ANALYSIS OF NETWORK TRAFFIC AND ITS APPLICATIONS

A Dissertation

by

CHENGZHI LI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 1999

Major Subject: Computer Engineering

ANALYSIS OF NETWORK TRAFFIC AND ITS APPLICATIONS

A Dissertation

by

CHENGZHI LI

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Wei Zhao
(Chair of Committee)

Riccardo Bettati
(Member)

Nitin Vaidya
(Member)

Garng Huang
(Member)

Wei Zhao
(Head of Department)

December 1999

Major Subject: Computer Engineering

ABSTRACT

Analysis of Network Traffic and Its Applications. (December 1999)

Chengzhi Li, B.S., Fuzhou University, P. R. China;

M.S., Xiamen University, P. R. China

Chair of Advisory Committee: Dr. Wei Zhao

Quality of service (QoS) is an important issue in high speed packet switching networks that support a diverse set of services, including videoconference, telemedicine, distance education, video on demanding, voice mail, and so on. The success in the deployment and management of such networks will critically depend on how well we know the network traffic behaviour. In this dissertation, we study the network traffic and related issues in the high speed packet switching networks which support real time communication. We concentrate on the techniques for the analysis of the network traffic and its applications in the network management.

The contributions of this dissertation are:

1. We develop a comprehensive methodology which can be used to evaluate upper bounds of connections' end-to-end delays for networks with static priority driven scheduling discipline and arbitrary topology. This methodology has been adopted in the NetEx project which collaborates with Honeywell Technology Center, Minneapolis in a DARPA funded Real-Time Adaptive Resource Management project.

2. We prove that to find an optimal static priority driven scheduling is an NP-hard problem if the network system has more than two switches. Furthermore, we propose a suboptimal static priority driven scheduling based on the location and traffic information of connections. Our simulation results show that

the proposed scheduling performs better than other well known static priority schedulings.

3. We develop a criteria for testing the stability of an ATM network with arbitrary topology. For a specialized ring topology (called Cruz-Gallager-Parekh ring), we prove that the network is stable if the total utilization of each link is less than 73.2%.

To MY WIFE, PARENTS, SISTER, and BROTHER

ACKNOWLEDGMENTS

Many people have participated in making the work reported in this thesis possible. First of all, I would like to thank my academic advisor Dr. Wei Zhao, who has provided me with the opportunity to pursue my Ph.D in the computer engineering and has been much more than an academic advisor for me. Dr. Zhao constantly motivated me to aim high and work hard. His teachings and concerns went beyond computer engineering and were aimed at preparing his students for challenging professional careers. I also wish to thank Dr. Bettati, Dr. Vadyia and Dr. Jennifer L. Welch for their guidance and advice through my study and serving on my advisory committee, Dr. Haung for serving my advisory committee, Dr. Olga Kocharovskara and Dr. Edward Fry for serving as my graduate council representative.

I would like to express my great appreciation to Professor Weichin Li, my former advisor, for his patience, guidance, and profound and thorough knowledge in Functional Analysis, Number Theory, Stochastic Analysis, and Operations Research.

I thank my fellow graduate students Badari Devalla, Yong Guan, Cen Li, Amitava Raha, Anirudha Sahoo, Dong Xuan, and many others for their friendship, help, and encouragement.

Finally, I would like to express my heartfelt appreciation to my wife, Xiaoying, for her continuous love, support and understanding. Xiaoying has inspired me, encouraged me and helped me in many ways during my studies at Texas A&M University. My profound gratitude goes to my father who showed himself to me a model of a man during fighting for his belief and my mother who has always shown her selfless support and constant love to me in my life.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Communication networks are in a transition period, moving from relatively slow speed (e.g. Mbps) and single service networks (e.g. telephone network for voice, cable TV network for video, and data network for computer communication) to high speed (e.g. Gbps) packet switching networks which support a diverse set of services, including videoconference, telemedicine, distance education, video on demand, voice mail, and so on. It is expected that a significant portion of future network services will come from multimedia applications which involve real time transportation of data, image, voice, and video. A real time application is characterized by the stringent deadline constraint imposed on its message delivery time. The bursty nature of the traffic of these applications and the statistic multiplexing mechanism used in the packet switching networks raise new issues and pose new challenge to the design, implementation, and management of future high speed packet switching networks. Therefore supporting the real-time communication over high speed packet switching network is the focus of this dissertation.

In this Chapter, we point out what is the motivation of this research. Then we summarize the results obtained in this research. We conclude the Chapter with the outline of the dissertation.

A. Motivation

To ensure a high speed packet switching network able to support numerous applications with stringent timing constraints, the following topics must be carefully inves-

The journal model is *IEEE Transactions on Automatic Control.*

tigated:

- *Packet Scheduling.*

  In high speed packet switching network, each switch has a packet scheduler at each output port. The packet scheduler determines the order of packet transmission and therefore controls queueing delays suffered by packets. Generally speaking, the packet scheduling discipline is the heart of the packet scheduler. The packet scheduling discipline should exploit the network resource as much as possible, i.e., it will make the network admit as many application requests as possible without violating their QoS requirements. Also, the complexity of the packet scheduling discipline should be kept minimal, such that the scheduling discipline can be implemented at link speed, potentially in hardware to eliminate or minimize the scheduling overhead. Finally, the packet scheduling discipline should be scalable, i.e., the complexity of the scheduling algorithm and required data structures does not increase with increasing of the number of application requests.

- *Traffic Characterization.*

  In order to obtain the QoS guarantee, the network users must characterize their applications' traffic by a set of the traffic parameters and submit it with the QoS requirements to the network. Furthermore, in order to manage the network, the internal network traffic must be estimated, otherwise, the performance of the network can not be analyzed and predicted. Therefore, concise and accurate traffic characterization plays a key role to improve the utilization of the network and to manage the network.

- *Delay Analysis.*

  The end-to-end delay suffered by the network traffic must be accurately estimated since it determines the number of applications of the real time communication that can coexist in the network. Overly pessimistic delay estimation will result in the poor utilization of the network. On the other hand, too optimistic estimation may result in the violation of QoS guarantees. Therefore, the tight evaluation of the end to end delay also plays a key role to maximize the utilization of the network.

All these topics concern the network traffic. Packet scheduling concerns how to arrange the network resource to support heterogeneous network traffic with different quality of service requirements. Traffic characterization concerns how to simply and accurately describe the network traffic to provide the scientific basis for network design and management. Delay analysis concerns how to determine the timing behaviour of the network traffic to guarantee in advance the quality of service to heterogeneous network traffic. Therefore we propose to study the network traffic and related issues. In particular, we analyze the queuing delay suffered by packets in a high speed packet switching network. We design a new simple static priority scheduling algorithm to improve the performance of the network. Finally, we examine the condition that can guarantee a network is stable, i.e., queuing delays suffered by the packets in the network is bounded. Because without determining the stability of the network, it does not make any sense to provide real time communication over the network.

B. Research Contributions

This dissertation develops the state of the art for real-time communication in following ways:

1. We develop a comprehensive methodology which can be used to evaluate upper bounds of end-to-end delays suffered by the traffic in the networks with static priority driven scheduling discipline and arbitrary topology [1, 2, 3]. This methodology has been adopted in the NetEx project which collaborates with Honeywell Technology Center, Minneapolis in a DARPA funded Real-Time Adaptive Resource Management project.

2. We prove that to find an optimal static priority driven scheduling is an NP-hard problem if the network has more than two switches [1]. Furthermore, we propose a suboptimal static priority driven scheduling based on the topology information and traffic information. Our simulation results show that the proposed scheduling performs better than other well known static priority schedulings.

3. We develop a criteria for testing the stability of an ATM network with arbitrary topology. For a specialized ring topology (called Cruz-Gallager-Parekh ring [3]), we prove that the network is stable if the total utilization of each link is less than 73.2%.

C. Dissertation Outline

The remainder of this dissertation is organized as following. In Chapter II, we categorize the networks into two classes based on their topology. We also make a brief survey of the source traffic modeling.

In Chapter III, we make a brief survey of the existing results about the packet scheduling disciplines, methodology for the analysis of queuing delay suffered by the network traffic, and the network stability.

In Chapter IV, we study the computation of the delay suffered by the network traffic in an ATM network. A numerical method is developed to compute the worst-case end-to-end delays suffered by the traffic in an ATM network with arbitrary topology. Convergence of the numerical method is formally proved and a closed form for the computing error is obtained.

In Chapter V, we propose a new method for deriving end-to-end delay bounds suffered by the traffic in a tandem network, which uses a FIFO scheduling discipline. Our new method takes into account delay dependencies in successive servers along the path traveled by the packets and achieves better performance than the method provided in [4, 5].

In Chapter VI, we study ATM networks with static priority scheduling. We address the problem of how to assign priorities to cells in such a network. Particularly, we analyzed five algorithms: FIFO, relative deadline monotonic (RDM), Cruz' Algorithm [5], and two new algorithms, namely *Partition* Algorithm and *Integrated* Algorithm. Performance evaluations show that two new algorithms outperform the other three. Sometimes, the performance difference is significant.

In Chapter VII, we address the issue of stability in the communication networks. A network is said to be *stable* if all the data packets experience bounded delays within the network. Obviously, unbounded packet delays will have a detrimental impact on the performance of any distributed application communicating via the network. Therefore, ensuring stability within the network has been a pivotal issue in the design and management of communication networks. We choose ATM network to address the issue of stability and find that a ring with large number of switches is stable if the total utilization of the links is less than or equal to 73.2%.

In Chapter VIII, we summarize the current research work and indicate future research directions.

CHAPTER II

MODELS

In this Chapter, we will give a brief description of Network Models and Traffic Models.

A. Network Model

A network consists of a set of packet switches which are interconnected by physical links. In this work we only consider connection-oriented model, i.e., packets from a source travel along a fixed, predetermined route to the destination. A connection consists of a set of ordered physical links and packet switches. It is set up during the connection establishment and remains fixed during the life time of the connection. Furthermore, in adherence to ATM specification [6, 7, 8, 9], we also assume that all packets have fixed size and all switches are nonblocking. When packets arrive at the input port, they are routed directly to the appropriate output ports without switching conflicts. Packets destined for different output ports do not interfere with each other, and the queueing delay may only occur at the output ports of the switch.

For a network with arbitrary topology, even though every connection in the network is acyclic, the union of several connections may result in virtual cyclic dependencies. The presence of virtual cycles can lead to feedback effects for network traffic, such that the queueing length of an output port of a switch not only depends on the arrival traffic, but also depends on the departure traffic. This phenomenon may cause network to be unstable or, even worse, chaotic. Based on the virtual feedback phenomenon, we classify the network topologies into two classes. One is called the tandem network and the other is called the feedback network. Definitions and simple examples for both network topologies are given below.

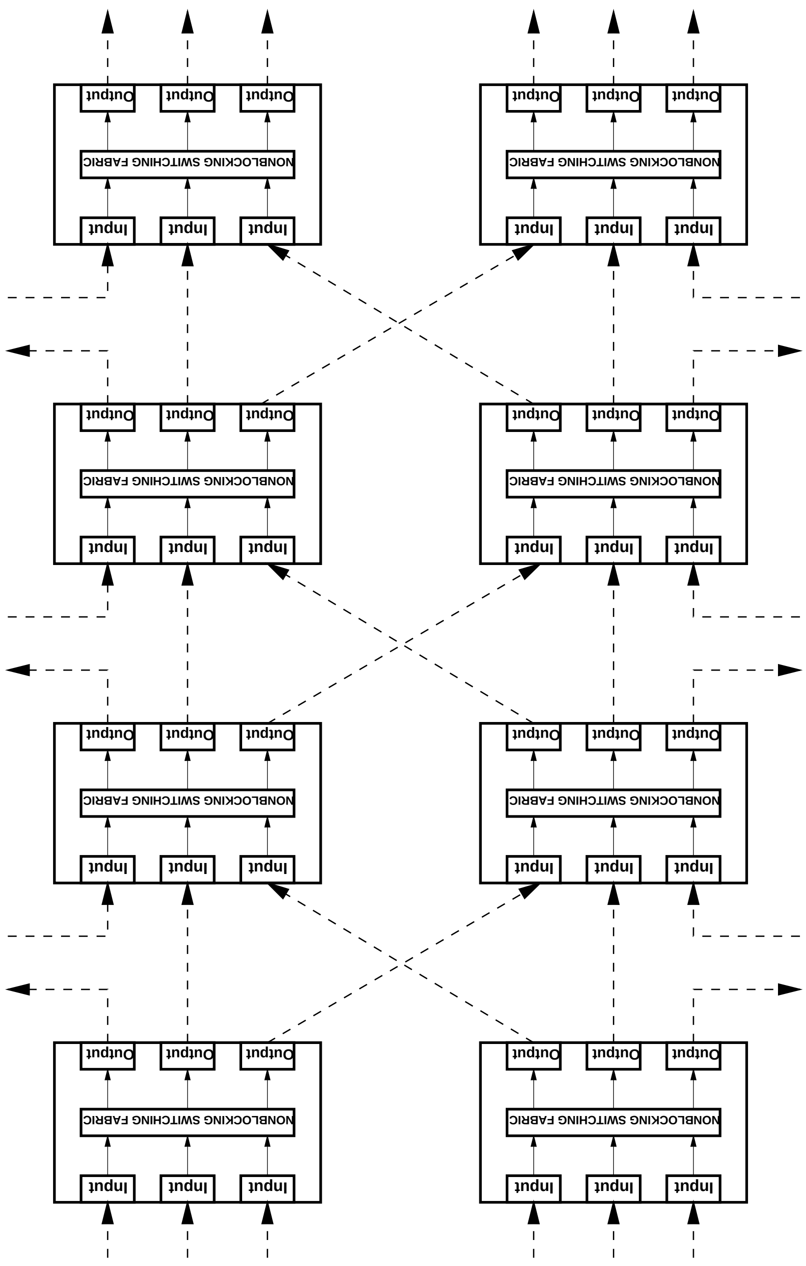


Fig. 1. A simple tandem network.

DEFINITION 1 *A network is called the tandem network if there exists an order for all switches such that*

1. *Every input traffic of the first switch is source traffic.*
2. *Every input traffic of the (i+1)-th switch can be estimated based on the information of all input traffic of switches with order less than i+1.*

See Figure 1 for a simple example. It is obvious that no virtual feedback phenomenon appears in any tandem network.

DEFINITION 2 *A network is called the feedback network if it is not a tandem network.*

See Figure 2 for a simple example. It also is obvious that there may be virtual cycles as the result of the union of several connections in a feedback network.

In order to simplify the analysis of network traffic, we model a network as a collection of servers. A server is an abstraction of a network component that is traversed by network traffic. Therefore, the input ports, the switching fabric, the output ports, and the physical links can be modeled as servers serving network traffic. Furthermore, the servers can be classified into two categories: *constant servers* and *variable servers* [4, 10, 11]. A constant server is the one that offers a constant delay to each cell that uses it and does not by itself change the traffic flow characteristics of a connection. For example, physical links and the switching fabric are constant servers. The function of an input port is to demultiplex the arriving cells based on the information in the cell header. This is achieved in constant time by the hardware associated with the input port. Thus, we can also model the input port of an ATM switch as a constant server. The functionality of an output port of a switch is more complicated. An output port may simultaneously receive cells belonging to different

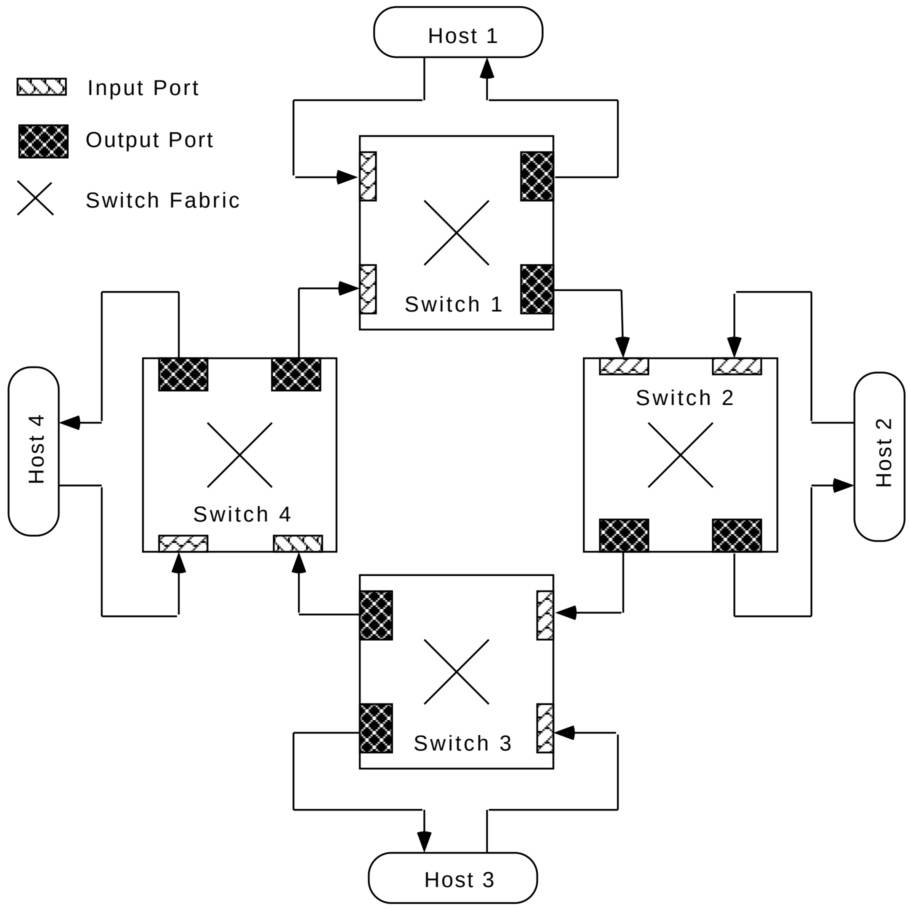


Fig. 2. A simple feedback network.

connections competing for transmission on the link associated with the output port. Thus, cells may be buffered at an output port and transmitted in an order that is determined by the scheduling discipline employed by the switch hardware. Note that a multiplexor server must be considered as a variable server since the delay suffered by a cell in this server varies depending upon the queue length in the buffer. Consequently, the traffic characteristics of a connection at the output of this server may differ from that at the input.

Figure 3 shows the network modeled as a collection of servers serving four con-

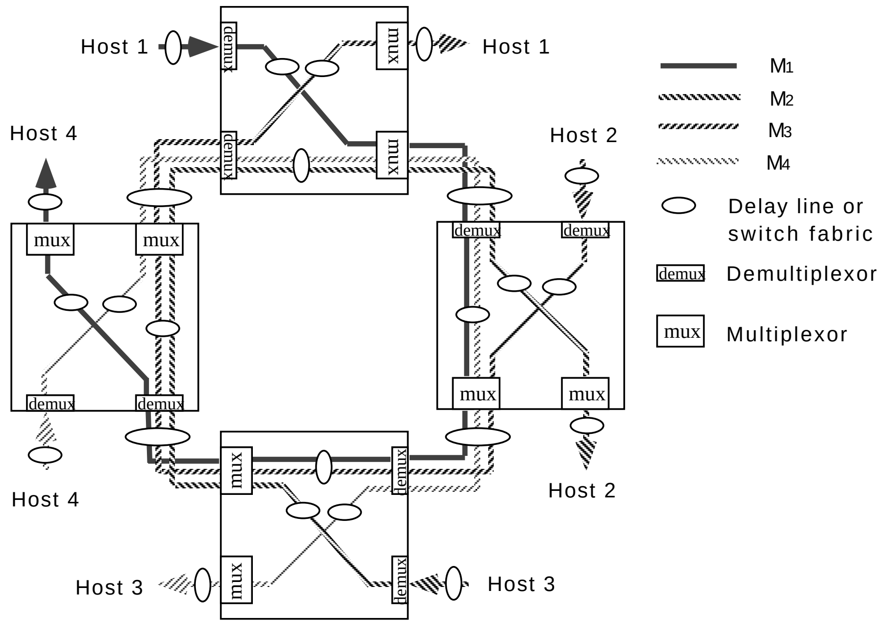


Fig. 3. Server representation of ATMnetwork.

nections M_1, M_2, M_3, and M_4.

Consider the connection M_1 from Host 1 to Host 4 shown in Figure 3. M_1 traverses 9 line servers (5 physical links and 4 switching fabrics) and 4 demultiplexor servers (input ports of 4 switches) all of which are constant servers. M_1 also traverses 4 multiplexor servers (output ports of 4 switches) which are variable servers. Recall that the constant servers serving M_1 only add a fixed amount of delay to M_1's cells and do not change M_1's traffic characteristics. Hence, their impact on M_1 can be accounted for by simply subtracting the total constant delays suffered by M_1 at these servers from M_1's delay requirement. The same holds for other connections. In the rest of this dissertation, we assume that the delay requirements of connections are modified in such a way. Consequently, we eliminate all the constant servers from further consideration and focus only on the variable servers in the remainder of this

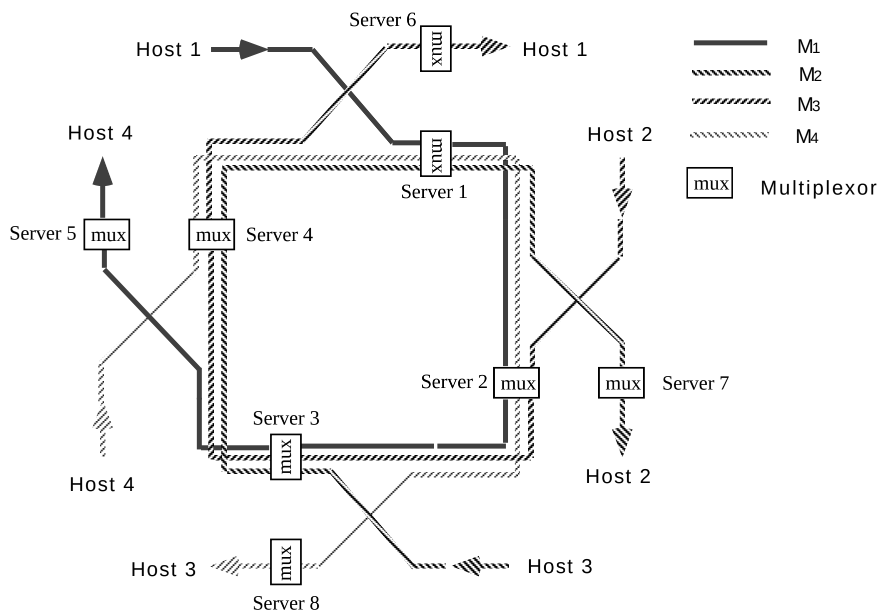


Fig. 4. Connection-server graph representation.

dissertation. Hence, from now on, we can view a connection as being served by a sequence of variable servers only. We will often omit the prefix ‘variable’ when referring to variable servers to avoid repetition. Further, we assume that each of these servers is given a unique integer identity.

We use the above abstraction to construct a connection-server graph. A connection-server graph is constructed as a labeled, directed graph with the servers as its nodes. A directed edge is introduced from server m to server n if there is a connection that is served by server m followed by server n. The edge is labeled by the connection that uses the servers in immediate sequence. Figure 4 shows the connection-server graph corresponding to the system shown in Figure 3. The sources and destinations of connections are also shown in the connection-server graph. The connection-server graph is used to facilitate the discussion of the queuing delay suffered by network traffic later.

