End-to-End Internet Quality of Service (QoS): An Overview of Issues, Architectures and Frameworks

Rajiv Chakravorty, Subrat Kar, Peyman Farjami[1]

Abstract:- **With the advent of high-end multimedia applications, the need to provide better Quality of Service (QoS) from the Internet has gained significant importance. Internet of the future will be characterized by diverse traffic sources – heavily overburdened by real time traffic right from voice to video and increasingly overwhelmed by the traffic generated through millions of e-commerce transactions. Therefore, QoS requirements of all such applications will introduce several new consequences along the way in which data is transmitted over the Internet. In this paper, we will attempt to evaluate all the issues that govern end-to-end Internet QoS. Highlighting all-important factors will then derive a unified approach for proper QoS management in the Internet. The paper also delves into the existing problems and describes means to maximize solutions towards better QoS from the Internet.**

Key Words: - **QoS, IntServ, RSVP, DiffServ, IPv6, MPLS.**

1. INTRODUCTION

OVER the years, end-to-end QoS have been perceived to be more or less governed by conventional parameters like bandwidth, packet loss, end-to-end delay and delay jitter. Although this is true for the Internet QoS, the end-to-end dependence is somewhat more diverse, essentially distributed to a wider set that actually characterizes end-to-end QoS. A sound description of the complete QoS architecture is imperative to understand its dependencies at various levels, and then highlight upon issues that require further investigation. Consideration of all the factors would then derive a relevant approach towards better understanding of proper QoS management in the Internet. The guiding principle of this paper is to provide a simplified overview on end-to-end QoS, investigate Internet QoS related problems and highlight their probable solutions. In addition, this paper will also discuss some future directions.

The article is organized as follows: Section 2 describes the QoS in a wider perspective. Here, a brief description on Host Level QoS is followed by details on Network QoS. Section 3 provides description on the IP QoS models - IntServ and DiffServ. Internet traffic self-similar nature is explained in section 4 while benefits available from link-sharing scheme are discussed in section 5. Some possible mechanisms that will support QoS (IPv6, MPLS) in the future are presented in section 6. Section 7 elaborates on a framework that enables end-to-end QoS. The complete paper has been summarized in the last section.

2. QoS: A WIDER PERSPECTIVE

THE QoS model (Figure 1) captures the abstraction required to enclose dependencies related to QoS in the Internet. It provides a reasonable insight towards better QoS management from two very different levels of QoS. Towards a broader outlook, we term them here as *Network QoS* and the *Host QoS*. In other words, end-to-end QoS not only depends on QoS available from the Network, but also depends upon QoS contributions from the Host level as well. We will briefly discuss host level contribution over end-to-end QoS in the next subsection.

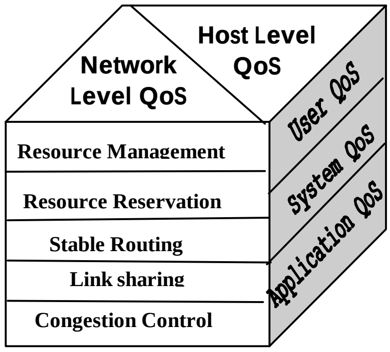

Figure 1: QoS Issues

In this paper, the emphasis will be clearly on network level QoS (or Internet QoS). The obvious reason to constrain the discussion on host level QoS is to validate the presence of increasingly available end terminal computing power. The discussion on Host QoS would have been more interesting some few years ago, but due to steadily increasing end-computing capacity over last couple of years, the contribution which host level QoS had on end-to-end QoS has been practically nullified. However, for devices those still or will likely have this disadvantage (future embedded network devices), a related discussion will be certainly meaningful. We justify this in the next subsection.

2.1 Host level QoS

This can be distinguished further into three levels of abstraction. i.e. user level, application level and system level.

