reduce the frequency of LSA, changes in link available bandwidth will be announced only when they are significant, e.g., more than 5% of link capacity or more than 10 Mbps. A hold-down timer should also be used to set an upper bound on the frequency on LSA propagation. A recommended minimum interval is 30 seconds [16].

2. **Path Constraint and Path Computation Complexity**
   The commonly used path constraints in constraint-based routing are bandwidth, hop count, delay, jitter, or monetary cost. The path computation algorithms select paths that optimize one or more of these constraints. If multiple constraints are to be optimized simultaneously, the complexity of the algorithms usually becomes very high. Some of them may even be NP-complete [12].

   Let $d(i, j)$ be a constraint for link (i, j). For any path $P = (i, j, k, \ldots, l, m)$, where $i, j, \ldots, l, m$ are the nodes along the path, constraint d is:

   (a) Additive if $d(P) = d(i, j) + d(j, k) + \ldots + d(l, m)$;
   (b) Multiplicative if $d(P) = d(i, j) * d(j, k) * \ldots * d(l, m)$;
   (c) Concave if $d(P) = mind(i, j), d(j, k), \ldots, d(l, m)$.

   For example, constraints such as delay, jitter, cost and hop count are of type (a), reliability is of type (b), and bandwidth is of type (c). As [13] proves, finding an optimal path with two or more additive and/or multiplicative constraints is NP-complete. The only computationally feasible combinations of constraints are bandwidth and any of the above. NP-completeness is based on the assumptions that:

   (a) All the constraints are independent; and
   (b) The delay and jitter of a link are known a priori.

   Since bandwidth, delay and jitter are not independent in packet networks, polynomial algorithms for finding paths with hop count; delay, jitter and buffer space constraints are possible. The complexity of such algorithms is $O(N * E * e)$, where N is the hop count, E is the number of links of the network, and e is the number of distinct reservable bandwidth values among all links [12].

If these constraints can be made sequential in terms of their importance, the path selection algorithm becomes much simpler because one constraint can be applied during iteration, the most important constraint being the first.

Delay and jitter constraints can be mapped to bandwidth and hop count constraints, if needed, with the queuing delay bound set to (application's delay bound - propagation delay). Propagation delay can be estimated from the distance between the source and the destination, and an inflation factor can be used to account for any detour.

A constrained based routing entails more overhead of frequent routing table computations because of significant changes in resource availability and residual bandwidth of the links even without a change in topology.

3. **Selection of Optimal Route**
   In the event of having multiple paths meeting constraints, optimal route should be chosen based on a good compromise between resource conservation and load balancing. The first and most important criterion is shortest path because it helps controlling the traffic path. The second is hop count because traversing fewer hops consumes less link bandwidth, buffer space, and processing power. The third criterion is the residual bandwidth on the path because selecting the widest paths helps in achieving lower value of highest link utilization.

3.8.2 Constraint-Based Routing with Delay and Jitter Constraints

Because real-time applications have stringent delay and jitter requirements, algorithms for selecting path with delay and jitter are important. A polynomial algorithm for selecting path with delay and jitter constraints is described [12]. Algorithms for selecting path with delay and jitter constraints are complex. Because queuing delay and jitter of packets in a flow can be determined by the amount of bandwidth reserved by the flow and the number of hops the flow traverses, algorithms for selecting path based on bandwidth and possibly hop count is practically more useful, except when propagation delay becomes a very significant part of the total delay. But because propagation delay is static for a link, algorithms for selecting path with propagation delay constraint is much simpler than algorithms that take queuing delay into consideration.