B. Traffic Models

Traffic models are at the heart of any performance evaluation and prediction of telecommunications networks. An accurate estimation of network performance is critical for the success of high speed packet switching networks. Such networks need to guarantee an acceptable quality of service level to the users. Therefore, traffic models need to be accurate and able to capture the characteristics of network traffic. During past decades, many traffic models have been proposed for the communication network (see several good survey papers [7, 12, 13, 14]). From the mathematical point of view, all network traffic models can be categorized into two classes: stochastic network traffic models and deterministic network traffic models. Stochastic network traffic models use the stochastic process to capture the statistical characteristics of the actual traffic. For example, *Renewal Traffic Models* use the Poisson processes; *Markov Modulated Traffic Models* use the *Markov* process; and *Self-Similar Traffic Models* use fractional *Gaussian* process. Though the stochastic network traffic models can make networks achieve higher utilization via statistical multiplexing mechanism, it is very difficult for the network service provider to enforce users to follow their stochastic traffic characterization and guarantee their QoS requirements. On the other hand, the deterministic network traffic models use the deterministic function to describe the realistic network traffic. Even though the deterministic network traffic models make networks achieve lower utilization via statistical multiplexing mechanism, it is very easy for the network service provider to enforce users to follow their deterministic traffic characterization and guarantee their QoS requirements. Since our object is the real time communication, we only study the deterministic traffic models. Furthermore, from the physical point of view, all deterministic traffic models can be classified into two categories: discrete event traffic models and fluid flow traffic models. The detail discuss of these two categories is given as following.

1. Discrete event traffic models:

   The discrete event traffic models view traffic as a stream of discrete entities (messages, frames, packets, cells, etc). It can be precisely determined by two sequences. One is for the entity arrival time and the other is for the correlative entity size. This traffic models are appropriate in the case where entity size is variable and the individual entity of traffic has non-ignorable impact on the performance of the networks. Two of the common used discrete event traffic models for real time communication are given as following:

   - (C,P) Models [15]: A traffic stream falls into this model if during every time period P, only one message with length C comes from this stream.
   - $(X_{min}, X_{ave}, I, S_{max})$ Model [16]: A traffic stream falls into this model if the inter arrival time between any two packets in the stream is more than X_{min}, the average packet inter arrival time during any time interval of length I is larger than X_{ave}, and the maximum packet size is less than S_{max}.

2. Fluid flow traffic models:

   The fluid flow traffic models views traffic as a stream of fluid, characterized by a flow rate (such as bits per second). This corresponds to the traffic which is infinitely divisible. Fluid flow traffic model is appropriate in the case where individual units of network traffic are numerous relative to a chosen time scale and have little impact on the performance of the networks, just as one molecule more or less in a water pipeline has an infinitesimal effect on the flow. In the ATM network, all packets are fixed-size cells of relatively short length (53 bytes); in addition, the high transmission speeds render the transmission impact of an individual cell negligible. Two of the common used fluid flow traffic models for real time communication are given as following:

- Cruz's Model [4]:

  A traffic stream can be described by this model if there is a monotonic increasing function $b(\cdot)$, called as a traffic constraint function of the traffic stream, such that during any time interval of length I, the number of bits coming from the traffic stream is no greater than $b(I)$. For a given traffic stream, there are many valid traffic constraint functions, out of which, the leaky bucket traffic constraint function, i.e., $b(I) = \alpha + \rho I$, is commonly used.

- Texas A&M Model [11]:

  A traffic stream can be described by this model if there is a monotonic increasing function $\Gamma(\cdot)$, called as a Maximum Rate Function of the traffic stream, such that during any time interval of length I, the maximum average rate of the traffic stream is less than $\Gamma(I)$. For a given traffic stream, there are many valid Maximum Rate Functions, out of which, the three points approximation of Maximum Rate Functions is commonly used. See [11] for details.

Since our target is for high speed packet switching networks, we only deal with the fluid traffic models. In the following discussion, we will assume that the traffic entering the network is bounded by a piecewise linear traffic function [4, 5] of the form

$$F(I) = \min(I, \beta + \rho * I), \tag{2.1}$$

which models a traffic that is shaped by a token bucket of size β and rate ρ, followed by a leaky bucket with the rate equal to the link speed, which we normalize to one.

CHAPTER III

PREVIOUS WORK

In this Chapter we survey relevant work in the areas of scheduling discipline, queuing delay analysis, and network stability.

A. Scheduling Disciplines

The scheduling discipline plays an important role in the network management. The scheduler at a switch determines the order in which packets in the waiting queue are transmitted. Hence, the scheduling discipline has a direct impact on the delays experienced by the network traffic as well as on the distortion of the network traffic, i.e., the connection's traffic may become more bursty after passing several switches. In order to deal with the network traffic with diverse characteristics and different QoS requirements, a good scheduling discipline should hold the following criteria [17]:

- Efficient Resource Utilization:

  The scheduling discipline should exploit the network resource as much as possible, i.e., it will make network admit as many connections as possible without violating their QoS requirements.

- Simplicity of Implementation:

  The scheduling discipline should be simple enough to be implemented at very high speed, especially in hardware, to eliminate or minimize scheduling overhead.

- Scalability:

  The scheduling discipline should be applicable to a large number of connections, i.e., the complexity of scheduling algorithm and the required data structure does not increase with the increasing of the number of connections.

- Fairness:

  The scheduling discipline should let server to serve connections proportional to their reservations and distribute the unused bandwidth left by idle connections proportionally among the active connections.

Unfortunately, there is no a scheduling discipline satisfying all criteria or superior to others in all situations. The design or selection of a particular scheduling discipline for a scheduler involves the tradeoff between the need to support a large number of connections with diverse QoS requirements and the need for simplicity in the scheduling operations; the tradeoff between to achieve the maximum utilization of the network resources and to achieve the fairness for all connections.

Many scheduling disciplines have been proposed in last decades. There are several ways to classify scheduling disciplines, for example, classifying in terms of the characteristics of the network performances, i.e., work conserving or non work conserving (see the survey paper [18]); classifying in terms of how to isolate network traffic, i.e., class isolating, intra class isolating, and inter class isolating (see the survey paper [19]); classifying in terms of scheduler based methods or rate based methods (see the survey paper [20]). Here, we classify the most representative scheduling disciplines in terms of their main functionalities, i.e., traffic shaping oriented, delay guarantee oriented, bandwidth guarantee oriented, and differentiating traffic oriented.

- Traffic shaping oriented scheduling disciplines:

  This class includes *Stop and Go* [21] (SG), *Jitter Earliest Deadline First* [16] (J-EDF), *Rate Controlled Static Priority* [22] (RCSP). One of the common characteristic of these scheduling disciplines is that the traffic distortion within the network is controlled.

- Delay guarantee oriented scheduling disciplines:

  This class includes *Earliest Deadline First* [23] (EDF), *Service Curve Earliest First* [24] (SCED). One of the common characteristic of these scheduling disciplines is that the deadlines of connections are used to determine the order of service for the packets of these connections.

- Bandwidth guarantee oriented scheduling disciplines:

  This class includes *Virtual Clock* [25] (VC), *Generalized Processor Sharing* [26] (GPS), *Weighted Fair Queueing* [27] (WFQ), *Worst Case Fair Weighted Fair Queueing* [28] (WF^2Q), *Self Clocked Fair Queueing* [29] (SCFQ), *Weighted Round Robin* (WRR), *Hierarchical Round Robin* [30] (HRR). One of the common characteristic of these scheduling disciplines is that the allocated bandwidths for connections are used to determine the order of service for packets of these connections.

- Differentiating traffic oriented scheduling disciplines: This class includes *Static Priority* [23] (SP), *Class Based Queueing* [31] (CBQ). One of the common characteristic of these scheduling disciplines is to provide some semblance of sharing for connections among the same class.

B. Methodology of Delay Analysis

The delay experienced by a packet in a network can be divided into four components:

1. Queueing Delay:

   Queueing delay is the time spent by a packet in a server queue while waiting for beginning of transmission.

2. Transmission Delay:

   Transmission delay is the length of the time interval between the beginning of transmission of the first bit and the end of transmission of the last bit of a packet. For the ATM switch with OC3 output link, the transmission delay suffered by an ATM cell is about 3 μs.

3. Propagation Delay:

   Propagation delay is the length of the time interval a bit takes to traverse a link connecting two server. This delay for electric and optical signals is between 4 and 5 μs/km.

4. Processing Delay:

   Processing delay is any packet delay resulting from processing overhead. At an ATM switch, a cell undergoes a fixed processing delay. This delay is due to the cell being copied into switch memory one or more times and the time taken for computing the CRC and for translating the VCI/VPI into route through the switch, etc..

On the other hand, from theoretical point of view, all delays can also be categorized in to two classes: constant delay and variable delay. For ATM network, after a connection is set up, the transmission delay, the propagation delay, and the processing delay are constants and can easily be evaluated. But the queueing delay is a variable and accurately computing queueing delay is very difficult.

To guarantee that all connections can meet their deadline requirements, an efficient and effective method to derive the upper bound for the end-to-end delay experienced by connection's traffic is needed. In terms of effective, we mean that the method of the delay analysis must be able to produce tight delay bounds. Otherwise expressively overestimating the delay bounds may make the network system underutilized. In terms of efficient, we mean that the method of the delay analysis should be simple in order to be used in on line connection admission control. Otherwise, overhead of delay analysis also deteriorates the utilization of the network. During the past decade, various delay analysis techniques have been invented to evaluate upper bounds for the end-to-end delays experienced by the traffic in a network environment. All these delay analysis techniques can be classified into two classes. One is called the Decomposed Method, and the other is called the Integrated Method. The brief description of these methods are presented as the following.

1. Decomposed Method.

   Basic idea for Decomposed Method is breaking up the network into isolated servers, characterizing the local traffic on per connection basis at the isolated server, analyzing the local delay suffered by the traffic at isolated server, and deriving the end to end delay by summarizing all local delay bounds at isolated servers passed by the connection. Fundamental work has been pioneered in [4, 5].

   Advantage: This method is simple and suitable to deal with network with arbitrary topology and any scheduling disciplines.

Disadvantage: This method often overestimates the end-to-end delay suffered by the connection's traffic and makes the network resource underutilized. Since a packet suffers the worst case delay at one server, it may not suffer the worst case delay at successive server. The upper bound for the end-to-end delay obtained by the Decomposed Method is always too loose. So some real time connections may be rejected even though network can guarantee their QoS requirements.

2. Integrated Method:

Basic idea for Integrated Method is analyzing all server involved in an integrated manner, taking into account delay dependencies in successive servers that a connection has to be served, and deriving the end-to-end delay directly. There are three kinds of Integrated Method.

- Service Curve Allocated Method: First, service curves (i.e. the minimum service offered by a server to a connection, see [24] for accurate definition of service curve) are assigned to every connection at each server. Then, scheduling disciplines at each server are designed according to these service curves, such that these service curves are guaranteed by servers. Finally, the end-to-end delay bound is derived based on the source traffic characterization and network service curve which is the evolution of service curves of all servers passed by the connection. See [24] for SCED scheduling discipline.

  Advantage: Theoretically this method fully utilizes the network resource and can deal with the network with arbitrary topology.

  Disadvantage: Unfortunately, it is too complicated to be implemented.

Since the deduced scheduling discipline is always dynamic priority algorithm and the scheduling overhead is not negligible and will impair utilization of the network resource.

- Service Curve Induced Method: First, scheduling disciplines are chosen for every server. Next, service curves are derived based on these server scheduling disciplines. Then, the network service curve is derived based on these local service curves. Finally, the end-to-end delay bound is derived based on the characterization of the source traffic and the network service curve. See [27] for GPS scheduling discipline.

  Advantage: The bounds for the end-to-end delays suffered by connections' traffic obtained by this method are very tight.

  Disadvantage: To derive the service curves is not trivial, particular for FIFO, SP scheduling disciplines.

- Mapping Method: First, the multi-hop connections are mapping into the connections in a reference system which only contains one server. Usually the delay analysis for connections in a single server system is easy. By comparing the original system and the reference system, the end-to-end delay experienced by network traffic is derived only on the source traffic characterization and the server scheduling discipline. See [32] for VC scheduling discipline.

  Advantage: The bounds for the end-to-end delays suffered by connections' traffic obtained by this method are very tight.

  Disadvantage: To design an appropriate reference system is not easy.

C. Network Stability

A network is said to be *stable* if all the packets experience bounded delays within the network. Obviously, unbounded packet delays will have a detrimental impact on the performance of any distributed application communicating via the network. Therefore, ensuring stability within the network has been a pivotal issue in the design and management of communication networks. For a tandem network, the criteria for testing stability is obvious, i.e., the total utilization of each output link is less than the capability of the link. But for a network with arbitrary topology, the answer is not so easy to find. Even though every connection in the network is acyclic, the union of several connections may result in virtual cycles in the network topology. The presence of these virtual cycles complicates the analysis of network performance considerably. Even worse, it can lead to feedback effects, i.e., the delay suffered by traffic on a server depends on not only the input traffic, but also the output traffic of the server. This phenomenon may drives the network towards unstable even though it is still not overloaded.

For a specialized ring topology (called Cruz-Parekh-Gallager ring and see Figure 2 for a simple example), a conjecture was provided by Cruz [5], and Parekh and Gallager [27], and we called it as C-P-G conjecture.

CONJECTURE 1 *A Cruz-Parekh-Gallager ring is stable for any work-conserving scheduling discipline if the total long term work load of each output link is less than the capability of the output link.*

Since it does not make any sense to determine delay bounds for traffic in a potentially unstable network, to prove C-G-P conjecture is an urgent and challenge task. For C-P-G ring with 4 servers (see Figure 2) and a special static priority scheduling (i.e., when entering C-P-G ring, traffic is assigned lower priority 0; otherwise, traffic is always assigned higher priority 1), Cruz proved that a C-P-G ring with 4 nodes is stable if the system load is less than 75%. Two years later, Parekh in [27] generalized Cruz's result. Given the same static priority scheduling as that in [5], he proved that C-P-G ring with arbitrary number of nodes is stable if the system load is less than 50%.

CHAPTER IV

QUEUING DELAY ANALYSIS BY DECOMPOSED METHOD

In this Chapter[1], we study delay computation problem for an ATM networks with static priority scheduling. Given an ATM network with arbitrary topology, it is possible that the delays suffered by the network traffic at different servers depend on each other due to the logical cyclic dependency of the traffic. We start by formally deriving the characteristic of traffic inside the network. We then use the Decomposed Method to estimate the end-to-end delay. In this approach, the network is decomposed into *servers*, and each ATM connection is viewed as traversing a sequence of servers. The worst-case end-to-end delays are obtained by summing up the upper bounds of the local delays suffered by a connection at each of the servers. Particularly, after setting up nonlinear algebra equations to determine relationships between the local delays and the characteristics of source traffic, we develop a numerical method to compute worst-case delays suffered by the traffic in an ATM network with arbitrary topology. Convergence of this method is formally proved and a closed form for the computing error is obtained.

[1]Part of this chapter is reprinted with permission from Proceedings of the *IEEE* INFOCOM'97, Kobe, Japan, 1997, vol. 1, pp. 160-167; Proceedings of the 18th IEEE Real-Time Systems Symposium, San Francisco, CA, Dec. 1997, pp. 264-273; and Proceedings of IEEE International Conference on parallel Processing, Minneapolis, MN, Aug. 1998, pp. 432-440. Copyright 1998 by the *IEEE*. (see Appendix A).

A. Some Definitions and Notations

In this section, we give the definitions and notations that will be used in the rest of this dissertation.

1. *Connections.*

   Let $\mathcal{M} = \{M_1, M_2, \ldots, M_N\}$ denote the set of N connections that compete for resources within the ATM network and $C(j)$ be the set of all connections that traverse Server j.

2. *Servers.*

   Servers in a connection-server graph are multiplexors with a single output link. The topology of a connection-server graph is therefore determined by subsets $P(j) \mid j = 1, 2, \cdots$ of servers, where $P(j)$ specifies the set of all servers whose output traffic enters Server j. We call the servers in $P(j)$ the *predecessor* servers of Server j.

3. *Routes.*

   The route of a connection is defined by the sequence of servers traversed by that connection. Let $s(i,j)$ denotes the identity of the j^{th} server in the route of Connection M_i. If S_i is the total number of servers serving Connection M_i, the *route* of M_i can be represented as the sequence G_i of servers serving that connection.

   $$G_i = < s(i,1), s(i,2), \ldots, s(i,j), \ldots, s(i,S_i) > . \tag{4.1}$$

   The *partial route* $G_{i,j}$ is the set of servers traversed by a cell of Connection M_i from the source *up to and including* Server j. In other words, $G_{i,j}$ contains all the servers used by M_i upstream from Server j, including Server j.

4. *Priority assignment.*

Let $\pi(i,j)$ denote the priority assigned to connection M_i at the server j. It is obvious that the priorities assigned to connections on the servers control their local delay bounds. Therefore the assignment should be sensitive to the deadline requirements of the connections in order to maximize the chance that all the deadline requirements of the connections can be met.

5. *Traffic Characterization.*

In order to allow for an analytical delay calculation, the traffic is characterized in form of a *traffic bounding function*, which defines the maximum number of bits that can arrive as function of the time interval. This bounding function $F_{i,j}(I)$ specifies the maximum number of bits that can arrive at Server j from Connection M_i during any interval of length I. In the following discussion, we will assume that the traffic entering the network is bounded by a linear traffic function [4, 5] of the form

$$F(I) = \min(I, \beta + \rho * I), \tag{4.2}$$

which models a traffic that is shaped by a token bucket of size β and rate ρ, followed by a leaky bucket with the rate equal to the link speed, which we normalize to one.

As illustrated in Figure 5, the linear bounding function consists of two linear segments, one with a rate of one, and one with rate ρ. Following the terminology in [33], we call the intersection point of the two segments the *flex* point, which is denoted as f in Figure 5. In the following we will call a traffic that is characterized by Equation (4.2) to be *bound by a* (β, ρ) *function*. Furthermore, we assume the traffic of Connection M_i at its entrance $s(i,1)$ to the network to

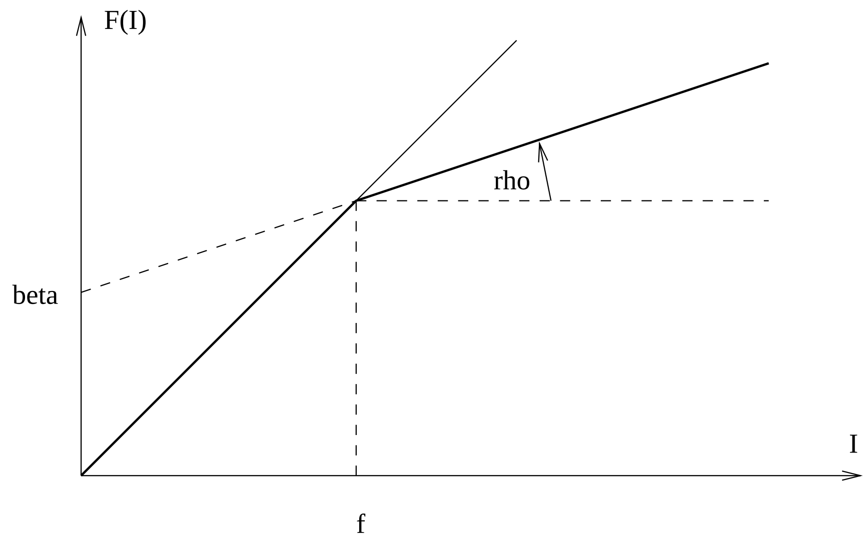


Fig. 5. Linear traffic bounding function F with parameters β and ρ.

be bound by a (β, ρ) function with parameters β_i and ρ_i. In order to quantify the effect that connections have on each other, we want to cluster the connections sharing a particular link that are assigned the same priority, and represent them by a single traffic bounding function. Hence, we define $\mathcal{F}_{p,j}(I)$ to be the aggregated traffic of connections with priority p on the output link of Server j. That is, $\mathcal{F}_{p,j}(I)$ is the maximum number of bits of connections with priority p that can leave Server j during any interval of length I. Similarly, we let $\mathcal{J}_{p,j}$ be the aggregated traffic of connections with priority *higher than or equal to* p on the output link of Server j. That is, $\mathcal{J}_{p,j}(I)$ is the maximum number of bits of connections with priority higher than or equal to p that can leave Server j during any interval of length I. These traffic bounding functions will be useful in the following delay analysis.

6. *Worst case delays.*

Let d_i be the *worst-case end-to-end delay* experienced by Connection M_i. We define $d_{p,j}$ to be the worst-case delay experienced at Server j by a connection with priority p. If Connection M_i is assigned priority $\pi(i,j)$ at Server j, then $d_{\pi(i,j),j}$ is the *worst case local delay* of Connection M_i at Server j, and the end-to-end delay for M_i can be formulated as the sum of the worst-case local delays on its route:

$$d_i = \sum_{j=1}^{S_i} d_{\pi(i,s(i,j)),s(i,j)}. \tag{4.3}$$

Assuming a system with K servers and P priorities per server, in the following discussion we will use the KP-dimensional vector $\vec{d}$ of delays at all priorities at all the servers in the system, i.e.,

$$\vec{d} = (d_{1,1}, d_{2,1}, \cdots, d_{P,1}, d_{1,2}, \cdots, d_{P,2}, d_{1,3}, \cdots, d_{P,K}) . \tag{4.4}$$

7. *Maximum link utilization.*

We will denote μ to be the maximum of average link utilizations in the network, i.e.,

$$\mu = \max_{i=1,\cdots,K} \sum_{j \in C(i)} \rho_j \tag{4.5}$$

where $C(i)$ is the set of connections that traverse server i.

B. Internal Network Traffic Bounding Functions

After knowing the source traffic characteristics, it is still difficult to estimate the delay suffered by the traffic inside the network. Without additional traffic regulators implemented at the servers, the shape of traffic is always distorted inside the network.

In order to determine delay bounds at servers inside the network, we need to study the characteristics of internal network traffic. The following theorem allows us to determine the traffic bounding function $F_{i,j}(I)$ for Connection M_i at the output of Server j, when the source traffic of M_i is bound by a (β, ρ) function as defined in Equation (4.2).

THEOREM 1 *For any connection M_i whose traffic at its entrance to the network is bound by a (β, ρ) traffic constraint function with parameters β_i and ρ_i, the traffic of M_i at the output of j^{th} server on its route is bound by a (β, ρ) function, denoted as $F_{i,j}(I)$ with parameters $\bar{\beta}$ and $\bar{\rho}$, where*

$$\begin{aligned} \bar{\beta} &= \beta_i + \rho_i \sum_{g \in G_{i,j}} d_{\pi(i,g),g} \\ \bar{\rho} &= \rho_i \quad . \end{aligned} \tag{4.6}$$

Proof: See the last Section of this Chapter. Q.E.D

According to Theorem 1, as the traffic moves along its route, it remains bounded by a (β, ρ) function. The rate ρ remains constant along the route, but the burstiness β increases as a function of the accumulated worst-case delay on the route. This theorem generalizes earlier results by Cruz [4] and Raha *et al.* [11], where the traffic characteristic function at the output of an FCFS server was obtained in terms of that at the immediate input to the server [4] or at the source [11]. In Theorem 1 we extend these results to servers with static-priority scheduling. As we will see, Theorem 1 facilitates the efficient computation of the worst-case delays in networks with static-priority scheduling.

Traffic from different connections in an ATM network may be multiplexed at the multiplexor of a switch and transmitted over its output link. In order to determine queuing delays, we must characterize the aggregate traffic over a single link. The following theorem describes the aggregate traffic for an arbitrary set of connections over a particular link.

THEOREM 2 *Let $\mathcal{M}^*$ be any subset of the connections that traverse Server j. Its aggregate traffic on the output link of Server j is bounded by a (β, ρ) function with parameters $\bar{\beta}$ and $\bar{\rho}$, where*

$$\begin{aligned} \bar{\beta} &= \sum_{i \in \mathcal{M}^*} \left(\beta_i + \rho_i \sum_{g \in G_{i,j}} d_{\pi(i,g),g} \right) \\ \bar{\rho} &= \sum_{i \in \mathcal{M}^*} \rho_i . \end{aligned} \tag{4.7}$$

Proof: See the last Section of this Chapter. Q.E.D

Theorem 2 indicates that every aggregate traffic inside the network is also bound by a (β, ρ) function, and the parameters can be determined by applying Equation (4.7). In particular, we can use Equation (4.7) to determine the traffic bounding functions $\mathcal{F}_{p,j}$ for the aggregate traffic with priority equal to the given priority p and $\mathcal{J}_{p,j}$ for the aggregate traffic with priority higher than or equal to the given priority p, at the output link of Server j. This is of use in analyzing the delays at the servers in the network, as we describe in the following section.

