APPROACHES FOR PROVIDING QUALITY OF SERVICE IN MULTIMEDIA NETWORKS

Dan Mancas[1], Adrian Neatu[1], Dan Tusaliu[1], Adina Basandica[2]

[1]*University of Craiova,* [2]*High School "G. Bibescu" Craiova*

Abstract: Quality of Service (QoS) is a fast growing area of technology, being driven by the growth of multimedia applications such as videoconference or voice over IP. In the first section of this paper there are presented different approaches for providing Quality of Services in networks with multimedia traffic. In the second section there are presented the results of the QoS performance evaluations performed on a test network based on Cisco routers.

Keyword: Quality of Service, multimedia traffic, active queue management, fair queuing, resource reservation.

1. APPROACHES ON QoS

It is commonly accepted, that the Quality of Service (QoS) concept is of central importance for distributed multimedia systems. From the communications point of view, the main concern is how to support QoS requirements by networks, protocols, and operating systems. However, with respect to QoS in multimedia systems it is necessary to consider more than communication system and operating system, because most multimedia systems do not only manipulate and transmit multimedia data. Additionally, they store and retrieve multimedia data, and present them in a proper way to the user. However, the relevance of database management system aspects for QoS in multimedia systems has not been broadly recognized.

Traditional computer architecture research has focused primarily on the design and performance of individual machines. With the increasing importance of network technology and multimedia interfaces, future architectures will span machine boundaries and will need to provide adequate support for time-critical operations.

Video Conferencing, Video-on-Demand and IP telephony are few successful commercial networked applications. Those that have timeliness constraints are called real-time applications. Among real-time applications there are those that are tolerant or intolerant depending on whether they can tolerate occasional loss. Such real-time applications are competing with traditional Internet applications – email, file transfer for network level resources such as bandwidth and queue buffers. They demand high bandwidth and assurance of timeliness of data delivery from the underlying network

The current Internet architecture, offers a simple point-to-point best-effort service. Multimedia applications (e.g., remote video-on-demand, multimedia conferencing etc.) require different type of services and need application specific Quality of Service (QoS) to perform to the required standards. Thus, Quality of Service (QoS) gains importance in the event of multimedia and real time applications. Many of multimedia applications are sensitive to the QoS rendered to it by the underlying network architecture and their performance degrades rapidly in the event of a failure to provide the required QoS. For a network to deliver appropriate quality of service, it must go beyond the best-effort service model and to allow multimedia traffic to reserve network resources and to provide preferential routing.

Today's computer users are becoming more and more sophisticated, demanding richer machine interfaces. This is evidenced by the fact that "multimedia" has become a trendy buzzword used to hype seemingly every new computer product to hit the marketplace. In response many researchers are investigating the best means to make desktop audio and video a reality, and this area is beginning to mature. The MPEG I, II, III and IV coding standards now enjoy widespread acceptance, and numerous vendors market hardware implementations. There are still unresolved issues, e.g. synchronization, quality of service (QoS) guarantees, transport and storage mechanisms, etc.

However, viewing and manipulating a single stream of full-size video along with its associated audio stream is becoming usual. This provides sufficient fluency for numerous applications, but many others will require a richer environment. Consequently, work has been done to investigate means for supporting high definition television (HDTV), as well as the use of two video streams to create three-dimensional television (3DTV). A natural extension is the need for numerous streams of audio and video. Multiple media streams will become a necessity to meet the increasing demands of future applications. An example which requires multiple audio and video streams is an application that supports multi-viewpoint audio and video. Stereo (3D) audio and video offer only a single perspective from which a scene may be viewed, but multi-viewpoint audio and video allow a user to observe a scene from many different perspectives so that a sense of immersion is experienced. Multi-viewpoint audio and video are composed of several different streams and thus represent a significant technical challenge over current desktop audio and video.

Quality of Service (QoS) can be broadly defined as the degree of user satisfaction. Network QoS refers to the ability of the network to handle traffic such that it meets the service needs of certain applications. This requires fundamental traffic handling mechanisms in the network, the ability to identify traffic that is entitled to these mechanisms and the ability to control these mechanisms. Any QoS assurance is only as good as the weakest link in the “chain” between sender and receiver. So QoS is fundamentally an end-to-end issue implying that QoS assurances have to be configurable, predictable and maintainable from source to destination. This means that it should be relevant over all architectural layers from source media devices down the protocol stack across the network element and up the receiver protocol stack to playback devices. Consequently, the issue of QoS can be addressed at different levels of the network protocol stack including:

- User level by specifying (qualitatively or quantitatively) user perceivable service parameters.
- Application level by ensuring that the application adapts according to the network and system resource availability.
- Network level by defining traffic models, classification of service disciplines, and resource reservation on a per-flow or flow aggregate basis to ensure that the applications’ resource requirements are met.