User QoS is mainly concerned with the requirements of a user that varies from a corporate user to that of a normal user. As such, *pricing* is an important entity to the service provider for providing QoS. A user can request resources only if it's available. This

[1] Rajiv Chakravorty is currently with Philips Advanced Systems and Applications Labs, Eindhoven, The Netherlands Email: rajiv.chakravorty@philips.com
Dr. Subrat Kar is an Assistant Professor at the Indian Institute of Technology, New Delhi Email: subrat@ee.iitd.ernet.in
Peyman Farjami is a Phd student at the Chair of Communication Networks, Aachen, Germany Email: pemi@comnets.rwth-aachen.de

makes resource *availability* an important constituent of User QoS. Moreover, the preference level of an end user is subjective and it's not always trivial to figure out what factors actually build up user preferences. The supporting environment has an important role in such decisions. Therefore, all parameters that usually govern user QoS may not always be deterministic and it is left to the user to decide if the perceptual quality (e.g. CD quality) is good enough to meet the (user) requirements.

Application QoS forms an important constituent to the host level QoS. It can be realized more accurately from the QoS support available at the static and at the dynamic level. At the static level, support is in terms of static compression and decompression techniques applied to the application data at the end hosts. Dynamic level application support caters mostly to make the application robust by adapting it to the prevailing network conditions. Adaptive nature of these applications can be based either on delay adaptation and/or bandwidth adaptation. Mbone tool such as *vat* (audio conferencing tool) uses a delay adaptive algorithm to adjust the stability of the playback points in order to adapt the application. In the other case, algorithms like hierarchical encoding to variable compression schemes can provide dynamic adaptation, mostly based on bandwidth adaptation. Most of the techniques are a function of the feedback available from the network so as to make the applications adapt to the network conditions. However, architectures that involve multicasting are still not scalable to large numbers of receivers because of the *feedback implosion* problem (encoder with large amount of feedback from the receivers) at the sender end. As a remedy, the use of weights for each individual session was suggested, but its use still remains questionable in case of low bandwidth session employing large number of receivers.

System QoS defines the QoS that includes all sorts of components located in between the applications and the underlying network. These include the operating system (kernel), filters, device drivers, libraries (static and dynamic) and other middleware components. Therefore, these are also termed as *reusable components*. Experiences here suggest that all parameters relating to CPU (computation time, cycle time, utilization etc.) and memory (e.g. memory requests) affect the system level QoS. In resource constrained environments, design optimizations contribute significantly towards the improvement of QoS at this level.

2.2 Network Level QoS

The ***Network Level QoS*** can be abstracted more clearly into two parameters i.e. *Device level QoS* and *physical network QoS*. The *device level QoS* depends largely on system components (hardware and software) that run these devices. The performance parameters of these components are then abstracted in terms of *Physical network QoS*. Some examples of the parameters to quantify physical network QoS, which we are aware of, include cells/second in ATM, frames/sec in a video capture etc.

However, device QoS attributes cannot be completely dedicated to the hardware that runs the device. Much of it also depends upon the implementation efficiency of the software component that enables the underlying hardware. These software components include tasks that run within the micro-kernel. These tasks can be sometimes quite complex ranging from scheduling, filtering to classification etc. mostly used to support the functionality of the device that runs the underlying hardware. Some important issues that influence network QoS have been considered.

2.2.1 Resource Management

Resource (bandwidth) management allows to successfully share resources across several contending applications. This is important for general purpose distributed environments where resources need to be shared across multiple applications. There are several techniques to achieve proper resource management. Two such techniques will be briefly discussed.

2.2.1.1 Using Agent Technology

The main objective here is to assess the practical implications and concepts of mobile agent technology for use in resource management. This includes tasks right from dynamic management of VPN's (virtual private networks) to all solutions related to network management software. The only target in all such cases is to use what is available and not always to provide an enhanced agent platform.