C. Expressions for Local Delays

Once the traffic bounding functions of both the traffic entering the network and the traffic inside the network are known, the local delays for every connection at each switch can be determined. Formula (4.8) indicates how long a new arriving cell with priority p can be delayed at a given switch j. As defined earlier, $P(j)$ denotes the set of predecessor servers to Server j.

$$d_{p,j} = \max_{0<t\leq T_{p,j}} (\sum_{k\in P(j)} \mathcal{J}_{p-1,k}(t+d_{p,j}) + \sum_{k\in P(j)} \mathcal{F}_{p,k}(t) - t) + 1. \tag{4.8}$$

In this formula, $T_{p,j}$ denotes the *maximum busy interval* for all connections with priority equal to or higher than p on Server j. Formula (4.8) describes the maximum delay of a cell as the time the cell is delayed by *higher-priority* cells, which arrived before or while the cell is queued, and *same-priority* cells, which were there before the arrival of the cell. If no higher-priority connections "join" a set of connections at a server, then the worst case delays at that server are one cell time; the traffic at most wait one cell time to flow through the server due to being blocked by one cell with lower priority. This is illustrated by the following lemma.

LEMMA 1 *If all the traffic with priority higher than or equal to p at Server j comes from only one previous server then $d_{p,j} = 1$.*

Lemma 1 holds because the cells have been ordered at the output link of the previous server. Server j now simply forwards the cells.

Given the shape of the traffic bounding functions, we can formulate the maximum busy interval $T_{p,j}$ using the following closed form.

THEOREM 3 *The maximum busy interval $T_{p,j}$ at Server j for all connections assigned priority higher than or equal to p is given by*

$$\begin{aligned} T_{p,j} &= \min_{t>0}\{t| \sum_{k\in P(j)} \mathcal{J}_{p,k}(t) - t < 0\} \\ &= \frac{\sum_{i\in C(j),\pi(i,j)\leq p} [\beta_i + \rho_i * \sum_{g\in G_{i,j}, g\neq j} d_{\pi(i,g),g}]}{1 - \sum_{i\in C(j),\pi(i,j)\leq p} \rho_i}. \end{aligned} \tag{4.9}$$

Proof: See the last Section of this Chapter. Q.E.D.

The summations in Equation (4.9) go over all connections that traverse Server j and are assigned a priority higher or equal to p. We note that the term $d_{p,j}$ occurs on both sides of Equation (4.8). This means that the local delays cannot be determined directly, but are solutions to the system of equations defined by Equation (4.8).

The following theorem provides a means to formally determine local delay. We use the term χ to indicate whether the delay for a connection's traffic with a given priority on a given server has an effect on the delay for the traffic with a given priority on downstream servers. Particularly, we define $\chi_{i,q,s,p,j}$ to be 1 if Server s is upstream from Server j on the route of Connection M_i, and the connection is assigned priority q and p on Server s and j, respectively. The value for $\chi_{i,q,s,p,j}$ is zero otherwise. For each server j we define a server j^* among the predecessor servers of Server j, whose output traffic flowing through Server j has the largest flex point. We say that Server j^* is *critical* for Server j.

THEOREM 4 *The worst case delay $d_{p,j}$ at Server j for cells of connections with priority p is*

$$d_{p,j} = \Pi_{p,j} + \sum_{s=1}^{K} \sum_{q=1}^{P} C_{q,s,p,j} d_{q,s}, \tag{4.10}$$

where K is the number of servers in the system, and $\Pi_{p,j}$ and $C_{q,s,p,j}$ are defined as follows:

$$\Pi_{p,j} = \frac{\sum_{i \in C(j), \pi(i,j) \leq p} \beta_i + 1}{1 - \sum_{i \in C(j), \pi(i,j) < p} \rho_i} + \frac{\sum_{i \in C(j), \pi(i,j) \leq p} \rho_i - 1}{1 - \sum_{i \in C(j), \pi(i,j) < p} \rho_i} * \frac{\sum_{i \in C(j^*), \pi(i,j^*) = p} \beta_i}{1 - \sum_{i \in C(j^*), \pi(i,j^*) = p} \rho_i}$$

where $j^* \in P(j)$ is a critical server for Server j, and if $q \neq p$ or $s \neq j$

$$C_{q,s,p,j} = \frac{\sum_{i \in C(j), \pi(i,j) \leq p} \rho_i * \chi_{i,q,s,p,j}}{1 - \sum_{i \in C(j), \pi(i,j) < p} \rho_i} + \frac{\sum_{i \in C(j), \pi(i,j) \leq p} \rho_i - 1}{1 - \sum_{i \in C(j), \pi(i,j) < p} \rho_i} * \frac{\sum_{i \in C(j^*), \pi(i,j^*) = p} \rho_i * \chi_{i,q,s,p,j^*}}{1 - \sum_{i \in C(j^*), \pi(i,j^*) = p} \rho_i},$$

otherwise

$$C_{p,j,p,j} = 0. \tag{4.11}$$

Proof: See the last Section of this Chapter. Q.E.D

The equations in Theorem 4 can be written as a system of equations as follows:

$$\vec{d} \;=\; \vec{\Pi} + \mathcal{C}\cdot\vec{d}\,, \tag{4.12}$$

where $\vec{\Pi} = (\Pi_{1,1}, \Pi_{2,1}, \cdots, \Pi_{P,1}, \cdots, \Pi_{P,K})^\top$, and $\mathcal{C}$ is given as following

$$\begin{bmatrix} 0 & C_{2,1,1,1} & \cdots & C_{P,1,1,1} & \cdots & C_{P,K,1,1} \\ C_{1,1,2,1} & 0 & \cdots & C_{P,1,2,1} & \cdots & C_{P,K,2,1} \\ \vdots & \ddots & \ddots & & \vdots & \\ & & \ddots & 0 & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & C_{P,K,P-1,K} \\ C_{1,1,P,K} & C_{2,1,P,K} & \cdots & C_{P,1,P,K} & \cdots & 0 \end{bmatrix}$$

We simplify the notation for the system of equations (4.12) by denoting

$$z_{p,j}(\vec{d}) \;=\; \Pi_{p,j} + \sum_{s=1}^{K}\sum_{q=1}^{P} C_{q,s,p,j} * d_{q,s} \tag{4.13}$$

and $\vec{Z}(\vec{d}) = [z_{1,1}(\vec{d}), \cdots, z_{P,K}(\vec{d})]^\top$, and

$$\vec{d} \;=\; \vec{Z}(\vec{d}) \;. \tag{4.14}$$

Although Equation (4.12) may at first sight look linear, it is not. The values of $\Pi_{p,j}$ and $C_{q,s,p,j}$ depend on the choice of Server j^* as critical server, and so indirectly depend on the delay d_{i,j^*} on that server.

Equation (4.14) can be solved by using a simple iterative procedure as follows:

Let $\vec{d}^{[0]}$ represent the KP-dimensional delay vector at the beginning of the first iteration, and let $\vec{d}^{[n]}$ the same vector at the end of the n^{th} iteration. Before the first iteration, vector $\vec{d}^{[0]}$ is initialized to be

$$\vec{d}^{[0]} := (1, \quad 1, \quad \ldots, \quad 1)^{\top}. \tag{4.15}$$

During the n^{th} iteration, the new value for $\vec{d}$ is computed as follows:

$$\vec{d}^{[n]} := \vec{Z}(\vec{d}^{[n-1]}). \tag{4.16}$$

In order to demonstrate the convergence of this iterative procedure, we need to estimate the error between $\vec{d}$ and $\vec{d}^{[n]}$, the vector at the end of the n^{th} iteration. That is, we need to establish the difference between the value of $d_{p,j}$ computed at the end of the n^{th} iteration and the real value of $d_{p,j}$. For the iteration procedure to converge, this difference must approximate zero for large values of n. The following theorem gives an estimation of $\vec{d}$ at the end of the n^{th} iteration. We use the notation $\|\cdot\|$ to denote the maximum norm.

In order to simplify the notations, we define $C^k_{q,s,p,j}$ to be the value for $C_{q,s,p,j}$, assuming that Server k in $P(j)$ were to be the critical server j^*, then the upper bound $\tilde{C}_{q,s,p,j}$ on $C_{q,s,p,j}$ can be defined as

$$\tilde{C}_{q,s,p,j} = \max_{k \in P(j)} \left\{ C^k_{q,s,p,j} \right\} \tag{4.17}$$

and, similarly

$$\tilde{\Pi}_{p,j} = \max_{k \in P(j)} \left\{ \Pi^k_{p,j} \right\}. \tag{4.18}$$

THEOREM 5 *For a given priority assignment, if*

$$\nu = \max_{p,j}\left(\sum_{s=1}^{K} \sum_{q=1}^{P} \tilde{C}_{q,s,p,j}\right) < 1 \tag{4.19}$$

then the iterative procedure defined by (4.15) and (4.16) can be used to solve (4.14), and at the end of the n^{th} iteration the following holds:

$$\|\vec{d} - \vec{d}^{[n]}\| \leq \frac{(\nu)^n}{1-\nu} * \|\vec{d}^{[1]} - \vec{d}^{[0]}\| \quad . \tag{4.20}$$

Proof: See the last Section of this Chapter. Q.E.D

The convergence of the iterative procedure follows as a corollary, given that we showed earlier (Theorem 5) that if $\nu < 1$ holds, then

$$\lim_{n\to\infty} \frac{\nu^n}{1-\nu} = 0. \tag{4.21}$$

As the value for n increases, the right-hand side of Equation (4.20) tends to go to zero. Hence, the iterative procedure converges.

As a result we have an effective scheme that - given a set of connections and their routes in a network with arbitrary topology and static-priority scheduling on the links - determines (1) the convergence of the iterative procedure, and (2) the local delays of connections at the switches. This scheme assumes (β, ρ) traffic bounding functions at the entrance to the network and does not rely on traffic regulation inside the network.

D. Final Remarks

In this Chapter, we have studied the worst case delays suffered by traffic in an ATM networks with static priority scheduling and arbitrary topology. We have developed an iterative procedure for computing worst case delays. The convergence of the numerical procedure is formally proved and a closed form for the computation error is derived.

Furthermore, with the proofs of these results, a comprehensive methodology is first introduced into real-time communication theory to analyze the worst case delay. This methodology includes how to take off the max operator in maximum problem; how to set up the nonlinear algebra equations for the mutual dependent local queuing delays; how to validate the existence of unique solution of nonlinear equations; and how to approximate the actual solution and estimate the error of approximation.

This work can be extended in a number of ways. For example, we are currently studying delay computation in connection-based *heterogeneous* networks. To analysis the delay suffered by the traffic in such networks it will be necessary to investigate characterizations of the traffic within the network. Utilizing a consistent traffic characterization function over a series of network segments is a key step in this process.

E. Proofs

1. Proof of Theorem 1

The linear traffic bounding function of connection M_i at the entrance to the network can be formulated as follows:

$$F_{i,s(i,0)}(t) \quad = \quad \min(t, \beta_i + \rho_i * t). \tag{4.22}$$

Since server j is on the route of connection M_i, the total worse case delays experienced by cells of connection M_i after exiting server j is given as $\sum_{g \in G_{i,j}} d_{i,g}$. According to [4], the traffic flow of connection M_i at the output link of server j is bounded by $F_{i,s(i,0)}(t + \sum_{g \in G_{i,j}} d_{i,g})$. Furthermore, the transmission rate of output link is 1, we have the tighter bounding function for traffic flow of connection M_i at the output link of server j:

$$F_{i,j}(t) \quad = \quad \min(t, F_{i,s(i,0)}(t + \sum_{g \in G_{i,j}} d_{\pi(i,g),g})),$$

$$
\begin{aligned}
&= \min(t, \min(t + \sum_{g \in G_{i,j}} d_{\pi(i,g),g}, \beta_i + \rho_i * (t + \sum_{g \in G_{i,j}} d_{\pi(i,g),g}))), \\
&= \min(t, \beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{\pi(i,g),g} + \rho_i * t), \\
&= \min(t, \bar{\beta} + \bar{\rho} * t). \qquad (4.23)
\end{aligned}
$$

where $\bar{\beta} = \beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{\pi(i,g),g}$ and $\bar{\rho} = \rho_i$.

Q.E.D

2. Proof of Theorem 2

For any $i \in \mathcal{M}^*$, according to Theorem 1, the traffic flow of connection M_i entering server j is bounded by $\min(t, \beta_i + \rho_i * \sum_{g \in G_{i,j}, g \neq j} d_{\pi(i,g),g} + \rho_i * t)$. Similar to the proof of Theorem 1, the aggregate traffic flow of subset $\mathcal{M}^*$ of connections on the output link is bounded by following function:

$$
\begin{aligned}
&\min(t, \sum_{i \in \mathcal{M}^*} \min(t + d_{\pi(i,j),j}, \beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{\pi(i,g),g} + \rho_i * t)), \\
&= \min(t, \sum_{i \in \mathcal{M}^*} (\beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{\pi(i,g),g} + \rho_i * t)), \\
&= \min(t, \sum_{i \in \mathcal{M}^*} (\beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{\pi(i,g),g}) + \sum_{i \in \mathcal{M}^*} \rho_i * t), \\
&= \min(t, \bar{\beta} + \bar{\rho} * t). \qquad (4.24)
\end{aligned}
$$

where

$$
\bar{\beta} = \sum_{i \in \mathcal{M}^*} \left(\beta_i + \rho_i \sum_{g \in G_{i,j}} d_{\pi(i,g),g} \right) \qquad (4.25)
$$

$$
\bar{\rho} = \sum_{i \in \mathcal{M}^*} \rho_i. \qquad (4.26)
$$

Q.E.D

3. Proof of Theorem 3

For $k \in P(j)$, let $\mathcal{M}_k = \{i | i \in C(j), \pi(i,j) \leq p, i \in C(k)\}$. According to Theorem 2, the aggregate traffic flow which come from server k and enter server j with priority no lower than p is bounded by

$$\mathcal{J}_{p,k}(t) = \min(t, \sum_{i \in \mathcal{M}_k} (\beta_i + \rho_i * \sum_{g \in G_{i,j}, g \neq j} d_{\pi(i,g),g}) + \sum_{i \in \mathcal{M}_k} \rho_i * t). \quad (4.27)$$

Hence the whole traffic flow entering server j with priority no lower than p is bounded by

$$\sum_{k \in P(j)} \mathcal{J}_{p,k}(t). \quad (4.28)$$

Therefore the maximum busy interval $T_{p,j}$ is the solution of following equation:

$$t = \sum_{k \in P(j)} \mathcal{J}_{p,k}(t). \quad (4.29)$$

We claim that such solution t must satisfy

$$\mathcal{J}_{p,k}(t) = \sum_{i \in \mathcal{M}_k} (\beta_i + \rho_i * \sum_{g \in G_{i,j}, g \neq j} d_{\pi(i,g),g}) + \sum_{i \in \mathcal{M}_k} \rho_i * t. \quad (4.30)$$

for all $k \in P(j)$. Otherwise, we have a contradiction

$$t < \sum_{k \in P(j)} \mathcal{J}_{p,k}(t). \quad (4.31)$$

Hence,

$$t = \sum_{k \in P(j)} (\sum_{i \in \mathcal{M}_k} (\beta_i + \rho_i * \sum_{g \in G_{i,j}, g \neq j} d_{\pi(i,g),g}) + \sum_{i \in \mathcal{M}_k} \rho_i * t). \quad (4.32)$$

i.e.,

$$T_{p,j} = \frac{\sum_{i \in C(j), \pi(i,j) \leq p} [\beta_i + \rho_i * \sum_{g \in G_{i,j}, g \neq j} d_{\pi(i,g),g}]}{1 - \sum_{i \in C(j), \pi(i,j) \leq p} \rho_i}. \quad (4.33)$$

Q.E.D

4. Proof of Theorem 4

First we prove that a solution of (4.8) must also be a solution of (4.13). Assume that $\vec{d}$ is a solution of (4.8). That is, $\vec{d}$ satisfies the follows equation :

$$d_{p,j} = \max_{0<t\leq T_{p,j}} (\sum_{k\in P(j)} \mathcal{J}_{p-1,k}(t+d_{p,j}) + \sum_{k\in P(j)} \mathcal{F}_{p,k}(t) - t) + 1. \tag{4.34}$$

In order to simplify notations, we use $\theta_{p-1,k,j}$ to denote the flex point of traffic constraint function for the traffic come from k-th input link of server j with priority higher than p and $\eta_{p,k,j}$ to denote the traffic constraint function for the traffic come from k-th input link of server j with priority equal to p. Let $\tau_1 = \max_{k\in P(j)}\{\theta_{p-1,k,j}\}$, $\tau_2 = \max_{k\in P(j)}\{\eta_{p,k,j}\}$ and $I_{p,j}^{max} = \max\{\tau_1 - d_{p,j}, \tau_2\}$. Furthermore, let $T_{p-1,j}$ be the maximum busy interval of server j for the aggregate traffic flow which enters the server j and is assigned priority higher than p at server j. It is obvious that $d_{p,j} \geq T_{p-1,j} \geq \tau_1$. Hence $I_{p,j}^{max} = \tau_2$. According the definition of $I_{p,j}^{max}$, it is easy to verify that when $t < I_{p,j}^{max}$,

$$\frac{d(\sum_{k\in P(j)} \mathcal{J}_{q,k}(t+d_{p,j}) + \sum_{k\in P(j)} \mathcal{F}_{p,k}(t))}{dt} > 1. \tag{4.35}$$

On the other hand, when $t > I_{p,j}^{max}$,

$$\frac{d(\sum_{k\in P(j)} \mathcal{J}_{q,k}(t+d_{p,j}) + \sum_{k\in P(j)} \mathcal{F}_{p,k}(t))}{dt} < 1. \tag{4.36}$$

Since $\tau_2 \leq T_{p,j}$, from (4.34), we have

$$d_{p,j} = \sum_{k\in P(j)} \mathcal{J}_{p-1,k}(\tau_2 + d_{p,j}) + \sum_{k\in P(j)} \mathcal{F}_{p,k}(\tau_2) - \tau_2 + 1. \tag{4.37}$$

According to Theorem 2, it is easy to be obtained

$$\tau_2 = \eta_{p,k^*,j}$$

$$= \frac{\sum_{i\in C(k^*),\pi(i,j)=p}\left[\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}\right]}{1-\sum_{i\in C(k^*),\pi(i,j)=p}\rho_i}. \tag{4.38}$$

Furthermore, $\tau_2 + d_{p,j} \geq \tau_1$, we have

$$\begin{aligned}
&\sum_{k\in P(j)} \mathcal{J}_{p-1,k}(\tau_2 + d_{p,j}) \\
=& \sum_{i\in C(j),\pi(i,j)<p} [\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g} + \rho_i * \tau_2 + \rho_i * d_{p,j}] \\
=& \sum_{i\in C(j),\pi(i,j)<p} [\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g} + \rho_i * \tau_2] + \sum_{i\in C(j),\pi(i,j)<p} \rho_i * d_{p,j}
\end{aligned} \tag{4.39}$$

and

$$\sum_{k\in P(j)} \mathcal{F}_{p,k}(\tau_2) = \sum_{i\in C(j),\pi(i,j)=p} [\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g} + \rho_i * \tau_2] \tag{4.40}$$

Now we can solve $d_{p,j}$ from (4.37) and get

$$\begin{aligned}
d_{p,j} =& \frac{\sum_{i\in C(j),\pi(i,j)\leq p}[\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g} + \rho_i * \tau_2] - \tau_2 + 1}{1-\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i} \\
=& \frac{\sum_{i\in C(j),\pi(i,j)\leq p}(\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}) + 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} \\
&+\frac{\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i - 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} * \tau_2 \\
=& \frac{\sum_{i\in C(j),\pi(i,j)\leq p}[\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}] + 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} \\
&+\frac{\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i - 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} * \frac{\sum_{i\in C(k^*),\pi(i,j)=p}\left[\beta_i + \rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}\right]}{1-\sum_{i\in C(k^*),\pi(i,j)=p}\rho_i} \\
=& \frac{\sum_{i\in C(j),\pi(i,j)\leq p}\beta_i + 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} + \frac{\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i - 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} * \frac{\sum_{i\in C(k^*),\pi(i,j)=p}\beta_i}{1-\sum_{i\in C(k^*),\pi(i,j)=p}\rho_i} \\
&+\frac{\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} \\
&+\frac{\sum_{i\in C(j),\pi(i,j)\leq p}\rho_i - 1}{1-\sum_{i\in C(j),\pi(i,j)<p}\rho_i} * \frac{\sum_{i\in C(k^*),\pi(i,j)=p}\left[\rho_i * \sum_{g\in G_{i,j},g\neq j} d_{\pi(i,g),g}\right]}{1-\sum_{i\in C(k^*),\pi(i,j)=p}\rho_i} \\
=& \Pi_{p,j} + \sum_{s=1}^{K}\sum_{q=1}^{P} C_{q,s,p,j} * d_{q,s}.
\end{aligned} \tag{4.41}$$

Hence, if $\vec{d}$ is a solution for (4.8), it is also a solution for (4.13). On the other hand, reversing the above reasoning, the rest of this Theorem is easy to be verified.

Q.E.D

5. Proof of Theorem 5

First, we have following estimation:

$$\begin{aligned} \|\nabla \vec{Z}(\vec{d})\| &= \max_{i=1,\cdots,K} \max_{p=1,\cdots,P} |\sum_{j=1}^{K}\sum_{q=1}^{P} \frac{\partial z_{q,i}}{\partial d_{q,j}}| \\ &\leq \max_{i=1,\cdots,K} \max_{p=1,\cdots,P} |\sum_{j=1}^{K}\sum_{q=1}^{P} \tilde{C}_{q,j,p,i}| \\ &\leq \nu. \end{aligned} \tag{4.42}$$

Hence,

$$\|\vec{Z}(\vec{d}_1) - \vec{Z}(\vec{d}_2)\| \leq \nu * \|\vec{d}_1 - \vec{d}_2\| \tag{4.43}$$

where

$$\|\vec{d}\| = \max_{i=1,\cdots,K} \max_{q=1,\cdots,P} |d_{q,i}|. \tag{4.44}$$

According to Contraction Mapping Principle [34], the solution of $\vec{d} = \vec{Z}(\vec{d})$ can be approximated by (4.16). and the computation error can be evaluated by(4.20) Q.E.D

CHAPTER V

QUEUING DELAY ANALYSIS BY INTEGRATED METHOD

In this Chapter[1], we continue to discuss the queueing delay analysis. As mentioned in previous Chapter, the main disadvantage of Decomposed Method is that it always overestimates the end-to-end delay due to ignore the delay dependency. Since a packet suffers the worst case delay at one server, it may not suffer the worst case delay at the successive server. In order to overcome this disadvantage, a delay analysis method, called as service-curve based method, has been provided [24]. The basic idea in service curve based method is to find a representation of a sequence of servers on the path of the connection as a single server. Successive servers are therefore integrated and dependencies between delays on successive servers can be taken into account. For guaranteed-rate scheduling algorithms, such as fair queueing, delay computation based on service curve based method performs very well. Many currently deployed networks, if they are packet-switched or ATM based, rely on non-guaranteed-rate disciplines, for example, FIFO and static-priority disciplines. We show that for this class of disciplines the service curve based method performs poorly. We propose a new Integrated Approach as alternative to the service curve based method to cluster servers for delay computation purposes, and show in a series of evaluations that this new approach outperforms approaches based on the service curve based method as

[1]Part of this chapter is reprinted with permission from Proceedings of the *IEEE* INFOCOM'97, Kobe, Japan, 1997, vol. 1, pp. 160-167; Proceedings of the 18th IEEE Real-Time Systems Symposium, San Francisco, CA, Dec. 1997, pp. 264-273; and Proceedings of IEEE International Conference on parallel Processing, Minneapolis, MN, Aug. 1998, pp. 432-440. Copyright 1998 by the *IEEE*. (see Appendix A).

well as Decomposed Method.

