

The Earliest Deadline First Scheduling with Active Buffer Management for Real-Time Traffic in the Internet

Xiaojun Hei and Danny H.K. Tsang
Dept. of Electrical & Electronic Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{heixj, eetsang}@ee.ust.hk

Abstract

We studied the problem of QoS guarantee for differentiated services. A two-level hierarchical scheduling framework was deployed to separate QoS metrics. Due to its desirable property of minimizing the maximum packet lateness, the Earliest Deadline First (EDF) scheduling was adopted to provide the in-class scheduling for the time-sensitive traffic. We employed an EDF scheduler combined with an active buffer management scheme (CHOKe) to improve the fairness of resource allocation and to maintain a good delay performance for real-time applications. Simulation results showed that the proposed scheme can achieve a better delay performance and make a more fair bandwidth allocation between the real-time TCP and UDP connections than the First Come First Served (FCFS) scheduling with the Drop-Tail buffer management which is commonly deployed in traditional IP routers.

I. INTRODUCTION

With the development of digital technology, the convergence of networks is occurring at both the media and the technological levels. Wavelength-Division Multiplexing (WDM) technology promises a large amount of bandwidth for optical networks so that it is possible for multimedia applications to run together with traditional data-oriented applications over a single common network. The Internet has become the dominant networking technology. However, multimedia communication with Quality-of-Service (QoS) guarantee continuously drives the need for substantial changes in the current Internet infrastructure.

Multimedia services such as voice, video and other applications demand not only high bandwidth, but also a stringent real-time delay constraint. The real-time requirement arises from the fact that the value of the communication depends on the time at which messages are successfully delivered to the recipient. The above services are commonly classified as either soft or hard real-time applications. Soft-real-time applications can tolerate some packet loss, while hard-real-time applications have zero loss tolerance (Aras *et al.*).

Internet telephony, one of the promising soft-real-time applications, can permit a few lost phonemes (critical units of speech) in a continuous speech (Hassan *et al.*). On the other hand, in the web-based visualization application, large data sets are processed in the central server and the results need to be transferred back to the client sides for

visualization (Bender *et al.*). Due to the interactive nature of the whole process, both stringent delay and loss-free data transmission are required. Furthermore, the dominant application running on the current Internet is the World Wide Web (WWW). The WWW can support a rich palette of media types such as pictures, audio and video instead of, as in the past, text-only Internet. Web surfing is a highly interactive activity; there is not only lossless data communication but lossable multimedia communication is also involved.

Because of the tight latency bound in the above multimedia applications, it has been suggested that it would be better not to use a reliable transport like TCP for these applications because, in some cases, the retransmission of the lost packet causes packets to arrive too late to be useful (Peterson and Davie). Instead, unreliable transport, UDP, is recommended. However, hard-real-time application requires both stringent delay and zero packet loss so, if it is built based on unreliable transport, the application must deal with the missing packets. Therefore, the complexity of the application software increases. The recently adopted Real-Time Transport Protocol (RTP) follows this approach. It provides mechanisms for dealing with impairment such as jitter and loss, as well as timing recovery and intermedia synchronization. In the recently proposed framework of Differentiated Service (DiffServ), besides the traditional best-effort traffic, two other traffic classes with different Per-Hop-Behavior (PHB), Expedited Forwarding (EF) and Assured Forwarding (AF), have been suggested to serve real-time applications. The transport in the framework has not yet been clearly defined, and it will be interesting to see whether the hard-real-time application can run over TCP. In this paper, soft-real-time and hard-real-time applications are proposed to run on UDP datagrams and TCP flows respectively and, based on the proper scheduling and buffer management schemes, these applications can be served in terms of better delay and packet loss performance.

In order to support communication services with QoS guarantee, network resources need to be managed in a systematic manner. The FCFS-DropTail scheme, which is commonly implemented in traditional IP routers, has some problems in regard to QoS provisions. The FCFS policy can achieve a tight delay bound by limiting the buffer size. With limited buffer size, packets will be dropped with a larger probability when they arrive at routers but there are not enough buffers to store them. The situation may be even worse if the traffic is highly bursty. In addition, the Drop-Tail buffer management can result in the global synchronization of multiple TCP connections which can underutilize the congested link because several connections may halve their congestion windows at the same time. Moreover, the FCFS-DropTail scheme cannot implement a fair resource allocation of the TCP and UDP flows (Pan *et al.*).

