MASTER'S THESIS

Bandwidth Allocation to Interactive Users in DBS-Based Hybrid Internet

by M. Stagarescu
Advisor: J.S. Baras

CSHCN M.S. 98-3
(ISR M.S. 98-6)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1998		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Bandwidth Allocation to Interactive Users in DBS-Based Hybrid Internet	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Research Laboratory,2800 Powder Mill Road,Adelphi,MD,20783**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**76**	

ABSTRACT

Title of Thesis: BANDWIDTH ALLOCATION TO
INTERACTIVE USERS
IN DBS-BASED HYBRID INTERNET

Degree candidate: Marian Stagarescu

Degree and year: Master of Science, Department of Electrical Engineering,
University of Maryland, College Park, 1998

Thesis directed by: Professor John S. Baras
Department of Electrical Engineering

We are motivated by the problem of bandwidth allocation to Internet users in DBS-based Hybrid Internet, where the Network Operations Center (NOC)-scheduler controls the amount of service provided to each user, by using packet scheduling and buffer management. Such a system exploits the ability of satellites to offer high bandwidth connections to large geographical areas, and it delivers low-cost hybrid (satellite-terrestrial) high-speed services to interactive Internet users. In this system, it is important to reduce the delay that users experience.

We analyze several bandwidth allocation policies at the Network Operations Center (NOC) of a DBS-based hybrid Internet network. We consider the problem of optimal scheduling of the services of interactive users in the DBS-based hybrid Internet configuration.

We show that, for the interactive Internet users, the Most Delayed Queue First (MDQF) policy, which serves the queues starting with the most delayed queue, is providing the minimum delay, when compared with the Equal Bandwidth (EB) and Fair Share (FS) allocation policies. The MDQF policy is shown to be optimal with respect to a performance metric of packet loss due to queuing time constraints.

The impact of the scheduling policies on the Hybrid Internet system's performance, is analyzed in the context of the interplay between the NOC queuing system (and its bandwidth allocation policies) and the underlying transport protocol (TCP), and we show the effectiveness of the MDQF policy in the presence of TCP congestion control algorithm.

We also present simulations in which the Internet server sources send self-similar ("fractal") data traffic to the NOC-scheduler. The results confirm our calculations, that the MDQF policy is a better performing policy for minimizing the mean delay at high load factors, when comparing it to the EB and FS policies. Finally, we propose two solutions, a buffer allocation policy and a "virtual delay" mechanism, which make the MDQF policy work in the presence of greedy sources.

BANDWIDTH ALLOCATION TO INTERACTIVE USERS IN DBS-BASED HYBRID INTERNET

by

Marian Stagarescu

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science

Advisory Committee:

Professor John S. Baras, Chairman/Advisor
Professor Prakash Narayan
Assistant Research Scientist M. Scott Corson

DEDICATION

To my wife, Flavia

To my mother and in the memory of my father

ACKNOWLEDGEMENTS

I am greatly indebted to my advisor, Dr. John S. Baras, for his guidance, support and encouragement through the course of this work. He introduced me to many new ideas through several illuminating discussions pertaining to this thesis. I would also like to thank Dr. P. Narayan and Dr. M. S. Corson for their valuable comments on the thesis. Also, I want to mention that the simulations were performed with a modified version of a simulator developed by G. Olariu.

This work was supported by the Army Research Laboratory (ARL), Federated Laboratory, Advanced Telecommunications and Information Research Program (ATIRP) under cooperative agreement DAAL01-96-2-0002, by NASA under cooperative agreement NASA-NCC3-528, by a contract from Lockheed Martin Telecommunications, and by a contract from Hughes Network Systems through the Maryland Industrial Partnership Program.

Finally, I wish to express my deepest gratitude to my wife for her love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

We are facing a dramatic shift in the nature of wide-area computer networks. We have moved into an era of commercial networking. While the Internet started out as a research network, it has gone through a rapid transition to a commercial service. The diversity of applications on the Internet is ever-increasing. With the rapid growth of Web-based Internet applications, such as Web servers and browsers, it has become crucial to understand the behavior of feedback based flow and congestion control protocols in a realistic scenario, using traffic that correspond to the one produced by dominant Internet applications. Moreover, best-effort service can be used for packet voice and video, in addition to its more traditional use for file transfer, electronic mail, and remote login. The nature of best-effort service precludes specifying the actual packet delay a user will experience. However, users of a commercial network are unlikely to accept having large delays in the service they receive. Thus, one of the challenge of designing current commercial networks is to develop a service model that can provide a variety of quality services to the user.

The Center for Satellite and Hybrid Communication Networks (CSHCN) and

Hughes Network Systems (HNS) have been working together to develop inexpensive hybrid (satellite and terrestrial) terminals that can foster hybrid communications as the most promising path to the Global Information Infrastructure [4], [5]. This hybrid service model ("Hybrid Internet Access") capitalizes on the existing installed base of cable and DBS-based network, and it allows information browsing and interactivity by the utilization of asymmetric channels. Its design concepts are presented in Chapter 2. $DirecPC^{TM}$, a commercial product of HNS is a typical example of Hybrid Internet Access. One of the services provided by a Hybrid Satellite Terrestrial Network (HSTN) is high speed Internet access based on an asymmetric TCP/IP protocol.

An effective flow and congestion control protocol for best-effort service is built upon four mechanisms: packet scheduling, buffer management, feedback and end adjustment. This protocol has two points of implementation. The first one is at the source, where flow control algorithms vary the rate at which the source send packets. The second point of implementation is at the Network Operations Center (NOC) of the HSTN. The two ways in which the NOC in the HSTN architecture can actively manage its own resources are packet scheduling (bandwidth allocation) and buffer management.

Packet scheduling algorithms, which control the order in which packets are sent and the usage of the NOC's buffer space, do not affect the congestion directly , in that they do not change the total traffic on the NOC's outgoing satellite link. Scheduling algorithms do, however, determine the way in which packets from different sources interact with each other which, in turn, affects the collective behavior of flow control algorithms. We shall argue that this effect makes packet scheduling algorithms a crucial component in effective congestion

control. In addition with being the most direct control by which the NOC serves every user, it is the only effective control within the scope of NOC, because buffer management alone cannot provide flexible and robust control of bandwidth usage.

Consequently, the main focus on this thesis is the investigation of the effectiveness of various packet scheduling (bandwidth allocation) policies at the NOC.

In Chapter 2 we introduce various bandwidth allocation schemes. One important component of the dramatic shift in the nature of Internet is the following: the assumption of user cooperation is no longer valid in the Internet [10]. Subsequently, discriminating queuing algorithms, which incorporates packet scheduling and buffer allocation must be used in conjunction with source flow control algorithms, to control congestion effectively in non-cooperative environments. We present first a simple policy, the Equal Bandwidth (EB) bandwidth allocation, in which the NOC maintains separate queues for packets from each individual data traffic source, and allocates an equal amount of service to each active source. This prevents a source from arbitrarily increasing its share of the bandwidth or the delay of other sources. In order to reduce the possible waste of bandwidth under EB policy we consider a more refined policy: The Fair Share (FS) bandwidth allocation [7]. the FS policy is superior to the EB policy both in satisfying connection requests, and minimizing the waste of bandwidth.

On the surface, these two scheduling policies appear to have considerable merit, but they are not designed to optimize a performance measure such as throughput or delay. A particular motivation for the investigations reported in

this thesis has been the issue of optimal packet scheduling at the NOC, in the sense of minimizing the queuing delay. In consequence, we were interested in obtaining improvements in the service quality, as perceived by the users. A better performing policy, which attempts to minimize the mean queuing delay at the NOC, especially at high load factors, may be one which serves the connections in the order of their queuing delay, starting with the most delayed queue. Such a scheme, formally introduced in [15], is appealing due to its similarity to Shortest Time to Extinction (STE) [1] scheduling policies. This thesis is a continuation of the research effort initiated in [15]. We describe this Most Delayed Queue First (MDQF) policy in section 2.1.3.

Chapter 3 deals with the analysis of these bandwidth allocation policies. We will consider the problem of optimal scheduling of the services of interactive users in a DBS-based Hybrid Internet configuration. Using a congested scenario, we derive approximate conditions under which the MDQF policy performs better than the EB and/or FS strategies. In heavy-traffic conditions these first-order approximations are analyzed, and the MDQF gain is obtained and exemplified.

The MDQF policy is shown to be optimal with respect to a performance metric of packet loss, when hard queuing time constraints are present: it minimizes, in the stochastic ordering sense, the process of packet loss, when compared with the EB/FS policies. This result is derived using a methodology introduced in [1], with variations specific to our case.

The different components of flow and congestion control algorithms introduced above, source flow control, NOC packet scheduling and buffer manage-

ment, interact in interesting and complicated ways. It is impossible to asses the effectiveness of any algorithm without reference to the other components of congestion control in operation. We will evaluate the proposed MDQF scheduling algorithm in the context of the underlying transport protocol of the DBS-based Hybrid Internet system, namely the TCP flow and congestion control algorithm. We use an estimate of the effective efficiency of the satellite gateway to show the effectiveness of the MDQF policy in the presence of the TCP congestion control algorithm, and we obtain an analytical expression for the MDQF gain over the EB and FS policies.

In Chapter 4 we investigate the impact of self-similar data traffic source models on hybrid Internet networks, including their effect on throughput and delay. This is done, in the context of various bandwidth allocation mechanisms, using simulations. Due to its correspondence to the Internet interactive users and WWW traffic, we first study the effects of the ON-OFF "heavy-tailed" data traffic source model on performance, when Equal Bandwidth (EB), Fair Share (FS) and the Most Delayed Queue First (MDQF) schemes are employed at the Network Operations Center (NOC) of a DBS-based Hybrid Internet network. We find that the MDQF policy performs better than the other bandwidth allocation strategies in congested scenarios.

Our investigations of the "cooperative" work among the sources in the MDQF scheme reveals an interesting phenomenon of "delay shifting" (section 4.1.2), due to the presence of "greedy" sources in the system. The degree of "delay shifting" can be controlled by buffer allocation, which exemplifies the interaction between the two control mechanisms available at the NOC; i.e. the packet scheduling

and the buffer management. Also, we propose another solution, a "virtual delay" mechanism imposed on the '"greedy" sources. This solution is intrinsically related to the dynamics of the MDQF policy.

One of our goals is to find a scheduling algorithm that functions well in current computing environments, where self-similar traffic is present [2]. The "heavy-tailed" ON-OFF model can generate self-similar traffic only when a "large" number of ON-OFF "heavy-tailed" (Pareto) distributed sources are aggregated [22], [11], [17]. This asymptotic result is not easily applicable in a simulation context. In addition, the ON-OFF model fails to capture at least one important characteristics of WWW traffic: the model assumes constant rate during the transmission, whereas in reality, the transmission rate of WWW traffic depends on the congestion status of the network.

To this end, we consider a more robust self-similar traffic model, the Fractional Brownian Traffic (FBT) model [13], [14]. The ability to capture rate fluctuations of the FBT source model, is a considerable improvement over the previous model. We present the Fractional Brownian Traffic generator and address its accuracy by comparing a synthesized FBT trace with real data from Bellcore. The hybrid Internet configuration for FBT experiments is presented in section 4.2. Similar findings of the "delay-shifting" phenomena and the effectiveness of the buffer allocation and the "virtual delay" solutions for making "greed" sources work with the MDQF policy, are presented in the context of FBT simulations.

Chapter 2

Hybrid Internet Access and Bandwidth Allocation Policies

The Internet is rapidly growing in number of users, traffic levels and topological complexity. Sustained network traffic and proliferation of multimedia applications have combined to challenge the Internet access solutions for home users and small enterprise.

Existing solutions as modem dial-up or "fiber-to-the-home" are either too slow or too expensive. A "Hybrid Internet Access" solution, exploiting the ability of satellites to offer high bandwidth connections to a large geographical area and the "asymmetric" Internet traffic of many users, so that they can use a receive-only VSAT, was developed by the Center of Satellite and Hybrid Communication Networks and Hughes Network Systems (HNS) [4], [5]. The structure of this Hybrid Satellite-Terrestrial Network (HSTN) is depicted next.