A. Integrated Delay Analysis for a Subsystem with Two Multiplexors

In this section, we first study a subsystem with two multiplexors, the topology for this subsystem is illustrated in Figure 6. An Integrated Method for delay analysis of FIFO scheduling discipline is presented.

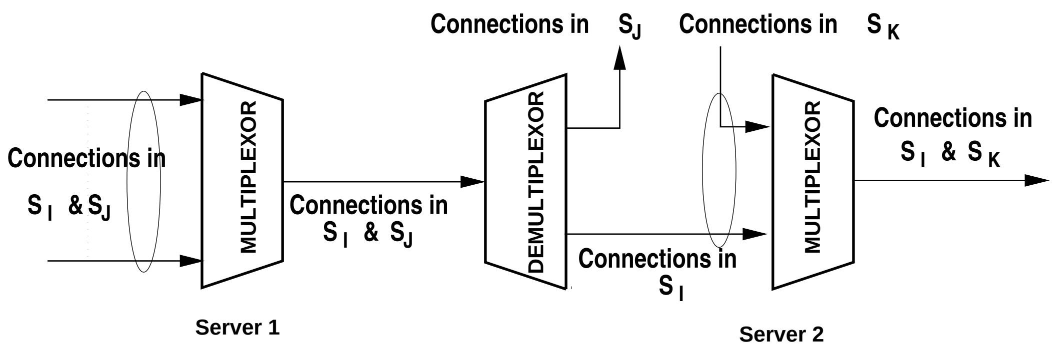


Fig. 6. A subsystem with two multiplexors.

1. Some Definitions and Notations

To evaluate the worst case delay suffered by traffic, the description for network traffic is needed. We give the following definitions and notations for this purpose.

DEFINITION 3 *The arrival traffic function $f_{i,j}(t)$ of connection i at server j is defined as the amount of data arriving at server j from connection i during time interval $[0, t)$.*

DEFINITION 4 *We call function $b_{i,j}(t)$ the constraint traffic function of $f_{i,j}(t)$ if* *for any* $t > 0$ *and* $s > 0$

$$f_{i,j}(t+s) - f_{i,j}(s) \le b_{i,j}(t). \tag{5.1}$$

We use $\mathcal{S}_I$ to denote the set of indexes of all connections which pass server 1 and server 2, use $\mathcal{S}_J$ to denote the set of indexes of all connections which only pass server 1, use $\mathcal{S}_K$ to denote the set of indexes of all connections which only pass server 2.

In this section, we assume that every source traffic is controlled by a leaky bucket, i.e., for $i \in S_I \cup S_J, j = 1$ or $i \in S_K, j = 2$

$$b_{i,j}(t) = \min\{t, \alpha_i + \rho_i * t\} \tag{5.2}$$

2. Internal Network Traffic Characteristics

To analyze the delay suffered by connections' traffic, it is expected to know the output traffic of servers. The following lemma, which was given in [35], solves this problem.

LEMMA 2 *For server i with FIFO scheduling discipline, if the aggregated arrival traffic function* $G_i(t)$ *is known, its output traffic* $W_i(t)$ *can be written as*

$$W_i(t) = \min_{0 \le s \le t}\{t - s + G_i(s)\}, \tag{5.3}$$

where

$$G_i(s) = \sum_{k \in \mathcal{M}_i} f_{k,i}(t), \tag{5.4}$$

and M_i *is the set of connections served by server i.*

On the other hand, if the output traffic of a server is known, it is also expected to know the arriving time for the just leaving data. The following lemma gives the relationship between the output traffic and the data arriving time.

LEMMA 3 *During time interval $[0, t)$, if the total amount of data leaving server i is $W_i(t)$, the time $H_i(t)$ when the $W_i(t)$-th bit comes to server i is given as*

$$H_i(t) = G_i^{-1}(W_i(t)), \tag{5.5}$$

where $G_i(t)$ is given in Lemma 2. Note: $H_i(t) \leq t$.

Proof: It is obvious from the definition of function $G_i(t)$. Q.E.D

In many situations, the characteristics of output traffic of the individual connection is very useful. The following lemma reveals the relationship between the output traffic of an individual connection and the input traffic of that connection.

LEMMA 4 *For server i with FIFO scheduling discipline, if the aggregated arrival traffic function $G_i(t)$ is known, the output traffic of connection k can be written as*

$$f_{k,i}^{out}(t) = f_{k,i}(G_i^{-1}(W_i(t))), \tag{5.6}$$

where $W_i(t)$ is given in Lemma 2.

Proof: For any time t, the amount of data departing the server during $[0, t)$ is $W_i(t)$, and $G_i^{-1}(W_i(t))$ is the time of the $W_i(t)$-th bit arrival at the server. Since the server using FIFO scheduling discipline, we know that $f_{k,i}(G_i^{-1}(W_i(t)))$ bits of data from connection i has departed the server during time interval $[0, t)$. So

$$f_{k,i}^{out}(t) = f_{k,i}(G_i^{-1}(W_i(t))). \tag{5.7}$$

Q.E.D

Furthermore, it plays a key role in delay analysis that when the arriving data leaves the server. The following lemma provides a theoretical solution.

LEMMA 5 *During time interval* $[0, t)$*, if the total amount of data having arrived at server* i *is* $G_i(t)$*, the* $G_i(t)$*-th bit leaves server* i *at time* $W_i^{-1}(G_i(t))$*, where* $W_i(t)$ *is given in Lemma 2.*

Proof: It is obvious from the definition of function $W_i(t)$. Q.E.D

3. Main Results

Now, we can provide the formula to accurately evaluate the end-to-end delay suffered by connections' traffic in the subsystem.

LEMMA 6 *End-to-end delays of connections in* $\mathcal{S}_I$ *are bounded by*

$$d_{\mathcal{S}_I} = \max_{t \geq 0}\{W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))\}, \tag{5.8}$$

where $G_i(t)$ *and* $W_i(t)$*,* $i = 1, 2$*, are given in Lemma 2.*

Proof: During time interval $[0, t)$, the total amount of traffic arriving at server 2 is $G_2(t)$. According to Lemma 5, the $G_2(t)$-th bit leaves server 2 at time $W_2^{-1}(G_2(t))$. Furthermore, these $G_2(t)$ contains $W_1(t)$ bits coming from server 1. According to Lemma 3, the $W_1(t)$-th bit arrives at server 1 at time $G_1^{-1}(W_1(t))$. Therefore, the delay suffered by connections in $\mathcal{S}_I$ at time t is given as $W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))$. So we have

$$d_{\mathcal{S}_I} = \max_{t \geq 0}\{W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t))\}, \tag{5.9}$$

Q.E.D

Since it is very difficult, if not impossible, to precisely describe internal network traffic. The information we have is only about the constraint function of source traffic. The formula provided in above lemma is only of theoretical value. In order to provide a useful Integrated Method, we need to deeply analyze equation (5.8).

Let

- $\bar{G}_1(t) = \sum_{i \in S_I \cup S_J} b_{i,1}(t)$.
- $\bar{W}_1(t) = \min_{0 \leq s \leq t}\{t - s + \bar{G}_1(s)\}$.
- $\bar{H}_1(t) = \bar{G}_1^{-1}(\bar{W}_1(t))$.
- $F_I(t) = \sum_{i \in S_I} b_{i,1}(t)$.
- $F_K(t) = \sum_{i \in S_K} b_{i,2}(t)$.

The following theorem is our important result. It provides an estimation for d_{S_I} in Lemma 6

THEOREM 6 *The delay suffered by connections in $\mathcal{S}_I$ in this subsystem is bounded by*

$$d_{S_I} \leq \max_{0 \leq s \leq \mathcal{B}_1}\{\max_{\mathcal{B}_1 + \mathcal{B}_2 \geq T \geq s}\{s + \min\{T - s, F_I(T - \bar{H}_1(s))\} + F_K(T - s)\} - \min\{T, \bar{G}_1^{-1}(T)\}\}, \tag{5.10}$$

where $\mathcal{B}_i$ is the length of maximum busy period of server i, $i = 1, 2$.

Proof: See the last Section of this Chapter. Q.E.D

From the equation (5.10), we can find that the end-to-end delay d_{S_I} can be computed by only using the constraint functions of source traffic. The drawback in equation (5.8) has been overcome.

B. New Delay Analysis Algorithm for Tandem Networks

Usually, in order to analyze the end-to-end delays suffered by connections in a network, a common technique that consists of two steps is adapted by many researcher. In the first step, a single server analysis technique is developed to estimate the local worst case delay and characterize the output traffic, provided characterizations of all input traffic of the server. In the second step, starting from characterizations of all sources traffic, an iterative process drives the local delay analysis from servers at the edge of the network to those inside the network. The main disadvantage of this technique is that the delay dependency in successive servers on a connection's path is ignored. So the obtained end-to-end delay bounds are very loose and the bursty of internal traffic inside the network is always overestimated.

In this section, we study a new technique which takes into account delay dependency. An accurate and comprehensive methodology for delay analysis of FIFO scheduling discipline is presented.

1. An New Algorithm

For a network, we can partition it into several subnetworks, each of them consists only of two servers. Based on the new method presented in previous section and the property of tandem network, we provide a new delay analysis algorithm for tandem networks. Figure 7 describes this algorithm in detail.

2. Evaluation

We conducted a series of simulations to study the performance of the proposed new method for analysis of end-to-end delay in a tandem network with FIFO scheduling discipline. We are going to demonstrate that our new method generates tighter bounds on end-to-end delays than the approach proposed in [4, 5].

a. Performance Evaluation

In this subsection, we evaluate the performance of the algorithm using the new method discussed in the previous subsection. We will first define a performance metric, then

Algorithm *:*

Step 1: *Partition the network into subnetworks, each of them consists at most of two servers..*

Step 2: *Chose the appropriate order for all subnetworks such that each input traffic of (i+1)-th subnetwork can be estimated by all input traffic of subsystems with order less than (i+1)-th.*

Step 3: *While loop from the first subnetwork to the last subnetwork, do following substeps:*

Step 3.1: *Computer the delay bounds suffered by connections in the i-th subnetwork.*

Step 3.2: *Estimate the all out put traffic of this subnetwork.*

Step 4: *Compute the end-to-end delays for each connection by summarize all local delays suffered at every subnetwork.*

Fig. 7. New delay analysis algorithm for FIFO scheduling.

describe the system configuration considered and present the simulation results.

Performance Metric. We quantify the performance of algorithms by measuring two quantities. One is the *end-to-end delay* $d_X(U)$ estimated by the algorithm X for the end-to-end delay suffered by the connection which travels the longest path in the network under the work load U. The other is called the *relative improvement*

$R_{X,Y}(U)$, which is used to compare two algorithms and is expressed as

$$R_{X,Y}(U) = \frac{d_X(U) - d_Y(U)}{d_X(U)}. \tag{5.11}$$

Topology and Traffic Descriptions. In our evaluation, we only consider a simple tandem network which consists of n 3×3 switches and these switches are connected into a chain. A example of such tandem network with 8 switches is showed in figure 8. There are $2n + 1$ sessions in this network. Session 0 is the longest, which enters the network by the middle input port of the first switch and exits the network from the middle output port of the n-th switch. For $k = 0$ to $n - 1$, the $(2k + 1)$-th session enters the network by the upper input port of the k-th switch and exists the network from the upper output port of the $(k + 1)$-th switch; the $(2k + 2)$-th session enters the network by the lower input port of the k-th switch and exists the network from the upper lower output port of the $(k + 2)$-th switch. It is easy to find that the middle output port of each switch, excepted the first one, is competed by four sessions including session 0. In order to simplify evaluation, we assume that every source traffic is controlled by a leaky bucket with the token arrival rate $\frac{U}{4}$, where U is the work load of the network.

b. Delay Computation

In Appendix, we summarize the formulas used for the delay calculation in the decomposition based and service-curve based approach as described in [36, 37, 38, 39, 40, 4, 5, 24, 41, 42, 43, 44, 45, 46]. We use these formulas to derive closed forms for the worst-case delay for Connection 0 in the topology used in these experiments. We call Algorithm *Decomposed* for the decomposed approach and Algorithm *Service Curve* for the service-curve based approach.

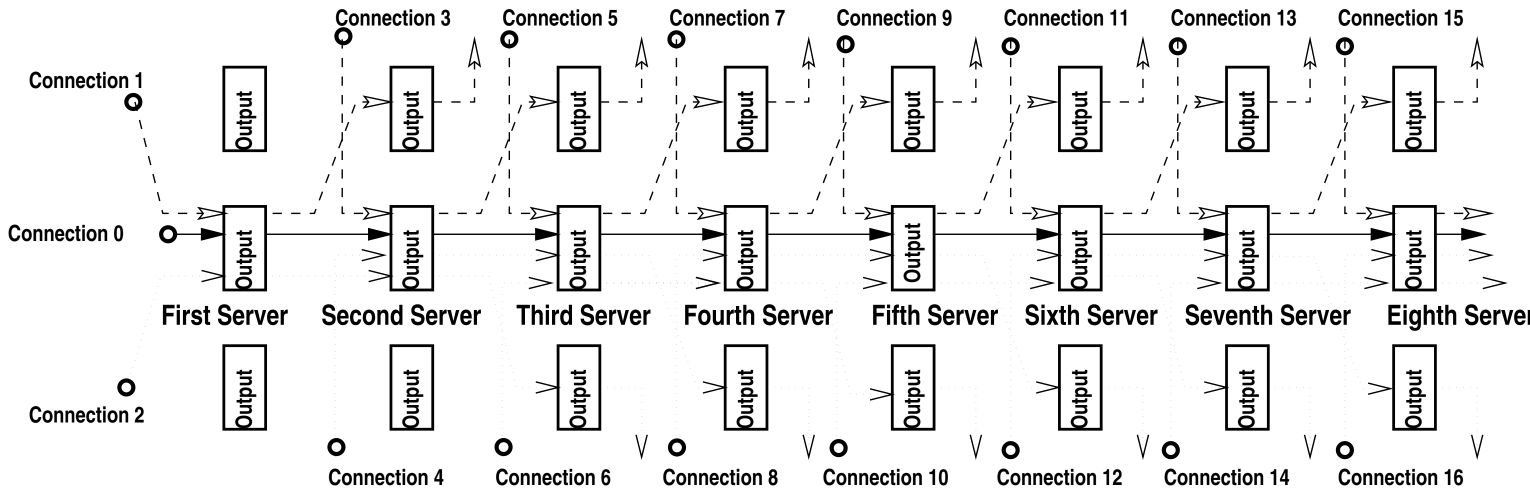


Fig. 8. Connection-server graph representation.

Algorithm *Decomposed.* We derive the worst-case end-to-end delay of Connection 0 by adding the local delays on the servers along its path. For this, we let E_k be the local delay suffered by traffic of Connection 0 at Server k. In the last Section of this Chapter, we derive the following equations for E_k:

$$E_1 = \frac{2\alpha}{1-\rho}; \qquad E_2 = \alpha\frac{3-\rho+4\rho^2}{(1-\rho)^2}$$

$$E_k = 3\alpha + \rho E_{k-1} + 3\rho\frac{\alpha + \rho\sum_{i=1}^{k-1} E_i}{1-\rho}, k \geq 3$$

The end-to-end delay D_0^D for Connection 0 using Algorithm *Decomposed* is then obtained by adding the local delays:

$$D_0^D = \sum_{k=1}^{n} E_k$$

Algorithm *Service Curve.* The delay calculation in this approach is based on the definition for the service curve given in [24]. As we compare the performance of the various approaches for a network with pre-defined servers (FIFO servers in this case), service curve for pre-defined non-guaranteed-rate disciplines is not available. We must use an induced service curve approach, where we derive the service curve from the scheduling policy used in the server. The performance of such a method, however, greatly depends on how tight service curves can be defined for a given service discipline. In the last Section of this Chapter, we derive an *upper bound* on the service curve for a FIFO server, which in turn provides a lower bound for the end-to-end delay D_0^{SC} for Connection 0 with the service curve method. As we derive in Appendix, the worst case delay D_0^{SC} is lower-bounded by the following expression:

$$D_0^{SC} \geq \frac{2\alpha}{1-2\rho} + \frac{\alpha(3-2\rho)}{(1-\rho)(1-3\rho)} + \frac{(n-2)\alpha(3-\rho)}{(1-\rho)(1-3\rho)}.$$

It is important to emphasize at this point that the following comparisons are between *upper bounds* on end-to-end delays for both Algorithm *Integrated* and Algorithm *Decomposed*, and *lower bounds* for Algorithm *Service Curve*. The results for the performance of Algorithm *Service Curve*, both in terms of end-to-end delays and in terms of relative performance, must therefore be considered as optimistic.

c. Numerical Results and Observations

The results of our experiments comparing the performance of the three approaches are depicted in Figures 9, 10, and 11. Figure 9 compares the service-curve based approach to the decomposition-based approach and illustrates how the former is not well suited (as was to be expected) for analyzing non-guaranteed-rate service disciplines, e.g., FIFO. As the network load increases, the inadequacy of modeling a FIFO server with a service curve becomes evident.

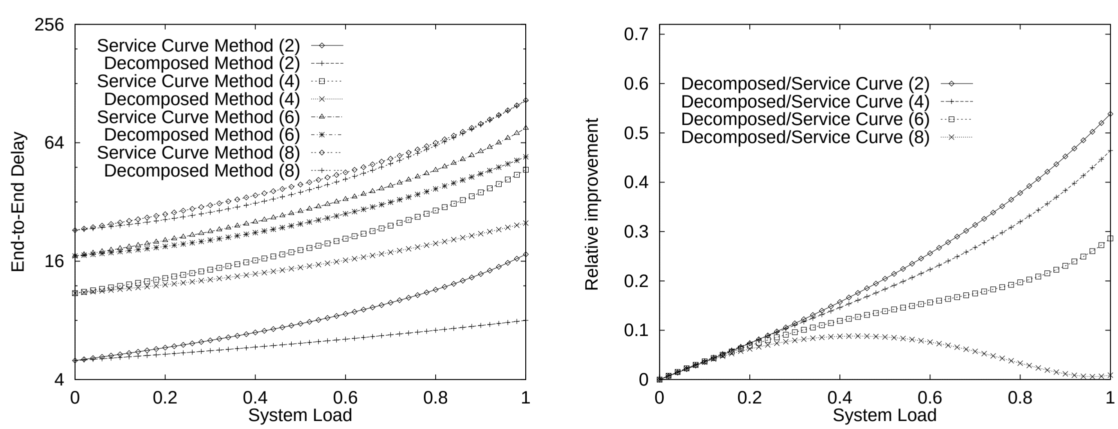


Fig. 9. Comparison between decomposed method and service curve based method.

From Figure 10 we see that Algorithm *Integrated* always outperforms Algo-

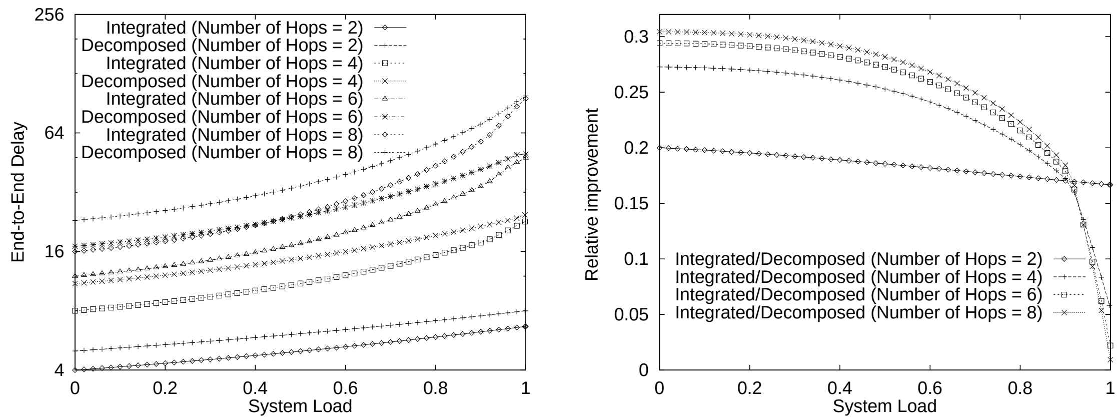


Fig. 10. Comparison between integrated method and decomposed method.

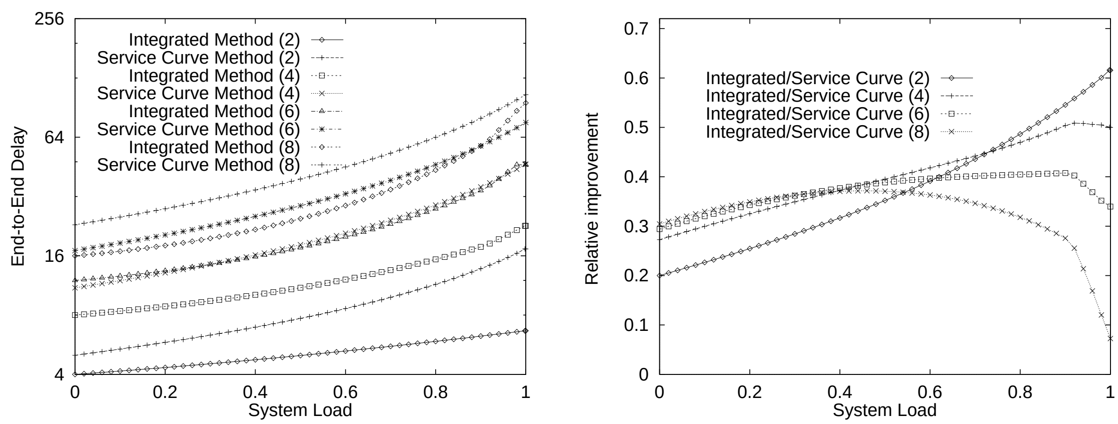


Fig. 11. Comparison between integrated method and service curve based method.

rithm *Decomposed*. Furthermore, for loads up to 80%, the performance improvement increases with growing network size. This is expected as Algorithm *Integrated* takes delay dependences within server pairs into account.

While the performance improvement of Algorithm *Integrated* over Algorithm *Service Curve* can be inferred by transitivity, we show a comparison in Figure 11 for illustrative purposes. The results of this experiment show that the performance gains are significant, except for large systems under high load.

C. Final Remarks

In this Chapter, we have proposed a new method for deriving the end-to-end delay bounds for connections in a tandem network, which uses a FIFO scheduling discipline. Our new method takes into account delay dependencies in successive servers along the path of a connection, which is in general very difficult for delay analysis, and achieves better performance than the method provided in [4, 5]. This can be observed through the extensive simulation experiments provided in previous section.

When servers do not have traffic regulation mechanisms (as is the fact with all work conserving servers), virtual circular dependencies among connections introduce feedback effects on local delays, which in turn show up as non-linearities in the local delay calculations. For this reason, the analysis method described in this paper is limited to sets of connections that do not generate cycles in the network. Based on our previous work on decomposition-based analysis with feedback effects of networks with both FIFO and static-priority servers, we are currently working on extending the approach proposed in this Chapter to general networks.

D. Proofs

1. Proof of Theorem 6

We only need to consider two cases.

1. Case 1: t belongs to a time interval in which busy periods of server 1 and server 2 do not overlap each other. (see Figure 12).