Some considerable research efforts directed in handling QoS at the network level are presented below.

Many commercial switches and routers today employ output-queuing. When a packet arrives at an output-queued (OQ) switch, it is immediately placed in a queue that is dedicated to its outgoing line, where it will wait until departing from the switch. This approach is known to maximize the throughput of the switch: so long as no input or output is oversubscribed, the switch is able to support the traffic and the occupancies of queues remain bounded.

Perhaps more importantly, the use of a separate queue for each output means that flows of packets for different outputs are kept separate, and cannot interfere with each other. By carefully scheduling the time that a packet is placed onto the outgoing line, a switch or router can control the packet’s latency, and hence provide quality-of-service (QoS) guarantees.

But output queuing is impractical for switches with high line rates, or with a large number of ports: the fabric and memory of an N x N switch must run N times as fast as the line rate. Unfortunately, at the highest line rates, memories with sufficient bandwidth are not available. For example, consider a 32x32 OQ switch operating at a line rate of 10 Gbps. If we use a 512-bit memory data path, we require memory devices that can perform both a write and a read operation every 1,6 ns.

On the other hand, the fabric and the memory of an input queued (IQ) switch need only run as fast as the line rate. This makes input queuing very appealing for switches with fast line rates, or with a large number of ports. That is, for a given speed of memory it is possible to build a faster switch; or for a given speed switch it is possible to use slower, lower-cost memory devices. For example, consider again the 32x32 switch operating at a line rate of 10Gbit/s. If the switch uses input-queuing instead of output-queuing, we can use memory devices that perform a write and a read operation every 51,2 ns. This is readily achievable with commercially available memories. For this reason, the highest performance switches and routers use input-queued crossbar switches (Partridge; McKeown *et al.*, 1996).

But IQ switches can suffer from head-of-line (HOL) blocking, which can have a severe effect on throughput. It is well-known that if each input maintains a single FIFO, then HOL blocking can limit the throughput to just 58,6% (Karol *et al.*, 1987).

One method that has been proposed to reduce HOL blocking is to increase the “speedup’’ of a switch. A switch with a speedup of S can remove up to S packets from each input and deliver up to S packets to each output within a time slot, where a time slot is the time between packet arrivals at input ports. Hence, an OQ switch has a speedup of N while an IQ switch has a speedup of one. For values of S between 1 and N packets need to be buffered at the inputs before switching as well as at the outputs after switching. We call this architecture a combined input and output queued (CIOQ) switch.

Both analytical and simulation studies of a CIOQ switch which maintains a single FIFO at each input have been conducted for various values of speedup (Iliadis and Denzel, 1990; Gupta and Georganas, 1991; Oie *et al.*, 1989; Chen and Stern, 1991). A common conclusion of these studies is that with S = 4 or 5 one can achieve about 99% throughput when arrivals are independent and identically distributed at each input, and the distribution of packet destinations is uniform across the outputs.

But it has been shown that a throughput of 100% can be achieved with a speedup of just one, if we arrange the input queues differently. That is, HOL blocking can be eliminated entirely using a scheme known as *virtual output queuing* in which each input maintains a separate queue for each output. It has been shown that for independent arrivals, the throughput of an IQ switch can be increased to 100% (McKeown *et al.*, 1996). We may draw the conclusion: Speedup is not necessary to eliminate the effect of HOL blocking.

In practice, the users are not only interested in the throughput of a switch, but also in the latency of individual packets. This is particularly important if a switch or router is to offer QoS guarantees. Packets in an IQ switch not only contend for an output, they also contend for entry into the switch fabric with packets that are destined for other outputs. We call this phenomenon *input contention*. Each input can deliver only one packet into the fabric at a time; if it has packets for several free outputs, it must choose just one packet to deliver, holding other packets back. This places a packet at the mercy of other packets destined for other outputs. This is in stark contrast with output-queuing, where a packet is unaffected by packets destined for other outputs. It can be drawn the conclusion: to control the delay, a mechanism to eliminate input contention is needed.