A hybrid terminal uses a modem connection for outgoing traffic and, through a second network interface receives incoming information from the Internet via VSAT. The hybrid terminal is attached to the Internet by using an Internet Service Provider. The traffic is transmitted to a hybrid gateway which is re-

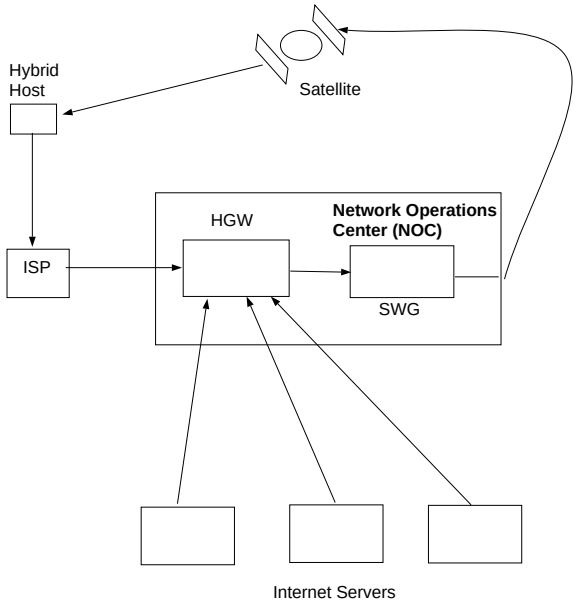

Figure 2.1: DBS-based Hybrid Internet Architecture

sponsible for splitting an end-to-end TCP connection from the hybrid terminal to any Internet application server and managing the data flow of the conventional terrestrial network and the hybrid network. By splitting the TCP connection the satellite channel is isolated from the Internet hosts. The Hybrid Internet Access solution copes with the long-delay effect of the satellite link by allowing the hybrid gateway to acknowledge packet reception from Internet hosts on the behalf of hybrid terminals; a technique called "TCP spoofing". $DirecPC^{TM}$, a commercial product of HNS is a typical example of Hybrid Internet Access. It employs a prioritization scheme which uses separate queues for Internet traffic and "push" traffic (package delivery and data feed traffic).

To fully utilize the satellite bandwidth and buffer space, the NOC must actively manage these resources and also provide feedback to the users. The hy-

brid users' acknowledgment packets are used to remove transmitted data from the hybrid gateway's buffers and they must respond to congestion signals from the NOC. Packet scheduling (bandwidth allocation) and buffer management are the two ways by which the NOC in the HSTN architecture manages its own resources. Scheduling is the most direct control by which the NOC serves every user. It is the only effective control within the scope of the NOC because buffer management alone cannot provide flexible and robust control of bandwidth usage.

The buffer management is used in conjunction with scheduling mechanisms, but need not be precisely tuned (a simple buffer allocation scheme is described in chapter 4, sections 4.1.2 and 4.2.3). In addition, the interplay between scheduling policies and the other two mechanisms for effective congestion control, namely feedback and end-adjustment, is studied and the results (section 3.6) show how an effective bandwidth allocation policy can help alleviate some of the TCP problems, the underlying transport protocol assumed in the HSTN configuration.

2.1 Bandwidth Allocation Policies

2.1.1 Equal Bandwidth Allocation Policy

One of the simplest packet scheduling algorithm is the **Equal Bandwidth Allocation (EB)** which attempts to split the available bandwidth evenly among the currently present flows. It provides each flow with a great degree of protection from other flows (in the sense of unfair capture of channel bandwidth). However, there can be a significant waste of bandwidth while operating under this scheme.

2.1.2 Fair Share Bandwidth Allocation Policy

An improvement (in the sense of less waste of bandwidth) over EB is offered by the **Fair Share Bandwidth Allocation (FS)** [7]. The Fair Share algorithm tries to cope with the waste of bandwidth while preserving the flow firewalls. The algorithm is described briefly below:

- **Fair Share Bandwidth Allocation Algorithm**

Step 1 Compute the **Fair Share** by dividing the total bandwidth to the number of active connections.

Step 2 Allocate bandwidth to connections with individual demand less than or equal to the **Fair Share** ("under-loading connections").

Step 3 Find the remaining bandwidth after the Step 2 allocation.

Step 4 Recompute the **Fair Share** excluding the set of "under-loading connections".

Step 5 The iteration is repeated from the allocation Step 2 unless it cannot be performed; in this case allocate the **Fair Share** to all connections in the current active set.

While these two algorithms attempt to provide a degree of fairness and reduce the packet clumping problem present in a FIFO queue they are not designed to optimize a performance measure such as throughput or delay.

2.1.3 Most Delayed Queue First Policy

Intuition suggests that a better performing policy, which myopically attempts to minimize the mean queuing delay at the NOC, especially at high load factors,

may be one which serves the connections in the order of their queuing delay, starting with the most delayed queue. One can find such a scheme appealing due to its similarity to Earliest Deadline First (EDF) [6], [12] or Shortest Time to Extinction (STE) [1] scheduling policies. Such a scheme was formally introduced in [15] and the algorithm is described below:

- **Most Delayed Queue First Bandwidth Allocation Algorithm**

Step 1 Sort the connections in decreasing order of the delay encountered by the Head-of-the-Queue (HoQ) packet.

Step 2 Allocate bandwidth to satisfy the current demand of the queues in the ordered set obtained from Step 1.

Step 3 Repeat Step 2 until the available bandwidth is exhausted or all connections are served.

All bandwidth allocation policies are analyzed in Chapter 3. Simulation results are reported in Chapter 4. For the simulations we used a revised version of a DirecPC flow prioritization and control simulator built by Olariu [15].

Chapter 3

Analysis of Bandwidth Allocation policies

In this chapter we will consider the problem of optimal scheduling of the services of interactive users in a Hybrid Internet configuration. The Network Operations Center (NOC) controls the amount of bandwidth allocated to each user by employing various control policies. We investigate the Equal Bandwidth (EB), Fair Share (FS) and Most Delayed Queue First (MDQF) policies. Using an ON-OFF source traffic model, with constant rate during the active period we present first a framework for the analysis and a quantification of the service quality as perceived by users. The merit of this approach, first introduced in [8], is the ability to capture the characteristics of the dominant application for interactive Internet users, namely the World Wide Web. The user quality depends critically on the rates allocated by the NOC controller. Using a congested scenario we derive approximate sufficient conditions under which the MDQF policy performs better than the EB and/or FS strategies. In heavy-traffic conditions these first-order approximations are analyzed and the MDQF gain is obtained and exemplified.

The MDQF policy is shown to be optimal with respect to a performance

metric of packet loss due to queuing time constraints. This result is derived by using a methodology introduced in [1], with variations specific to our case.

Finally the impact of the scheduling policies on the Hybrid Internet system performance is analyzed from a different perspective. The interplay between the NOC queuing system and its bandwidth allocation policies, and the underlying transport protocol (TCP) is investigated in section 3.6. We use an estimate of the effective efficiency of the satellite gateway to show the effectiveness of the MDQF in the presence of the TCP congestion control algorithm.

3.1 Framework for queuing analysis: ON/OFF source traffic model

In a Hybrid Internet configuration the queuing system of interest is represented by the satellite gateway (SGW) of the Network Operations Center (NOC). The data traffic sources, represented by the Internet servers are modeled with an alternating renewal process, i.e. the source alternates between active and idle periods. The active periods represent time intervals when the source is sending packets at a constant rate. The source is silent during the idle periods. If we denote with B the distribution function of the active periods, with mean $1/\mu \leq \infty$ and with I the distribution function of the idle periods, with mean $1/\lambda \leq \infty$, the long-run probability that the source is active is given by:

$$a_{on} = \frac{1/\mu}{1/\mu + 1/\lambda}. \tag{3.1}$$

Let N be the number of sources and suppose the capacity of the available satellite link is C [packets/sec]. The number of simultaneous sources that can send data

at rate c [packets/sec] is given by $s = C/c$. Let P_j be the steady-state probability that j sources are active. When $N \leq s$ there is no contention among the sources for the server capacity and P_j has the binomial distribution:

$$P_j = \left(\frac{N}{j}\right) a_{on}^j (1 - a_{on})^{N-j}, j = 0, 1, \ldots, N \leq s. \tag{3.2}$$

In [8] P_j was shown to be insensitive to the distributions of B (the busy period) and of I (the idle period). P_j is insensitive because a_{on} depends on the distributions of B and I only through their means. This insensitivity property can be used to compute P_j, for the case when $N > s$, based on exponential on/off distributions and then apply the solution to ON/OFF Pareto distributions with the same means.

If we define by w the mean size of a Web page, when there is no congestion at the service facility, the average time to complete the Web page retrieval is $1/\mu = w/c$. When the number of active users is $j \geq s$ each source will receive service at a rate $r(j)$. The formula for $r(j)$ depends on the bandwidth allocation strategy. For example if the Equal Bandwidth allocation strategy is used $r(j)$ equals c if $j \leq C/c$ and $r(j) = C/j$ if $j \geq C/c$. The throughput of the service facility is given by

$$throughput = \sum_{j=1}^{N} P_j j r(j) = E[Jr(J)] \tag{3.3}$$

where J(t) is the random number of active sources at time t.

3.2 Service quality

In this section we are interested to quantify the service quality. For users browsing the Web a primary measure of inconvenience is the time it takes the network

to complete the delivery of a page. In [15] the primary measure of service quality was the average delay defined as:

$$d = \frac{\Sigma \ \ delay\ of\ ACKed\ packets}{number\ of\ ACKed\ packets}$$

Let T be the average time it takes the NOC (service facility) to complete the delivery of a WEB page requested by a user, and let $T_0 = \frac{w}{c}$ be the "ideal" average time of this transfer. To quantify the service quality we use the following "figure of demerit" introduced in [8].

$$D = \frac{T}{T_0}. \tag{3.4}$$

In order to compute D we have to take in consideration various average rates. The average aggregate rate at which users receive complete WEB pages is $E[Jr(J)]/w$. By Little's law, the average number $E[J]$ of users in the active phase equals the product of the above average aggregate rate, $E[Jr(J)]/w$, with the average time T that a user stays in active phase. Hence,

$$E[J] = \frac{E[Jr(J)]}{w} \cdot T \tag{3.5}$$

and the final formula for D is ([1]):

$$D = \frac{cE[J]}{E[Jr(J)]} \tag{3.6}$$

The computation of D requires analytical expressions for $r(J)$. In the next section we derive approximate expressions for $r(J)$ for the investigated bandwidth allocation schemes, under various conditions.

3.3 Observations on the Dynamics of MDQF policy

We want to derive conditions under which the following property holds:
(P) The Most Delayed Queue First (MDQF) strategy performs better than Equal Bandwidth (EB) and/or Fair Share (FS) with respect to the D criterion for service quality.

We consider the following scenario. The sources present an amount of c data packets to the service facility, at each time instant during their active (ON) period. All sources are assumed to submit the same amount of workload X [packets] to the service facility. Consider that all sources start their first ON period at time $t = 0$ and we have a number N of active sources. We restrict our attention to the congestion regime where $c \geq C/N$, with C the NOC capacity [packets]. The amount of service allocated to a source under the EB and FS strategies is $r = C/N$ at each service time.
When the MDQF strategy is used, the service for a typical source operates in cycles. Let us denote by $n_1, n_1 + n_2, \ldots, n_1 + \ldots + n_k$ the time instants of initiation of service periods for a source and by $r_{n_1}, \ldots, r_{n_1+\ldots+n_k}$ the amount of service (number of packets served) allocated to the source, at those instants, under MDQF policy.