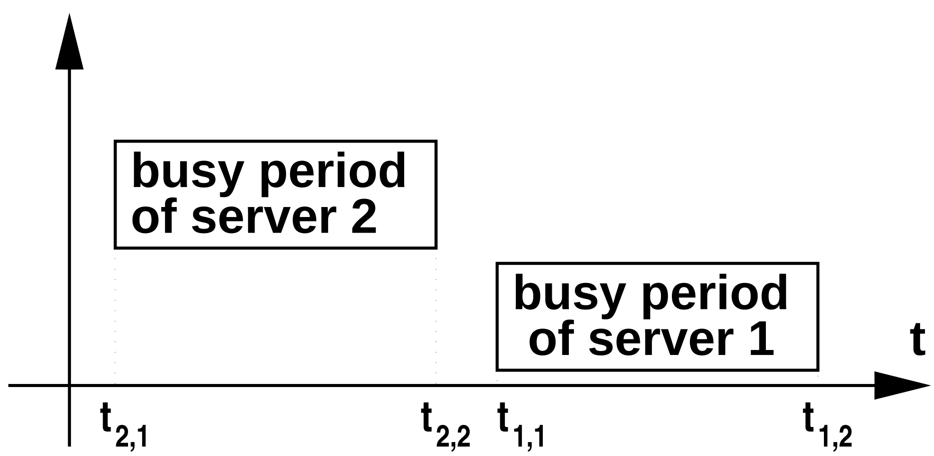


Fig. 12. Case 1.

If $t \in [t_{1,1}, t_{1,2}]$, because server 2 is not at busy state, the time of the $G_2(t)$-th bit leaving server 2 can be evaluated as

$$W_2^{-1}(G_2(t)) \quad \leq \quad t. \tag{5.12}$$

On the other hand, the time of the $W_1(t)$-th bit arriving at server 1 can be evaluated as

$$G_1^{-1}(W_1(t)) \quad \geq \quad t_{1,1} + \bar{G}_1(t - t_{1,1}). \tag{5.13}$$

By using $\bar{H}_1(0) = 0$, we have

$$\begin{aligned}
& W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t)) \\
\leq \; & t - t_{1,1} - \bar{G}_1(t - t_{1,1}) \\
& let \;\; T = t - t_{1,1} \\
= \; & T - \bar{G}_1^{-1}(T) \\
\leq \; & \max_{\mathcal{B}_1 \geq T \geq 0} \{T - \bar{G}_1^{-1}(T)\} \\
\leq \; & \max_{0 \leq s \leq \mathcal{B}_1} \{ \max_{\mathcal{B}_1 + \mathcal{B}_2 \geq T \geq s} \{s + \min\{T - s, F_I(T - \bar{H}_1(s))\} + F_K(T - s)\} \\
& - \min\{T, \bar{G}_1^{-1}(T)\}\}.
\end{aligned} \tag{5.14}$$

If $t \in [t_{2,1}, t_{2,2}]$, because server 1 is not at busy state, the delay suffered by the traffic of connections in $\mathcal{S}_I$ is bounded by

$$\begin{aligned}
d \;\; \leq \; & \max_{t \geq 0} \{t_{2,1} + \min\{t - t_{2,1}, F_I(t - t_{2,1})\} + F_K(t - t_{2,1}) - t\} \\
& let \;\; T = t - t_{2,1} \\
\leq \; & \max_{\mathcal{B}_2 \geq T \geq 0} \{\min\{T, F_I(T)\} + F_K(T)\} - T\} \\
\leq \; & \max_{0 \leq s \leq \mathcal{B}_1} \{ \max_{\mathcal{B}_1 + \mathcal{B}_2 \geq T \geq s} \{s + \min\{T - s, F_I(T - \bar{H}_1(s))\} + F_K(T - s)\} \\
& - \min\{T, \bar{G}_1^{-1}(T)\}\}.
\end{aligned} \tag{5.15}$$

If $t \in [t_{2,2}, t_{1,1}]$, the queuing delay suffered by connections in $\mathcal{S}_I$ at this subsystem is zero.

2. Case 2: t belongs to a time interval in which busy periods of server 1 and server 2 overlap each other. (see Figure 13).

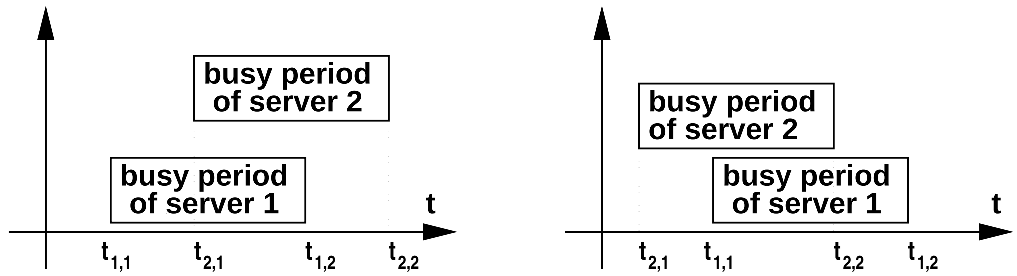


Fig. 13. Case 2.

First, we consider the situation $t_{1,1} \leq t_{2,1}$. If $t \in [t_{2,1}, t_{2,2}]$, the time of $G_2(t) - th$ bit leaving server 2 can be evaluated as

$$\begin{aligned} W_2^{-1}(G_2(t)) &= t_{2,1} + G_2(t) - G_2(t_{2,1}) \\ &\leq t_{2,1} + \min\{t - t_{2,1}, F_I(t - \bar{H}_1(t_{2,1})) + F_K(t - \bar{H}_1(t_{2,1}))\} \\ &\quad + b_{4,2}(t - t_{2,1}). \end{aligned} \tag{5.16}$$

On the other hand, the time of the $W_1(t) - th$ bit arriving at server 1 can be evaluated as

$$G_1^{-1}(W_1(t)) \geq t_{1,1} + \bar{G}_1^{-1}(t - t_{1,1}). \tag{5.17}$$

So, we have

$$\begin{aligned} & W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t)) \\ \leq\ & t_{2,1} + \min\{t - t_{2,1}, F_I(t - \bar{H}_1(t_{2,1}))\} + F_K(t - t_{2,1}) - t_{1,1} - \bar{G}_1^{-1}(t - t_{1,1}) \\ & let \;\; s = t_{2,1} - t_{1,1} \;\; and \;\; T = t - t_{1,1}, \\ =\ & s + \min\{T - s, F_I(T - \bar{H}_1(s))\} + F_K(T - s) - \bar{G}_1^{-1}(T) \end{aligned}$$

$$\leq \max_{0\leq s\leq \mathcal{B}_1}\{\max_{\mathcal{B}_1+\mathcal{B}_2\geq T\geq s}\{s+\min\{T-s, F_I(T-\bar{H}_1(s))\}+F_K(T-s)\}$$
$$-\min\{T, \bar{G}_1^{-1}(T)\}\}\}. \tag{5.18}$$

where $\mathcal{B}$ is the length of maximum busy period of server 1. If $t \in [t_{1,1}, t_{2,1}]$, this situation has been discussed in Case 1.

Finally, we consider the situation $t_{1,1} > t_{21}$. Similarly, if $t \in [t_{1,1}, t_{1,2}]$, we have

$$\begin{aligned} W_2^{-1}(G_2(t)) &= t_{2,1} + G_2(t) - G_2(t_{2,1}) \\ &\leq t_{2,1} + \min\{t - t_{2,1}, F_I(t - t_{2,1})\} + F_K(t - t_{2,1}). \end{aligned} \tag{5.19}$$

and

$$G_1^{-1}(W_1(t)) \geq t_{2,1} + \bar{G}_1^{-1}(t - t_{2,1}). \tag{5.20}$$

So,

$$\begin{aligned} & W_2^{-1}(G_2(t)) - G_1^{-1}(W_1(t)) \\ \leq\ & t_{2,1} + \min\{t - t_{2,1}, F_I(t - t_{2,1})\} + F_K(t - t_{2,1}) - t_{2,1} - \bar{G}_1^{-1}(t - t_{2,1}) \\ & \quad T = t - t_{2,1}, \quad we \quad have \\ =\ & \min\{T, F_I(T)\} + F_K(T) - \bar{G}_1^{-1}(T) \\ \leq\ & \max_{0\leq s\leq \mathcal{B}_1}\{\max_{\mathcal{B}_1+\mathcal{B}_2\geq T\geq s}\{s+\min\{T-s, F_I(T-\bar{H}_1(s))\}+F_K(T-s)\} \\ & -\min\{T, \bar{G}_1^{-1}(T)\}\}\}. \end{aligned} \tag{5.21}$$

If $t \in [t_{2,2}, t_{1,2}]$, this situation has been discussed in Case 1.

Q.E.D

2. An Application of Service Curve Method

We summarize the service curve method presented in [37, 24, 46, 17]. Consider server k serving n+1 connections. Without loss of generality, let $f_{i,k}^{in}(t)$ be the input traffic function of connection i at server k and $F_{i,k}^{in}(t) = \min\{t, a_{i,k} + \rho_{i,k} t\}$ be the constraint function of input traffic $f_{i,k}^{in}(t)$, and let $f_{i,k}^{out}(t)$ be the output traffic of connection i at server k and $F_{i,k}^{out}(t)$ be the constraint function of output traffic $f_{i,k}^{out}(t)$.

DEFINITION 5 *[37, 24] A non-negative function $s_{i,k}(t)$ is called as the service curve of connection i offered by server k if for any time t*

$$f_{i,k}^{out}(t) \geq \min_{0 \leq s \leq t} \{f_{i,k}^{in}(s) + s_{i,k}(t-s)\}. \tag{5.22}$$

One of the advantages of using service curve is to effectively evaluate the end-to-end delay suffered by the traffic. The following theorem reveals this fact.

THEOREM 7 *[24] Suppose that connection i passes through m servers and the k-th server offers the connection a service curve $s_{i,k}(t)$, $k = 1, \cdots, m$, and suppose that the constraint function of the input traffic of connection i to the network system is $F_i(t)$. Then the end-to-end delay of connection i is bounded by*

$$D_i = \max_{t \geq 0} \{S_i^{-1}(t) - F_i^{-1}(t)\}, \tag{5.23}$$

where $S_i(t)$ is called as the network service curve of connection i and is defined as

$$\begin{aligned} S_i(t) &= \min\{\sum_{k=1}^{m} s_{i,k}(t_k) \;\; \| \;\; t_k \geq 0, \sum_{k=1}^{m} t_k = t\} \\ &= s_{i,1} \otimes s_{i,2} \cdots \otimes s_{i,m}(t). \end{aligned} \tag{5.24}$$

Now, we use the above results to study the example presented in Section B. Let

$$g_k(t) = \sum_{i=0}^{n} f_{i,k}^{in}(t), \tag{5.25}$$

and

$$w_k(t) = \min_{0 \le s \le t}\{t - s + g_k(s)\}. \tag{5.26}$$

LEMMA 7 *A non-negative function $s_{i,k}(t)$ is a service curve of connection i offered by server k if for any time t,*

$$f^{in}_{i,k}(g_k^{-1}(w_k(t))) \ge \min_{0 \le s \le t}\{f^{in}_{i,k}(s) + s_{i,k}(t-s)\}. \tag{5.27}$$

Proof: For any time t, the amount of data departing the server during $[0, t)$ is $w_k(t)$, and $g_k^{-1}(w_k(t))$ is the time of the $w_k(t)$-th bit arrival at the server. Since the server using FIFO scheduling discipline, we know that $f^{in}_{i,k}(g_k^{-1}(w_k(t)))$ bits of data from connection i has departed the server during time interval $[0, t)$. So

$$f^{out}_{i,k}(t) = f^{in}_{i,k}(g_k^{-1}(w_k(t))). \tag{5.28}$$

Therefore, by the definition, we know that if

$$f^{in}_{i,k}(g_k^{-1}(w_k(t))) \ge \min_{0 \le s \le t}\{f^{in}_{i,k}(s) + s_{i,k}(t-s)\}, \tag{5.29}$$

then $s_{i,k}(t)$ is a service curve of connection i offered by server k. Q.E.D

Let

$$\bar{G}_{i,k}(t) = \sum_{j=0, j \ne i}^{n} F^{in}_{j,k}(t), \tag{5.30}$$

and

$$B_{i,k} = \max_{t \ge 0}\{t \quad \| \quad \bar{G}_{i,k}(t) > t\} = \frac{\sum_{j=0,j\ne i}^{n} a_{j,k}}{1 - \sum_{j=0,j\ne i}^{n} \rho_{j,k}}. \tag{5.31}$$

The definition for service curve is very loose. For a connection, there are infinite functions which satisfy the inequality (5.22). For example, a function with zero value is a service curve by the definition. After given the scheduling discipline for a server, how to find the maximum service curves for each connection is an interesting and challenge problem. Since it is very difficult, if not impossible, to obtain the maximum service curve when the server uses FIFO scheduling discipline, we try to estimate the service curve.

THEOREM 8 *If $s_{i,k}(t)$ is a service curve for connection i offered by server k, then $s_{i,k}(t) = 0$ when $t \in [0, B_{i,k}]$.*

Proof: For any $\epsilon > 0$, if $s_{i,k}(t) \geq \epsilon t$ for $t \in [0, B_{i,k}]$, let $f_{i,k}^{in}(t) = \min\{\epsilon t, F_{i,k}^{in}(t)\}$, we have that

$$f_{i,k}^{in}(t) = \min_{0 \leq s \leq t}\{f_{i,k}^{in}(s) + s_{i,k}(t-s)\}. \tag{5.32}$$

On the other hand, let $f_{j,k}^{in}(t) = F_{j,k}^{in}(t)$ for all $j \neq i$, it is easy to know that

$$g_k(t) = \sum_{j=0}^{n} f_{j,k}^{in}(t) = \sum_{j=0, j\neq i}^{n} f_{j,k}^{in}(t) + \epsilon t \geq (1+\epsilon)t, \qquad 0 \leq t \leq B_{i,k}, \tag{5.33}$$

and

$$w_k(t) = t, \qquad t \in [0, B_{i,k}]. \tag{5.34}$$

Therefore, we have that

$$g_k^{-1}(w_k(t)) = g_k^{-1}(t) \leq \frac{t}{1+\epsilon}, \qquad 0 \leq t \leq B_{i,k}, \tag{5.35}$$

and

$$\begin{aligned} f_{i,k}^{out}(t) &= f_{i,k}^{in}(g_k^{-1}(w_k(t))) \\ &\leq f_{i,k}^{in}(\frac{t}{1+\epsilon}) \\ &\leq \epsilon\frac{t}{1+\epsilon} \\ &< \epsilon t \\ &= f_{i,k}^{in}(t) \\ &= \min_{0 \leq s \leq t}\{f_{i,k}^{in}(s) + s_{i,k}(t-s)\}. \end{aligned} \tag{5.36}$$

This is a contradiction to Lemma 7. Q.E.D

COROLLARY 1 *The service curve* $s_{i,k}(t)$ *of connection* i *is upper bounded by* $s^*_{i,k}(t)$, *where*

$$s^*_{i,k}(t) = \begin{cases} 0, & t \leq B_{i,k} \\ t - B_{i,k}, & t \geq B_{i,k} \end{cases} \quad (5.37)$$

and $B_{i,k}$ *is defined in (5.31).*

Proof: According to Theorem 8, we have that $s_{i,k}(t) = 0,\ for\ t \in [0, B_{i,k}]$. For $t \geq B_{i,k}$, the maximum available service for connection i during time interval $[0, t)$ is $t - B_{i,k}$. Therefore $s_{i,k}(t) \leq s^*_{i,k}(t)$ for all $t \geq 0$. Q.E.D

In order to use the above results, we need to accurately estimate the internal network traffic. Following Lemma 8 gives a tight estimation about the constraint function of output traffic of each connection.

LEMMA 8 *The constraint function* $F^{out}_{i,k}(t)$ *of output traffic* $f^{out}_{i,k}(t)$ *of connection* i *at server* k *is lower bounded by* $\min\{t, F^{in}_{i,k}(t + q_{i,k})\}$, *i.e., for any* $t \geq 0$,

$$F^{out}_{i,k}(t) \geq \min\{t, F^{in}_{i,k}(t + q_{i,k})\}, \quad (5.38)$$

where

$$q_{i,k} = \max_{t \geq 0}\{\sum_{j \neq i} F^{in}_{j,k}(t) - t\}. \quad (5.39)$$

Proof: Let t^* be the time such that $q_{i,k} = \sum_{j \neq i} F^{in}_{j,k}(t^*) - t^*$. Without loss of generality, we assume that $f^{in}_{j,k}(t) \equiv F^{in}_{j,k}(t)$ for $t \in [0, t^*)$ and $j \neq i$, $f^{in}_{i,k}(t) \equiv 0$ for $t \in [0, t^*)$. After time t^*, we assume that $f^{in}_{j,k}(t) \equiv f^{in}_{j,k}(t^*)$ for $t \in [t^*, \infty)$ and $j \neq i$, $f^{in}_{i,k}(t) \equiv F^{in}_{i,k}(t - t^*)$ for $t \in [t^*, \infty)$. Since at time t^*, the queue length of server k is $q_{i,k}$, according to FIFO scheduling discipline, during $[0, t^* + q_{i,k})$, the output traffic of connection i is zero. But after time $t^* + q_{i,k}$, the output traffic of connection j ($j \neq i$) is zero. Hence

$$f_{i,k}^{out}(t) = \min\{t - t^* - q_{i,k}, F_{i,k}^{in}(t - t^*)\}, \tag{5.40}$$

and

$$f_{i,k}^{out}(t) - f_{i,k}^{out}(t^* + q_{i,k}) = \min\{t - t^* - q_{i,k}, F_{i,k}^{in}(t - t^* - q_{i,k} + q_{i,k})\}. \tag{5.41}$$

Therefore we have that

$$F_{i,k}^{out}(t) \geq \min\{t, F_{i,k}^{in}(t + q_{i,k})\}. \tag{5.42}$$

Q.E.D

Now we can approximate the service curves offered by servers to connection 0 in the example presented in Section B.

THEOREM 9 *The service curve $s_{0,k}(t)$ of connection 0 offered by the k-th server is bounded by $s_{0,k}^*(t)$, where*

$$s_{0,k}^*(t) = \begin{cases} 0, & t \leq B_{0,k} \\ t - B_{0,k}, & t \geq B_{0,k} \end{cases} \tag{5.43}$$

and

$$B_{0,k} \geq \begin{cases} \frac{2\alpha}{1-2\rho}, & if\ k = 1 \\ \alpha\frac{3-2\rho}{(1-\rho)(1-3\rho)}, & if\ k = 2 \\ \alpha\frac{3-\rho}{(1-\rho)(1-3\rho)}, & if\ k > 2 \end{cases} \tag{5.44}$$

Proof: According to Theorem 8, for $k = 1$, we have that

$$B_{0,1} = \max\{t \ \ \| \ \ 2F(t) > t\}$$

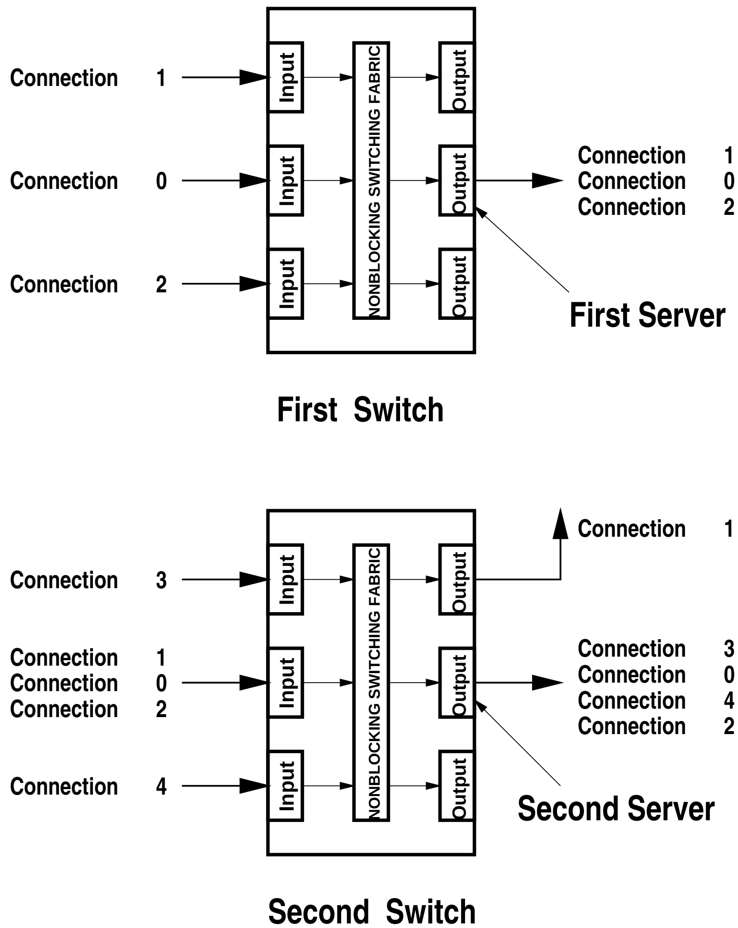


Fig. 14. Connections at the first switch and the second switch.

$$= \frac{2\alpha}{1-2\rho}. \tag{5.45}$$

For $k = 2$, we know that the traffic constraint functions for connection 3 and 4 are the same as $F(t) = \min\{t, \alpha + \rho t\}$. Furthermore, by Lemma 8, the constraint function of connection 2 is lower bounded by $\min\{t, F(t + q_{2,1})\}$, where $q_{2,1}$ is the maximum queue length of the middle output port of the first switch when it only serves connection 0 and connection 1 and can be easy evaluated as

$$\begin{aligned} q_{2,1} &= \max_{t\geq 0}\{F^{in}_{0,1}(t) + F^{in}_{1,1}(t) - t\} \\ &\geq \max_{t\geq 0}\{2F(t) - t\} \\ &= \frac{\alpha}{1-\rho}. \end{aligned} \tag{5.46}$$

Substitute $q_{2,1} = \frac{\alpha}{1-\rho}$ into $F(t + q_{2,1})$, we have that $\min\{t, F(t + q_{2,1})\} = \min\{t, \alpha + \rho\frac{\alpha}{1-\rho} + \rho t\}$. Therefore by Theorem 8, we obtain that

$$\begin{aligned} B_{0,2} &\geq \max\{t \quad \| \quad 2F(t) + \min\{t, F(t + q_{2,1})\} > t\} \\ &= \frac{3\alpha + \rho q_{2,1}}{1-3\rho} \\ &= \alpha\frac{3-2\rho}{(1-\rho)(1-3\rho)}. \end{aligned} \tag{5.47}$$

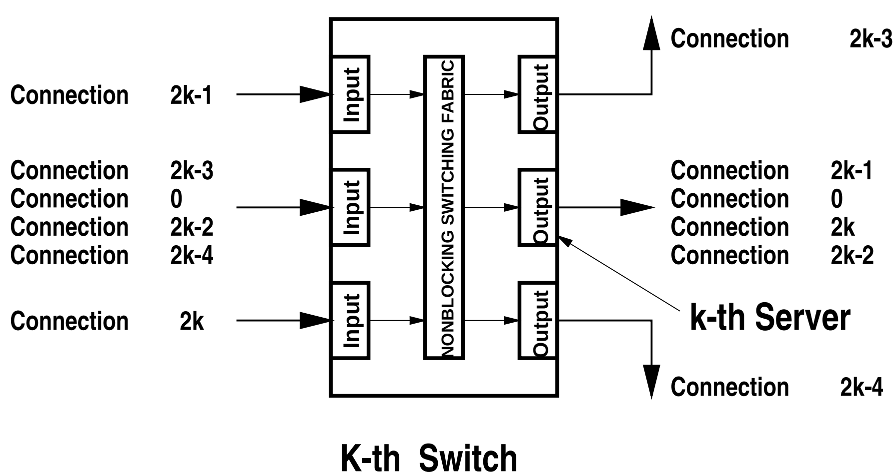


Fig. 15. Connections at the k-th switch.