2.2.1.2 Active network

The concept emerged because the lead user applications could perform user-driven computations at the nodes within the network[4][5]. This has been possible due to the emergence of the mobile code technology that makes dynamic network service invocation attainable. Networks are active in the sense that routers and switches can perform customized operations on a per-user or per-application basis. This idea of carrying procedures and data is seen as a step beyond the conventional switching or routing, which can in turn rapidly adapt the network to the changing requirements.

2.2.2 Resource Reservation

Reservation is one of the efforts to characterize QoS at the network level. Once resources are reserved, the application is assured of a minimum amount of acceptable QoS. A protocol that enables reservation in the Internet is RSVP (more description in section [3.1]).

2.2.3 Routing Stability

Many types of instabilities related to routing exist in the Internet. Route fluttering, routing asymmetry, routing loops are some of the common problems in the Internet. Consequently, this results into considerable degradation of the available QoS, reducing the overall QoS available to the end user.

Constraint based routing can offer some panacea to the problem discussed above. In addition to selecting routes that meet QoS requirements, it also provides increased network resource utilization. However, it can also lead to increased communication and computation overhead, larger routing table size and may sometimes result into potential instability.

2.2.4 Congestion Avoidance

It amounts to network support by employing effective techniques to avoid congestion within a given network. The best way to control congestion is to avoid it. One such technique to avoid congestion in gateways is *RED* (Random Early Detection)[9]. It avoids congestion by controlling the average queue size of the gateway buffers. It avoids TCP global synchronization problem and maintains its steady bias against burst traffic. The gateway calculates an average queue size that is compared to two thresholds, one denoting the minimum queue size while another denoting the maximum size. Below the minimum limit, no packet is marked while every packet is marked above the maximum threshold. In between the two, the packets are marked with a probability that remains a function of the average queue size. The packets are then treated according to their marked probability. Such type of control can effectively avoid congestion and thereby improve QoS in the network.

3. INTERNET QoS ARCHITECTURES

TWO different architectures with its applicability to two different groups in IETF (Internet Engineering Task Force) - IntServ (Integrated Services) and DiffServ (Differentiated Services) are discussed.

3.1 IntServ IP QoS Model

IntServ architecture [1] uses Resource Reservation Protocol (RSVP) as its working protocol to achieve its end-to-end signaling. In RSVP, resources are reserved at each router along the path between sender and receiver through explicit signaling for a flow that demands QoS. The benefit of end-to-end per-hop signaling is its ability to convert a connection-less Internet into a more reliable connection-oriented network. A router implementing RSVP would look functionally more complex than an ordinary router. It would have a daemon that would remember each session state with a session identifier that would map uniquely to a <source-destination address, protocol> pair. It would also have complex admission control for each session and equally complicated scheduling strategy that would implement typical variants of fair scheduling schemes for sessions demanding guaranteed QoS. However, based on the scheduling schemes incorporated within a router, IntServ formulates three types of service class:

1. *Guaranteed Quality Service*: Applications with rigid end-end delay bounds use this type of service.
2. *Controlled Load Service*: For applications with looser performance criterion than Guaranteed Quality Services but requiring higher attributes than normal best effort services.
3. *Best Effort Service*: Services provided by the Internet as of today.