The following condition must be satisfied in order to guarantee a better service quality under the MDQF policy, as compared to the EB and FS policies:

$$\frac{r_{n_1} + \ldots + r_{n_1+\ldots+n_k}}{n_1 + \ldots + n_k} \geq \frac{C}{N} \tag{3.7}$$

This condition requires that the average service under MDQF policy is larger

than the average service under EB and FS. By using the workload X we can express the condition (3.7) as follows. If we require that the workload X [packets] will be met by the sum of the service allocated at the instants $n_1, n_1+n_2, \ldots, n_1+\ldots+n_k$, under the MDQF policy, i.e.

$$X - \sum_{p=n_1}^{n_1+\ldots+n_k} r_p = 0 \tag{3.8}$$

then (3.7) is equivalent to:

$$X - (n_1 + \ldots + n_k)\frac{C}{N} \geq 0 \tag{3.9}$$

In order to obtain specific conditions derived from equations (3.8), (3.9), we study how these equations propagates, at each time instant of service initiation, i.e. instead of the "integral condition" (3.7) we will derive "instantaneous conditions". For this, we will work with the residual work (denoted res), i.e. the unfinished work counted right after the completion of a service period. For example the residual work after the completion of the service initiated at time instant n_1 is :

$$res_{n_1}^{MDQF} = X - r_{n_1}$$

In terms of residues we can write:

$$res_{n_1+\ldots+n_k}^{MDQF} = res_{n_1+\ldots+n_{k-1}}^{MDQF} - r_{n_1+\ldots+n_k} \tag{3.10}$$

$$res_{n_1+\ldots+n_k}^{FS} = res_{n_1+\ldots+n_{k-1}}^{FS} - n_k\frac{C}{N} \tag{3.11}$$

Assuming a stronger condition than (3.7), namely that the residual workload of MDQF is smaller than that of EB and FS at one time instant of initiation of service periods:

$$res_{n_1+\ldots+n_{k-1}}^{MDQF} \leq res_{n_1+\ldots+n_{k-1}}^{FS} \tag{3.12}$$

the following inequality must be satisfied in order to guarantee 3.12:

$$r_{n_1+\ldots+n_k} \geq n_k \frac{C}{N} \tag{3.13}$$

Applying backwards these stronger conditions we obtain the following set of sufficient conditions for property (P) to hold:

$$r_{n_1} \geq n_1 \frac{C}{N} \tag{3.14}$$

$$r_{n_1+n_2} \geq n_2 \frac{C}{N} \tag{3.15}$$

$$\ldots$$

$$r_{n_1+\ldots+n_k} \geq n_k \frac{C}{N} \tag{3.16}$$

It is clear that conditions (3.14) to (3.16) imply condition (3.7); that is (3.14) - (3.16) are stronger sufficient conditions than (3.7) for property (P) to hold.
At time instant n_1 we must have

$$res_{n_1}^{MDQF} = X - r_{n_1} \leq X - n_1 \frac{C}{N} \tag{3.17}$$

because of condition (3.14). Up to this time the source has sent $n_1 c$ data packets, since its total ON time cannot exceed n_1. Therefore the service allocated to this source at n_1 has to satisfy

$$r_{n_1} \leq C \quad and \quad r_{n_1} \leq n_1 c \tag{3.18}$$

We choose $r_{n_1} = min(C, n_1 c)$ which satisfies (3.18) and from (3.17) we get

$$X - min(C, n_1 c) \leq X - n_1 \frac{C}{N} \tag{3.19}$$

or

$$min(C, n_1 c) \geq n_1 \frac{C}{N} \tag{3.20}$$

If $C \geq n_1 c$ then (3.20) requires $c \geq C/N$, which is satisfied by the assumption made that we operate in the congestion regime.

If $C \leq n_1 c$ then (3.20) requires

$$n_1 \leq N. \tag{3.21}$$

Therefore, for the congestion regime (3.14) implies (3.21). However (3.21) does not necessarily imply (3.14).

At time instant $n_1 + n_2$ we must have:

$$res_{n_1+n_2}^{MDQF} = X - r_{n_1} - r_{n_1+n_2} \leq X - (n_1 + n_2)\frac{C}{N} \tag{3.22}$$

which implies

$$r_{n_1+n_2} \geq n_2 \frac{C}{N} \tag{3.23}$$

The allocated service at time instant $n_1 + n_2$ is chosen to be the minimum of C and the current demand $d_{n_1+n_2} = d_{n_1} - r_{n_1} + n_2 c$, where d_{n_1} was the demand at time instant n_1, $d_{n_1} = n_1 c$. The inequality (3.23) becomes

$$r_{n_1+n_2} = min(C, d_{n_1+n_2}) \geq n_2 \frac{C}{N} \tag{3.24}$$

If $C \leq d_{n_1+n_2}$ then (3.24) requires

$$n_2 \leq N. \tag{3.25}$$

If $C \geq d_{n_1} - r_{n_1} + n_2 c$ we must have

$$d_{n_1} - r_{n_1} + n_2 c \geq n_2 \frac{C}{N} \tag{3.26}$$

Replacing $d_{n_1} = n_1 c$ (3.26) is equivalent with:

$$n_1 c - r_{n_1} + n_2 c \geq n_2 \frac{C}{N} \tag{3.27}$$

which is obviously satisfied since $r_{n_1} \leq n_1 c$, and $c \geq C/n$ from the congestion regime's assumption.

The following result demonstrates that the "instantaneous conditions" (3.14)-(3.16) propagates in time if the corresponding service cycle length is less than or equal to the number of active sources N.

Fact: Suppose

$$r_{n_1+\ldots+n_p} \geq n_p \frac{C}{N}$$

Then

$$r_{n_1+\ldots+n_{p+1}} \geq n_{p+1} \frac{C}{N} \tag{3.28}$$

if

$$n_{p+1} \leq N$$

Proof:

$$r_{n_1+\ldots+n_{p+1}} = min(C, d_{n_1+\ldots+n_p} - r_{n_1+\ldots+n_p} + n_{p+1}c)$$

If $C \leq d_{n_1+\ldots+n_p} - r_{n_1+\ldots+n_p} + n_{p+1}c$ (3.28) becomes

$$C \geq n_{p+1} \frac{C}{N} \Rightarrow n_{p+1} \leq N.$$

If the minimum is $d_{n_1+\ldots+n_p} - r_{n_1+\ldots+n_p} + n_{p+1}c$ we must have

$$d_{n_1+\ldots+n_p} - r_{n_1+\ldots+n_p} + n_{p+1}c \geq n_{p+1} \frac{C}{N}$$

which is obviously satisfied since $c \geq \frac{C}{N}$ and $d_{n_1+\ldots+n_p} \geq r_{n_1+\ldots+n_p}$.

This result holds also for the case when the time instants of service initiation $n_1 + \ldots + n_{p+1}$ corresponds with a time instant when the source considered has finished its sending period, i.e.

$$d_{n_1+\ldots+n_{p+1}} = d_{n_1+\ldots+n_p} + r_{n_1+\ldots+n_p}$$

$\triangledown\triangledown\triangledown$

Let us now analyze further the conditions we have obtained for the time instants

$$n_1, n_2, \ldots n_k \leq N \tag{3.29}$$

We consider the following case : $c = \frac{C}{p},\ p > 1$

Suppose that, when we have the same delay at two or more queues, we tie-break based on the source index. In this situation the worst-case analysis is for source indexed N. We make the following observation regarding the values $n_p,\ p = 1, 2, \ldots, k$.

The amount of service allocated under MDQF strategy in "heavy-traffic" condition will have a transient period followed by a stationary value of C packets. In this situation the gain obtained from the MDQF policy, when compared with the EB and the FS policies, is obtained only during the transient period since after that, in one cycle of N steps, the allocation rate for a source under the MDQF policy is C packets, and for EB and FS we have $N \cdot \frac{C}{N} = C$ packets.

The length of the transient period is p time-clocks since after that, the current demand to be resolved is C packets. During this transient period only sources $1, \ldots, p$ will obtain a gain of $\frac{C}{p}$ under MDQF strategy. We present next a set of simulation results that exemplify the gain of the MDQF scheme in heavy-traffic.

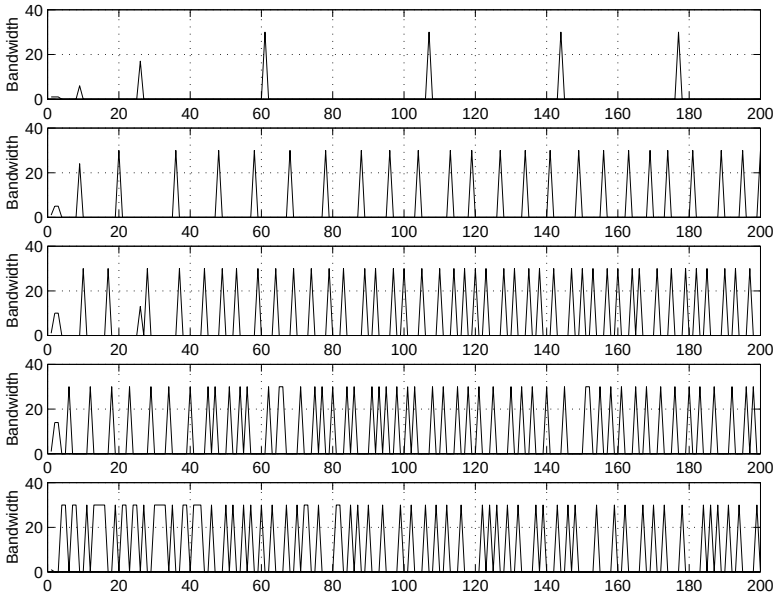

Figure 3.1: Transients in bandwidth allocation for MDQF scheme: $C = 30,\ N = 5$ ON/OFF Pareto sources

Bandwidth/Clock	Goodput	D
15	13.4	1.08
10	9.6	1.5
5	4.86	2.95

Table 3.1: Fair Share allocation

Bandwidth/Clock	Goodput	D
15	14	1.0
10	10	1.4
5	5	2.8

Table 3.2: MDQF allocation

3.4 Conditional Optimality of the MDQF Policy

In this section, we will consider the problem of optimal scheduling of the queues at the Network Operation Center (NOC).

We will consider the case in which the packets have constraints on their waiting times. We will prove the conditional optimality of MDQF policy, with respect to Equal Bandwidth/Fair Share when hard deadline constraints are used, i.e. packets are considered eligible for service at NOC any time before the deadline and become obsolete as soon as the deadline is missed.

Consequently, the metric of interest is the number of packets lost and we prove that the MDQF policy minimizes this metric, when compared with the EB/FS

policies.

The packets arriving at the service facility are time-stamped and enqueued into the corresponding flow (connection) queues. Also, upon their arrivals the packets will be allocated a deadline (fixed for all packets) corresponding to the delay imposed to queued packets. For each time-stamp we have a corresponding extinction time defined as the value of time-stamp plus the deadline. The MDQF policy corresponds to the Shortest Time to Extinction (STE) policy. The STE policy and its optimality in the class of non-preemptive policies is investigated in [1] and [16].

Our result is different from the one in [1] in the following sense: the exponential service requirement is relaxed and, in order to compare the MDQF and the Equal Bandwidth/ Fair Share policies, we consider a congested scenario where both Equal Bandwidth and Fair Share policies allocate the same equal (or fair) share of service to each source.

The active queues are ordered based on their Head-of-the-Queue extinction times. The MDQF policy will be denoted $\tilde{\pi}$ and the EB/FS policy by π. The system's evolution is depicted in figure 3.2.

A message sent from a source to the service facility is composed of packets with the same time-stamp. At a particular time instant a message is declared lost if its extinction time expired. Otherwise is considered eligible. The set of eligible messages at time t is denoted $E(t)$ or $E(t, \pi)$ to specify the policy π used. e_k will refer to both the extinction time and to the corresponding message with this extinction time. If we consider p the number of packets into the equal share, the service time granularity corresponds to the time needed to transmit p packets.

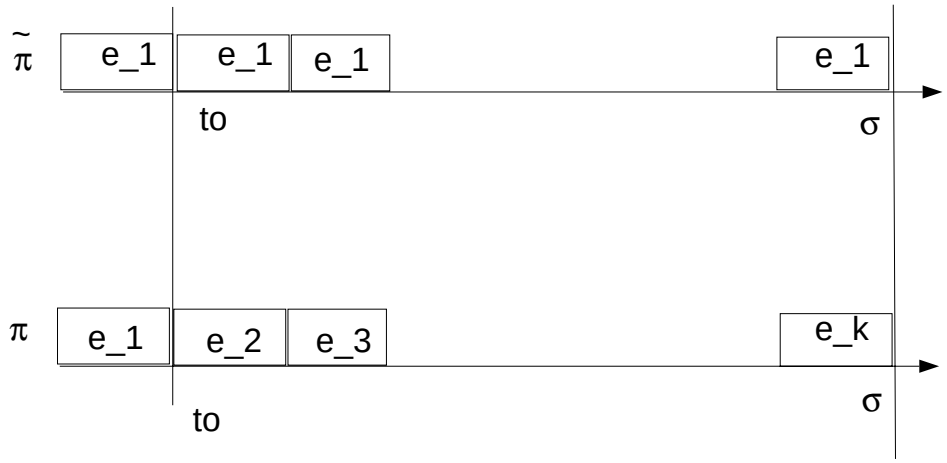

Figure 3.2: System evolution under MDQF ($\tilde{\pi}$) and EB/FS (π) policies

The state of the system is $z(t) = (E(t), A(t), d)$ where $A(t)$ is the set of arrivals up to time t and d is the deterministic, fixed deadline. We denote by $\{L_t^{\pi}(z), t \geq 0\}$ the process of the number of messages lost by time t when applying the policy π.