In the last several years, new scheduling schemes have been proposed to address some of the above issues. They are basically variants of two fundamental disciplines, Generalized Processor Sharing (GPS) and Earliest Deadline First (EDF). The packetized version of the GPS scheduler (PGPS) can guarantee the minimum per-connection throughput and delay bound with flow protection, but it needs to maintain per-connection states and reserve large bandwidth for a small delay bound. Due to the desirable property of the EDF scheduler, the ability to minimize the maximum lateness of packets (Guerin and Peris), EDF has an advantage over PGPS in that it is able to schedule real-time traffic in terms of system scalability and utilization. Furthermore, TCP performance can be improved by using active queue management (Braden *et al.*) such as Random Early Detection (RED) in which the average queueing delay can be controlled while transient queue-size fluctuation is allowed. However, like Drop Tail, RED is also unable to penalize unresponsive flows (Pan *et al.*). Recently, embedded in RED, a stateless active queue management scheme with the name of CHOKe, was proposed to work with the FCFS scheduling for approximating fair bandwidth allocation. CHOKe tries to bridge fairness and simplicity in the scheduler (Pan *et al.*).

A lot of studies have been done in respect to the scheduling schemes and the buffer management schemes. In fact, the queue management strategy can be used along with any scheduling scheme. Thus, it was suggested to study these schemes in an integrated fashion in order to get the optimal performance (Suter *et al.*). In our research, the EDF scheme is proposed to schedule real time traffic in order to get the optimal delay performance. The active buffer management CHOKe works together with the above EDF scheduling to improve the fairness of resource allocation between UDP-based and TCP-based real-time communications.

The paper is structured as follows. In Section 2, we introduce the two-level hierarchical scheduling framework which supports differentiated traffic in the Internet, and we describe how the EDF scheduler cooperates with the active buffer management scheme (CHOKe). Then, the performance of the EDF scheduler combined with CHOKe is discussed in Section 3. Finally a conclusion of the paper and a discussion of future work are presented in Section 4.

II. HIERARCHICAL SCHEDULING FRAMEWORK

The aim of the hierarchical scheduling framework is to meet the goals of sharing link capacity and of providing differentiated services, such as real-time service, best-effort service, and others (Bennett and Zhang). Link-sharing, first proposed by (Floyd and Jacobson), is one in which network resources are shared among traffic streams and are grouped according to administrative affiliation, protocol, traffic type, or other criteria. The concept of link-sharing

is implemented as a particular resource management scheduling scheme called Class Based Queueing (CBQ).

In CBQ, the user traffic is organized into a tree, or hierarchy, of classes, and traffic classes are differentiated by the network. It is suggested that, because of its simplicity, the FCFS-DropTail scheme is still used to serve the in-class packets. The bandwidth allocation is the major concern in CBQ. However, multimedia applications also require a tight delay bound. Therefore, the fluid Hierarchical Generalized Processor Sharing (H-GPS) system, a general and flexible framework which supports hierarchical link sharing and traffic management for different classes was proposed (Bennett and Zhang). The H-GPS scheme provides a more fine-grained link-sharing structure which can provide a guaranteed end-to-end delay bound for a session if the traffic in that session is leaky-bucket constrained. However, this delay bound is very conservative. On the other hand, without a large bandwidth reservation in EDF, applications are still able to obtain a small delay bound by assigning packets with more urgent time-tags as their deadlines. In our research an EDF scheduler and its variants were used to replace the corresponding GPS scheduler for time-sensitive traffic in the H-GPS framework.

The following hierarchical scheduling framework is introduced in order to fairly allocate bandwidth among classes in the upper level by the GPS scheduler, and to maintain, by the proper selection of the GPS or EDF scheduler in the lower level (shown in Fig. 1 below), the particular throughput or delay QoS guarantee. Due to the scalability concern, the scheduling framework tries to provide service with a class-based QoS.