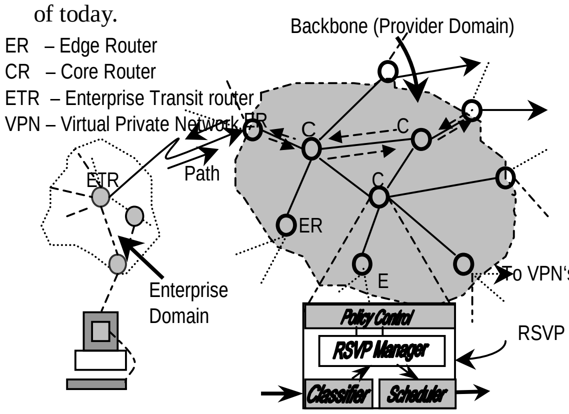

Figure 2: Integrated Services (IntServ) IP-QoS Model

RSVP has advantages that include its soft-state adaptive nature, flexibility of receiver initiating reservation and possibility to merge reservation requests. Indeed, benefits with IntServ are available at the expense of certain shortcomings inherent in this model. Internet is rather heterogeneous with majority of the end-user networks being constituted by Ethernet. Ethernet cannot inherently provide any temporal performance guarantees. It does not support deterministic network access delay because of its contention-based access to the physical medium. Therefore, use of RSVP with Ethernet makes it contrary to the preferential resource requirement of real time traffic. Another major drawback with IntServ is its scalability. A router implementing RSVP cannot scale with large number of sessions. Therefore, its applicability is always questioned in domains that have connections in order of millions. Even if that were possible, maintaining this number of sessions through explicit signaling would have its own associated network overhead. Therefore, the use of RSVP has been practically ruled out for the backbones that carry enormous traffic.

3.2 DiffServ IP QoS Model

The main problem that makes IntServ infeasible is scalability. Maintaining thousands or millions of connections in an IntServ router is impractical. With large number of sessions existing in the backbone, the amount of computation complexity involved by using RSVP will simply overburden the router.

Towards a solution, a new working group called DiffServ was formed to look into this issue. The group proposed a model that completely eliminates storage of session states in the router. Consequently, the model became quite popular, as it was computationally inexpensive to a router even in the backbone. In DiffServ, implementation of the concepts of aggregation of flows and per-hop behavior is applied to a network-wide set of traffic classes. Flows are classified according to pre-determined rules, so that many application flows can be aggregated to a limited set of class flows.

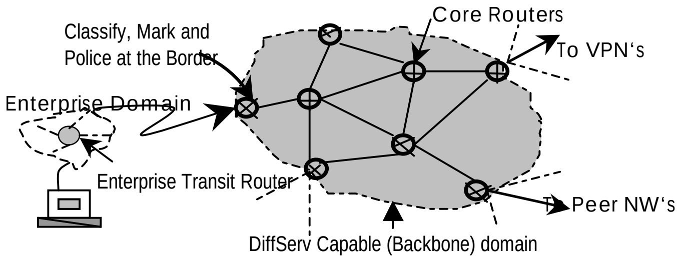

Figure 3: Differentiated Services (DiffServ) IP-QoS Model

Traffic entering the network domain at the border router is first classified for consistent treatment at each transit or core router [Figure 3] inside the network. The classification is based on IPv4 (or IPv6) Type of Service (TOS) Header field also called as DS (differentiated services) field. Currently two bits are engaged in for standardization while rest six reserved for use in the future. The representation of the code points for DS field defines the standard code point. Each of the core routers within the DiffServ domain will read the DS field of the packets and treatment will usually be applied by separating the traffic into queues according to the class of traffic. This type of framework is more scalable than IntServ mainly due to a stateless design, flow aggregation and minimization of signaling requirements. In all cases, it is also suitable for Enterprise domains (Intranets) provided all hosts and legacy routers mark the DS field according to the policies of the domain. However, even with these advantages, DiffServ model is still devoid of qualitative QoS guarantees. Unlike IntServ, once a packet enters into a DiffServ domain, its behavior is unpredictable because of the absence of explicit connection oriented end-end per-hop signaling. All the routers in DiffServ are expected to trustfully mark the packets and provide them the required level of treatment. Therefore in such cases, it is completely left to the network operator to provide the best possible services.

Defining service level agreements (SLA) between network providers implementing such services are always considered a serious challenge. SLA requirements between two different domains implementing DiffServ would necessarily involve thousands of rules that govern their inter-working. These issues become more complicated if the network providers have different number of service classes. Effective QoS *Mapping* between the service classes of one network provider with the other would require many considerations beforehand. As such, these issues regarding DiffServ are still under investigation within the aegis of IETF. However to reiterate, the focus of DiffServ has been entirely on how to support QoS in IP.