While the scheduling schemes can provide a fair bandwidth allocation with complicated structures, the queue management mechanisms, although simple and easy to implement, usually fail to provide fair service. The queue management schemes, such as Drop Tail and RED, are designed to control the queue length by dropping packets when necessary. It is well known that the widely deployed Drop Tail method can cause the “Lock Out” and “Full Queue” problems (Braden *et al.*, 1998). RED, an active queue management algorithm, was proposed to solve these issues (Floyd and Jacobson, 1993). By keeping the average queue size small, RED avoids the bias against burst traffic and reduces the delays experienced by most flows. RED drops the arrival packets randomly based on the average queue size. In this way, it avoids the problem of global synchronization, where many flows reduce their window size at the same time. However, like Drop Tail, RED doesn’t penalize unresponsive traffics. A flow's fraction of the aggregate packet drops roughly equals to its fraction of the aggregate arrival rate. In other words, the percentage of packets dropping for each flow over a period of time is almost the same. Consequently, misbehaving traffics could take up a large percentage of the link bandwidth and starve out the TCP friendly flows.

To improve RED's capability of handling unresponsive users, a few methods have been proposed, for instance, Flow Random Early Detection (FRED) (Lin and Morris, 1997) and RED with penalty box (Floyd and Fall, 1997). Even though these two router mechanisms do not require per-flow states, they all need extra data structures to collect certain types of state information. FRED keeps per-active-connection states and RED with penalty box stores information about unfriendly flows.

All the schemes discussed so far drop packets without examining them. A new RED mechanism called Stabilized RED (SRED) suggests an idea of comparing the arriving packet with a randomly chosen packet that recently preceded it into the buffer (Ott *et al.*, 1999). The scheme maintains a data structure, "Zombie List", which keeps the information regarding recently seen flows. The SRED router estimates the number of active flows based on “Zombie List”. When a packet arrives at the SRED router, it is compared against a randomly chosen packet from the “Zombie List”. The arriving packet is dropped randomly based on the packet comparison result and the estimation of the number of active flows. The dropping probability increases if the active flow estimation is bigger or packet comparison ends in matching. In this way, SRED penalizes the unfriendly flows and controls the buffer length independent of the number of active flows. However, this scheme needs a large “Zombie List” data structure.

From the discussions above, we can see that there are generally two trends in search of solutions to the problem of unfriendly flows. One trend is based on the almost perfect fair queuing. In this group, router algorithms are proposed to reduce FQ’s design complexity while keeping FQ’s good feature of max-min fair bandwidth allocation. The other trend is based on RED. Several methods are suggested to improve RED’s capability of handling misbehaving users. However, all the schemes discussed so far fail to provide fairness with a minimum overhead.

2. PERFORMANCE EVALUATION TESTS ON QoS

All QoS performance evaluation tests were performed in a Cisco based network environment, shown in figure 1.