There are two types of schedulers in the framework, the general scheduler and the link-sharing scheduler. The link-sharing scheduler allocates bandwidth between classes and the general scheduler serves a traffic class with its allocated bandwidth share. They, exclusively, determine the packet scheduling in the absence of congestion. In the presence of congestion, the link-sharing scheduler controls the scheduling of the packets from different classes. The GPS link-sharing scheduler ensures that each interior or leaf class in the multilevel structure receives its allocated bandwidth over appropriate time intervals, and distributes any 'excess' bandwidth fairly between classes.

Traditional best-effort traffic will still run on the Internet of the future and more services may yet emerge. In this paper, we considered how to provide services only for the best-effort and real-time traffic in the two-level hierarchical scheduling framework. This is depicted in Fig. 2. If one more traffic class appears, it should be easy to add one more branch in the hierarchical scheduler tree.

To simplify the problem of how the EDF scheduler cooperates with the CHOCe scheme, we focused our attention only on the EDF-CHOCe branch in the tree. The real-time traffic can be guaranteed in terms of the worst case of

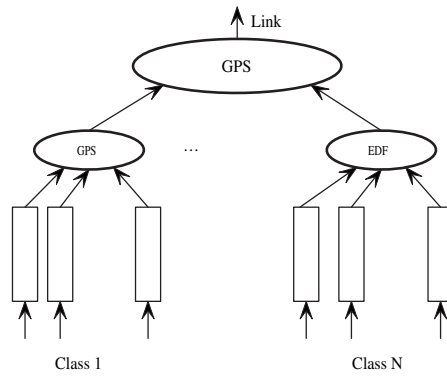


Fig. 1. Hierarchical Scheduling Framework.

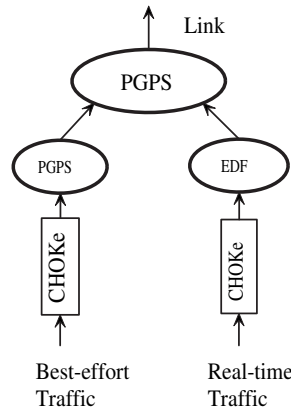


Fig. 2. Hierarchical Scheduling Structure.

minimum bandwidth by using the upper PGPS scheduler. If the best-effort traffic has not used up its bandwidth, the PGPS scheduler has the ability to allocate this excess bandwidth to the real-time traffic. Therefore the QoS provision for real time traffic will be better.

In the following section, we discuss the performance of the various combinations of the service policies (FCFS and EDF) and buffer management schemes (DropTail, RED and CHOKe) via the simulation experiments.

III. PERFORMANCE EVALUATION

A. Simulation Configuration

Simulations were carried out using the Network Simulator (NS). We considered the following simulation scenario with a single congested link, as shown in Fig. 3, in order to study how much bandwidth a single nonadaptive UDP source could obtain when routers used different schemes. The congested link in the network was between the routers R1 and R2. The link, with a capacity of 1 Mbps, was shared by 1 UDP and 32 TCP flows. Each source and destination node was connected to the router using a 10 Mbps link. This was ten times the bottleneck link

bandwidth. All the links had a small propagation delay of 1 *ms* so that the delay experienced by the packet was mainly caused by the queuing delay in the buffer rather than the transmission delay or propagation delay. Different scheduling and buffer management schemes were deployed in the congested link for the comparison studies. The maximum window size of TCP was set at 300 so that it did not become the limiting factor of the TCP flows' throughput. The TCP flows were derived from FTP sessions which transmit a very large file, and the UDP source sent packets at a constant bit rate of 2 *Mbps*. The link between the routers R1 and R2 became the bottleneck link in the network. All the packets were assumed to have the same fixed size of 1000 bytes.

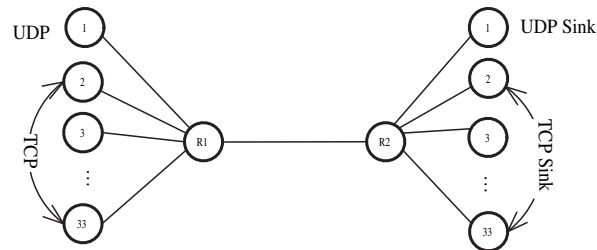


Fig. 3. Network Topology.