3.3 Inter-working IP QoS Models

In clear technical viewpoints, IntServ and DiffServ are complementary tools in the pursuit of end-end QoS. Each has an important role in a QoS enabled network. A manner in which both the models can be simultaneously used has been identified[3].

Here, the IntServ model is limited to the enterprise domain that is usually a small network compared to the backbone, which is a DiffServ domain [Figure 4]. The direct implication of it results into considerable reduction of maintaining session states within an IntServ router. The mapping of the IntServ service type to a PHB (per-hop behavior) of DiffServ should be necessarily justified by considering the service specific type requirements of IntServ to the appropriate PHB of DiffServ to invoke a similar service. All SLA's essentially needs to be taken into account for such a mapping.

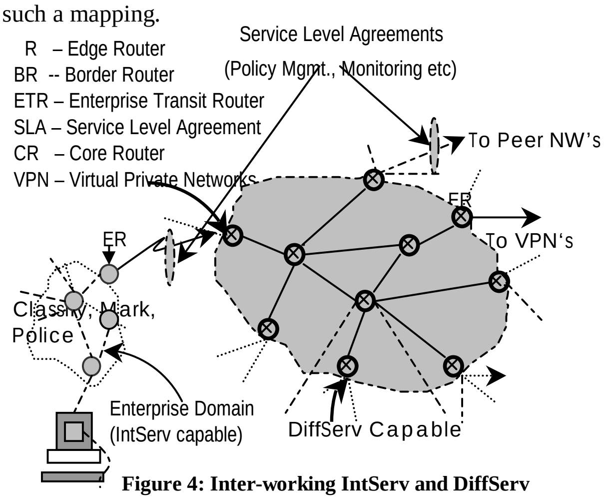

Figure 4: Inter-working IntServ and DiffServ

One of the main aspects of establishing end-end QoS is to ensure need-based availability of both quantitative as well as qualitative guarantees. While IntServ can provide both, DiffServ can only source qualitative guarantees. In addition, one of the main differences between both models is admission control. In DiffServ, admission is provided implicitly while in IntServ the admission control is explicit (through control messages). The former therefore breaks down the end-to-end support for QoS since neither the application traffic nor the intermediate nodes take any corrective action in case of admission failure occurring within the DiffServ domain. In case that happens, information about the failure should be explicitly communicated to the RSVP capable IntServ nodes accessible from this domain. SLA's if statically provisioned, admission control functionality is provided by static configuration. Otherwise dynamically variable SLA's can be propagated to inform about resource availability within the DiffServ domain. This can be explicitly communicated as an admission control information to the IntServ networks.

4. SELF SIMILAR INTERNET TRAFFIC

AN event that displays long term co-relations for packet arrivals across a wide range of time scales in an aggregated Internet traffic is a self-similar process. Usually burst traffic can be described using the notion of self-similarity. Such long-range dependence can also lead to drastic reductions in the effectiveness of deploying *buffers* in the Internet routers to absorb transient increase in the traffic load.

4.1 Effect on QoS

Studies have revealed that queue length distribution decayed more slowly for long-range dependent sources than for short-range dependent source[6]. The implication of such self-similar behavior of Internet traffic results in increased sizes of buffers at switches and multiplexes than those predicted by analysis and simulations. Once the buffer size is increased, it leads to higher queuing delays than originally anticipated. This in turn has obvious effects on the quality of multimedia applications requiring higher bandwidth and lower latency, such as video conferencing and other distributed real-time applications. Self-similar behavior of the Internet has therefore added further complexities to the problem of optimizing performance in the Internet.

4.2 Re-engineering the Internet

Self-similar traffic has highlighted the importance of efficient Tele-traffic modeling for the Internet. Detailed investigation in this direction will provide a reasonable approach to redesigning and re-engineering of the network devices that can be used even in the presence of self-similar nature of the Internet.