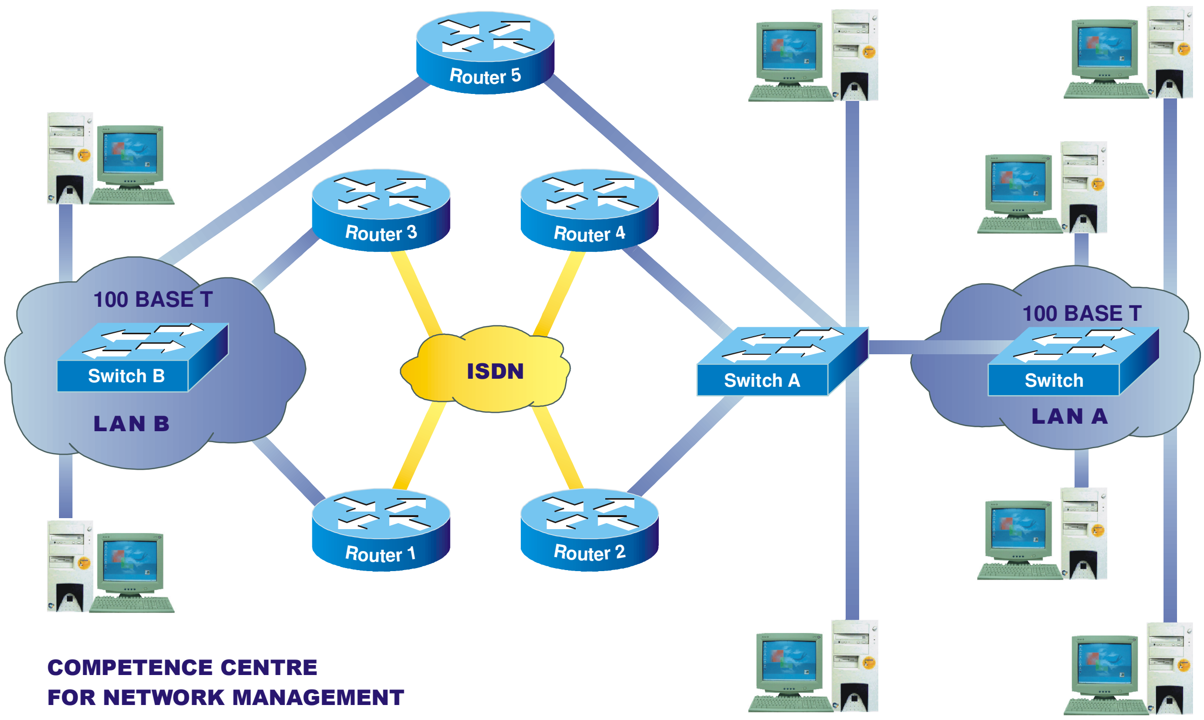

Fig 1. Test network

Router 1 to router 4 are Cisco 2610, router 5 is a Cisco 2611, and switches A and B are Cisco Catalyst 2900 XL. QoS was configured on each router in different scenarios. The multimedia traffic was generated using real time applications such as videoconference, as well as a traffic generator developed within our competence centre. The measurements were made using a sniffer machine. The results of the performance evaluation for Custom Queuing and Weighted Fair Queuing are presented below.

2.1 Custom Queuing

Tabel 1.

Port addr.	transmitted traffic rate A Kbps		transmitted traffic rate B Kbps		transmitted traffic rate C Kbps	
	Tx	Rx	Tx	Rx	Tx	Rx
20	30	30	35	35	60	45
80	10	10	20	14	50	7
25	5	5	15	6	45	3

Router configuration command:
Global configuration mode:
queue-list 1 protocol UDP 1 port 20
queue-list 1 protocol UDP 2 port 80
queue-list 1 protocol UDP 3 port 25
queue-list 1 default 10
queue-list 1 queue 1 limit 100 byte-count 70*packet_length
queue-list 1 queue 2 limit 100 byte-count 20*packet_length
queue-list 1 queue 2 limit 100 byte-count 10*packet_length

ISDN Interface configuration mode:
custom-queue-list 1

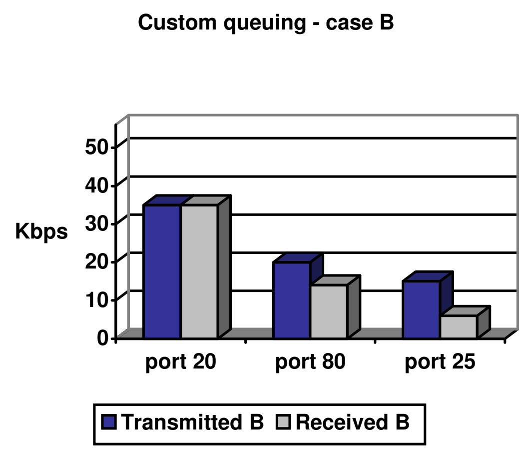

Fig. 2. CQ – Protocol Distribution B

The queuing is „weighted fair“. The ftp flow passes with the highest priority. Although, email will not longer starve for bandwidth, the email will work (slower than http because has lower queuing priority). For each flow, a maximum bandwidth is defined. But no bandwidth remains unused.

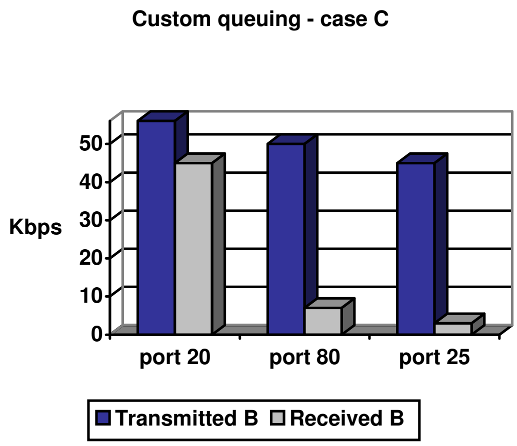

Fig. 3. CQ – Protocol Distribution C

The ftp port has the highest priority, although the bandwidth it uses can nod be greater than the maximum imposed by the CQ parameters (byte count) configuration. The rest of the bandwidth is distributed "fair" to the other flows in such a way, that no one will starv for bandwidth.

2.2 Weighted Fair Queuing

Table 2.