The FCFS-DropTail, FCFS-RED, FCFS-CHOKe, EDF-DropTail, EDF-RED and EDF-CHOKe schemes were studied for comparison. The minimum threshold min_{th} in both RED and CHOKe was set at 100 packets, and the maximum threshold max_{th} was set at twice the min_{th} . The physical queue size in the above schedulers was set at 300 packets. The delay requirements for both UDP and TCP flows were set at 600 *ms*.

B. Simulation Results

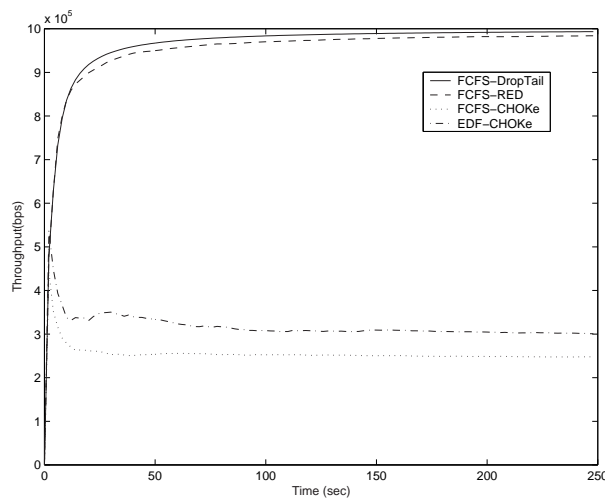


Fig. 4. UDP Throughput Comparison.

The throughputs of the UDP flow of different schemes: FCFS-DropTail, FCFS-RED, FCFS-CHOKe and EDF-CHOKe are plotted in Fig. 4. Fig. 4 clearly shows that the FCFS-DropTail and FCFS-RED schemes do not discriminate against the unresponsive UDP flow. The UDP flow takes away more than 95% of the bottleneck link capacity and all the TCP connections only share the remaining 5% bandwidth. The FCFS-CHOKe scheme provides fairly good resource allocation; the total TCP goodput takes up around 750 *Kbps*, while the EDF-CHOKe scheme has a little less TCP goodput than CHOKe, about 700 *Kbps*. The individual throughputs of the 33 connections in the above FCFS-CHOKe and EDF-CHOKe schemes along with their ideal fair shares are plotted in Fig. 5.

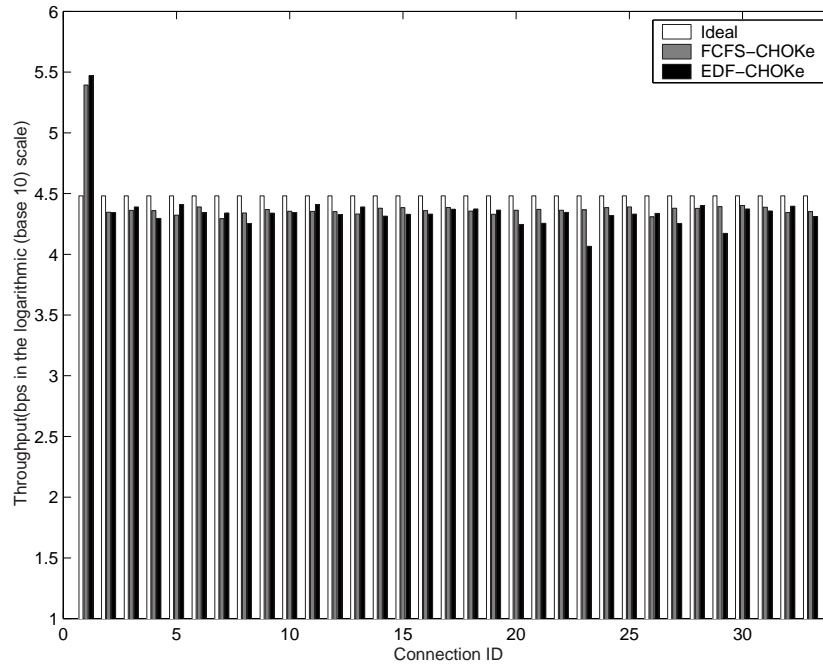


Fig. 5. Per Flow Throughput Comparison.