As explained earlier, delay is a useful constraint for path [17]. The path selection algorithm is complex because queuing delay of each router is not static. However, because propagation delay has become a significant part of the delay, and because the propagation delay of a link is static, path selection based on propagation delay is fairly simple. One approach to compute paths with propagation delay constraint is to use propagation delay as link metric, and use normal path selection algorithms such as the BF or Dijkstra's algorithm to compute the shortest path. To account for the queuing delay, the propagation delay requirement can be computed as follows:

$$propagationdelay = requiredapplicationdelay - queuingdelay * hopcount$$

The value of queuing delay at each hop can be set based on experience, e.g., 1 ms. This is an over-estimation of normal queuing delay. The hop count can also be over estimated. This will ensure that the propagation delay supplied to the path selection algorithm is smaller than the actual tolerable propagation delay. If the path selection algorithm can find a path that meets the propagation delay constraint, then the path will meet the application's need.

3.9 QoS-based Multicast Routing

Multicasting is a technical term that means that you can send a piece of data (a packet) to multiple sites at the same time. A multicast connection can substitute for many unicast connections carrying the same information, while reducing the load on the network.

Multicasting is supported in the Internet with an overlay network architecture known as multicast backbone (MBone). Today, the MBone is significant for distributing data, voice and video conferencing to a group of people anywhere in the Internet in a cheap and convenient way.

The QoS approaches discussed so far suffer scalability problems because of the need to maintain connection-specific information. The reason being the links are point-to-point in nature when end-to-end routing is considered and routers do not keep connection-specific state. Since multicasting groups several addresses in the form of a connection, multicast-based routing tables maintain connection-specific state. This makes adding QoS specific requirements for a connection in a routing table much easier.

3.9.1 Multicast Routing with Delay Constraint and Optimum Cost

[19] presents heuristic for NP-hard problem of finding path with bounded delay and minimum cost for multicast routing. It claims that the heuristic proposed performs very well if accurate network status information is available at the source.

3.9.2 Alternate Path Routing

As mentioned in [18], when opportunistic default route is not capable of supporting QoS requirements, we need alternate routing path to avoid a bottleneck link and route pinning to avoid service disruption.

3.9.3 QoS Sensitive Protocols

Each new member is offered multiple routes to select the best from all. QoSMIC (QoS Sensitive Multicast Internet protoCol) uses dynamic routing information without relying on a link-state exchange protocol [18]. It considers multiple paths and selects the most promising one using dynamic information to support QoS.

3.9.4 Internet Related Proposals

QoS extensions to CBT (Core Based Trees) protocol imposes QoS requirements on tree construction and maintains them by restricting new members from joining. When a new member wants to join the group it checks if the QoS level required by a new member can be met.

3.10 Other Traffic Management Schemes

In this section, we discuss the applicability of policing, shaping and Random Early Detection (RED) [14].

RED is a buffer management scheme designed to prevent tail-drop caused by traffic burst [16]. Backbone routers can generally buffer traffic for up to 100ms per port. If traffic bursts over output line-rate for longer than

that, buffer will become full and all subsequent packets will be dropped. Tail-drop causes many TCP flows to decrease and later increase their rate simultaneously and can cause oscillating network utilization. By preventing tail-drop, RED is widely believed to be useful for enhancing network performance. Weighted RED (WRED) is a more advanced RED scheme. It depends on other mechanism(s) such as classification/marking/policing to mark packets with different drop priorities. WRED will then drop them with different probabilities (which also depends on the average queue length).

However, it is difficult to set the (W)RED parameters (e.g. different drop probabilities at different queue lengths) in a scientific way to make it work well. Besides, when the backbone is over-provisioned and link utilization is maintained at around 50%, transient burst is unlikely to cause tail-drop. But (W) RED can be useful during the transient congestion caused by network failure for non-real-time traffic. In general, Traffic Engineering is more effective in controlling traffic distribution in a network and has a bigger impact on network performance, compared to traffic management schemes such as policing, shaping, and WRED. It should be invoked for QoS purpose before traffic management schemes.

3.11 Summary of IP-QoS Approaches

In this section, we are giving an overview of QoS approach proposed by [16].

[16] approach emphasizes the importance of good network design. After a network is carefully designed, Diff-Serv, a traffic protection scheme and Traffic Engineering can be deployed to make the network multi-service capable and to increase the network's survivability in emergency.