5. LINK SHARING

LINK sharing refer to sharing of bandwidth into portions that are then used by different *entities*. Entities here can be referred to in aspects of sharing bandwidth. Bandwidth can be shared among agencies, different protocol suites (SNA, IP, IPX etc.), different classes of traffic or for that matter types of protocol (Ftp, Telnet etc.) within a given protocol suite[10]. This can also be looked into hierarchically. For this reason it is also called as *hierarchical link sharing.* Several rules and considerations are followed to provide this sharing. Studies are ongoing in the direction to involve link-sharing issues within the framework of reservation. However, this can cause problem only in one aspect of reservation i.e. Admission Control. If it is assumed that most of the future traffic will be variable bit rate (VBR) traffic, use of a statistical admission control can provide a high degree of bandwidth utilization. However the use of such link sharing techniques clearly violates the use of a statistical admission control based algorithms, as it lowers the degree of statistical gains obtained (see [11], pg. 95-96). This can however be overcome by allowing the scheduler or traffic shaper to ignore the changes in the link bandwidth provided by link sharing. A study on these issues will be definitely a matter of interest in the future.

6. TOWARDS BETTER INTERNET QoS

THERE are several steps taken by the Internet community in support of QoS. We elaborate on some of the areas where the ongoing work will have an implication on the developmental work relevant to QoS in the future.

6.1 IPv6

IPv6 is an enhanced version of IPv4 protocol specification. Incorporating QoS is not the only reason for transiting from IPv4 to IPv6, there are other reasons as well. Security considerations, mobility issues and increased address usage in the network domain, were some of issues considered while formulating IPv6[2].

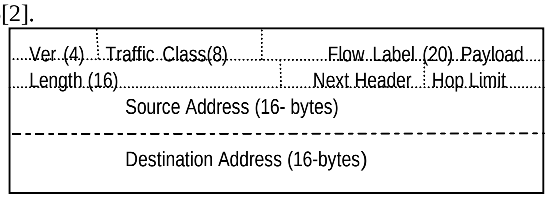

Figure 5: Internet Protocol (version 6) header [RFC-2460]

Essentially it consists of two QoS related fields [RFC 2460] in its header [Figure 5]. First component is 8-bit *Traffic Class field* while the second one is a 20-bit *Flow Label*. The purpose of Traffic Class field is to identify and distinguish between different types of classes or priorities of IPv6 packets. The Class field is intended to allow some sort of differentiated services similar to that of type of service (TOS) field of IPv4. Much of it will depend upon the way TOS field undergoes eventual standardization. The flow label is a 20-bit field that is used to characterize flows accordingly. More specifically it is meant for special handling of the flows (for e.g. in case of RSVP to differentiate between default service and other real-time services). *Flow Label* is assigned to a flow by the flow's source node or the node originator. Hosts or routers that do not support the functions should set the field to zero.

6.2 Advance Reservations

Reservations are needed because of the scarcity of network resources (bandwidth). If resources were plentiful, one need not make any immediate reservation at all. Therefore, some issues concerning advance reservation are considered here. In advance reservations, duration of the reservations from the starting time to the finishing time must be included. Renegotiations can be possible if a session requires a shorter or longer duration. Admission control techniques are used, which work solely on the basis of currently active flows as well as requests to be available in the future. Bandwidth is provided for normal data traffic in addition to sessions making immediate reservation requests. States are maintained for all the current and future requests. The number of states established (in the router) will be proportional to the total number of current and future requests. The domain has been investigated and two such techniques – using RSVP messages[8] and agents[7] can be used for advanced reservations.

6.3 Multi-Protocol Label Switching (MPLS)

MPLS is a non-destination based forwarding strategy used to forward IP packets. Forwarding is done by a Label Switching Router (LSR) that performs a match on a label associated with a packet in an MPLS domain [Figure 6]. This switching stands in between layer-2 and layer-3 of the OSI layer. For this it needs a protocol to distribute label switched paths (LSP's) which in turn can be used as tunnels. Usually Label distribution Protocol (LDP) is used for the purpose. A LSP once set-up, path of a packet can be known by label assigned by the ingress LSR.