In [1] the following result was obtained : $L^{STE} \leq_{st} L^{\pi}$ for all policies π in the class of non-preemptive and non-idling policies, under the assumption of exponential distribution of service times. The proof is based on the construction of a policy $\tilde{\pi}$ that improves over π at one decision instant (t_0)(by choosing the STE message). Two coupled processes are then constructed, $(L_t^{\tilde{\pi}}, \overline{L}_t^{\pi})$ such that $(L_t^{\tilde{\pi}} \leq \overline{L}_t^{\pi})$ a.s. for $t \geq t_0$. The same construction is repeated for a number of decision points along $\tilde{\pi}$'s trajectory. We use the same line of reasoning as in [1] but we modify the proof in [1] to reflect the EB/FS policy as π; we consider further approximations needed to avoid the cases where the exponential distribution of service times was used in the proof in [1].

We start from t_0 where $E(t_0) = \{e_1, \ldots, e_n\}$ and we proceed with the construction of $\tilde{\pi}$. At time instant t_0, $\tilde{\pi}$ chooses to serve e_1 and π serves e_2. At the next time instant $\tilde{\pi}$ schedules e_1 (from the most delayed queue 1) and π served e_3. The service under $\tilde{\pi}$ ends at σ, when e_1 is exhausted. We consider σ the next decision instant for both policies, i.e. we stop the service under π at σ as well. Based on the value of σ three cases are possible.

- Case 1: $\sigma \geq e_2$

  In this case, under $\tilde{\pi}$ all packets eligible at t_0, with extinction times less than or equal to σ are lost, except e_1. Under π the same messages will be lost. e_1 is lost under π because $\sigma \geq e_2 \geq e_1$. $E(\sigma)$ is identical under both policies. The states are matched at σ and in $[\sigma, \infty)$ we let $\tilde{\pi}$ follows π. If $t \in [e_1, e_2)$ the policy π looses e_1 by choosing not to schedule it.

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}}, \quad t \in [t_0, e_1) \bigcup [e_2, \infty)$$

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}} + 1, \quad t \in [e_1, e_2)$$

- Case 2: $\sigma < e_1$

  We examine the situation at time σ. Under policy π, $e_1 \in E(\sigma)$, as well as $\{e_2, ...e_k\}$ if they were not exhausted by the service received in $[t_0, \sigma)$, i.e. the messages $\{e_2, ...e_k\}$ contained more than one Equal Share number of packets (reasonable assumption).

  Under policy $\tilde{\pi}$ the extinction set $E(\sigma, \tilde{\pi}) = \{e_2, e_3, \ldots, e_n\}$. Let $\tilde{\pi}$ follows π except that it schedules $\{e_2, e_2, \ldots, e_k\}$ when π schedules $\{e_1, e_2, \ldots, e_k\}$.

  We let the time evolve until π arrives at e_1 and denote τ this moment.

If e_1 meets its deadline under π, e_2 will be scheduled by $\tilde{\pi}$ at τ and the states are matched at τ. Letting $\tilde{\pi}$ follow π in $[\tau, \infty)$ we have $L(\pi, t) = L(\tilde{\pi}, t)$ for all $t \in [\sigma, \infty)$.

If e_1 expired when π arrived at it, we have $\tau \geq e_1$. If $\tau \geq e_2$ then e_2 is lost under $\tilde{\pi}$. Again by letting $\tilde{\pi}$ follow π in $[\tau, \infty)$ we have:

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}}, \quad t \in [\sigma, e_1) \bigcup [e_2, \infty)$$

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}} + 1, \quad t \in [e_1, e_2)$$

If $\tau < e_2$, $E(\tau, \tilde{\pi})$, the set of extinction times under $\tilde{\pi}$ is given by the set $\{e_2, \ldots, e_k, \ldots\}$. The same set is also available to π and again the states are matched at τ.

There is one distinctive sub-case here. When visiting the queue corresponding to e_1, the head-of-the-line message in this queue was expired but is possible that new packets arrived at this queue. If this is the case, the EB/FS changes under π. Let denote this new arrival by $e_1^{'}$. The situation is depicted in figure 3.3

In order to have the same end time for the service under both policies we need the following approximation. If we denote by $C/(n-1)$ the equal share of packets when the service starts at τ and by m the residual number of equal shares when $e_1^{'}$ arrives, we consider $m\frac{C}{n-1} \simeq (m+1)\frac{C}{n}$, which holds for a big number of current active users. With this approximation the service under both policies ends at the same time denoted σ_1.

The following cases are possible.

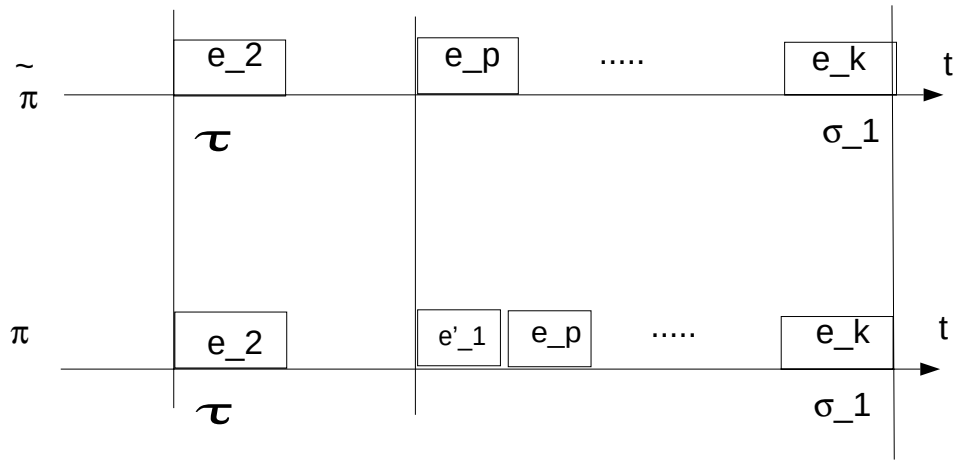

Figure 3.3: Case 2: new arrivals at queue 1 under EB/FS

If $\sigma_1 \geq e_1^{'}$, $e_1^{'}$ is lost under $\tilde{\pi}$ and the states are matched at σ_1.

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}}, \quad t \in [t_0, e_1) \bigcup [e_1^{'}, \infty)$$

$$\overline{L}_t^{\pi} = L_t^{\tilde{\pi}} + 1, \quad t \in [e_1, e_1^{'})$$

If $\sigma_1 < e_1^{'}$ we let $\tilde{\pi}$ follow π and we are back to a situation previously described.

- Case 3: $e_1 \leq \sigma < e_2$ For $t \in [t_o, e_1)$ we have $\overline{L}_t^{\pi} = L_t^{\tilde{\pi}}$ and for $t \in [e_1, \sigma)$ $\overline{L}_t^{\pi} = L_t^{\tilde{\pi}} + 1$. At time σ, under $\tilde{\pi}$, e_2 is eligible for service in addition to all messages that are eligible under π. Consequently, we can proceed as in case 2.

The analysis in Case 1,2 and 3 implies:

$$\overline{L}_t^{\pi} \geq L_t^{\tilde{\pi}}, \quad \forall t \in [t_0, \infty) \tag{3.30}$$

The observation that the processes $\{\overline{L}^{\pi}(t), t \geq t_0\}$ and $\{L^{\pi}(t), t \geq t_0\}$ have the same low, enable us to make statements about the optimality in the sense of stochastic order, which is strictly stronger than that in the sense of expected values. We present next a set of equivalent characterizations of the notion of stochastic order ([1]):

Theorem 1

Let $X = \{X(t), t \in \Lambda\}$ and $Y = \{Y(t), t \in \Lambda\}$ be two processes, where $\Lambda \subset \mathcal{R}$. Let $D = D_{\mathcal{R}}[0, \infty)$, the space of right continuous functions from $\mathcal{R}^+$ to $\mathcal{R}$ with left limits at all $t \in [0, \infty)$ be the space of their sample paths. We say that the process X is stochastically smaller than the process Y, and write $X \leq_{st} Y$ if $P[f(X) > z] \leq P[f(Y) > z], \quad \forall z \in \mathcal{R}$, where $f : D \to \mathcal{R}$ is measurable and $f(x) \leq f(y)$, whenever $x, y \in D$ and $x(t) \leq y(t), \quad \forall t \in \Lambda$.

The following equivalence will be used to stochastic order relations:

- 1) $X \leq_{st} Y$
- 2) $P(g[X(t_1), \ldots, X(t_n)] > z) \leq P(g[Y(t_1), \ldots, Y(t_n)] > z)$
  for all $(t_1, \ldots, t_n), z, n, g : \mathcal{R}^n \to \mathcal{R}$, measurable and such that $x_j \leq y_j, 1 \leq j \leq n \Rightarrow g(x_1, \ldots, x_n) \leq g(y_1, \ldots, y_n)$.
- 3) There exists two stochastic processes $\overline{X} = \{\overline{X}(t), , t \in \Lambda\}$ and $\overline{Y} = \{\overline{Y}(t), , t \in \Lambda\}$ on a common probability space such that
  $\mathcal{L}(\mathcal{X}) = \mathcal{L}(\overline{\mathcal{X}}), \mathcal{L}(\mathcal{Y}) = \mathcal{L}(\overline{\mathcal{Y}})$ and $\overline{X}(t) \leq \overline{Y}(t), \ \forall t \in \Lambda \ \ a.s.$, where $\mathcal{L}(\cdot)$ denotes the law of a process on the space of its sample paths.

From 1) and 3) in Theorem 1 and from (3.30) we conclude that:

$$\overline{L}_t^{\pi} \geq_{st} L_t^{\tilde{\pi}}, \quad \forall t \in [t_0, \infty)$$

By repeating the same construction n times we have a policy that schedules according to the MDQF-rule at these n decision points along its trajectory and satisfies:

$$L^{\tilde{\pi}_n} \leq_{st} L^{\tilde{\pi}_{n-1}} \leq_{st} \cdots \leq_{st} L^{\pi} \tag{3.31}$$

Consider the time instants $t_i,\ 1 \leq i \leq k$ and $g : \mathcal{R}^n \to \mathcal{R}$ as in Theorem 1. Consider also the policy $\tilde{\pi}_{t_k}$ previously defined. Let us denote by L_0 and $L^{\tilde{\pi}_{t_k}}$ the process of the number of packet lost under policies MDQF and $\tilde{\pi}_{t_k}$ respectively. The variables $L^{\tilde{\pi}_{t_k}}(t_1), \ldots, L^{\tilde{\pi}_{t_k}}(t_k)$ have the same joint probability distributions with the variables $L_0(t_1), \ldots, L_0(t_k)$. Hence, for all z, we have:

$$P(g[L^{\tilde{\pi}_{t_k}}(t_1), \ldots, L^{\tilde{\pi}_{t_k}}(t_k)] > z) = P(g[L_0(t_1), \ldots, L_0(t_k)] > z) \tag{3.32}$$

From (3.31), $L^{\tilde{\pi}_{t_k}}(t) \leq_{st} L^{\pi}(t),\ t \geq 0$, therefore, we have

$$P(g[L^{\tilde{\pi}_{t_k}}(t_1), \ldots, L^{\tilde{\pi}_{t_k}}(t_k)] > z) \leq P(g[L^{\pi}(t_1), \ldots, L^{\pi}(t_k)] > z) \tag{3.33}$$

From equations (3.32), (3.33) and Theorem 1, we conclude:

$$L^{MDQF} \leq_{st} L^{\pi} \tag{3.34}$$

From the above result of the MDQF optimality in presence of hard deadline constraints, one can consider the MDQF policy effective for applications with hard deadlines since it discard the packets having missed their deadline, which would waste a fraction of the NOC bandwidth if they were serviced. However, not all applications are hard. Those packets which have "soft" deadlines can also be kept even if they miss their deadlines. For the NOC-scheduler the deadline, or the queuing delay, is used to select the first queue to be serviced and so, the deadline semantic can be different from extinction-time.