Complex schemes such as the per-flow queuing/scheduling approach are considered undesirable in [16] approach. Since routers and switches are already complex, more complex mechanisms may hurt equipment reliability and network performance. But the most fundamental problem of any complex schemes is that, even if these schemes do not compromise router reliability, it is very difficult for network operators to specify the parameters (e.g., output rate and size of a large number of queues) for these schemes to work properly. And these parameters must be adjusted from time to time to keep up with traffic shift too.

The essence of DiffServ is to divide traffic into a few classes and handle them differently, especially under adverse network conditions (e.g. caused by equipment failure) [6]. Some premium services can therefore be created to increase network revenue. Protection schemes such as fast reroute promptly switch traffic to an alternative path when there is network failure to minimize packet loss and delay. Traffic Engineering is to map traffic onto the network in such a way that congestion caused by uneven traffic distribution is avoided or can be quickly relieved. Compared to the traditional traffic management schemes such as policing, shaping and buffer management, Traffic Engineering can affect traffic and network performance at a much larger scale. Traffic Engineering therefore can be considered as macro control compared to traffic management schemes.

4 QoS and ATM

As we mention earlier, IP supports one class of service, known as the "Best Effort" point-to-point delivery service, in which reliable and fair delivery is guaranteed. This kind of services works fine for traditional data applications, like file transfer and World Wide Web access etc. However, due to the emergence and the increasing demand for QoS-sensitive applications with voice and video, in which timeliness is more important than correctness, the weakness of IP, being unable to provide QoS, makes it fail to provide quality real time traffic.

To come up with a solution to this problem, we aware that the architecture of ATM has been designed in such a way that it can be used effectively by any agent that needs to take care of QoS provisioning. So in order to incorporate QoS in IP, new classes of service, namely IntServ (Integrated Services) and DiffServ (Differentiated Services), and a QoS signaling protocol, RSVP, are being introduced in the Internet. Before we go to see how these new mechanisms utilize the inherent QoS properties of ATM, we first examine some basic principles of ATM to see what makes ATM so best suited to be a sophisticated underlying layer for QoS provisioning to support real time traffic in the Internet.

4.1 QoS Provisioning in ATM

4.1.1 Cell Structure and Hardware Switching

As compared to packet or frame which is of a variable size, fixed sized ATM cell has a predictable delay. In addition, the ATM cell structure with relatively small information field length reduces the internal buffers in the switching node, thus limits the queuing delays in those buffers, and as a result, it guarantee a small delay and a small delay jitter as required in supporting voice and video applications. Besides, as a cell is of fixed size, there is no need to trace the beginning and ending of the cell with any software, making fast and flexible hardware switching feasible to handle both constant rate traffic (voice and video) and variable rate traffic (data) easily.

4.1.2 Virtual Circuits (VCs) and Signaling Protocols

Although ATM is a packet switching technology, it is also connection oriented. That means, a logical or virtual path has to be set up before any information can be sent from the source to the destination. Supposed that a connection is not set up, each cell will has to be individually routed when it is sent, resulting in significant delays. So a virtual path helps speed up the network traffic as there involves no routing decisions. ATM offers two types of connections, permanent virtual circuits (PVCs) and switched virtual circuits (SVCs). A PVC is like a leased line such that a connection is always there, while a SVC is set up with a signaling protocol. The User-to-Network-Interface (UNI) signaling protocol handles the signaling to the network and the Network Node Interface (NNI) signaling protocol handles signaling inside of the network.

Both of these virtual circuits can be established with pre-specified QoS in point-to-point and point-to-multipoint configurations. Together with a rich set of QoS mechanisms with a wide variety of service categories or QoS descriptors and parameters, which will be described below, ATM offers a fine control over the traffic parameters requested and managed.

4.1.3 Traffic Management

Traffic management has been implemented to control traffic in the ATM network. It enables the network to provide different levels of QoS for each specific application, so as to realize the coexistence of voice, video and all kinds of data traffic in a network. Besides, traffic management helps to avoid congestion, so that bandwidth can be fully utilized and network performance does not degraded.