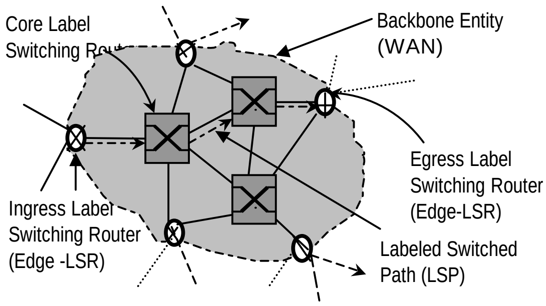

Figure 6: MPLS Domain

Packets are classified at the ingress with an MPLS header attached. The header has 20 bit label along with 3 bit Class of Service (COS) field, 1 bit stack indicator and 8-bit time-to-live (TTL) field. The label acts as an index to search through the forwarding table. After processing of the packet, the incoming label is changed to an outgoing label and the packet is then switched to the next LSR. In this way, a number of flows can be aggregated into a traffic trunk (corresponding to a LSP) of a given service class. Advantages of MPLS include:

1. Faster packet classification and forwarding.
2. Efficient tunneling mechanism.

QoS in such a scheme depends upon the labels and Class of Service (COS) fields in an MPLS header. It provides a simple mechanism by which number of flows can be aggregated into a single service class. It is this property that enables MPLS to work coherently with differentiated services (DiffServ) IP QoS model. In future, we can expect DiffServ to play a dominant role in providing QoS for WAN's, often in conjunction with MPLS.

7. FRAMEWORK FOR END-END QoS

THIS section details the framework that can enable end-to-end Internet QoS. Figure 7 shows a content server that streams real time data to a remote player. The framework enables end-to-end QoS for streaming media over the Internet. Here, the media data stored in some universal file format (e.g. Microsoft ASF, QuickTime etc.) is retrieved from the content server and directly streamed over the Internet.

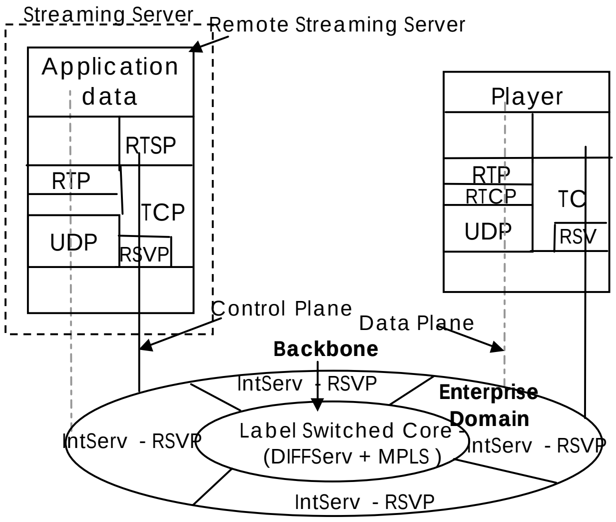

Figure 7: End-to-End Internet QoS

The protocol used for controlling, accessing and retrieving media data from the content server is provided through an application level protocol like RTSP (Real Time Streaming Protocol). Using available controls, commands can be activated by the player to control the display of streams that are directly available from the server. The user can play, stop, pause, or record the stream. This also depends to the extent access restrictions exercised by the content server for a particular stream. For each of the streams, there can be a separate network connection used. Or multiple streams can be logically multiplexed over a given network channel. Data is delivered using simple RTP (Real Time Protocol) over UDP. The use of RTP is to provide sequencing and timing information for the data, but this cannot ensure in-time data delivery by itself. Sequencing only helps to detect packet loss while timestamp is used during presentation of media data. Moreover, control protocols like RTCP (Real Time Control Protocol) can be used to provide QoS reports either to the sender or to the receiver. This information can be about packet loss, inter-arrival jitter etc. Using this report, the sender may shift to a lower bit rate encoding in case there is excess packet loss. RTCP is a duplex protocol; both sender and receiver can exchange control information. However, RTCP generates too many control messages that can act as an extra overhead to the network. This problem can be solved to some extent by decreasing the rate of transmission of these control messages with increase in the number of receivers.