3.5 Scheduling policies and TCP effects

In this section we use a quantitative estimate of the effective efficiency for the NOC-satellite gateway and we will show that the Most Delayed Queue First (MDQF) scheduling strategy improves over the Fair Share (FS) and Equal bandwidth (EB) by desynchronizing the TCP congestion windows. We assume a fixed number of file (HTML page) transfers from Internet Servers to the satellite gateway (SGW) in a congested scenario, when due to buffer overflows and retransmissions, the data transfers under-utilize the NOC-SGW capacity C [packets], i.e the data transfers will fill only a fraction $\rho \cdot C,\ \rho \leq 1$ of the service facility capacity. The estimate of the effective efficiency ρ is based on how congestion windows evolve in TCP when a particular scheduling policy is in use at the SGW. This investigation extends earlier results on the calculation of TCP effects ([8]) and TCP congestion windows synchronization ([19]), to the case of interactions between TCP and the bottleneck scheduling policies.

3.5.1 TCP and window/population dynamics

In this section we provide a quick overview of the TCP congestion control algorithm and then we describe the synchronization phenomenon of window adjustment. At TCP connection setup, the receiver specifies a maximum window size *maxwnd*. The sender has a variable called the congestion window *cwnd* which is increased when new data is acknowledged and is decreased when a packet drop is detected. The actual window used by the sender is

$$wnd = \lfloor MIN(cwnd, maxwnd) \rfloor$$

The congestion window adjustment algorithm has two phases, the slow-start where the window is increased rapidly and the congestion-avoidance where the window is increased more slowly. A connection can be in one phase or another, depending on a control threshold *ssthresh*. The next figure indicates how the window evolves during congestion and the corresponding number of packets (population).

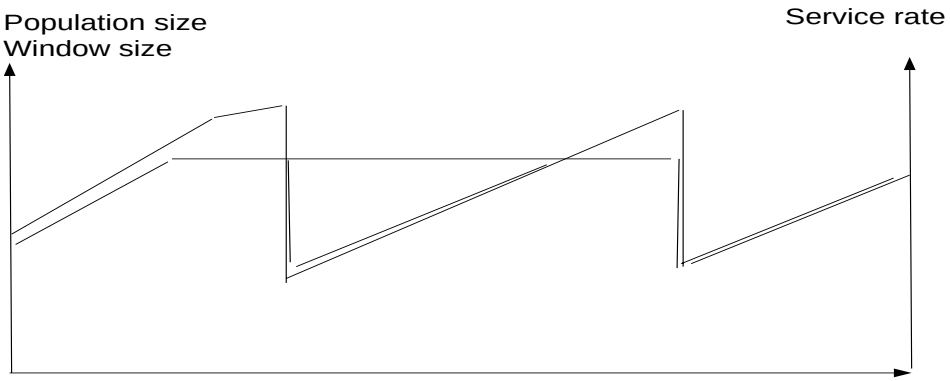

Figure 3.4: Service rate, population and window size evolution

An important phenomenon detected ([19]) is the synchronization of window adjustment, i.e. the windows of different TCP connections competing for a congested resource turn out to be in phase. The population of packets is in proportion to the window of any connection. The service that a connection receives equals its window divided by the round-trip time. In case of multiple TCP connections the short periods of population collapse are synchronized and the resource spend much of the time at less than full capacity. If this in-phase window adjustment can be desynchronized we will have a lower inefficiency by increasing the population of packets at any time. The rest of the subsections explains the influence of MDQF scheduling policy in increasing the effective

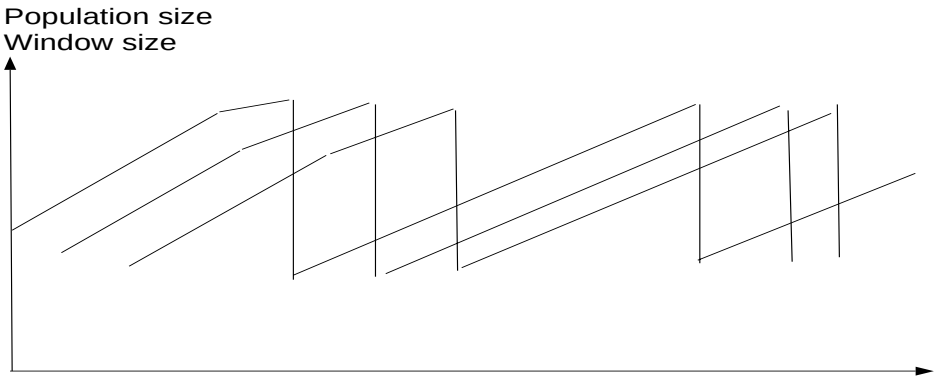

Figure 3.5: Out of phase window adjustment

efficiency of the system.

The following notations are used in the next subsections:

- W_k = current window of connection k
- $P = \sum W_k$ = total outstanding packet population
- RTT = round-trip time (IS to NOC) with empty connection queue
- τ = work time of the server for one data unit(packet)

3.5.2 Queue dynamics under scheduling policies

We assumed a fixed number N of file transfers from Internet Servers to NOC-SGW. The transfers are modeled as ON-OFF periods and, during a file transfer a constant amount c packets are presented to the service facility at each time instant of the ON (active) period of a source. The SGW capacity is denoted C [packets], i.e. the NOC scheduler can put a maximum of C packets on the transmission line (satellite link) at one time instant. Each transfer proceeds over

its own TCP connection and the corresponding buffer at SGW has a length of B packets. Furthermore, we assume a congested scenario were the number of sources send c packets at each time instant during their active periods, exceeding the capacity C: $c \geq \frac{C}{N}$. In addition, the round-trip times (RTT) of all IS-to-NOC connections is assumed to be the same. Under these assumptions we have:
Fact: Under the Fair Share and Equal Bandwidth scheduling policy all the active connections will reach their buffer capacity at the same time instant k, where:

$$kR_{inc} = B$$

with R_{inc} the increase rate of queues length, measured as $c - \frac{C}{N}$ packets per service cycle.

Proof: The analysis is carried during the last part of the "congestion-avoidance" phase when the bottleneck has growing queues on all active connections. When using the FS/EB policy at each service time instant the queue length is increasing with $c - \frac{C}{N}$ packets. The buffer capacity will be reached simultaneously at a time instant k where $k(c - \frac{C}{N}) = B$.

The main implication of this result is the synchronization of TCP congestion windows assuming the same RTT for all connections. At $k + RTT$ all *cwnd* will be halving and the de-population from the peak population value P_{high} will be

$$P_{low} = \frac{1}{2^h} P_{high}$$

where h is an average estimate of the number of packets lost per connection.

Next we consider the MDQF policy. The queues are indexed in increasing order of the Head-of-the-Queue delays. Let n_p be a time instant of service initiation

for queue 1. The queue 1 will evolve according to the following equation:

$$Q^1_{n_p+k} = Q^1_{n_p+k-1} - r^1_{n_p} + k \cdot c$$

until the next time instant of service initiation.

The next queue in the set of ordered queues will evolve as follows:

$$Q^2_{n_p+1} = Q^2_{n_p} + c$$

$$Q^2_{n_p+2} = Q^2_{n_p+1} - r^2_{n_p+1} + c$$

$$Q^2_{n_p+m} = Q^2_{n_p+m-1} - r^2_{n_p+1} + m \cdot c$$

Replacing the amount of service allocated $r_{n_k} = min(C, Q_{n_k})$, where n_k are the time instants of service initiation of an active queue, we obtain a lower bound of c packets for the separation in queue lengths. In a previous analysis of MDQF policy (section 3.3) we obtained sufficient conditions for MDQF gain over FS/EB in terms of the length of a service cycle. Based on these conditions, the upper bound for the separation in queues length is given by $max(C, (N-1) \cdot c)$ [packets]. We summarize these observations in the following result:

Fact: Under MDQF policy the active queues will reach their buffer capacity at different times. The time-distance d between the moments when any two queues reach their allocated buffer is bounded by:

$$t_c \leq d \leq t_C \tag{3.35}$$

where t_c and t_C are the transmission times of c and $max(C, (N-1) \cdot c)$ packets respectively.

Based on this result we will compute the effective efficiency at the SGW when MDQF is used and we will quantify the gain over FS/EB schemes.

3.5.3 Effective efficiency estimation

To estimate the service being delivered by the SGW the approach in [8] was to estimate the evolution of population P over the congestion-avoidance phase. The peak of the population P_{high} is the sum of the packets in queues and the packets that received service in the last RTT time units. When using FS/EB policy the population size when the congestion-avoidance phase begins is $P_{low} = 2^{-h}P_{high}$ since all the windows are cut at the same time and by the same factor. The de-population factor h is estimated by $h = l \cdot d$ where l is the location-influenced average number of collisions created per connection and d is the average damage done by a collision in the form of lost and damaged packets. The combined effect on efficiency is computed in [8] as:

$$\rho = 1 - \frac{((1 - p_{low})^+)^2}{2(p_{high} - p_{low})} \tag{3.36}$$

where p_{high} and p_{low} are the normalized populations:

$$p_{high} = \frac{P_{high}}{RTT/\tau} = b + 1, \quad p_{low} = \frac{P_{low}}{RTT/\tau} = 2^{-ld}(b + 1), \; b = \frac{B\tau}{RTT}$$

Under MDQF policy we have

$$P_{low}^{MDQF} = \frac{1}{2}W_{high} + (\frac{1}{2}W_{high} + \frac{d_1}{RTT} + 1) + \cdots + (\frac{1}{2}W_{high} + \frac{d_N}{RTT} + 1) \tag{3.37}$$

where d_k is the time-distance between the moments when the most delayed queue and the queue k (numbered according to the Head-of-the-Queue delays) reach their buffer capacity. The difference in queue length between the most delayed queue and queue k is $k \cdot c$ packets. Therefore, the time-distance d_k refers to the transmission time of these $k \cdot c$ packets and is given by:

$$d_k = \frac{k \cdot c}{C}, \; k \in [1, N - 1] \tag{3.38}$$

Using (3.38) in (3.37) we obtain:

$$P_{low}^{MDQF} = P_{low}^{FS} + \frac{N(N+1)\cdot c}{2\cdot C\cdot RTT} + N$$

where $P_{low}^{FS} = P_{low}^{EB} = \sum \frac{W_{high}}{2}$. By replacing P_{low}^{MDQF} in the formula for effective efficiency (3.36) we obtain the MDQF gain over the EB/FS policies:

$$\frac{((1-p_{low})^{+})^2}{2(p_{high}-p_{low})} - \frac{((1-p_{low}^{MDQF})^{+})^2}{2(p_{high}-p_{low}^{MDQF})} \tag{3.39}$$

Chapter 4

Experiments with Bandwidth Allocation Policies

Recent measurements of network traffic have shown that self-similarity is a phenomena present in wide-area traffic traces [2]. In this chapter we investigate the impact of long-range dependency, peculiar to self-similar stochastic processes, on hybrid Internet networks, including its effect in throughput and delay. This is done in the context of various bandwidth allocation mechanisms using simulations.

First we study the effect of ON-OFF "heavy-tailed" source model on performance when the Equal Bandwidth (EB), Fair Share (FS) and the Most Delayed Queue First (MDQF) schemes are employed at the Network Operations Center (NOC) of a hybrid Internet network. We find that MDQF performs better than the other bandwidth allocation strategies in congested scenarios. An interesting phenomena of "delay shifting" shows the "cooperative" work among the sources in MDQF scheme. The degree of "delay shifting" is controlled by buffer allocation and a "virtual delay" mechanism imposed on the '"greedy" source.