4.1.4 Service Categories

The various kinds of traffic ATM networks can carry are classified into four categories [24]. These ATM service categories are defined using three QoS parameters, namely CDV, CTD and CLR, which will be addressed next.

- **Constant Bit Rate (CBR)** CBR service transfers information at a consistent bit rate between the source and destination. CBR service category is tailored for applications that require smooth transitions and cannot tolerate cell loss and delay.
- **real time-Variable Bit Rate (rt-VBR) & non-real time-Variable Bit Rate (nrt-VBR)** VBR service transfers information with a variable bit rate to make the most efficient use of network resources. Both rt-VBR and nrtVBR are intended for applications that are time sensitive. The difference is that rt-VBR can tolerate occasional cell loss, while nrt-VBR can accept certain amount of jitter.
- **Available Bit Rate (ABR)** ABR service transports at a variable bit rate. In addition, it uses a feedback control mechanism to dynamically regulate the amount of traffic flows from the source, achieving flexible exploitation of available network resources. ABR service is sensitive to loss, but it makes no guarantees on accurate timing.
- **Unspecified Bit Rate (UBR)** UBR service also transports at a variable bit rate, but it gives no feedback about congestion. This category is well suited for applications that are very tolerant of delay and cell loss.

4.1.5 ATM QoS Parameters and QoS Classes

During VC connection setup, both the customer and the carrier must agree on a contract defining the service. So as to specify the quality of service desired by the customer and accepted by the carrier, and to make negotiation possible, the ATM standard defines a number of parameters [24] to quantify the QoS. Some of them are listed here. Peak Cell Rate (PCR), Sustained Cell Rate (SCR) and Minimum Cell Rate (MCR) are traffic parameters that describe the inherent characteristic of a traffic source. Cell Delay Variation (CDV), Cell Transfer Delay (CTD), Cell Loss Ratio (CLR), Cell Error Ratio (CER), Severely-Errored Cell Block Ratio (SECBR) and Cell Misinsertion Rate (CMR) are the six QoS parameters that correspond to a network performance objective. CDV, CTD and CLR are negotiable, while the other three are not. These parameters characterize the performance of an ATM layer connection and also quantify end-to-end network performance at the ATM layer.

QoS classes are associated with a particular connection and specify a set of performance parameters and objective values for each performance parameter specified. There are two types of ATM QoS classes: a Specified QoS and an Unspecified QoS class. With a specified QoS class, performance parameters are explicitly specified. On the other hand, an unspecified QoS class has no objective specified for the performance parameters. The network may have a set of internal QoS objectives for the performance parameters, thus introducing an implicit QoS class.

4.2 IP QoS over ATM

When discussing IP QoS over ATM, our focus is on IntServ and DiffServ, but we should not leave behind the ATM support for IP unicast and multicast Best Effort Services, if reliability has to be concerned.

Classical IP over ATM [20] is based on a Logical IP Subnetwork (LIS). In a LIS, routers and hosts are separately configured. Within a LIS, hosts communicate directly using the ATM network. However, if hosts in different

subnets want to communicate with each other, the connection must go through an IP router and an Address Resolution Protocol, namely ATMARP, is needed for ATM edge devices to resolve IP addresses to native ATM addresses. And similarly, to handle IP multicast, the Multicast Address Resolution Server (MARS) is applied to map an IP address into a list of native ATM addresses, rather than just a single address.

4.2.1 IntServ with RSVP over ATM

To ensure interoperability when implementing IntServ with RSVP over ATM, several issues are being discussed below [22].

IntServ uses RSVP to provide QoS. The integration of RSVP signaling and ATM signaling in support of the Integrated Service model involves QoS translation and VC management.

- **QoS translation** concerns the mapping of IntServ models to ATM QoS models. As the QoS specifications in the Integrated Services model is not identical to those in ATM, such a mapping is essential to provide effective inter-operation and end-to-end quality of service for IP Integrated Service networks containing ATM subnetworks.