For $k > 2$, similarly, we can obtain that

$$\begin{aligned} B_{0,k} &\geq \max\{t \quad \| \quad 2F(t) + \min\{t, F(t + q_{2k-2,k-1})\} > t\} \\ &= \frac{3\alpha + \rho q_{2k-2,k-1}}{1 - 3\rho}. \end{aligned} \tag{5.48}$$

where $q_{2k-2,k-1}$ is the maximum queue length of the middle output port of the $(k-1)$-th server when it only serves connection 0, connection (2k-3), and connection (2k-4) and can be easy evaluated as

$$\begin{aligned} q_{2k-2,k-1} &= \max_{t \geq 0}\{F^{in}_{0,k-1}(t) + F^{in}_{2k-3,k-1}(t) + F^{in}_{2k-4,k-1}(t) - t\} \\ &\geq \max_{t \geq 0}\{3F(t) - t\} \\ &= \frac{2\alpha}{1 - \rho}. \end{aligned} \tag{5.49}$$

Therefore, we have

$$B_{0,k} \;\geq\; \alpha\frac{3 - \rho}{(1 - \rho)(1 - 3\rho)}. \tag{5.50}$$

Q.E.D

According to Theorem 8, network service curve $S_0(t)$ of connection 0 can be estimated as following:

$$\begin{aligned} S_0(t) &= s_{0,1} \otimes s_{0,2} \cdots \otimes s_{0,n}(t) \\ &\leq s^*_{0,1} \otimes s^*_{0,2} \cdots \otimes s^*_{0,n}(t) \\ &= \begin{cases} 0, & t \leq \sum_{k=1}^{n} B_k \\ t - \sum_{k=1}^{n} B_k, & t \geq \sum_{k=1}^{n} B_k \end{cases} \end{aligned} \tag{5.51}$$

Therefore

$$D_0^{SC} = \max_{t \geq 0}\{S_0^{-1}(t) - F^{-1}(t)\}$$

$$\geq \sum_{k=1}^{n} B_k$$
$$\geq \frac{2\alpha}{1-2\rho} + \alpha\frac{3-2\rho}{(1-\rho)(1-3\rho)} + \alpha\frac{(n-2)(3-\rho)}{(1-\rho)(1-3\rho)}. \tag{5.52}$$

3. An Application of Decomposed Method

In order to compare our new method with the decomposed method, we summarize the decomposed method discussed in [4, 5]. Assume that server k serves $n+1$ connections. Similarly, let $f_{i,k}^{in}(t)$ and $f_{i,k}^{out}(t)$ be the input and output traffic functions of connection i, and $F_{i,k}^{in}(t) = \min\{t, \alpha_{i,k} + \rho_{i,k}t\}$ and $F_{i,k}^{out}(t)$ be the constraint functions of input and output traffic of connection i, for $i = 0, 1, \cdots, n$. Furthermore, we denote E_k as the local delay bound suffered by all input traffic at server k.

THEOREM 10 *[4, 5] The local delay bound suffered by all input traffic at server k can be evaluated as*

$$E_k = \max_{t\geq 0}\{\sum_{i=0}^{n} F_{i,k}^{in}(t) - t\}$$
$$= \sum_{i=0, i\neq i^*}^{n} (\alpha_{i,k} + \rho_{i,k}X^*), \tag{5.53}$$

where

$$X^* = \frac{\alpha_{i^*,k}}{1-\rho_{i^*,k}} = \max\{\frac{\alpha_{i,k}}{1-\rho_{i,k}} \| i = 0, \cdots, n\}. \tag{5.54}$$

Furthermore, the constraint function of output traffic of connection i can be evaluated by

$$F_{i,k}^{out}(t) = F_{i,k}^{in}(t + E_k). \tag{5.55}$$

Now we use the decomposed method to study the example presented in Section 4.

According to the mechanism of the output port of a switch, we assign the middle output port of the k-th switch as the k-th server in the network. We also denote E_k

as the local delay bound suffered by all traffic at the k-th server. Using the above theorem, we have that

$$\begin{aligned} E_1 &= \max_{t \geq 0}\{3F(t) - t\} \\ &= \frac{2\alpha}{1-\rho}, \end{aligned} \tag{5.56}$$

and

$$\begin{aligned} E_2 &= \max_{t \geq 0}\{2F(t) + 2F(t+E_1) - t\} \\ &= 2\alpha + \frac{1+2\rho}{1-\rho}(\alpha + \rho E_1) \\ &= \alpha\frac{3-\rho+4\rho^2}{(1-\rho)^2}, \end{aligned} \tag{5.57}$$

and for $k > 2$

$$\begin{aligned} E_k &= \max_{t \geq 0}\{2F(t) + F(t+E_{k-1}) + F(t + \sum_{i=1}^{k-1} E_i) - t\} \\ &= 3\alpha + \rho E_{k-1} + 3\rho\frac{\alpha + \rho\sum_{i=1}^{k-1} E_i}{1-\rho}. \end{aligned} \tag{5.58}$$

Therefore, the end-to-end delay bound suffered by connection 0 traffic and evaluated by the decomposed method is obtained as

$$D_0^* = \sum_{k=1}^{k=n} E_k. \tag{5.59}$$

CHAPTER VI

NEW STATIC PRIORITY ASSIGNMENT ALGORITHM

In this Chapter[1], we study *static-priority scheduling.* Traditionally, FIFO (first-come-first-served) has been the discipline used for scheduling cell transmission. Although inexpensive, FIFO is not sensitive to delay requirements of applications. Hence it is impossible for FIFO scheduling discipline to support multiple levels of quality of service as proposed in the ATM standard. On the other hand, *dynamic-priority* scheduling is capable of providing delay-sensitive traffic transmission. However, dynamic-priority schemes are known to be difficult to implement in high-speed networks. At present there is no ATM product that provides dynamic link scheduling. *Static-priority* scheduling can be considered a compromise between FIFO and dynamic-priority scheduling. It is relatively inexpensive in comparison with dynamic-priority scheduling while providing delay-sensitive communication to applications.

One key problem that must be addressed when using static-priority scheduling in ATM networks is *Priority Assignment.* Since the delay bounds for connections' traffic depend on the priorities assigned to the connections, priority assignment should be sensitive to the deadline requirements of the connections in order to maximize the chance that the entire connection set can be admitted, i.e., all the deadlines of connections can always be satisfied.

---

[1]Part of this chapter is reprinted with permission from Proceedings of the *IEEE* INFOCOM'97, Kobe, Japan, 1997, vol. 1, pp. 160-167; Proceedings of the 18th IEEE Real-Time Systems Symposium, San Francisco, CA, Dec. 1997, pp. 264-273; and Proceedings of IEEE International Conference on parallel Processing, Minneapolis, MN, Aug. 1998, pp. 432-440. Copyright 1998 by the *IEEE*. (see Appendix A).

Differently from the previous work [23], where various priority assignment methods were examined and compared for systems with a single server or with traffic regulation mechanisms, we concentrate on ATM switches with static-priority scheduling discipline. In addition, we allow for networks with arbitrary topology and do not assume that traffic regulation mechanisms are in place.

Based on the delay-computation method developed in Chapter IV, we study five priority assignment algorithms. As the base method for comparison, we assign the same priority to all connections, effectively scheduling cells in FIFO manner. The second algorithm assigns the priority in a deadline-monotonic manner [47, 48]. That is, the smaller the relative deadline of a connection, the higher is its priority. The third algorithm, called as *Partition Algorithm*, combines the first two by recursively partitioning the connection set into subsets of connections. Priorities are assigned at subset level, based on deadline information, and the connections in each subset are assigned the same priority. Our performance comparison shows that this partition method outperforms the first two methods in all the situations tested. It is naturally expected that an algorithm that is aware of the topology or the load of the system performs better than the algorithms based only on deadline information. We therefore go on to study a basic priority assignment approach described by Cruz [5] for a limited class of system configurations. We integrate the scheme proposed by Cruz with our partition scheme, in fact proposing an algorithm that is both delay aware and topology aware. Our evaluations show that the integrated approach by a large margin outperforms the remaining four algorithms.

A. Complexity of Priority Assignment Algorithm.

Given static-priority schedulers in the servers, the probability that a set of connections can be established depends on the way the priorities on the servers are assigned to connections. Unfortunately, the following theorem indicates that it is very unlikely that an efficient optimal priority assignment algorithm can be found.

THEOREM 11 *Given a general-topology connection-server graph and a set of connections $\mathcal{M}$, the problem of finding a priority assignment $\pi(i,j)$ of connections to servers so that every connection M_i meets its end-to-end deadline D_i is $\mathcal{NP}$-hard.*

Proof: See the last Section of this Chapter. Q.E.D

Since an efficient optimal assignment algorithm is unlikely to exist, in next section, we compare a number of heuristic algorithms, which assign priorities with consideration for either the deadlines of connections, or the topology of the underlying network, or both.

B. The Algorithms

The trivial approach simply assigns the same priority to all connections on all servers. The scheduling policy on all servers then degenerates to FIFO scheduling. FIFO does not take into account deadline information, and therefore the performance can be expected to be poor. In the following discussion we will be using FIFO as a baseline.

1. Deadline Based Heuristics

We can expect that the performance improves if the priority assignment reflects the message urgency. Intuitively, the smaller the relative deadline of a connection is, the higher its priority should be.

Algorithm *Partition:*

Step 1: *Assign the same priority to all connections. All connections are initially in the same subset.*

Step 2: *Compute the delays for each connection. If all connections pass the deadline test, stop; return the current priority assignment.*

Step 3: *For each subset in which a connection fails to pass the deadline test, perform the following steps:*

- **Step 3.1:** *If the subset consists of a single element, stop; the algorithm was not able to find a feasible priority assignment.*

- **Step 3.2:** *Partition the subset of connections into two subsets and assign connections to the subsets in increasing order of their laxity (defined to be the difference between the deadline and the computed delay).*

Step 4: *For Server j, assign priority p to a connection if the server is on the path of the connection, and the connection is in the pth subset.*

Step 5: *Return to Step 2.*

Fig. 16. Algorithm Partition for assigning priorities to connections.

The *relative deadline monotonic* (RDM) algorithm [47, 48] assigns priorities in this way. For single-server systems and periodic workload, RDM is known to be an optimal static-priority assignment algorithm. Interestingly, our evaluations show that RDM does not perform well when connections traverse multiple servers. In some cases it even underperforms FIFO! This effect is particularly strong when deadline variations are small, and deadlines do not provide a sufficient decision basis for fixed-priority assignments. The two algorithms FIFO and RDM can be combined into a simple scheme that starts with an FIFO assignment and successively modifies priorities of connections to take message urgency into consideration. This leads to Algorithm *Partition*, which is described in Figure 1. This algorithm repeatedly partitions the connection set into an increasing number of subsets in accordance with message deadline laxity. It then assigns the different priorities to the connections in the different subsets. The iteration stops when all connections pass the deadline test, and the whole set of connections is admissible, or when a subset with only one connection needs to be further partitioned because the connection does not meet the deadline. In that case, no more partitions can be done, and the algorithm declares failure. Because the size of the smallest subset of connections is halved at every iteration step, the worst-case cost of the algorithm is the order $O(\lg n)$ in the number of delay computations.

In its basic form, Algorithm *Partition* compares relative deadlines for deciding how to partition the connection set into urgent and non-urgent connections. Connections that traverse a larger number of servers tend to experience more delay, which is not considered when the algorithm simply compares relative deadlines. In the evaluations described below we therefore make the decisions how to partition the connection set by using the *modified relative deadline* D'_i, which is defined as the relative deadline D_i of connection M_i divided by the number of servers traversed by M_i: $D'_i := D_i/S_i$, where S_i is the number of servers on the route of the connection M_i. In this way the length of connections is accounted for when making priority assignments.

2. Topology Based Heuristics

The priority assignment algorithms described above share the following two characteristics: (1) Each connection is assigned the same priority on all the servers on its route, and (2) the priority assignment is independent of the topology of the network or the load of individual servers. Better results should be expected when priorities are allowed to vary between servers, and are assigned with regard to the underlying topology.

A very simple priority assignment that takes into account the network topology was described by Cruz [5] for the case of a ring. Cruz proposed a two-priority scheme, in which connections are assigned a low priority on the first server when they join the network. On all the other servers, connections are assigned a high priority. In other words, cells already in the ring have higher priority than cells that just want to join the ring. Cruz argued that assigning priorities in this way leads to less disturbance of traffic and hence improves delay bounds. We call this method the "Cruz Algorithm".

We now consider an integrated algorithm, which not only uses the location and topology information, but also the timing information (i.e., laxities) to assign priorities. It can be considered as an integration of Cruz's algorithm and Algorithm *Partition*. Figure 2 describes this approach, which we call Algorithm *Integrated*. Except for Step 4 and Step 5, Algorithm *Integrated* is identical to Algorithm *Partition*. After repartitioning the connections in Step 3, but before re-assigning the priorities in Step 6, Algorithm *Integrated* slightly lowers the priorities of connections at their entrance to the network, essentially following the idea of Cruz's algorithm. This is done by assigning a lower priority $(p+1)$ to a connection in the pth subset at Server j if the latter is the server at the entrance of the connection to the network. If this assignment is not successful, Algorithm *Integrated* resorts to the priority assignment

Algorithm *Integrated:*

Step 1: *Assign the same priority to all connections. All connections are initially in the same subset.*

Step 2: *Compute the delays for each connection. If all connections pass the deadline test, stop; return the current priority assignment.*

Step 3: *For each subset in which a connection fails to pass the deadline test, perform the following steps:*

Step 3.1: *If the subset consists of a single element, stop; the algorithm was not able to find a feasible priority assignment.*

Step 3.2: *Partition the subset of connections into two subsets and assign connections to the subsets in increasing order of their laxity (defined to be the difference between the deadline and the computed delay).*

Step 4: *For Server j, increase the priority by one for the connections that are not in the first (highest-priority) subset and join the network at Server j.*

Step 5: *Compute the delays for each connection. If all connections pass the deadline test, stop; return the current priority assignment.*

Step 6: *For Server j, assign priority p to a connection if the server is on the path of the connection, and the connection is in the pth subset.*

Step 7: *Return to Step 2.*

Fig. 17. Algorithm Integrated for assigning priorities to connections.

of Algorithm *Partition* by assigning priority p to every connection in the pth subset.

C. Performance Evaluation

In this subsection, we evaluate the performance of the five priority assignment algorithms discussed in the previous subsection. We first define a performance metric, then describe the system configuration and present the performance results.

1. Performance Metric

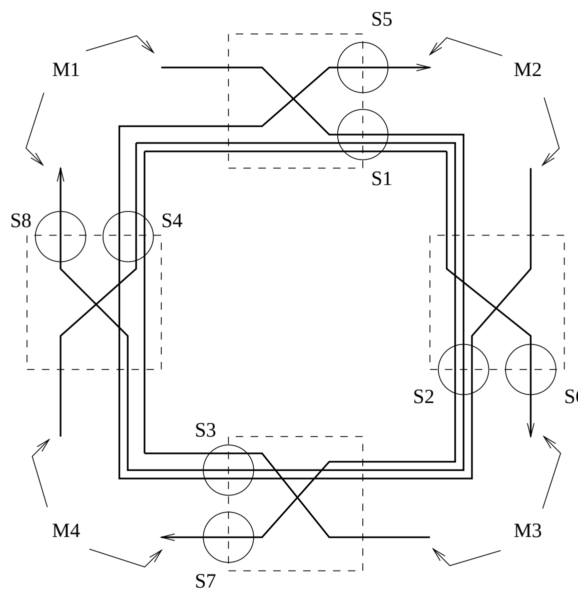
We quantify the performance of an algorithm by measuring the *Admission Probability* ($AP(U)$) for a given link utilization U, that is, the probability that a set of randomly chosen connections can be admitted given that the average utilization of the links in the network is U.

2. Topology and Traffic Load

We consider ATM networks with a specialized ring topology. This topology has been used as an representative benchmark by Cruz, Gallager, and Parekh to study the problem of stability in packet switching networks [4, 5, 27, 49, 50]. Henceforth, we shall refer to this topology as the *Cruz-Gallager-Parekh* (C-G-P) ring.

The architecture of the C-G-P ring is described as follows. The system consists of K 2×2 switches and K connections, $M_1, \ldots, M_i, \ldots, M_K$. Each server has a distinct identity id, where $id = 1, 2, \ldots, 2 * K$ and every connection has an acyclic path that traverses K servers. For connection M_i, the first server, $s(i, 1)$, is Server i and the following servers are

$$s(i,j) = \begin{cases} 1 + (i+j-2) \bmod K & 1 \le j \le K-1 \\ \\ K + i & j = K. \end{cases} \tag{6.1}$$



	$s(i,1)$	$s(i,2)$	$s(i,3)$	$s(i,4)$
M_1	1	2	3	5
M_2	2	3	4	6
M_3	3	4	1	7
M_4	4	1	2	8

Fig. 18. Cruz-Gallager-Parekh ring with 4 switches.

Figure 18 shows an example of a C-G-P ring, which is the connection server graph of the ring with four switches depicted in Figure 2. The source traffic for each connection in the C-G-P ring is constrained by a (β, ρ) traffic bounding function as defined in Equation (4.2).

3. Simulation Methodology

We measured the performance by simulating the behavior of the five algorithms with randomly generated connection sets. For each data point $1,000$ connection sets were randomly generated, and each was was tested for admission. For each connection, β_i and ρ_i were chosen from uniform distributions, and the relative deadlines of connections were chosen from a general exponential distribution. Statistics were collected from the sample set to estimate the admission probability defined earlier. For all measurements, the 99-percentile confidence intervals are below 1% of the admission probability range.

4. Numerical Results and Observations

Figure 19 shows the admission probability results for our example network. The performance figures are corresponding to the case when the average relative deadline is 40, and the standard deviation of deadlines (STD(D)) ranges from 30 to 37. From these figures, we can make the following observations:

1. In general, we found that the admission probability is sensitive to the average link utilization. As the utilization increases, the admission probability decreases. This is expected because a higher network utilization makes it more difficult for the system to admit a set of connections.

2. In all the cases tested, it is seen that Algorithm *Partition* performs far better

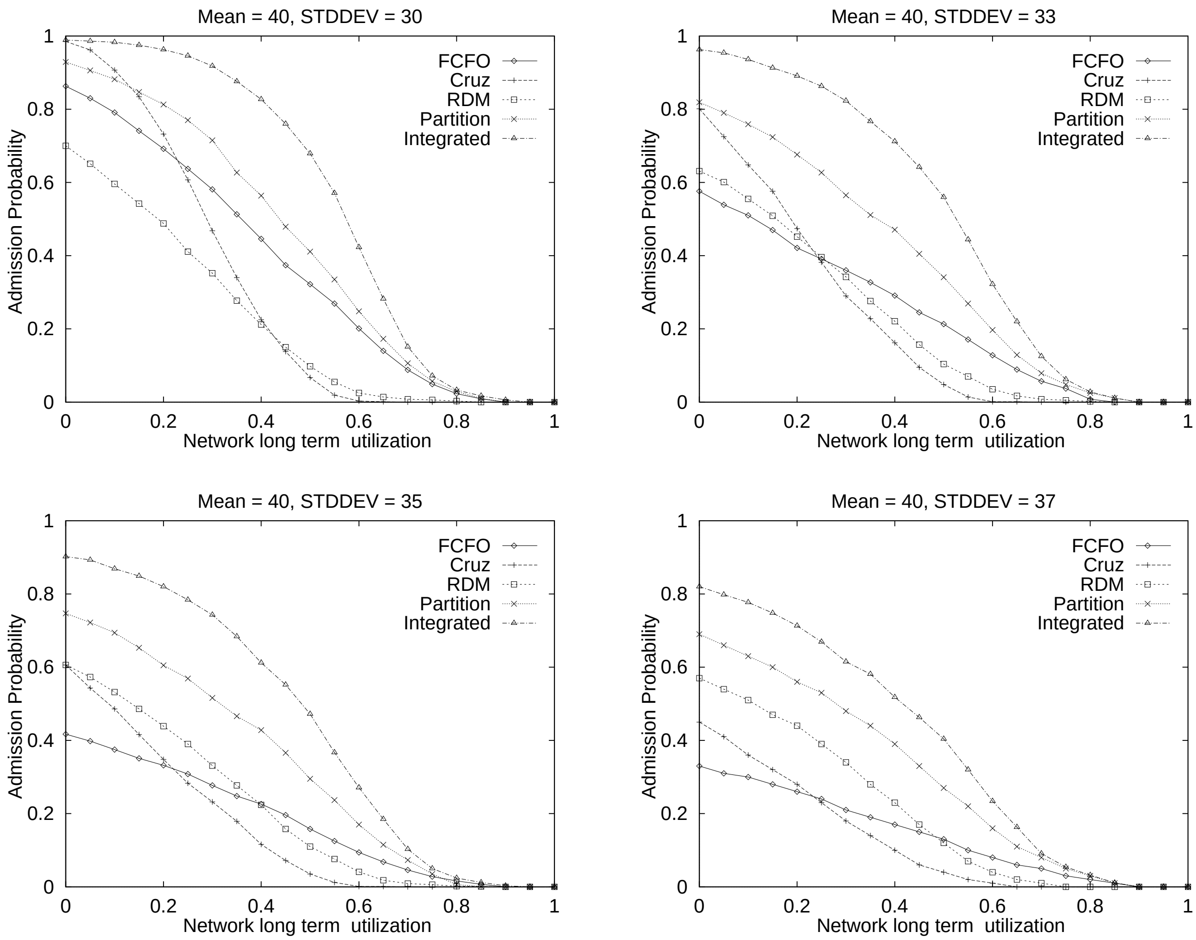


Fig. 19. Admission probability vs. long-term utilization.

than both the FIFO and the RDM methods. This comes from the fact that, whenever a feasible assignment can be found by either FIFO or RDM, Algorithm *Partition* finds it as well. Sometimes, the improvement is significant. For example, when link utilization is 0.4 and STD(D) = 33 (Figure 19), either FIFO or RDM admitted no more than 30% of connection sets while our new algorithm can admit around 50% of connection sets.

3. Furthermore, we observe that the integrated method out-performs all the other

four methods, namely FIFO, RDM, Cruz, and Algorithm *Partition*. For example, when the utilization is 50%, the integrated method admits up to 25% more message sets than the partition method, and up to seven times more than RDM. While the results are encouraging, they are not surprising. Recall from the design of this algorithm that it inherently considers all the possible priority assignments that would be examined by any of the other four methods while maintaining the same order of complexity as the partition method. This demonstrates that by properly integrating the timing and topology information, performance can indeed be improved without introducing much overhead.

D. Final Remarks

We addressed the problem of how to assign priorities to connections in an ATM network with arbitrary topology. In particular, we analyzed five algorithms: FIFO, relative deadline monotonic (RDM), Cruz' Algorithm, Algorithm *Partition*, and Algorithm *Integrated*. The first three have been proposed in previous studies, while the latter two are new. Performance evaluations show that the two new algorithms outperform the other three. Sometimes, the performance difference is significant.

This work can be extended in a number of ways. For example, we are currently studying priority assignment in connection-based *heterogeneous* networks.

E. Proofs

1. Proof of Theorem 11

The proof is by reduction from JOB-SHOP SCHEDULING [51]: Assume that we have a set of processors and a set of jobs, each consisting of a set of nonpreemptive subjobs of identical length δ. Each subjob of the job needs to be executed on a predefined processor. Can we find a schedule that meets an overall deadline D?

We reduce this problem to the following priority-assignment problem in a connection-server graph: Map the processors in the job shop onto servers in the connection-server graph, and the jobs onto connections with traffic bounding functions $(\delta, 0)$. Let δ be the time to – nonpreemptably – transmit a single cell on a server. Find a priority assignment to make each connection meet the overall deadline D. Since the solution of the priority assignment clearly solves the job-shop scheduling problem, it follows that the priority assignment problem for connection-server graphs is $\mathcal{NP}$-hard.

CHAPTER VII

NETWORK STABILITY CRITERIA

In this Chapte[1], we address the issue of stability in communication networks. A network is said to be *stable* if all packets experience bounded delays within the network. Obviously, unbounded packet delays will have a detrimental impact on the performance of any distributed application communicating via the network. Therefore, ensuring stability within the network has been a pivotal issue in the design and management of communication networks. Network stability has been a research problem studied by many researchers. For example, it was established that for the satellite packet switching using the ALOHA protocol if the network throughput is pushed much above 36% then the network can be potentially unstable [52].