Second we consider a more robust self-similar traffic model, the Fractional

Brownian Traffic (FBT) [13], [14] . The ON-OFF model can generate self-similar traffic only when a "large" number of ON-OFF "heavy-tailed" (Pareto) distributed sources are aggregated [22]. This asymptotic result is not easily applicable in a simulation context. This is one of the reasons we choose the FBT model.

In addition the ON-OFF model fails to capture at least one important characteristics of WWW traffic: the model assumes constant rate during the transmission, whereas in reality, the transmission rate depends on the congestion status of the network. The ability to capture, to some degree, rate fluctuations in FBT model is a considerable improvement over the previous aggregated model. The hybrid Internet configuration for FBT experiments is presented in section 4.2. Similar findings of the "delay-shifting" phenomena and the effectiveness of the buffer allocation and the "virtual delay" solutions are presented in the context of FBT simulations. The rest of this chapter present the simulation results.

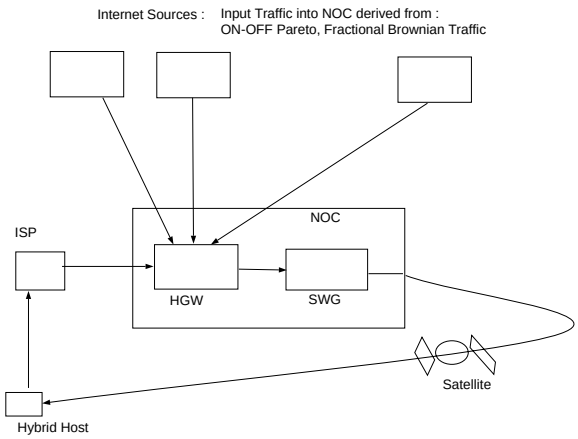

Figure 4.1: Simulation System Configuration

4.1 Experiments with ON/OFF Pareto distributed sources models

In this section we motivate the choice of the ON-OFF Pareto distributed source traffic model, we briefly describe its relation to self-similar traffic and we present simulation results for the investigated bandwidth allocation policies.

In the context of interactive users in the hybrid Internet network configuration, the dominant application is WWW. We consider Web-user-like transfers where the source alternates between "transfer" (ON) periods followed by "think" (OFF) periods. Consequently, we use an ON-OFF source traffic model. Within the scope of the NOC queuing system the ON state starts when a file (a HTML page for example) requested by a user is transmitted from an Internet Server (IS) and arrives at NOC. When the transfer completes the source enters the OFF period. Evidence of the self-similarity in WWW traffic was reported in [2] where it was shown that self-similarity in such traffic can be explained based on the underlying distribution of WWW document sizes, user preference in file transfer and the effect of "think time". Empirically measured distributions from client traces and WWW servers have shown heavy-tailed transmission times (ON) and OFF times.

Next, we expose the mathematical formulation of heavy-tailed distribution and its relationship with self-similarity.

A distribution is heavy-tailed if

$$P[X > x] \sim x^{-\alpha}, \quad as \quad x \to \infty \quad and \quad 0 < \alpha < 2 \tag{4.1}$$

The simplest heavy-tailed distribution is the Pareto distribution. From equation (4.1) we see that if the asymptotic shape of the distribution is hyperbolic, it

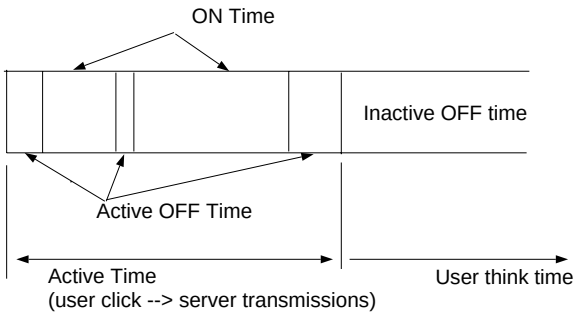

Figure 4.2: WWW applications: ON-OFF times

is heavy-tailed. The Pareto distribution is hyperbolic over its entire range; its density function is

$$f(x) = \alpha k^{\alpha} x^{-\alpha-1} \; ; \; \alpha, \; k > 0; \; x \geq k$$

and its distribution function is given by

$$F(X) = P(X \leq x) = 1 - (k/x)^{\alpha}$$

Heavy-tailed distributions have a number of properties depending on the parameter α. If $\alpha \leq 2$, then the distribution has infinite variance; if $\alpha \leq 1$, the distribution has infinite mean. As α decreases, an arbitrarily large portion of the probability mass may be present in the tail of the distribution.

A self-similar process may be constructed by superimposing many simple renewal reward processes, in which the inter-renewal times are heavy-tailed. To visualize this construction, consider a large number of processes that are each either ON or OFF. The ON and OFF periods for each process strictly alternate and the distribution of ON and/or OFF period is heavy-tailed. This model corresponds to a network of Internet Servers sending data to the NOC at constant rate during their ON periods and being silent during the OFF periods. For this model it

has been shown that the result of aggregating many such sources is a self-similar process. For the mathematical formulation of this result we need to introduce several definitions ([21]).

Self-Similar data traffic

A stochastic process $X(t)$ is statistically self-similar with Hurst parameter $H \in [0.5, 1]$ if, for any real $a > 0$, the process $a^{-H}X(at)$ has the same statistical properties as $X(t)$. The parameter H (Hurst parameter) is a measure of persistence of a statistical phenomenon. H increases from a value of 0.5 which indicates the absence of self-similarity and the closer H is to 1 the greater the degree of persistence (or long-range dependence).

Long-range Dependence

One of the most significant properties of self-similar processes is referred to as long-range dependence. This property is defined in terms of the behavior of the autocovariance $C(\tau)$.

A long-range dependent process has a hyperbolically decaying autocovariance:

$$C(\tau) \sim |\tau|^{-\alpha}, \quad as \quad |\tau| \to \infty, \quad 0 < \alpha < 1$$

where the parameter α is related to the Hurst parameter by $H = 1-(\alpha/2)$. Long-range dependence reflects the persistence phenomenon in self-similar processes, i.e. the existence of clustering and bursty characteristics at all time scales.

The self-similarity of the aggregated ON-OFF Pareto process was established in [22] and it can be stated as follows:

Fact ([22]): The superposition of many ON-OFF sources with strictly alternating ON- and OFF-periods, whose ON-periods or OFF-periods exhibit the *Noah effect* (i.e. high variability or infinite variance) can produce aggregate network traffic that exhibits the *Joseph effect* (i.e. is self-similar or long-range dependent).

In this section, we pursue our experiments with this model, i.e. ON-OFF Pareto distributed source traffic models, in order to test the behavior of bandwidth scheduling techniques at NOC. We consider next the parameters used in simulation.

4.1.1 Description of experiments and simulation results

The simulation configuration is as follows. For the Pareto distribution of ON-OFF sources we use a value of 0.001 for the parameter k, which specifies the minimum value that the random variable can take. The "burst" length distribution has a heavy-tailed distribution with parameter $\alpha = a_{ON} = 1.99$. At the end of burst generation, the source becomes silent for a random time, Pareto distributed, which, in average, is greater than the length of data traffic generation interval, i.e. we choose as a rule for the OFF periods a parameter $\alpha = a_{OFF} = 1.005 < a_{ON}$, which imply an average length of the OFF period greater than the average length of the ON period. The rest of the simulation parameters are listed in the next table.

Pareto model	**k=0.001** $a_{ON} = 1.99$ $a_{OFF} = 1.005$
Buffer per Connection	**1000 packets**
Number of Connections	**5 connections**
Constant Arrival Rate	**10 packets / unit time**

Table 4.1: Simulation Configuration

Bandwidth	Equal Bandwidth	Fair Share	MDQF
20	**1.16**	**1.12**	**0.87**
15	**8.90**	**8.88**	**6.30**
10	**21.36**	**21.24**	**19.72**

Table 4.2: On/Off Pareto sources : Average Delays

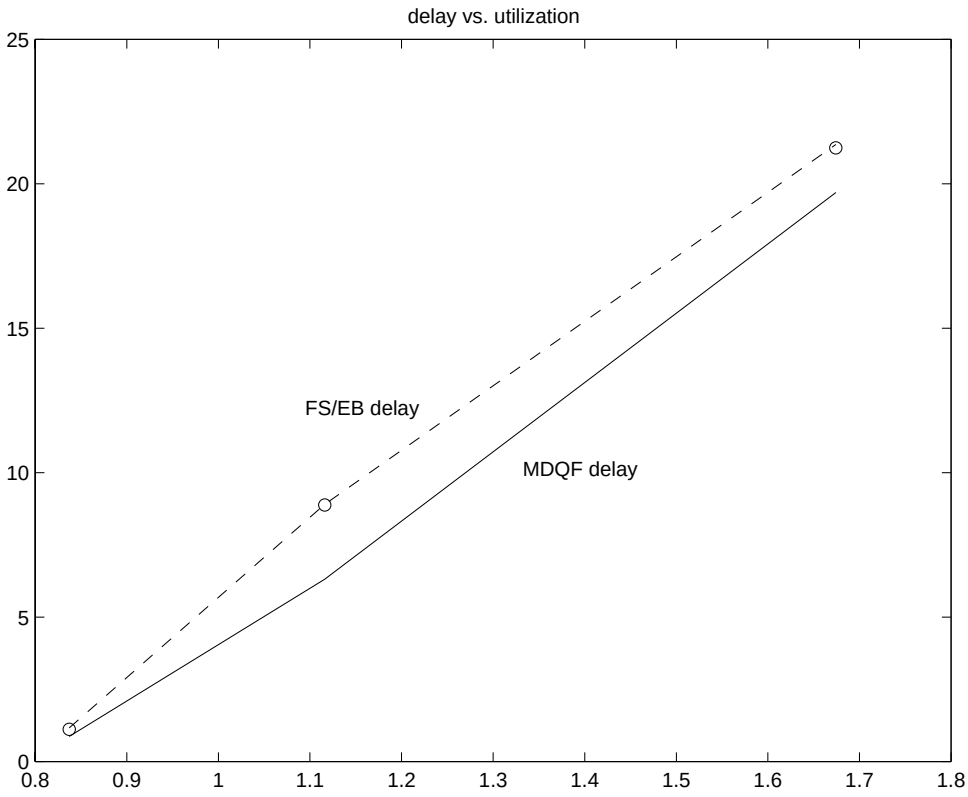

Figure 4.3: Delay vs. Utilization: ON/OFF Pareto sources

4.1.2 Making Greed Sources work with MDQF policy

Under a basic premise that users are independent and selfish, the delays of packets in queues depend crucially on the service discipline implemented at NOC. When the Equal Bandwidth and / or Fair Share policies are used we have bandwidth firewalls which protect well-behaved sources from the greedy ones. However, the limitation imposed by these service disciplines cause large delays and buffer overflows for the sources with input rates bigger than their equal and/or fair share.

Our challenge is to design a service discipline, so that the system exhibits good performance in spite of the selfishness of individual users. In order to achieve this goal the MDQF service discipline employs a "cooperative" scheme where the sources are served based on their Head-of-Queue(HoQ)-delays, without building bandwidth share firewalls among competing connections. While this policy was proven to be optimal in the sense of minimizing the number of packets lost due to ACK delay threshold, it is possible that the well-behaved sources will suffer a bigger delay performance degradation than acceptable for a particular user.

In MDQF, the greedy source will improve its delay performances over EB/FS at the expense of performance degradation of well-behaved users. We call this phenomena "delay shifting" because the delay is shifting from the greed source to the rest of sources. To illustrate the "delay shifting" we consider the case of a greedy source (source 1) which sends at a rate of 20 packets/unit time during its active periods. The rest of the sources are sending at 10 packets/unit time. The available bandwidth is 15 packets.

We present next two simple solution to the problem of "delay-shifting".

Connection	Buffer	Rate	Average Delay	Rate	Average Delay
1	**1000**	**10**	**8.69**	**20**	**12.04**
2	**1000**	**10**	**5.59**	**10**	**7.81**
3	**1000**	**10**	**3.73**	**10**	**4.07**
4	**1000**	**10**	**7.34**	**10**	**11.05**
5	**1000**	**10**	**6.21**	**10**	**8.78**

Table 4.3: "Delay Shifting": MDQF / Average Delays

Influence of Buffer Allocation

The first solution imposes a bound on the buffer allocated to a greedy source and this translates in bounding the performance degradation of well-behaved sources, as illustrated in the next set of simulation results.