  On one side, classical IP supports Best Effort (BE) service and IntServ provides two classes of QoS: the Guaranteed Services (GS) and the Controlled Load Services (CLS). On the other side, ATM provides four categories of services: CBR, rt-VBP/nrt-VBR, ABR and UBR. The following QoS translations have been recommended in [21]: i) GS maps to CBR or rt-VBR; ii) CLS maps to nrt-VBR or ABR with a minimum cell rate; and iii) BS maps to ABR or UBR. Implementation variables, policy mismatches and cost would complicate the translation.

- **VC management** is targeted to match the dynamics of the RSVP protocol. The issues of how many VCs are needed and which traffic flow goes into which VC are discussed.

  RSVP works in the following way. Both the RSVP control packets and data packets follow the same IP path through the network. So the RSVP PATH messages can install appropriate PATH state in the routers along the path for data, allowing the RSVP RESV messages to retrace the hops from the marked PATH states to make resource allocations.

  Both PVCs and SVCs can be used to set up a connection for RSVP. With PVCs, the end devices must provide traffic control services to multiplex several flows over a given VC. There is no setup time required for PVCs as they are pre-dedicated. Nevertheless, resources are limited to what has been pre-allocated and resource underutilization may result due to idle PVCs. In addition, PVCs are set up in such a way that the link would be over-subscribed if all the PVCs are simultaneously utilized, and this make it fail to satisfy the granted QoS. On the contrary, in the case of SVCs, it brings about flexibility but also increased complexity, setup time and cost.

  Point-to-multipoint VCs helps provide QoS for IP multicast. As defined in UNI 3.x and 4.0 that it can only have one service class for all the VCs' recipients, a separate VC for each recipient would still be needed if the recipients want to request different QoS. As a result, this gives rise to scaling and resource problems. ATM 4.0 provides a solution to this problem by adding a Leaf Initiated Join (LIJ) functionality. The advantage of this new capability is the reduced burden on the source because a new end user can join into an existing point-to-multipoint VC without contacting the VC source. Nonetheless, the new branches have to inherent the same QoS from the existing branches and the same scaling problem remains unsolved.

  Inspired by the idea of a multicast server, multiple senders can send cells through a point-to-multipoint VC to multiple receivers. This does not solve the VC scaling problem, but propagates it to the server.

Besides, it requires the server to be able to handle RSVP packet interpretations and implement additional functionality.

Next, we briefly address some other issues that we should also be aware of when implementing RSVP and IntServ model over ATM.

RSVP supports both unicast and IP multicast applications, in other words, it means it relies closely on both unicast and multicast routings. Although ATMARP and MARS have already provided some solutions for IP over ATM network, solution for multicast still resides on multicast routing protocols to connect multiple senders and receivers on different subnets.

To address the scaling issues, a possible solution is to aggregate several RSVP flows for the same destination over a single VC. Nevertheless, this solution increases the complexity in VCs management and packets scheduling at the VC root.

When implementing RSVP and IntServ over ATM, considerations must be given to hosts that are directly connected to ATM networks. This is because the host may not possess all of the RSVP functionalities of a router.

Some form of administrative control and cost allocation is needed to control access to different classes of preferential service created by RSVP and IntServ. Policies unique to ATM and IP over ATM and those developed for IP and IntServ are currently studied to determine if they can be inter-operated with one another.

It is obvious that ATM, the RSVP protocol and IntServ models will keep on changing in the future. It is worth to briefly list a numerous issues that will bring about improvements to the current implementations. These include:

- Short-cut-route by Next Hop Resolution Protocol (NHRP)
- A new routing protocol utilizing QoS information available in RSVP
- Support for ”heterogeneous receivers” with variegated VC
- A lighter weight signaling mechanism to replace Q.2931
- QoS renegotiation to support dynamic changes
- Many-to-many connectionless feature by group addressing
- Optimizations with MultiProtocol Label Switching (MPLS).

4.2.2 DiffServ over ATM

To ensure interoperability when implementing DiffServ over ATM, several issues are being discussed below [25, 26].