Ip prec.	transmitted traffic rate A Kbps		transmitted traffic rate B Kbps		transmitted traffic rate C Kbps	
	Tx	Rx	Tx	Rx	Tx	Rx
7	30	30	40	37	60	33
3	10	10	13	13	50	17
0	5	5	15	4	45	4

Router configuration command:

ISDN Interface configuration mode:

fair-queue

WFQ is QoS signaling aware.

8 + 4 + 1 = 13.

The bandwidth is ~56Kbps. Therefore, the reserved bandwidths are:

56 x 8/13 = ~34;

56 x 4/13 = ~17;

56 x 1/13 = ~4;

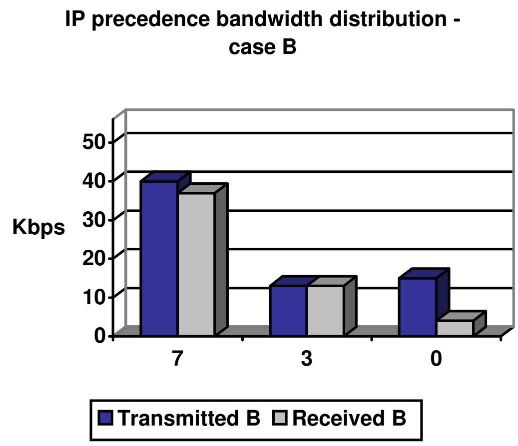

Fig. 4. WFQ - Distribution B.

The second flow (IP priority = 3) does not use the entire bandwidth that was reserved for him, so this remaining bandwidth is used by the first flow (IP priority = 7). This was an automated WFQ decision.

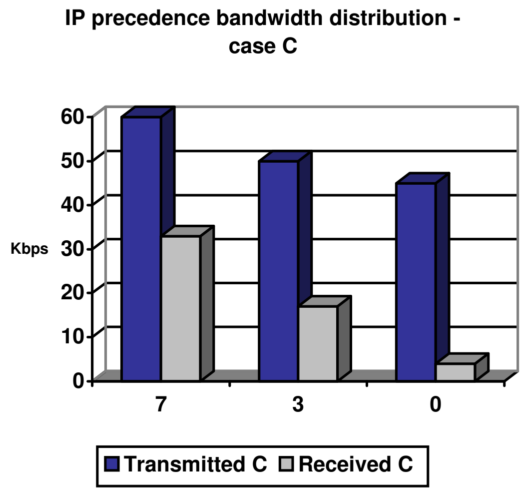

Fig. 5. WFQ - Distribution C.

No flow can use more than the bandwidth that was reserved for it at the interface configuration time. So, no flow will starv, if the device was properly configured.

REFERENCES

Braden, B. etc, (1998) Recommendations on queue management and congestion avoidance in the Internet. *IETF RFC* (Informational) 2309.

Chen, J.S.-C. and Stern, T.E.(1991). Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch. *IEEE J. Select.* Areas Communications, **vol. 9**, no. 3, p. 439-49.

Floyd, S. and Jacobson, V. (1993). Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transaction on Networking*, 1(4), pp 397-413.

Floyd, S., and Fall, K., (1997) Router Mechanisms to Support End-to-End Congestion Control. *LBL Technical report*.

Partridge, C., et al. A fifty gigabit per second IP router. *IEEE/ACM Transactions on Networking*.

Gupta, A.L. and Georganas N.D.(1991) Analysis of a packet switch with input and output buffers and speed constraints. *Proc. InfoCom '91*, Bal Harbour, FL, p.694-700.

Iliadis I. and Denzel W.E. (1990) Performance of packet switches with input and output queuing. *Proc. ICC '90*, Atlanta, GA, p.747-53.

Karol M., Hluchyj M., Morgan S. (1987). Input versus output queuing on a space-division switch. *IEEE Transactions on Communications*, **vol. 35**, pp. 1347-1356.

Lin, D. and Morris, R. (1997). Dynamics of random early detection. *Proceedings of ACM SIGCOMM'97*, pp 127-13.

McKeown, N., Izzard, M., Mekkittikul, A., Ellersick, W., and Horowitz, M. (1996). The Tiny Tera: A Packet Switch Core. *Hot Interconnects V, Stanford University.*

McKeown, N., Anantharam, V., Walrand, J. (1996) Achieving 100% Throughput in an input-queued switch. *Infocom '96*.

Oie Y., Murata M., Kubota K., and Miyahara H. (1989). Effect of speedup in nonblocking packet switch. *Proc. ICC '89*, Boston, MA, p. 410-14.

Ott, T., Lakshman, T. and Wong, L. (1999). SRED: Stabilized RED. *Proceedings of INFOCOM'99*.