Remark It may appear that the hard bound in the buffer allocation will cause a too large number of packet losses for the source 1 in the above example. However, the congestion control mechanism of the underlying transport protocol will limit this losses by reducing the sending rate after the first series of buffer overflows.

Influence of "Virtual Delay"

Instead of limiting the buffer allocated to a greedy source we approach the problem of 'delay-shifting' from a different angle. If we consider the queuing systems as allocating three quantities: bandwidth, promptness and buffer space, the previous solution can be thought as an interplay between bandwidth and

Connection	Buffer	Average Delay
1	**500**	**9.33**
2	**1000**	**6.50**
3	**1000**	**4.07**
4	**1000**	**8.39**
5	**1000**	**7.00**

Table 4.4: Influence of Buffer Allocation: MDQF / Average Delays

buffer space. We now look at the role played by a promptness factor ("virtual delay") into the MDQF bandwidth allocation policy. Promptness allocation (when packets get transmitted) is based on the data already available at the NOC. The MDQF policy tend to favor (more promptness, less delay) the user who utilize more bandwidth. The separation, in a limited sense, of the promptness allocation from the bandwidth allocation, can be accomplished by introducing a non-negative parameter δ ("virtual delay") and time-stamping the greedy source with the actual arrival time plus this "virtual delay". The rest of the parameters are unchanged. The MDQF will exercise its control based on the new time-stamps and will penalize the greedy source reducing its allocated bandwidth. The role of this "virtual delay" can be seen more clearly from the next simulation results.

When greed work is present on the network the MDQF can be enhanced with two simple solutions (buffer allocation bound and "virtual delay") in order to be able to provide low delay to low throughput sources. This is one important

Connection	Virtual Delay(penalty)	**MDQF Average Delay**
1	**20**	**4.75**
2	**0**	**3.31**
3	**0**	**3.11**
4	**0**	**5.08**
5	**0**	**3.89**

Table 4.5: Influence of "Virtual delay" imposed on greedy source : MDQF / Average Delays

feature of MDQF algorithm.

4.2 Experiments with Fractional Brownian Traffic

4.2.1 Fractional Brownian Traffic generation

The Fractional Brownian Traffic (FBT) model was introduced in [13]. It has the advantage of being a parsimonious model that capture the "self-similar" nature of the aggregated connection-less Internet traffic. The abstract model of FBT has the ability to capture rate fluctuations which is a considerable improvement over the previous ON-OFF model, where only the aggregation of a large number of ON-OFF sources with a Pareto distributed activity and idle period yields a limiting behavior identical to M/G/∞ input stream ([11], [17]).

In our hybrid Internet configuration we consider that a large number of Internet Servers (IS) is sending data to the NOC. As a result of numerous interactions with the network, this traffic exhibits fractal characteristics, i.e. it is network-level fractal traffic. The NOC is receiving this traffic into a limited number of queues (without preserving a one-to-one TCP connection - NOC queue mapping) and is employing various bandwidth allocation schemes. The input traffic presented to the NOC queues is generated according to the following Fractional Brownian Traffic (FBT) process [13]:

$$A_t = mt + \sqrt{am}Z_t \tag{4.2}$$

where Z_t is a normalized Fractional Brownian motion (FBM) process with Hurst parameter $H \in [\frac{1}{2}, 1)$ and is characterized by the following properties:

1 Z_t has stationary increments;

2 $Z_0 = 0$ and $E[Z_t] = 0$ for all t;

3 $E[Z_t]^2 = |t|^{2H}$ for all t;

4 Z_t has continuous paths;

5 Z_t is Gaussian.

The parameter m is the mean input rate and a is a variance coefficient which depends on H and on the burst transmission rate.

To generate Fractional Brownian Traffic we use a Fractional Gaussian Noise (FGN) traffic generator as described in [18]. The FGN process is an exactly (second-order) self-similar process defined as follows:

A Fractional Gaussian Noise $X = (X_k : \ k = 0, 1, 2, \ldots)$ is a stationary Gaussian process with mean $\mu = E[X_k]$, variance $\sigma^2 = E[(X_k - \mu)^2]$, and autocorrelation function

$$r(k) = 1/2(|k+1|^{2H} - |k|^{2H} + |k-1|^{2H}), \ k = 1, 2, \ldots$$

Asymptotically

$$r(k) \sim H(2H-1)|k|^{2H-2}, \ k \to \infty, \ 0 < H < 1$$

One can see that the resulting aggregated processes $X^{(m)}$ have the same distribution as X, $\forall H \in (0, 1)$. We need the following definition: ([21])

Exactly Self Similarity

For a stationary time series x, define the m-aggregated time series $x^{(m)} = \{x_k^{(m)}, k = 0, 1, 2, \ldots\}$ by summing the original time series over non-overlapping, adjacent blocks of size m. This may be expresses as:

$$x_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i$$

A process x is said to be exactly self-similar with parameter β, $0 < \beta < 1$ if, for all $m = 1, 2, \ldots$ we have:

$$Var(x^{(m)}) = \frac{Var(x)}{m^\beta}$$

and the autocorrelation satisfies:

$$R_{x^{(m)}}(k) = R_x(k)$$

Using the exactly self-similarity definition, one can conclude that Fractional Gaussian Noise is exactly second-order self-similar with self-similarity Hurst parameter $1/2 < H < 1$.

The method for synthesizing FGN is based on the Discrete Time Fourier Transform (DTFT) and can be summarized as follows:

Assuming that the power spectrum of the FGN process is known, a sequence of complex numbers z_i, corresponding to this spectrum, is then constructed. The time-domain sample path x_i is obtained from the frequency-domain counterpart z_i, by using an inverse-DTFT.

x_i has, by construction, the power spectrum of FGN and, because autocorrelation and power spectrum form a Fourier pair, x_i is guaranteed to have the autocorrelation properties of an FGN process.

The difficulty of computing the power spectrum of the FGN process is addressed in [18] and is briefly described here.

For on FGN process the power spectrum is

$$f(\lambda; H) = A(\lambda; H)[|\lambda|^{-2H-1} + B(\lambda; H)].$$

for $0 < H < 1$ and $-\pi \leq \lambda \leq \pi$.

$$A(\lambda; H) = 2\ sin(\pi H)\Gamma(2H+1)(1 - cos\lambda)$$

$$B(\lambda; H) = \sum_{j=1}^{\infty}[(2\pi j + \lambda)^{-2H-1} + (2\pi j - \lambda)^{-2H-1}]$$

$$B(\lambda; H) \sim a_1^d + b_1^d + a_2^d + b_2^d + a_3^d + b_3^d + \frac{a_3^{d'} + b_3^{d'} + a_4^{d'} + b_4^{d'}}{8H\pi}$$

where

$$d = -2H - 1 \quad d' = -2H$$

$$a_k = 2k\pi + \lambda \quad b_k = 2k\pi - \lambda$$

A Fractional Gaussian Noise sample path generated with this method is given in the next figure:

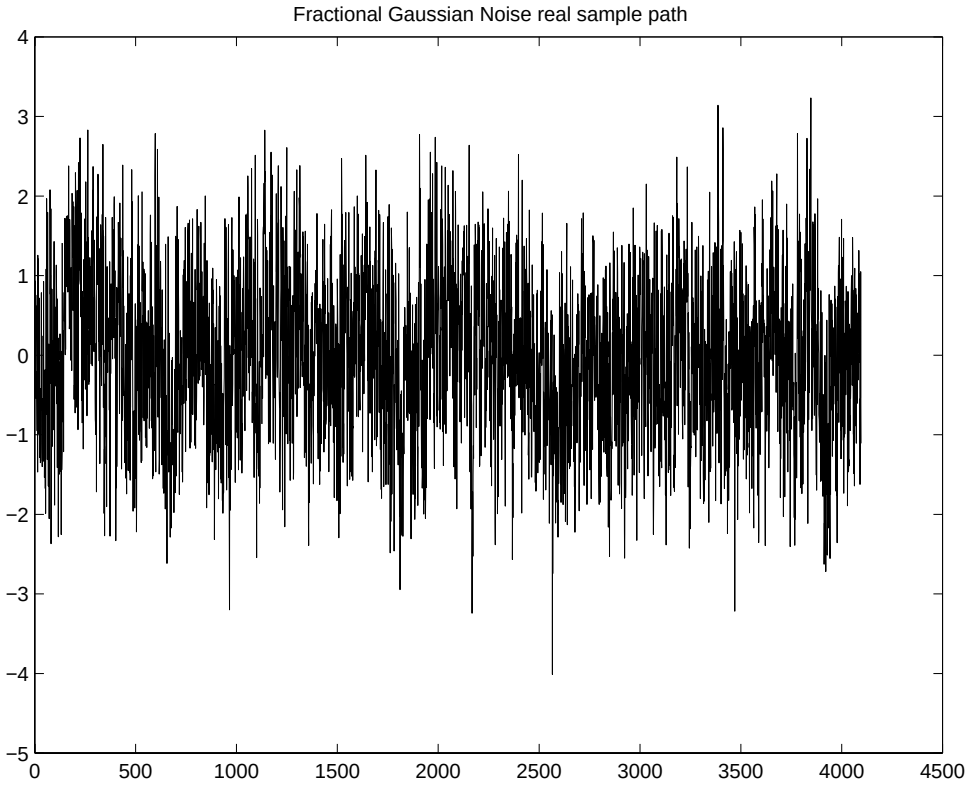

Figure 4.4: Fractional Gaussian Noise sample path: H=0.78

In order to compare the accuracy of the method previously described, we use the FGN sample path to compute the corresponding FBM trace. The FBM process is the sum of the FGN increments. The FBT is next computed with parameter taken from pOct.TL Bellcore traffic data ($thumper.bellcore.com/pub/lantraffic/pOct.TL$).

- $m = 2279$ kbit/sec
- $a = 262.8$ kbit.sec
- $H = 0.76$

These parameter were estimated in [13] and were used to compare the simulated Fractional Brownian Traffic with the true trace from Bellcore. The fractional Brownian motion process synthesis used by Norros was based on a bisection method, different from the one we use here. We present next our simulated Fractional Brownian Traffic and the Bellcore trace.

The similarity is considerable in the profiles of both traces and we consider the accuracy of the traffic generator satisfactory for our purposes.

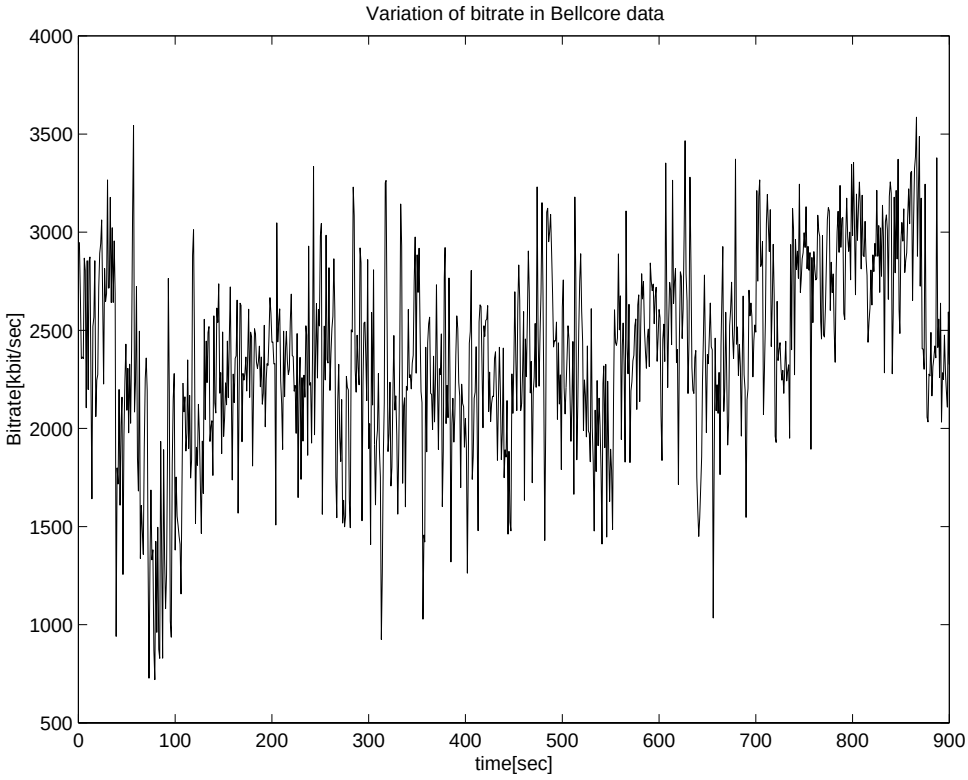

Figure 4.5: Bit-rate variation in Bellcore trace (pOct.TL)