To run DiffServ over ATM, first of all, a mapping of ATM QoS parameters to DiffServ behaviors is again needed. DiffServ defines two PHBs, Expeditited Forwarding (EF) [23] and Assured Forwarding (AF). EF PHB provides a kind of service that is of low loss, low delay and low latency. This resembles the CBR or rt-VBR service in ATM. AF PHB provides a kind of service that ensures high priority packets are forwarded with a greater degree

of reliability than packets of a lower priority. This resembles the ABR or nrt-VBR service in ATM. In fact, the mapping is not that simple and identical. FIrst, DiffServ deploys PHBs, which do not really specify the QoS requested, they just mark the traffic flows. However, ATM specifies QoS specifically with traffic parameters. Second, ATM guarantees end-to-end bandwidth, but both EF and AF do not do so. So these complicate the mapping process. Details of the proposed mapping can be read from [25, 26].

Next, we consider VC management issues. DiffServ carries aggregate flows. There are two ways to handle this in ATM. Permanent ATM QoS VPs (Virtual Paths) is one that is relatively easy to implement and do not require much maintenance, but with the disadvantage of a wastage of reserved bandwidth if the path is not fully utilized. Or alternatively, when an aggregate comes across an ATM network, a separate SVC will be set up to handle each new IP DS connection. This strategy makes full use of available bandwidth and is thus cost effective, but will give rise to a high VC consumption and increased delay due to connection setup time.

As far as Connection Admission Control (CAC) is concerned, the more complicated the router topology is, the more complex and inefficient CAC is, as more information has to examined when making routing decisions.

Making use of the idea of accumulated flows in DiffServ, best efforts traffic can consume ATM bandwidth reserved for DS high quality ATM traffic provided that the bandwidth is not completely occupied by the high quality traffic. This results in economic use of resources.

DiffServ deploys a Bandwidth Broker system to keep information about QoS requests from the members of a domain, and manage and facilitate communication with external domain, which may be an ATM network, with respective to resource management and traffic control. The ATM management system needs some ways to communicate with the bandwidth broker to exchange information for providing smooth DiffServ/ATM interoperability.

Both the IP destination and the demanded IP QoS now have to be considered when forwarding a packet. It is suggested that, with permanent ATM connection, every IP combination of aggregated path address information and DSCP has to be mapped to a specific ATM VPI/VCI; with switched ATM connection, a single VP per ATM link is reserved to allow the establishment of SVCs within it. The DSCP will have to be mapped to an ATM QoS for call setup, while the aggregated path address information has to be mapped to a single VCI. The switched connection is more advantageous and offers better scalability. Nevertheless, the resources allocated for a VP limits the summed resources available for all VCs within that particular VP.

FInally, efforts need to be taken at the ATM switch as well. These include issues about bandwidth provisioning, relative resource allocation, and priority control. Much more details can be found in [25, 26].

Just as mentioned in the discussion about IntServ, MPLS mechanisms can optimize the implementation of DiffServ over ATM. This requires efforts be put on ATM switches to enable label switching. More in depth discussion on interoperability with MPLS and ATM can be read from [27].

5 QoS and Future Possibilities

Although there has been significant progress in the area of Quality of Service (QoS) architectures for the Internet, but still a lot of work still needs to be done in porting different QoS tools to a coherent platform for end-to-end service delivery.

The Internet inherently provides *best effort* service. Currently, most of the QoS architectures try to augment this base service with a number of QoS tools to provide some sort of reliability to the application. There is still an

absence of a widespread QoS-aware infrastructure that deals with QoS related issues directly.

Following are the key ingredients identifies by [28] as the next steps for QoS architecture:

1. **QoS-Enabled Applications** Currently most of the applications developed do not take advantage of any QoS infrastructure present. There is a need to define a standard API for QoS related services that the developers can use to develop QoS aware real-time applications.