Figure 7 shows that a small enterprise domain can implement IntServ. In such a case, QoS support for the application at the network level will be provided by RSVP. Depending upon the application requirements, the required QoS service class will be invoked from the ISP providing the service. This session can then get mapped appropriately to the edge of the backbone implementing DiffServ. There may be one or more such QoS mappings for a given session before the application data is finally available to the receiver.

8. SUMMARY

AN approach towards achieving better end-to-end QoS from the Internet has been discussed. This approach provides a complete control of QoS parameters; emphasizing on end host capability for additional support to end-to-end QoS.

At the host-level, we stressed the importance of adaptive applications that have a higher contribution towards achieving improved QoS in the Internet. In addition, we discussed the existing IP QoS models and more importantly the existence of both the model i.e. IntServ and DiffServ for their desired inter-working. In IntServ, we described how resource reservation protocol (RSVP) is used for reservations. We also elaborated at the network level to issues concerning resource management, congestion avoidance, routing stability etc. that are vital for assuring QoS support from the Internet. Our view about self-similar nature of Internet traffic is largely based on to review the design constraints that should be kept in mind while designing networks (and supporting network components) of the future. We investigated issues like IPv6, MPLS etc. that will most likely enable support for Internet QoS in the future. Finally, we discussed a framework to enable end-to-end Internet QoS for real-time multimedia applications. This framework uses all existing Internet standards and protocols, whcih can be successfully coupled to provide better end-to-end Internet QoS.

References

[1] R.Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin “Resource Reservation Protocol RSVP -- Version 1 Functional Specification“, Internet Engineering Task Force RFC2205, Category: Standards Track September, 1997

[2] S. Deering, R. Hinden “Internet Protocol, Version 6 (IPv6) Specification“, Internet Engineering Task Force RFC 2460, Category: Standards Track December, 1998 (Related: RFC 2374/5, 2461/2)

[3] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, “Integrated Services Operation Over Diffserv Networks ” Internet Draft, draft-ietf-issll-diffserv-rsvp-02.txt, June, 1999.

[4] Ulana Legedza, David J. Wetherall, and John Guttag “Improving The Performance of Distributed Applications Using Active Networks” IEEE INFOCOM, San Francisco, 1998

[5] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden “A Survey of Active Network Research”, IEEE Communications Magazine, Vol. 35, No. 1, pp80-86. January 1997

[6] S.Zafer and T. Sirin “On Multimedia Networks: Self-Similar traffic and Network Performance“, IEEE Communications Magazine, pp. 48-52, January 1999.

[7] O. Schelen and S. Pink “An Agent-based architecture for advance reservation”, Proceedings of the IEEE Conference on Computer Networks Minneapolis, Minnesota, and November 1997.

[8] M. Degermark, T. Kohler, S. Pink and O. Schelen “Advance Reservations for Predictive Services in the Internet” Proc. 5th Intl. Workshop on Network and Operating System Support for Digital Audio and Video (NOSDAV), Durham, New Hampshire, April 18-21, 1995, pp. 3-14.

[9] Sally Floyd and Van Jacobson, “Random Early Detection gateways for Congestion Avoidance” IEEE/ACM Transactions on Networking, V.1 N.4, August 1993,pp. 397-413.

[10] Floyd, S., and Jacobson, V “Link-sharing and Resource Management Models for Packet Networks” IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp. 365-386, August 1995

[11] Sugih Jamin “A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks“, Phd Desertation, University of Southern California, SC Tech Report, USC-CS-96-639, 1996.