To provide a quantitative comparison, we adopted the concept of fairness index (Jain). The fairness index always results in a number between 0 and 1, with 1 representing the greatest fairness. Based on the results in Table I, it can be seen that both FCFS and EDF working together with DropTail and RED cannot provide a fair bandwidth allocation. Though the EDF-CHOKe scheme is less fair than the FCFS-CHOKe scheme, it is shown that with the active buffer management scheme, for example, CHOKe, the EDF scheduling provides a more fair allocation than the traditional EDF with the Drop-Tail buffer scheme and the EDF with the active buffer management of RED.

The packet delay distribution at the congested link between R1 and R2 of the TCP connections in the FCFS-CHOKe and EDF-CHOKe schemes are plotted in Fig. 6. Because the FCFS-CHOKe scheme makes the scheduling decision without considering the delay constraint for the communication, most of the TCP packets in EDF-CHOKe

TABLE I
FAIRNESS INDEX COMPARISON

	Fairness Index
Ideal	1.0
FCFS-DropTail	0.0305
FCFS-RED	0.0309
FCFS-CHOke	0.3744
EDF-DropTail	0.0304
EDF-RED	0.0304
EDF-CHOke	0.2838

are transmitted within their delay constraint (0.6 sec) while, in comparison, most of the TCP packets in FCFS-CHOke suffer deadline violation.

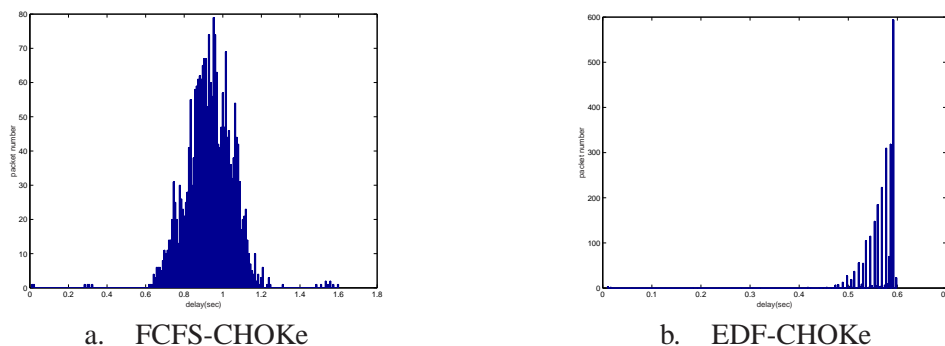


Fig. 6. Packet Delay Distribution of the TCP Connections

Furthermore, the statistics in respect to the packet delay in the congested link between R1 and R2 for the FCFS-CHOke and EDF-CHOke schemes are tabulated in Table II. As seen in Table II, because of its aggressive nature, UDP traffic captures more bandwidth than TCP flows and thus it has a good delay performance. TCP traffic obtains a very different treatment in terms of delay in the above two schemes. In Table II, *avg* refers to the mean packet delay and *std* refers to its standard deviation.

TABLE II
DELAY STATISTICS (SEC)

	Total TCP Traffic		UDP Traffic	
	<i>avg</i>	<i>std</i>	<i>avg</i>	<i>std</i>
FCFS-CHOke	0.9378	0.1203	0.0859	0.2743
EDF-CHOke	0.5665	0.0398	0.0476	0.1564

We investigated the effectiveness of the above scheduling schemes in terms of throughput and delay separately. However, these two principal network metrics are closely related. To describe this relationship, we adopted the *power* of the network (Jain), the ratio between the throughput and the delay. The *powers* of the TCP and UDP connections in the FCFS-CHOke and EDF-CHOke schemes are shown in Table III. Note that the *powers* of the

TCP connections are calculated using the method of the mean of ratio introduced in (Jain). Generally, it is expected to have as much throughput and as little delay as possible. So, with a higher *power* index, the scheme is more effective. Table III shows that the *power* indexes for both TCP connections and UDP connection in EDF-CHOke are larger than those in FCFS-CHOke.