2. **The Service Environment** Up till now there is no single complete service environment that provides both service accuracy and scaling properties. The IntServ model does provide accuracy but is not cost effective or scalable. Similarly, the DiffServ model offers excellent scaling properties but it is not appropriate for applications that require high level of accuracy. What we require is a compromise between the two.

3. **QoS Discovery** There is no robust mechanism for path discovery with specific service performance attributes. The assumption within both IntServ and DiffServ architectures is that the best effort routing path is used, where the path is either capable of sustaining the service load, or not. There is still further refinement that may be required in the model of service discovery and the associated task of resource reservation.

4. **QoS Routing and Resource Management** Do date there is no explicit architectural option to allow the network service path to be aligned along other than the single best routing metric path, so that available network resources can be efficiently applied to meet service requests. This area of active network resource management, coupled with dynamic network resource discovery, and the associated control level signaling to network admission systems appear to be a topic for further research at this point in time.

5. **TCP and QoS** TCP is a *self clocking* protocol that uses the acknowledgments to pace the transmission of packets. So, even if the forward link is fine and there is some problem in the reverse direction, the performance of TCP will suffer. Because of this asymmetry of TCP flow, the path taken by the forward packets of a TCP session may not exactly correspond to the path used by the reverse packet flow. Hence using some sort of symmetric service is still an open issue.

6. **Measuring Service Delivery** There is a strong requirement within any QoS architecture for network management approaches that provide a consistent view of the network, and map this view to an admission control function which can determine whether to admit to a service flow or not. The presence of network measuring methodology also allows clients to measure the delivered service quality so that any additional expense can be justified. The QoS architecture still needs to be refined to incorporate such measuring methodologies.

7. **QoS Accounting** This topic has been addressed previously within the IETF under the topic of "Internet Accounting". Any further refinements as they apply to QoS accounting may prove to be a productive initial step in formulating a standards-based model for QoS accounting.

8. **QoS Deployment Logistics** How should QoS-enabled networks should be deployed? This is a question that has not been answered yet. No network operator will make any investment in deployment of infrastructure unless there are immediate clients and applications to make use of it. Clients will not make investments unless they see performance gains in applications. Application designers will not integrate QoS features into their applications unless there is an infrastructure present to support it. So, we are in a sort of deadlock and somehow need to get out of it.

6 Conclusion

There are basically two different approaches to solve the QoS problem. These can be characterized as either state-based or stateless QoS.

The architecture of Integrated Services model provides state-based QoS. It is an application-driven process (also known as application-signaled QoS) that negotiates with the network to obtain and reserve resources that can support the application's requirements. The Differentiated Service model, on the other hand, provides stateless QoS. It decouples the network load control functions from the application, i.e. the network implements a load control function as a part of the function of admission of traffic into the network, admitting no more traffic than it can support. Integrated Services model provides the highest level of QoS while Differentiated Services model provides a greater ability to manage network resources and a greater ability to ensure the integrity of network services.

In this paper we have introduced the requirements for the support of real-time traffic in the Internet and related QoS issues. We briefly mentioned the issues and surveyed important approaches to achieve real-time traffic guarantees. Since IP is connectionless network and routing tables do not maintain state specific information, inventing QoS sensitive routing algorithms is essential. So we surveyed constraint-based routing algorithms, their complexity and other issues. Multicasting has major role in today's Internet applications, so we briefly mentioned multicasting tree construction approaches which are QoS aware. Our study reveals that an efficient QoS routing scheme which addresses all the problems is still a distant goal.

As far as ATM is concerned, it is well-designed to support real traffic, as well as its large variety of QoS provisioning, it is a good candidate to provide the currently high demand of both state-based and stateless QoS implementations in the Internet.

References

[1] Andrew Campbell, Cristina Aurrecoechea, and Linda Hauw: *A Review of QoS Architectures*, Proceedings of the 4th International Workshop on Quality of Service (IWQoS), pages 173-95, Jan de Meer and Andreas Vogel, editors, 7-8 March 1996.