We choose ATM networks to address the issue of stability. ATM networks are expected to provide guaranteed quality of services (QoS). With the proliferation of multimedia applications, bounded delay has become an important quality of service requirement. Hence, it is very important to *ensure* stability in ATM networks.

The stability problem for ATM networks has been addressed by several researchers in the context of a specialized ring topology (called Cruz-Gallager-Parekh ring) [4, 5, 27, 53]. This network presents (in some sense) the worst case scenario and becomes a "benchmark" to compare the techniques and results in the study of

[1]Part of this chapter is reprinted with permission from Proceedings of the *IEEE* INFOCOM'97, Kobe, Japan, 1997, vol. 1, pp. 160-167; Proceedings of the 18th IEEE Real-Time Systems Symposium, San Francisco, CA, Dec. 1997, pp. 264-273; and Proceedings of IEEE International Conference on parallel Processing, Minneapolis, MN, Aug. 1998, pp. 432-440. Copyright 1998 by the *IEEE*. (see Appendix A).

stability. We first consider an ATM network in which first come first service (FIFO) is the scheduling policy employed at the multiplexors, (for example the output link schedulers of an ATM switch). We consider the FIFO scheduling policy because it is widely available in practical networks. By applying theorems from fixed point theory and by utilizing the link transmission constraints we found that Cruz-Gallager-Parekh rings with large ring size are stable if the total utilization of the links is less than or equal to 73.2%. Our result also validates Parekh's claim that for Cruz-Gallager-Parekh ring's with ring size less than or equal to four the system is stable when the total utilization of the links is less than 100% [27]. Further, we show that the criteria for stability in a FIFO based ATM network also applies to a network using any work conserving scheduling policy. We also addressed the stability problem for a network with arbitrary topology and multiplexors employing a priority driven scheduling mechanism. The stability of such networks depends on the priority assignment mechanism employed at the servers within the network. We found that one class of priority assignment mechanism, namely a static, fixed, and globally distinct priority assignment, will guarantee the stability as long as the utilization of the individual links is less than 100%.

A. Related Work

Considerable progress has been made recently towards solving the stability problem in ATM networks. ATM networks with arbitrary topology can have cycles in their connection server graphs. The presence of cycles in the connection server graph may lead to feedback dependency loops in the system. Due to the presence of these loops the system may be potentially unstable, i.e., connections in such a system can have unbounded cell delays. There have been several approaches to solve the problem. The first approach considers ATM networks with specialized mechanisms (both hardware and software) so as to prevent cyclic inter-dependencies between connections. Much of the previous studies using this approach have concentrated on designing specialized scheduling policies for ATM switches [54, 55, 27, 22, 53]. In [55, 22, 53], non-working conserving systems were studied where traffic regulation and restoration are used to ensure the network stability. In [27], system stability was ensured by considering a specialized weight assignment for the PGPS scheduling policy in order to restore the connections traffic at the output of a sever.

The other approaches deal with the problem using information on network topology or application semantics. For example, [4, 5, 27] considered ATM networks with the specialized ring topology. In [11], a CAC algorithm for real-time applications was developed. The CAC algorithm addresses the traffic dependence issue and determines if all the deadlines of real-time messages can be met. In this Chapter, we consider general ATM networks and establish stability criteria for the networks with arbitrary topology.

B. FIFO Driven Scheduling

In this section, we establish the stability criteria for an FIFO based ATM network. Due to their implementation efficiency and cost, ATM networks with FIFO servers are widely prevalent in the market. An FIFO server transmits cells on its output link in the order they arrive at its input. Therefore, the worst case delay experienced by any cell at the server is the same for any connection traversing it.

Particularly we study ATM networks with a specialized ring topology. This topology has been used as an representative benchmark by Cruz, Gallager, and Parekh to study the problem of stability in ATM networks [4, 27]. Henceforth, we refer to this topology as the *Cruz-Gallager-Parekh* (C-G-P) ring.

The architecture of the C-G-P ring is described as follows. The system consists of K 2×2 switches and K connections, $M_1, M_2, \ldots, M_i, \ldots, M_K$. Each server has a distinct identity id, where $id = 1, 2, \ldots, 2 * K$ and every connection has an acyclic path which traverses K servers. For connection M_i, $s(i,1) = i$ and

$$s(i,j) = \begin{cases} 1 + (i + j - 2) \bmod K & 1 \leq j \leq K - 1 \\ \\ K + i & j = K. \end{cases} \tag{7.1}$$

Figure 2 shows an example of a C-G-P ring with 4 switches.

The source traffic in the C-G-P ring is constrained by a piecewise linear function. For connection M_i in the C-G-P ring, the traffic at the source is given by

$$F_{i,s(i,1)}(I) = \min(I, \beta_i + \rho * I), \tag{7.2}$$

where β_i and ρ are positive real-numbers. Note that the value of ρ is the same for all the connections.

The following theorem gives the criteria for stability in an FIFO based ATM network with a C-G-P ring topology.

THEOREM 12 *A C-G-P ring with FIFO servers is stable if*

$$\mu < \begin{cases} 1, & \textit{if } 2 \leq K \leq 4 \\ \\ \sqrt{1 + \frac{2*(K-1)}{K-2}} - 1, & \textit{if } K \geq 5 \end{cases} \tag{7.3}$$

Proof: See the last Section of this Chapter. Q.E.D

Figure 20 shows the plot of the upper bound of μ given by (7.3). We observe

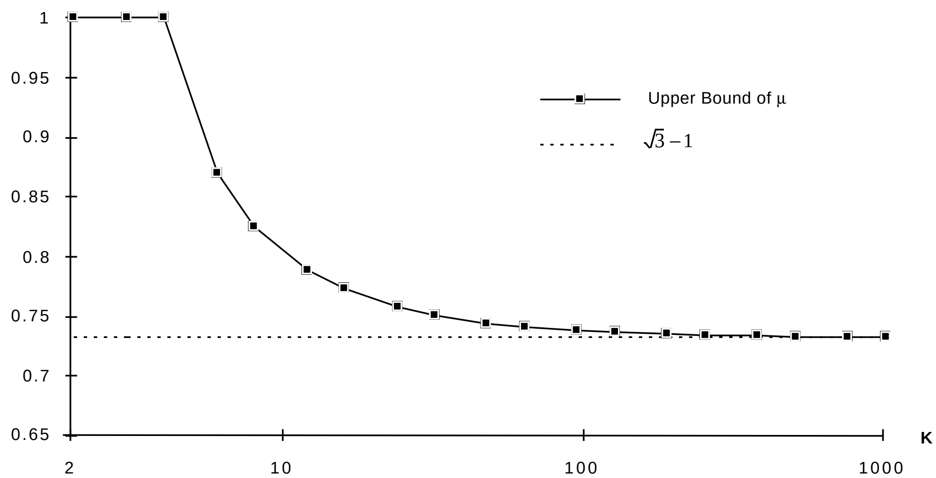


Fig. 20. Plot of the upper bound of μ given by (4.5).

that as K is increased the upper bound of μ decreases. Further, for large values of K, the upper bound of μ converges to $\sqrt{3} - 1$. This observation is formalized in the following corollary.

COROLLARY 2 *An ATM network with the C-G-P topology and FIFO servers is stable if*

$$\mu < \sqrt{3} - 1 \approx 0.732 \tag{7.4}$$

Proof: See the last Section of this Chapter. Q.E.D

The results of Corollary 2 means that any ATM network with the C-G-P ring topology is stable if the maximum link utilization in the network is less than 73.2%. This is an efficient criteria to determine stability.

C. Priority Driven Scheduling

In this section, we study the stability problem in ATM networks with priority driven scheduling and arbitrary topology. In some ATM networks servers with priority driven scheduling policies are used to provide different levels of services. In priority driven scheduling, every connection traversing a server is assigned a priority. The server transmits the cells waiting in its queue in an order given by the priorities of the connections associated with the cells. For example, if connection M_1 has a higher priority than connection M_2, then M_1's cell will always be transmitted before connection M_2's cell. Thus, the worst case cell delay of a connection at the server depends only on the traffic of the connections with a higher priority.

Let $\pi(i, j)$ be the priority assigned to connection M_i at server j. The following theorem provides a criteria of stability for a system with arbitrary topology.

THEOREM 13 *For a given static priority assignment, if*

$$\nu = \max_{p,j}\left(\sum_{s=1}^{K}\sum_{q=1}^{P}\tilde{C}_{q,s,p,j}\right) < 1 \tag{7.5}$$

then the system is stable, where $\tilde{C}_{q,s,p,j}$ *is defined in (4.17).*

Proof: See the last Section of this Chapter. Q.E.D

DEFINITION 6 *A static priority scheduling is said to be fixed if for* $j \neq j'$,

$$\pi_(i, j) = \pi_(i, j'). \tag{7.6}$$

If the static priority scheduling is fixed, let P_i be the priority for M_i at all the servers. $\vec{P}$ is the priority assignment vector for the set of N connections. $\vec{P}$ is given by

$$\vec{P} = (P_1,\ P_2,\ \ldots,\ P_i,\ \ldots,\ P_N)^{\top}_{N\times 1}. \tag{7.7}$$

DEFINITION 7 *A fixed static priority scheduling is said to be globally distinct if for* $i \neq i'$

$$P_i \neq P_{i'}. \tag{7.8}$$

It must be noted that if the connection priorities in a static fixed system are not distinct then in the worst case the performance of the static fixed priority scheduling based system can reduce to that of a system using FIFO servers.

Without loss of generality, we assume that in the system of N connections, $M_1, M_2, \ldots, M_N$, we have

$$P_1 > P_2 > \ldots > P_i > \ldots > P_N. \tag{7.9}$$

The following theorem, gives the criteria for stability of the ATM network with static, fixed, and globally distinct priority assignment (SFGDP).

THEOREM 14 *For an ATM network with arbitrary topology and SFGDP based servers, if* $\mu < 1$*, then the system is stable.*

Proof: See the last Section of this Chapter. Q.E.D

Theorem 14 establishes the criteria for stability of an ATM network with SFGDP servers. It is worth to point out that because of our assumption (7.9), the delay of connection M_i at its j^{th} server is dependent only on the delays experienced by connections $M_1, \ldots, M_{i-1}$ and the delays experienced at upstream servers. Therefore, delays can be solved in a sequential order for connection 1 then connection 2 and so on. For a given connection M_i, the delays at the servers are also computed in a sequential order, i.e., $s(i,1)$ then $s(i,2)$ and so on.

D. Extensions

Recall that the stability criteria established in the previous sections were based on the following two assumptions.

1. The source traffic description function of all the connections in the system is piecewise linear.

2. The scheduling policy used in the servers within the network is either FIFO or static priority scheduling.

In this section, we relax the above two assumptions and establish the stability criteria for a general ATM network. Specifically we extend our results to encompass systems in which the source traffic description function is not piecewise linear. We also extend our results to ATM networks with servers employing work conserving scheduling policies other than FIFO and static priority.

1. General Source Traffic

Recall that in Theorems 12 and 14 we assumed that the source traffic of the connections were constrained by the piecewise linear traffic description function. Although the source traffic of many connections can be characterized by such piecewise linear functions, it is useful to establish the criteria for stability in a system without this constrain. The following theorem establishes the stability criteria for a general source traffic description function.

THEOREM 15 *The criteria for stability given in Theorems 12, 13 and 14 hold if the source traffic satisfies the following condition:* $\forall i$, $M_i \in \mathcal{M}$, *there are positive real numbers* ρ_i, *and* T *such that for* $I > T$,

$$F'_{i,s(i,1)}(I) \leq \rho_i, \tag{7.10}$$

where $F'_{i,s(i,1)}(I)$ *is the derivative of* $F_{i,s(i,1)}(I)$ *about the variable* I.

Proof: See the last Section of this Chapter. Q.E.D

This theorem says that as long as the long term average rate of the source traffic is bounded, all the results for stability established using the piecewise linear function are also applicable for the general source traffic function.

2. Work Conserving Scheduling

Here, we consider an ATM network which consists of servers using any work conserving scheduling policy. A server employing a work conserving scheduling policy always transmits a cell if its buffer is not empty. The FIFO and priority driven scheduling policies are examples of work conserving scheduling policy. In the following theorem we establish the stability criteria for an ATM network with servers employing a work conserving scheduling policy.

THEOREM 16 *An ATM network with servers employing a work conserving scheduling policy is stable if it is stable with all servers employing FIFO scheduling policy.*

This theorem can be easily proved by observing that for the given source traffic the length of the maximum busy interval for any work conserving server is the same. Now if the system using FIFO servers is stable then the length of the maximum busy interval at the FIFO servers is bounded. Therefore, the length of the maximum busy interval at the servers is also bounded if the system uses some other work conserving scheduling policy at its servers. Further, since for any work conserving server the maximum queue length at the server is no more than the length of its maximum busy interval, the queue length at the work conserving servers is bounded when the length of its maximum busy interval is bounded. Therefore, when the system with

FIFO servers is stable then the system with any other work conserving servers is also stable. Hence the criteria for stability given in Theorems 12, 13, and 14 hold if the scheduling policy at the servers are work conserving.

Theorems 12 and 16 together generalize the claim made by Gallager and Parekh [27] that that any C-G-P ring of 4 switches with work conserving servers is stable when $\mu < 1$. Our result indicates that when the ring size is large (≥ 5), the network with any work conserving scheduling policy is stable if $\mu < \sqrt{3} - 1$.

E. Final Remarks

In this Chapter we addressed the stability problem in ATM networks. We have focused on the development of criteria for testing the stability of an ATM network. The problem of stability in ATM networks was studied by many researchers [4, 27, 11]. However, our work differs from the previous work by making the following contributions:

For the static priority scheduling based networks with arbitrary topology, we develop the criteria for network stability. We also generalize the result of stability in an FIFO based ATM network to the one using any work conserving scheduling policy.

In previous work, the Cruz-Gallager-Parekh ring has been a "benchmark" architecture to study the stability problem. For example, Gallager and Parekh claimed a C-G-P ring is stable if the total number of switches is no more than 4 [27]. We validated this result. Furthermore, we found that a large size ring is stable if the total utilization of the links is less than or equal to 73.2%. For ATM networks with priority driven scheduling policies and arbitrary topology, we found that one class of priority assignment mechanism, namely a static fixed globally distinct priority assignment, will guarantee the stability as long as the utilization of the individual links is less than 100%.

We also showed that the main results on stability holds not only for piecewise linear source traffic model (as assumed in most previous work) but also for general source traffic as long its long term average rate exists.

This work can be extended in several ways. It would be interesting to consider the stability problem in connection based heterogeneous networks. To establish the criteria of stability in such networks it will be necessary to investigate characterizations of the traffic within the network. Utilizing a consistent traffic characterization function over a series of network segments is a key step in this process.

F. Proofs

1. Proof of Theorem 12

When $K = 2$, this is a trivial case. So we only consider cases of $K \geq 3$. According to Theorem 5, we only need to verify if $\nu < 1$.

Without loss of generality, we assume the source traffic of connection M_j is given as

$$\begin{aligned} F_{j,s(j,1)}(I) &= \min_I(I, \beta_j + \rho * I) \\ &= \begin{cases} I, & I \leq \frac{\beta_j}{1-\rho}, \\ \\ \beta_j + \rho * I, & \frac{\beta_j}{1-\rho} \leq I. \end{cases} \end{aligned} \tag{7.11}$$

and the direction of connection is counterclockwise. Hence, by the Theorem 1, we have for $h = 1, \cdots, K-1$

$$F_{i,s(i,h)}(I) = \begin{cases} I, & I \leq \frac{\beta_i + \rho * \sum_{g=1}^{(h-1)} d_{1,s(i,g)}}{1-\rho}, \\ \\ \beta_i + \rho * \sum_{g=1}^{(h-1)} d_{1,s(i,g)} + \rho * I, & \frac{\beta_i + \sum_{g=1}^{(h-1)} \rho * d_{1,s(i,g)}}{1-\rho} \leq I. \end{cases} \tag{7.12}$$

For the delays of all the servers, we only need to consider servers $1, 2, \cdots, K$. These servers form a loop. If the delays at them are bounded, then the delays at other servers are also bounded. For any server $j, j = 1, \cdots, K$, there are two input links, i.e. $L_j = 2$. One of them is used by connection M_j. Thus, the traffic on this link is given by

$$\begin{aligned} \mathcal{F}_{1,j}(I) &= \min_I(I, \beta_j + \rho * I) \\ &= \begin{cases} I, & I \leq \eta_{j,1}, \\ \\ \beta_j + \rho * I, & \eta_{j,1} \leq I. \end{cases} \end{aligned} \tag{7.13}$$

where

$$\eta_{j,1} = \frac{\beta_j}{1-\rho}. \tag{7.14}$$

The other link is used by connections $M_{1+((j+h-1) \bmod (K))}$ for $h = 2, 3, \cdots, K-1$. Thus, the traffic on that link is given by

$$\begin{aligned} \mathcal{F}_{2,j}(I) &= \min(I, \sum_{h=2}^{K-1} F_{1+((j+h-1) \bmod (K)),j}(I)) \quad (7.15) \\ &= \begin{cases} I, & I \leq \eta_{j,2}, \\ \\ \sum_{h=2}^{K-1} [\beta_{1+((j+h-1) \bmod (K))} \\ +\rho * (h-1) * d_{1,1+((j+h-1) \bmod (K))}] + (K-2) * \rho * I, & \eta_{j,2} \leq I. \end{cases} \end{aligned}$$

where

$$\eta_{j,2} = \frac{\sum_{h=2}^{K-1}[\beta_{1+((j+h-1) \bmod (K))} + \rho * (h-1) * d_{1,1+((j+h-1) \bmod (K))}]}{1-(K-2)*\rho}. \quad (7.16)$$

It is easy to verify

$$\eta_{j,2} \quad = \quad \max\{\eta_{j,1}, \eta_{j,2}\}. \quad (7.17)$$

Therefore $d_{1,j}$ is given by

$$\begin{aligned} d_{1,j} &= \beta_j + \rho * \eta_{j,2} \\ &= \beta_j + \rho * \frac{\sum_{h=2}^{K-1} \beta_{1+((j+h-1) \bmod (K))}}{1-(K-2)*\rho} \\ &\quad + \rho * \frac{\sum_{h=2}^{K-1} \rho * (h-1) * d_{1,1+((j+h-1) \bmod (K))}}{1-(K-2)*\rho}. \end{aligned} \quad (7.18)$$

Thus,

$$\begin{aligned} \nu &= \max_j\{\sum_{h=1,h\neq j}^{K} C_{1,h,1,j}\} \\ &= \rho * \frac{\sum_{h=2}^{K-1} \rho * (h-1)}{1-(K-2)*\rho} \\ &= \rho * \rho * \frac{(K-1)*(K-2)}{2*(1-(K-2)*\rho)}. \end{aligned} \quad (7.19)$$

On the other hand,

$$\begin{aligned} \mu &= \max_{i=1,\cdots,K} \sum_{j\in C(i)} \rho \\ &= (K-1)*\rho \end{aligned} \quad (7.20)$$

Hence $\rho = \frac{\mu}{K-1}$.

Substituting $K = 3$ and $\rho = \frac{\mu}{K-1} < \frac{1}{2}$ in to (7.19), we have

$$\nu < \frac{1}{2} * \frac{1}{2} * \frac{(3-1)*(3-2)}{2*(1-(3-2)*\frac{1}{2})} = 1. \quad (7.21)$$

Similarly for the case of $K = 4$, it is obvious that $\nu < 1$.

Now considering the case of $K \geq 5$.

$$\rho = \frac{\mu}{K-1} < \frac{\sqrt{1 + 2 * \frac{(K-1)}{(K-2)}} - 1}{K-1}. \tag{7.22}$$

Substituting (7.22) into (7.19), we have

$$\begin{aligned} \nu &< \frac{\sqrt{1 + 2 * \frac{(K-1)}{(K-2)}} - 1}{K-1} * \frac{\sqrt{1 + 2 * \frac{(K-1)}{(K-2)}} - 1}{K-1} * \frac{(K-1) * (K-2)}{2 * (1 - (K-2) * \frac{\sqrt{1+2*\frac{(K-1)}{(K-2)}}-1}{K-1})} \\ &= \frac{(K-2) * (1 + \frac{K-1}{K-2} - \sqrt{1 + 2 * \frac{K-1}{K-2}})}{(K-1) * (1 - \frac{K-2}{K-1} * (\sqrt{1 + 2 * \frac{(K-1)}{(K-2)}} - 1))} \\ &= \frac{(K-2) + (K-1) - (K-2) * \sqrt{1 + 2 * \frac{(K-1)}{(K-2)}}}{(K-2) + (K-1) - (K-2) * \sqrt{1 + 2 * \frac{(K-1)}{(K-2)}}} \\ &= 1. \end{aligned} \tag{7.23}$$

Q.E.D

2. Proof of Theorem 13

Let

$$A = \max_{p,j}\{\tilde{\Pi}_{p,j}\}, \tag{7.24}$$

and

$$B = \max_{p,j}\{\sum_{s=1}^{K}\sum_{q=1}^{P} \tilde{C}_{q,s,p,j}\}, \tag{7.25}$$

By (4.17), we have $0 \leq C_{q,s,p,j} \leq \tilde{C}_{q,s,p,j}$. Then, according to condition (7.5), $B < 1$. Let

$$O_{max} = \frac{A}{1-B}. \tag{7.26}$$

and

$$\Omega = \{(x_{1,1}, x_{2,1}, \cdots, x_{P,1}, \cdots, x_{P,K})^\top \mid 0 \le x_{q,s} \le O_{max}, q = 1, \cdots, P, s = 1, \cdots, K\}. \tag{7.27}$$

Then, Ω is a nonempty compact set. For any (p, j) and $\vec{x} \in \Omega$, we have

$$\begin{aligned} \Pi_{p,j} + \sum_{s=1}^{K}\sum_{q=1}^{P} C_{q,s,p,j} * x_{q,s} &\le A + \sum_{s=1}^{K}\sum_{q=1}^{P} \tilde{C}_{q,s,p,j} * O_{max} \\ &\le A + B * O_{max} \\ &\le A + B * \frac{A}{1-B} \\ &\le \frac{A}{1-B} \\ &= O_{max}. \end{aligned} \tag{7.28}$$

That is,

$$\vec{Z} : \Omega \to \Omega. \tag{7.29}$$

According to Schauder Theorem [34], there exists $\vec{x} \in \Omega$ such that

$$\vec{x} = \vec{Z}(\vec{x}). \tag{7.30}$$

This means that there always exists a solution for (4.14).

On the other hand, for any

$\vec{x} = (x_{1,1}, x_{2,1}, \cdots, x_{P,K})^\top \ge \vec{0}$, if there is $x_{p*,j*} > O_{max}$, such that

$$x_{p*,j*} = \max\{x_{1,1}, x_{2,1}, \cdots, x_{P,K}\} > O_{max}. \tag{7.31}$$

If $x_{p*,j*} = \Pi_{p*,j*} + \sum_{s=1}^{K}\sum_{q=1}^{P} C_{q,s,p*,j*} * x_{q,s}$, we have

$$\begin{aligned} x_{p*,j*} &= \Pi_{p*,j*} + \sum_{s=1}^{K}\sum_{q=1}^{P} C_{q,s,p*,j*} * x_{q,s} \\ &\le \Pi_{p*,j*} + \sum_{s=1}^{K}\sum_{q=1}^{P} \tilde{C}_{q,s,p*,j*} * x_{p*,j*} \end{aligned}$$

$$
\begin{aligned}
&\leq A + B * x_{p*,j*} \\
&\leq \frac{A}{1-B} \\
&= O_{max} \\
&< x_{p*,j*}. \qquad (7.32)
\end{aligned}
$$

This is a contradiction. Hence, if $\vec{x} \notin \Omega$

$$\vec{x} \neq \vec{Z}(\vec{x}). \qquad (7.33)$$

This means that any positive solution of (7.30) must be within Ω. That is, $\vec{d}$ defined by (4.14) is bounded. Specifically, $||\vec{d}||$ is bounded by O_{max}.