TABLE III
POWER COMPARISON

	TCP connections	UDP connection
FCFS-CHOke	2.45×10^4	3.44×10^6
EDF-CHOke	3.78×10^4	7.65×10^6

Fig. 4, Fig. 6, Table I, and Table II, reveal that the EDF-CHOke scheme can maintain a good delay performance, as well as make a more fair bandwidth allocation between real-time TCP and UDP connections.

Although the EDF-CHOke scheme has a better performance than the traditional FCFS-DropTail scheme, the former is more complex than the latter in terms of implementation and control. The complexity arises because the EDF-CHOke scheme has to select the packet with the smallest deadline for transmission on the link. The scheduler needs to maintain a priority list of deadlines and an insertion or a deletion from this list has a complexity of $O(\log K)$ operations, where K is the number of packets awaiting transmission. In addition, the CHOke buffer management also adds more operating complexity because it needs to calculate the average queue length for dropping decisions whenever a new packet arrives.

IV. CONCLUSION

With more and more real-time multimedia applications running on the current Internet, the next generation Internet is expected to support a wide range of applications with heterogeneous QoS requirements. In this paper, a hierarchical scheduling framework, which maintains particular throughput or delay QoS guarantees for multimedia applications by building the two-level hierarchical scheduling structure with an inter-operation between the GPS scheduler in the upper level and the EDF scheduler in the lower level, was proposed.

In the above framework, the Earliest Deadline First (EDF) scheduler schedules the real-time traffic. Simulation results show that the proposed EDF scheduler working with an active buffer management scheme can achieve a better delay performance and, at the same time, make a more fair bandwidth allocation between real-time TCP and UDP connections than the First Come First Served (FCFS) scheduler with the Drop-Tail buffer management.

There are still many interesting issues for future study. The first is how to develop the theoretical model for the

EDF scheduling so that it can cooperate with the GPS scheduler in the higher hierarchy. This may result in a better traffic management scheme. Second, in the simulation study, the UDP traffic model is assumed to be a constant bit stream, but recent studies show that the Internet traffic is self-similar (Paxson and Floyd). The scheduling schemes proposed in this paper were only studied in a simple network scenario of one congested link. We intend to explore these mechanisms in more complicated scenarios with multiple congested links.

REFERENCES

- C.M. Aras, J.F. Kurose, D.S. Reeves, and H. Schulzrinne. (1994). real-time communication in packet-switched network. *Proceedings of the IEEE*, 82(1):122-39.
- M. Hassan, A. Nayandoro, and M. Atiquzzaman. (2000). Internet telephony: services, technical challenges, and products. *IEEE Communications Magazine*, 38(4):96-103.
- M. Bender, R. Klein, A. Disch, and A. Ebert. (2000). A functional framework for Web-based information visualization systems. *IEEE Transactions on Visualization & Computer Graphics*, 6(1):8-23.
- Larry L. Peterson and Bruce S. Davie. (2000). *Computer networks : a systems approach*. 2nd edition, San Francisco : Morgan Kaufmann Publishers.
- R. Pan, B. Prabhakar, and K. Psounis. (2000). CHOCkE: a stateless active queue management scheme for approximating fair bandwidth allocation. *Proceeding of INFOCOM'2000*.
- R. Guerin and V. Peris. (1999). Quality-of-service in packet networks: basic mechanisms and directions. *Computer Networks*, 31:169-189.
- B. Braden, et al.. (1998). Recommendations on Queue Management and Congestion Avoidance in the Internet. RFC 2309, Internet Engineering Task Force.
- B. Suter, T.V. Lakshman, D. Stiliadis, and A.K. Choudhury. (1999). Buffer management schemes for supporting TCP in gigabit routers with per-flow queueing. *IEEE Journal on Selected Areas in Communications*, 17(6):1159-69.
- Jon C. R. Bennett, Hui Zhang. (1997). Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675-89.
- Sally Floyd, Van Jacobson. (1995). Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4):365-86.
- Raj Jain. (1991). *The art of computer systems performance analysis : techniques for experimental design*,

measurement, simulation, and modeling. New York : Wiley.

NS (Network Simulator): <http://www.isi.edu/nsnam/ns/>.

Vern Paxson, Sally Floyd. (1995). "Wide area traffic: the failure of Poisson modeling". *IEEE/ACM Transactions on Networking*, 3(3):226-44.