[2] Paul Ferguson and Geoff Huston: *Quality of Service in the Internet: Fact, FIction or Compromise?*, INET98, July 1998.

[3] White Paper: *QoS Protocols and Architectures*, Quality of Service Forum, http://www.qosforum.com/.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin: *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*, RFC 2205, September 1997.

[5] USC Information Science Institute (ISI): *RSVP Project*, http://www.isi.edu/div7/rsvp/, March 1999.

[6] K. Nichols, S. Blake, F. Baker, and D. Black: *Definition of the Differentiated Services FIelds (DS FIelds) in the IPv4 and IPv6 Headers*, RFC 2474, December 1998.

[7] E. Rosen, A. Viswanathan, and R. Callon: *Multiprotocol Label Switching Architecture*, RFC 3031, January 2001.

[8] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, M. Speer: *SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style Networks*, RFC 2814, May 2000.

[9] XiPeng Xiao: *Providing QoS in the Internet*, PhD thesis, Michigan State University.

[10] Xipeng Xiao and Lionel Ni: *Internet QoS: A Big Picture*, IEEE Network Magazine, March/April, pp. 8-18, 1999.

[11] C. Aras, J. Kurose, D. Reeves, and H. Schulzrinne: *Real-time communication in packet switched networks*, Proceedings of the IEEE, vol. 82, pp. 122-139, January 1994.

[12] Qingming Ma and Peter Steenkiste: *Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks*.

[13] Q. Ma and P. Steenkiste: *On path selection for traffic with bandwidth guarantees*, Proceedings of IEEE International Conference on Network Protocols, (Atlanta, GA), October 1997.

[14] S. FLoyd and V. Jacobson: *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, 1(4), August 1993.

[15] E. Crawley, R. Nair, B. Jajagopalan and H. Sandick: *A framework for QoS-based Routing in the Internet*, RFC 2386, August 1998.

[16] R. Guerin, S. Kamat, A. Orda, T. Przygienda, and D. Williams: *QoS Routing Mechanisms and OSPF extensions*, Internet draft, January 1998.

[17] XiPeng Xiao and Lionel M. Ni: *A Pratical Approach for Providing QoS in the Internet Backbone*, Michigan State University, http://www.cse.msu.edu/ xiaoxipe/presentations/mplsForum.0308.pdf.

[18] M. Faloutsos, A. Banerjea, R. Pankaj: *QoSMIC: Quality of Service sensitive Multicast Internet protoCol*, ACM SIGCOMM'98, September 1998.

[19] Vachaspathi P. Kompella, Joseph C. Pasquale and George C. Polyzos, *Multicast routing for multimedia communication*, IEEE/ACM Transactions on Networking, volume 1, pp. 286-292, 1993.

[20] M. Laubach and J. Halpern: *Classic IP and ARP over ATM*, RFC 2225, April 1998.

[21] Borden, M., and M. Garrett: *Interoperation of Controlled-Load and Guaranteed-Service with ATM*, RFC 2381, August 1998.

[22] Crawley, E., Berger, L., Berson, S., Baker, F., Borden, M., and J. Krawczyk: *A Framework for Integrated Services and RSVP over ATM*, RFC 2382, August 1998.

[23] V. Jacobson, K. Nichols, K. Poduri: *An Expedited Forwarding PHB*, RFC 2598, June 1999.

[24] The ATM Forum: *ATM Traffic Management Specification, Version 4.0*, April 1996.

[25] Sitaraman VR: *IP QoS Over ATM*, http://www.cis.ohio-state.edu/ jain/cis788-99/ip_qos_atm/.

[26] Rudiger Geib: *Differential Services for the Internet and ATM*, http://www.internet2.edu/qos/wg/papers/I2QoS-geib-difs-atm-02.html.

[27] *MPLS using LDP and ATM VC Switching*, April 1999, 19 Pages, http://info.internet.isi.edu:70/in-drafts/files/draft-ietf-mpls-atm-02.txt.

[28] G. Huston: *Next Steps for the IP QoS Architecture*, RFC 2990, November 2000.