Q.E.D

3. Proof of Corollary 2

We have

$$\sqrt{1 + 2 * \frac{(K-1)}{(K-2)}} - 1 > \sqrt{3} - 1 \approx 0.732. \qquad (7.34)$$

By Theorem 5, the corollary follows.

Q.E.D

4. Proof of Theorem 14

Due to the property of SFGDP scheduling discipline, it is easy to know that

$$d_{1,s(1,j)} = 1, \quad j = 1, \cdots, S_1. \qquad (7.35)$$

Since connection M_i is assigned lower priority than connections $M_1, \cdots, M_{i-1}$, at the j^{th} server in the route of connection M_i, we have

$$d_{i,s(i,j)} = \max_{0 \leq I \leq T_{i,j}} \sum_{k \in P(s(i,j))} \mathcal{J}_{i-1,k)}(I + d_{i,s(i,j)}) + F_{i,s(i,j)}(I) - I). \tag{7.36}$$

where $T_{i,j}$ is the maximum busy interval of server $s(i,j)$ for connection $M_1, M_2, \cdots, M_i$ and is given in Theorem 3 in Chapter IV.

Let

$$\zeta_{i,j} = \frac{\beta_i + \rho_i * \sum_{g=1}^{j-1} d_{i,s(i,g)}}{1 - \rho_i} \tag{7.37}$$

and for $k \in P(s(i,j))$

$$\eta_{k,s(i,j)} = \frac{\sum_{h \in C(k),\ h<i} [\beta_h + \rho_h * \sum_{g=1}^{j_h - 1} d_{h,s(h,g)}]}{1 - \sum_{h \in C(k),\ h<i} \rho_h}. \tag{7.38}$$

and

$$\tau_{i,s(i,j)} = \max_{k \in P(s(i,j))} \{\eta_{k,s(i,j)}\}. \tag{7.39}$$

and

$$I^{max}_{i,s(i,j)} = \max\{\tau_{i,s(i,j)} - d_{i,s(i,j)}, \zeta_{i,j}\}. \tag{7.40}$$

According to Theorem 4 in Chapter IV, we have

$$I^{max}_{i,s(i,j)} = \zeta_{i,j}, \tag{7.41}$$

and

$$d_{i,s(i,j)} = \frac{\sum_{k \in C(s(i,j)),\ k \leq i} [\beta_k + \rho_k \sum_{g \in G_{k,j},\ g \neq j} d_{k,\ g}] + 1}{1 - \sum_{k \in C(s(i,j)),\ k<i} \rho_k} \frac{\sum_{k \in C(s(i,j)),\ k \leq i} \rho_k - 1}{1 - \sum_{k \in C(s(i,j)),\ k<i} \rho_k} * \frac{\beta_i + \rho_i \sum_{g=1}^{j-1} d_{i,s(i,g)}}{1 - \rho_i}. \tag{7.42}$$

Since $\nu < 1$ and $\sum_{k \in C(s(i,j)),\ k<i} \rho_k \leq \nu$ and $\rho_i \leq \nu$, we have

$$\frac{1}{1 - \sum_{k \in C(s(i,j)),\ k<i} \rho_k} \leq \frac{1}{1-\nu}, \tag{7.43}$$

and

$$\frac{1}{1-\rho_i} \leq \frac{1}{1-\nu}. \tag{7.44}$$

Furthermore, due to the property of SFGDP scheduling discipline, before considering $d_{i,s(i,j)}$, we can obtain $d_{k,s(k,g)}$, $k = 1, \cdots, i-1$, $g = 1, \cdots, S_k$ and $d_{i,s(i,g)}$, $g = 1, \cdots, j-1$. Hence, all local delay suffered by the network traffic can be obtained and are bounded, provided $\nu < 1$. So system is stable. Q.E.D

5. Proof of Theorem 15

Let

$$F^{new}_{i,s(i,1)}(I) = \begin{cases} I, & I \leq \frac{F_{i,s(i,1)}(T)}{1-\rho_i}, \\ F_{i,s(i,1)}(T) + \rho_i * I, & \frac{F_{i,s(i,1)}(T)}{1-\rho_i} \leq I. \end{cases} \tag{7.45}$$

It is easy to know that

$$F^{new}_{i,s(i,1)}(I) \geq F_{i,s(i,1)}(I), \forall I \geq 0. \tag{7.46}$$

Hence the ATM network is stable if for the new traffic functions the ATM network is stable. Q.E.D

6. Proof of Theorem 16

The worse case delay experienced by a cell from any connection at the server j using any work conserving scheduling is less or equal to the maximum busy interval of the server. We only need to prove that for any server j the maximum busy interval T_j^{max}

is bounded if the system using FIFO scheduling policy at its servers is stable.

First, we have

$$T_j^{max} = \min_{I \geq 0}\{I; \sum_{i \in C(j)} F_{i,s(i,1)}(I + \sum_{g \in G_{i,j}} d_{1,g})) - I < 0\}, \tag{7.47}$$

Since $\mu < 1$, we have

$$\sum_{i \in C(j)} [\beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{1,g} + \rho_i * T_j^{max}] - T_j^{max} = 0. \tag{7.48}$$

and

$$T_j^{max} = \frac{\sum_{i \in C(j)}[\beta_i + \rho_i * \sum_{g \in G_{i,j}} d_{1,g}]}{1 - \sum_{i \in C(j)} * \rho_i}. \tag{7.49}$$

where $d_{1,j}$ is the worse case delay at server j with FIFO scheduling policy. If the ATM network using FIFO scheduling policy at its servers is stable, $d_{1,1}, \cdots, d_{1,K}$ are bounded. So $T_1^{max}, \cdots, T_K^{max}$ are bounded. Hence, an ATM network with servers employing a work conserving scheduling policy is stable if it is stable with servers employing FIFO scheduling policy.

Q.E.D

CHAPTER VIII

CONCLUSION

A. Summary of Results

The fundamental theme of this dissertation is the analysis of network traffic for supporting real-time communication in high speed packet switching network. Surrounding this theme, we investigated the internal network traffic, scheduling discipline, worst case delay suffered by the network traffic, and the network stability.

In Chapter I, we introduced and motivated the problem. In Chapter II, we overviewed the network models and network traffic models. In Chapter III, we presented surveys for the existing related work in following areas: scheduling disciplines, methodology for estimation of worse case delays, and the network stability.

In Chapter IV and Chapter V, we presented a through and comprehensive analysis for the worst case queuing delay suffered by the network traffic. In Chapter IV, after introducing several useful Theorems to accurately describe the internal network traffic, we study delay computation problem for an ATM networks with static priority scheduling. Given an ATM network with arbitrary topology, it is possible that the local delays suffered by the network traffic at different servers depend on each other due to the potential cyclic dependency of the traffic. Therefore we developed a numerical method to compute these local delays. Convergence of the method is formally proved and a closed form for the approximation error is obtained. In Chapter V, after studying the relationship between input traffic and output traffic on the individual connection base, we proposed a new method for deriving end-to-end delay bounds for connections in tandem network, which uses a FIFO scheduling discipline. Our new method takes into account delay dependencies in successive servers along the path of a connection, which is in general very difficult for delay analysis, and achieves better performance than the method provided in [4, 5].

In Chapter VI, we addressed the problem of how to assign priorities to connections in an ATM network. We proved that to find an optimal priority assignment algorithm is an NP hard problem. Furthermore, we developed two heuristic algorithms for priority assignment: Algorithm *Partition* and Algorithm *Integrated.* We campared these two new algorithms with other three well known algorithms: FIFO, relative deadline monotonic (RDM), and Cruz's Algorithm. Simulations show that these two new algorithms outperform the other three. Sometimes, the performance difference is significant.

In Chapter VII, we addressed the stability problem in ATM networks. We focused on the development of criteria for testing the stability of an ATM network. For the static priority scheduling based networks with arbitrary topology, we developed the criteria for network stability. We also generalized the result of stability in an FIFO based ATM network to the one using any work conserving scheduling policy and found that a large size ring is stable if the total utilization of the links is less than or equal to 73.2%. Furthermore, we found that one class of priority assignment mechanism, namely a static fixed globally distinct priority assignment, will guarantee the stability as long as the utilization of the individual links is less than 100%. We also showed that the results on stability hold not only for piecewise linear source traffic model (as assumed in most previous work) but also for general source traffic as long as its long term average rate exists.

B. Directions for Future Research

1. Real Time Communication in Wireless Networks

It is no doubt that wireless communications will play an important role in future telecommunication. Similar to the wired network, the wireless network is expected to provide a diverse set of services to transport voice, data, image, video, and other media. To enable these services, it is essential that wireless network be able to support multiple classes of traffic with distinct QoS requirements and traffic characteristics. The QoS is defined in terms of data rate, bit error rate, delay, throughput, etc.. For example, the voice traffic generally requires a data rate of 8 kbps, bit error rate less than 10^{-2}, and the end-to-end delay less than 100 millisecond, while the video traffic requires data rate of 64 kbps, bit erro rate less than 10^{-6}, and the end-to-end delay less than 200 millisecond. However, wireless communications face more obstacles than wired communications as described in the following.

- First, the radio spectrum, and therefore the capacity available for wireless access service, is generally limited by regulation. Thus, unlike wireline communications wherein an increasing user population can easily be served by deploying additional wire (or fiber) facilities to conneeect those users to the network (thereby increasing the total capacity available to serve that increasing population), the available radio spectrum can not arbitrarily be expanded.

- Second, unlike the wireline network, wherein the physical link between a remote user terminal and the end-office switch or remote concentrator are of time invariant high quality, the radio link is subject to several time varying impairments arising from inherent user mobility and unavoidable changes caused by motion of the surrounding enviroment (closing doors and passing trucks are two examples). These impairments are manifested in a time varying bit error rate performance of radio link, with the bit error rate often too high to meet the needs of the application.

- Third, unlike a wireline connection, wherein users attach via fixed network ports, wireless access implies that the user's point of attachment is unknown at the time of connection establishment and may change throughout the duration of the connection.

As a result, wireless communication is characterized by lower bandwidths, higher bit error rate, and more frequent spurious disconnecting.

Even though there are considerable research efforts for the real-time communication in wireless networks [57, 58, 59, 60], the solid and systematic framework for QoS provisioning by wireless networks still has not been formed. One of the challenges for the wireless network is how to generalize the theory of the network traffic that has worked very well for the wire network and has deeply been studied in this dissertation to wireless communication. Generally speaking, the wireless network traffic depends on two factors, the telecommunication traffic and the mobile host traffic. In order to precisely characterize the wireless network traffic, we need to introduce two coupling equations, one for the telecommunication traffic and the other for mobile host traffic, to capture the time and space dynamics of the traffic. Furthermore, the equation for the telecommunication traffic needs to take into account the retransmission of packets due to bit error. It is obvious that these two equations are more complicated than the traffic function used in the wire network.

2. Application in Network Security

The main theme of the analysis of network traffic in this work is to provide performance guarantee. There is another feature of QoS provisioning by the high speed network, i.e., security guarantee.

Generally speaking, there are two kinds of attacks to a network. One is the passive attack in which an intruder gains the sensitive information by traffic analysis even though the message has well been encrypted. It is well known that the traffic pattern within the communication network exhibits different features under different circumstances. For example, the pattern of the traffic comes in and goes out of a military headquarters depends on the activity of military force. Although the encryption can provide the privacy of the message, the pattern of the traffic will still give some sensitive information to a stand by intruder. Therefore, how to hide the traffic patterns without violating the delay guarantee of real time communication is a challenge and interesting problem. The theory of network traffic analysis provided in this work can be used to aid to decide when and where to reroute packets or pad messages with dummy packets so that the pattern of network traffic is independent of communication activities and the deadline requirements of these real-time communications are still met.

The other is the active attack in which an intruder enters network to delete, modify, or insert messages. The notable feature of such attack is that the pattern of normal network traffic has been changed. For example, when an intruder deletes or modifies packets, the network traffic flow will slow down; when an intruder inserts packets, the network traffic flow will speed up. Therefore, how to quickly detect the active intrusion is a challenge and interesting problem. The theory of network traffic analysis provided in this work can be used to accurately describe the traffic profile so that any irregular fluctuation of network traffic will quickly be detected.

REFERENCES

[1] C. Li, R. Bettati, and W. Zhao, "Static priority scheduling for ATM networks," in *Proceedings of the 18th IEEE Real-Time Systems Symposium*, San Francisco, CA, 1997, pp. 264-273.

[2] C. Li, R. Bettati and W. Zhao, "Response time analysis for distributed real-time systems with bursty job arrivals," in *Proceedings of IEEE International Conference on parallel Processing*, Minneapolis, MN, 1998, pp. 432-440.

[3] C. Li, A. Raha, and W. Zhao, "Stability in ATM networks," in *Proceedings of the IEEE INFOCOM'97*, Kobe, Japan, 1997, vol. 1, pp. 160-167.

[4] R. L. Cruz, "A calculus for network delay," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114-131, 1991.

[5] R. L. Cruz, "A calculus for network delay, part II: Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132-141, 1991.

[6] ATM Forum, *ATM User-Network Interface Specification Version 3.1*. Mountain View, CA, 1994; available http://www.atmforum.com/atmforum/specs/approved.html.

[7] ATM Forum, *Traffic management specification Version 4.0*. Mountain View, CA, 1995; available http://www.atmforum.com/atmforum/specs/approved.html.

[8] ATM Forum, *Private Network-Network Interface specification Version 1.0*. Mountain View, CA, 1996; available http://www.atmforum.com/atmforum/specs/approved.html.

[9] J.-Y. Le Boudec, "The asynchronous transfer mode: A tutorial," *Computer Networks and ISDN Systems*, vol. 24, pp. 279-309, 1992.

[10] A. Raha, S. Kamat, and W. Zhao, "Guaranteeing end-to-end deadlines in ATM networks," in *Proceedings of the 15th IEEE International Conference on Distributed Computing Systems*, Vancouver, Canada, 1995, pp. 60-68.

[11] A. Raha, S. Kamat, and W. Zhao, "Admission control for hard real-time connections in ATM LAN's," in *Proceedings of the IEEE INFOCOM'96*, San Francisco, CA, 1996, pp. 111-120.

[12] A. Adas, "Traffic models in broadband networks," *IEEE Communications*, vol. 35, no. 7, pp. 82-89, 1997.

[13] J. Filipiak, *Modelling and control of dynamic flows in communication networks.* New York: Springer-Verlag, 1988.

[14] V. Frost and B. Melamed, "Traffic modeling for telecommunication networks," *IEEE Communications*, vol. 32, no. 3, pp. 70-81, 1994.

[15] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing synchronous message deadlines in high speed token ring networks with timed token protocol," in *Proceedings of the 12th IEEE International Conference on Distributed Computing Systems*, Yokohama, Japan, 1992, pp. 468-475.

[16] H. Zhang and S. Keshav, "Comparison of rate based service disciplines," in *Proceedings of ACM SIGCOMM'91*, Zurich, Switzerland, 1991, pp. 113-121.

[17] H. Sariowan, "A service-curve approach to performance guarantees in integrated-service networks," Ph.D. dissertation, University of California, San Diego, CA, 1996.

[18] H. Zhang, "Service disciplines for guaranteed performance service in packet switching networks," *Proceedings of the IEEE*, vol. 83, no. 10, pp. 1374 -1396, 1995.

[19] D. Towsley, "Providing quality of service in packet switched networks," in *Performance Evaluation of Computer and Communication Systems*, (ed. L. Donatiello and R. Nelson), New York: Springer-Verlag, pp. 560-586, 1993.

[20] C. M. Aras, J. Kurose, D. S. Reeves, and H. Schulzrinne, "Real-time communication in packet-switched networks," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 122-139, 1994.

[21] S. J. Golestani, "Congestion-free communication in high-speed packet networks," *IEEE Transactions on Communications*, vol. 39, no. 12, pp. 1802-1812, 1991.

[22] H. Zhang and D. Ferrari, "Rate-controlled static priority queueing," in *Proceedings of the IEEE INFOCOM'93*, San Francisco, CA, 1993, pp. 227-236.

[23] J. Liebeherr, D.E. Wrege, and D. Ferrari, "Exact admission control in networks with bounded delay services," *IEEE/ACM Transactions on Networking*, vol. 46, no. 12, pp. 885-901, 1996.

[24] R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048-1056, 1995.

[25] L. Zhang, "VirtualClock: a new traffic control algorithm for packet-switched networks," *ACM Transactions on Computer Systems*, vol. 9, no. 2, pp. 101-124, 1991.

[26] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proceedings of ACM SIGCOMM'89*, Austin, TX, 1989, pp. 1-12.

[27] A. K. J. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[28] J. C. R. Bennett and H. Zhang, "WF^2Q: worst-case fair weighted fair queueing," in *Proceedings of the IEEE INFOCOM'96*, San Francisco, CA, 1996, vol. 1, pp. 120-128.

[29] S. J. Golestani, "A self-clocked fair queueing scheme for high speed applications," in *Proceedings of the IEEE INFOCOM'94*, Toronto, Canada, 1994, vol. 2, pp. 636-646.

[30] E. L. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1024-1039, 1991.

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

[32] 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, pp. 399-408, 1995.

[33] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control for EDF schedulers," in *Proceedings of the IEEE INFOCOM'97*, Kobe, Japan, 1997, vol. 1, pp.

310-317.

[34] N. Dunford and J. Schwartz, *Linear operators, Part I : General theory*. New York: Interscience publishers, 1958.

[35] R. Agrawal and R. Rajan, "Performance bounds for guaranteed and adaptive services," IBM Research Report RC 20469(91385), 1996.

[36] J.-Y. Le Boudec, "Network calculus make easy," Institute for Computer Communication and Applications, Lausanne, Switzerland, Technical Report no. 218, 1996; available http://icawww.epfl.ch.

[37] J.-Y. Le Boudec, "Application of network calculus to guaranteed service networks," *IEEE Transactions on Information Theory*, vol. 44, no.3, pp. 1087-1096, 1998.

[38] J.-Y. Le Boudec, "A note on time and space methods in network calculus," Institute for Computer Communication and Applications, Lausanne, Switzerland, Technical Report no. 224, 1997; available http://icawww.epfl.ch.

[39] J.-Y. Le Boudec, G. D. Veciana and J. Walrand, "QoS in ATM: Theory and Practice," in *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, 1996, vol. 1, pp. 773-778.

[40] J.-Y. Le Boudeci and P. Thiran, "Network calculus viewed as a min-plus system theory applied to communication networks," Institute for Computer Communication and Applications, Lausanne, Switzerland, Technical Report no. 016, 1998; available http://icawww.epfl.ch.

[41] R. L. Cruz, "Service guarantees for window flow control," in *Proceedings of the 34th Allerton Conference on Communication, control & Computing*, Monticello,

IL, 1996, pp. 1-12.

[42] R. L. Cruz, "Quality of service management in integrated service networks," in *Proceedings of 1st Semi-Annual research review*, San Diego, CA, 1996, pp. 1-9.

[43] R. L. Cruz and C.M. Okino, "Service guarantees for joint scheduling and flow control," in *Proceedings of the 36th IEEE conference on decision & control*, San Diego, CA, 1997, vol. 2, pp. 1804-1809.

[44] R. L. Cruz, "SCED+: efficient management of quality of service guarantees," in *Proceedings of the IEEE INFOCOM'98*, San Francisco, CA, 1998, vol. 2, pp. 625-634.

[45] R. L. Cruz, "An analysis of traffic clipping," in *Proceedings of the Conference on Information Sciences and Systems*, Princeton, NJ, 1998, pp. 1-6.

[46] H. Sariowan, R. L. Cruz and G. C. Polyzos, "Scheduling for quality of service guarantees via service curves," in *Proceedings of ICCCN'1995*, Las Vegas, NE, 1995, pp. 512-520.

[47] N. C. Audsley, A. Burns, M. F. Richardson, and A. J. Wellings, "Hard real-time scheduling: the deadline-monotonic approach," in *Proceedings of the 8th IEEE Workshop on Real-Time Operating Systems and Software*, Atlanta, GA, 1991, pp. 15-23.

[48] J. Y.-T. Leung and J. Whitehead, "On the complexity of fixed-priority scheduling of periodic, real-time tasks," *Performance Evaluation*, vol. 2, pp. 237-250, 1982.

[49] A. K. J. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344-357, 1993.

[50] A. K. J. Parekh and R. G. Gallage, "A generalized processor sharing approach to flow control in integrated services networks: the multiple-node case," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 137-150, 1994.

[51] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theorey of NP-Completeness.* San Francisco: Freeman, 1979.

[52] L. Kleinrock, *Queueing Systems*, volume II: Computer Applications. New York: Addison-Wesley, 1994.

[53] H. Zhang, "Providing end-to-end performance guarantees using non-work-conserving disciplines," *Computer Communications: Special Issue on System Support for Multimedia Computing*, vol. 18, no. 10, 1995.

[54] C. S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 913-931, 1994.

[55] S. J. Golestani, "A framing strategy for congestion management," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1064-1077, 1991.

[56] A. Acampora, "Wireless ATM: a perspective on issues and prospects," *IEEE Personal Communications*, pp. 8-17, August 1996.

[57] R. Ramjee et al., "Performance evaluation of connection rerouting schemes for ATM-Based wireless networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, pp. 249-261, 1998.

[58] M. Elaoud and P. Ramanathan, "Adaptive use of error correcting codes for real-time communication in wireless networks," in *Proceedings of the IEEE INFOCOM'98*, San Francisco, CA, 1998, vol.2, pp. 548-555.

[59] J. M. Capone and I. Stavrakakis, "Determining the call admission region for real-time heterogeneous applications in wireless TDMA networks," *IEEE Network*, vol. 12, no. 2, pp. 38-47, 1998.

[60] J. M. Capone and I. Stavrakakis, "Delivering QoS requirements to traffic with diverse delay tolerances in a TDMA environment," *IEEE/ACM Transactions on Networking*, vol. 7, no. 1, pp. 75-87, 1999.

APPENDIX A

IEEE REPRINT PERMISSION.

Sep 10 98 11:37a Brian 732-981-8062 p.1

Copyrights, Trademarks & Permissions
Phone 732.562.3966
Fax 732.981.8062

www.ieee.org/copyright
copyrights@ieee.org
pubs-permissions@ieee.org

IEEE
Networking the World™

9 September 1998

Chengzhi Li
Texas A&M Univ.
Dept. of Computer Science
College Station, TX 77843-3112

SUBJECT: 1) "Stability in ATM Networks" by Li et al.
2) "Static Priority..." by Li et al.
3) "Response Time Analysis..." by Li et al.

Dear Chengzhi Li:

This is in response to your letter of 30 July in which you have requested permission to reprint, in your upcoming dissertation, your above IEEE copyrighted papers. We are happy to grant this permission.

Our only requirement is that the following copyright/credit notice appears prominently on the first page of each reprinted paper, with the appropriate details filled in:

© 19xx IEEE. Reprinted, with permission, from *Proceedings of* (full conference name; place and date of conf.; page numbers).

Sincerely yours,

William J. Hagen/ln
William J. Hagen, Manager
IEEE Intellectual Property Rights

WJH:ln

The Institute of Electrical and Electronics Engineers, Inc.
445 Hoes Lane • P.O. Box 1331 •

VITA

Chengzhi Li received his Bachelor of Science degree in Applied Mathematics from Fuzhou University in China in 1982 and Master of Science degree in Operations Research from Xiamen University in China in 1985. During 1985 to 1991, he was a member of faculty in the department of computer science and system science in Xiamen University in P. R. China. He began graduate studies in the department of computer science in Texas A&M University in August 1996.

His permanent address is 5 Fuli Road, Zhangzhou City, Fujian 363000, P. R. China.