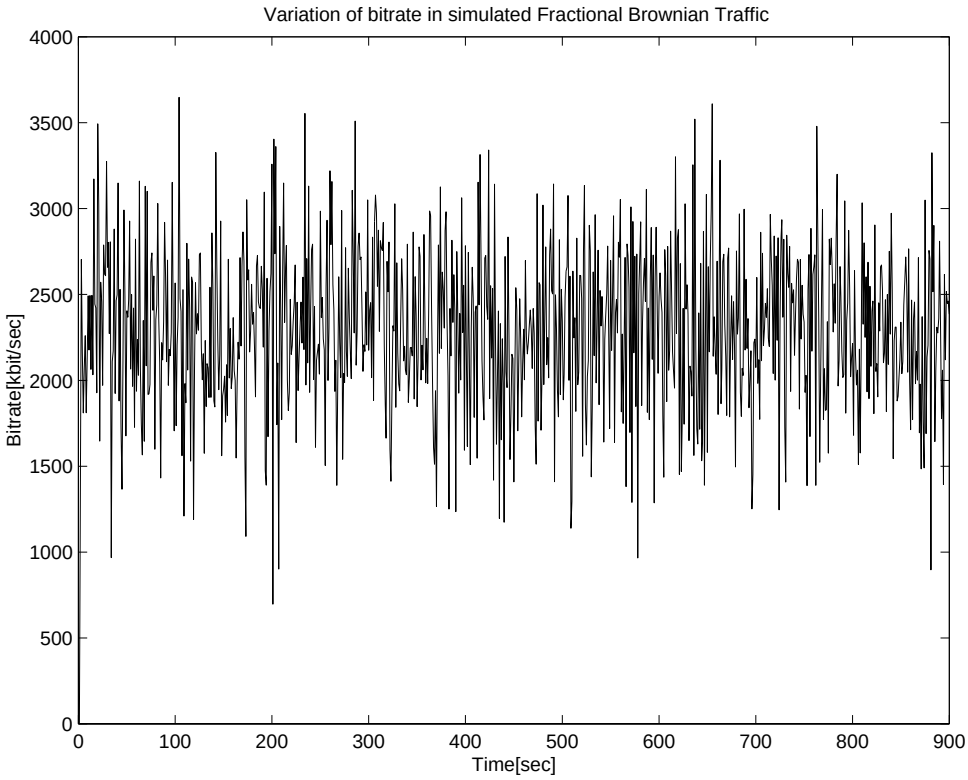

Figure 4.6: Bit-rate variation in simulated Fractional Brownian Traffic

4.2.2 Description of experiments

The experiments with Fractional Brownian Traffic will address the "delay shifting" problem in the MDQF setting. We used the following simulation configuration:

- Buffer per Connection = 500 packets
- Total Bandwidth = 15 packets /unit time
- Number of Connection = 5 connections
- Mean Arrival Rates = 1 2 3 4 5 packets/unit time
- Hurst parameter H = 0.7 0.8 0.9 0.95 0.98

The simulation results summarized next illustrate the presence of "delay shifting".

Connection	**Fair Share**	**MDQF**
1	**0.00**	**3.78**
2	**0.02**	**4.11**
3	**0.07**	**4.20**
4	**6.96**	**4.25**
5	**10.55**	**4.32**

Table 4.6: Fractional Brownian Traffic : Average Delays

One can see clearly the bandwidth firewalls provided by the Fair Share strategy comparing with the "co-operative" work employed by the Most Delayed Queue First policy.

4.2.3 Greed Work: Influence of Buffer Allocation

We repeat the previous experiment with a modified buffer allocation. The "greed" sources (sources 4 and 5 as compared with source 1 which has the smallest rate) are penalized by a buffer reduction of 300 packets and 400 packets respectively. This corresponds to a reduction in the work presented to server by these sources. Consequently, the delays are reduces as follows:

Connection	**Buffer**	**MDQF average delays**
1	**500**	**1.67**
2	**500**	**1.93**
3	**500**	**2.04**
4	**200**	**2.10**
5	**100**	**2.06**

Table 4.7: Buffer allocation for "delay shifting" with FBT: Average Delays

As in the experiments with ON/OFF Pareto sources we used a reduction of the buffer allocation in order to limit the degradation of service quality as perceived by sources with slower input rates.

4.2.4 Influence of "Virtual Delay"imposed on greed work

The next set of simulation results shows the effect of the "virtual delay" scheme in the presence of greed work. We penalize the greedy sources (4 and 5) with a virtual delay of 5 and 10 time-units respectively in order to be able to provide low delay to low throughput sources. This important feature is exemplified next.

Connection	Virtual Delay(penalty)	**MDQF Average Delay**
1	**0**	**2.72**
2	**0**	**2.21**
3	**0**	**2.81**
4	**5**	**2.85**
5	**10**	**2.89**

Table 4.8: "Virtual Delay" imposed on greedy sources with FB Traffic: MDQF/Average Delays

Chapter 5

Conclusions

In the analysis of the MDQF policy we have attempted to address a crucial issue of service quality, as perceived by interactive users in a DBS-based Hybrid Internet architecture, namely minimum latency. The MDQF algorithm distinguishes between users, and allocates bandwidth and buffer space independently. Moreover, the bandwidth allocation is not uniform across users, as in the case of EB policy.

Our calculations showed that the MDQF is able to deliver lower delay to the sources than the EB and FS policies do. The analysis in Chapter 3 showed that, when combined with hard deadline constraints, the MDQF policy delivers the optimal (minimum) delay to the users, improving over the EB and FS policies.

Furthermore, we analyzed the effectiveness of the MDQF policy in the presence of the TCP transport protocol used in DBS-based HSTN. The NOC-scheduler was shown to alleviate the TCP window synchronization phenomenon, when the MDQF algorithm is used. The MDQF gain over EB and FS policies was derived and analytically expressed as a function of the number of active sources, the service facility capacity, the transfer rate from sources, and the

equally assumed Round Trip Time (RTT) of connections.

By using simulations in Chapter 4, we exposed the NOC-scheduler to self-similar traffic from ON-OFF Pareto distributed and Fractional Brownian Traffic data traffic source models. We performed tests that show the improvement of MDQF over the EB and FS.

Moreover, since the MDQF policy does not have built-in firewalls to guarantee good performance in the presence of "greed" work, we presented two solutions to this problem. The first one is to use a buffer allocation policy, in conjunction with the MDQF packet scheduling algorithm, to make the greedy sources work with low throughput sources. The second solution is intrinsically related to the MDQF dynamics. It imposes a "virtual delay" that penalizes the greedy sources. These solutions showed that MDQF is tunable through the "virtual delay" parameter or through a buffer allocation policy, and can protect well-behaved sources.

Our conclusion is that the MDQF policy, when employed at the NOC-scheduler of a DBS-based Hybrid Internet Network, creates an environment where the service quality, as perceived by interactive users, is improved compared with other bandwidth allocation policies.

We hope, in the future, to investigate the performances of MDQF under real load conditions, with background traffic such as video-conferencing or package delivery, interacting with asymmetric TCP/IP protocols, on DBS-based Hybrid Networks. Also, we hope to explore extensions of flow and congestion control algorithms that are more attuned to the properties of the MDQF NOC-scheduler.

Bibliography

[1] P. P. Bhattacharya and A. Ephremides, *Optimal Scheduling with Strict Deadlines*, IEEE Trans. Automatic Control, Vol. 34, No. 7 (July 1989), pp. 721–728.

[2] M.E. Crovella, A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE/ACM Transactions on Networking, Vol. 5, No. 6 (December 1997), pp. 835–846.

[3] A. Demers, S. Keshav, S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, Proc. ACM SIGCOMM, 1989, pp. 3–12.

[4] A.D. Falk, N. Suphasindhu, D. Dillon, J.S. Baras, *A System Design for a Hybrid Network Terminal Using Asymetric TCP/IP to Support Internet Applications*, Proceedings of Technology 2004 Conference, Washington, D.C., 1994.

[5] A.D. Falk, V. Arora, N. Suphasindhu, D. Dillon, J.S. Baras, *Hybrid Internet Access*, Proceedings Conference on NASA Center for Commercial Development of Space, M.S. El-Genk and R.P. Whitten, eds., American Institute of Physics, New York, AIP Conf. Proc. No. 325, 1:69-74, 1995 (See also: CSHCN TR 95-7).

[6] N. R. Figueira and J. Pasquale, *An Upper Bound on Delay for the Virtual-Clock Service Discipline*, IEEE/ACM Transactions on Networking, Vol. 3, No. 4 (August 1995), pp. 399-408.

[7] R. Jain, K. Ramakrishnan, D. Chiu, *Congestion Avoidance in Computer Networks with a Connectionless Network Layer*, DEC-TR-506, 1988.

[8] D.P. Heyman, T.V. Lakshman, A.L. Neidhardt, *A New Method for Analysing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet*, Performance Evaluation Review, Vol. 25, No. 1 (June 1997), pp. 24–38.

[9] S. Keshav, *A Control-Theoretic Approach to Flow Control*, Proc. SIGCOMM '91, September 1991, pp. 3–15.

[10] C. Lefelhocz, B. Lyles, S. Shenker, L. Zhang, *Congestion Control for Best-Effort Service: Why We Need a New Paradigm*, IEEE Network, Vol. 10, No. 1 (January/February 1996), pp. 10–19.

[11] N. Likhanov, B. Tsybakov, N.D. Georganas, *Analysis of an ATM Buffer with Self-Similar ("Fractal") Input Traffic*, Proc. IEEE INFOCOM '95, pp. 985–992.

[12] S. Morgan, *Queueing Disciplines and Pasive Congestion Control in Byte-Stream Networks*, Proc. INFOCOM '89, 1989, pp. 711–720.

[13] I. Norros, *A Storage Model with Self-Similar Input*, Queueing Systems 16 (1994), pp. 387–396.

[14] I. Norros, *On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks*, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6 (August 1995), pp. 953–962.

[15] G. Olariu, *Flow Control in a Hybrid Satellite-Terrestrial Network: Analysis and Algorithm Design*, M.S. Thesis, Systems Engineering Program, Institute for Systems Research, University of Maryland, College Park, 1997, Technical Report ISR TR MS 97-7.

[16] S. S. Panwar, D. Towsley and J. K. Wolf, *Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service*, Journal of A.C.M, Vol. 35, No. 4 (Oct. 1988) , pp. 832–844.

[17] M. Parulekar, A.M. Makowski, emphBuffer Provisioning for $M|G|\infty$ Input Processes: A Versatile Class of Models for Network Traffic, Institute for Systems Research, University of Maryland, College Park, 1996, Technical Report ISR TR 96-59.

[18] V. Paxson, *Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic*, Computer Communication Review/ACM SIGCOMM, Vol. 27, No. 5 (October 1997), pp. 5–17.

[19] S. Shenker, L. Zhang and D. D. Clark, *Some Observations on the Dynamics of a Congestion Control Algorithm*, Computer Communication Review/ACM SIGCOMM, Vol. 20, No. 4 (October 1990), pp. 30–39.

[20] S. Shenker, *Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines*, IEEE/ACM Trans. on Networking, Vol. 3, No. 6 (December 1995), pp. 819–831.

[21] W. Stallings, *High-speed networks: TCP/IP and ATM design principles*, Prentice Hall, 1998.

[22] M.S. Taqqu, W. Willinger, R. Sherman, *Proof of a Fundamental Result in Self-Similar Traffic Modeling*, Computer Communication Review/ACM SIGCOMM, Vol. 27, No. 2 (April 1997), pp. 5–23.

[23] J. Walrand, *An Introduction to Queueing Networks*, Prentice Hall, 1988.