박 사 학 위 논 문

Doctoral Thesis

인터넷 망에서의 가용대역 예측과 측정 기반의 수락 제어

Available Bandwidth Estimation and Measurement-Based Admission Control in IP Networks

남 승 엽 (南 昇 燁 Nam, Seung Yeob)

전자전산학과 전기및전자공학전공

Department of Electrical Engineering and Computer Science

Division of Electrical Engineering

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

2004

인터넷 망에서의 가용대역 예측과 측정 기반의 수락 제어

Available Bandwidth Estimation and Measurement-Based Admission Control in IP Networks

Available Bandwidth Estimation and Measurement-Based Admission Control in IP Networks

Advisor : Professor Sung, Dan Keun

by

Nam, Seung Yeob
Department of Electrical Engineering and Computer Science
Division of Electrical Engineering
Korea Advanced Institute of Science and Technology

A thesis submitted to the faculty of the Korea Advanced Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical Engineering and Computer Science , Division of Electrical Engineering

Daejeon, Korea
2004. 6. 10.
Approved by

Professor Sung, Dan Keun
Advisor

인터넷 망에서의 가용대역 예측과 측정 기반의 수락 제어

남 승 엽

위 논문은 한국과학기술원 박사학위논문으로 학위논문심사위원회에서 심사 통과하였음.

2004년 5월 28일

심사위원장 성 단 근 (인)

심사위원 이 황 수 (인)

심사위원 정 송 (인)

심사위원 황 강 욱 (인)

심사위원 이 형 호 (인)

DEE 995119 남 승 엽. Nam, Seung Yeob. Available Bandwidth Estimation and Measurement-Based Admission Control in IP Networks. 인터넷 망에서의 가용대역 예측과 측정 기반의 수락 제어. Department of Electrical Engineering and Computer Science , Division of Electrical Engineering . 2004. 148p. Advisor Prof. Sung, Dan Keun. Text in English.

Abstract

The capacity of core networks has increased tremendously due to recent technology development in optical transmission and high-speed router/ethernet switches. However, IP networks originally designed to provide best-effort services can not guarantee strict or statistical quality-of-service (QoS) requirements for real-time traffic flows because resources are not reserved and all packets are treated equally in most nodes. Thus, it is very important to monitor network status and manage network resources in order to guarantee QoS for flows with real-time performance requirements. Since the delay performance strongly depends on the available bandwidth of the path among many network resources, this dissertation is concerned with monitoring the available bandwidth and proposing an admission control scheme of internet flows based on the estimated available bandwidth.

First, a new mechanism is proposed to estimate the available bandwidth of a queueing system, whose service rate and the load of input traffic are not known in advance. In order to estimate the available bandwidth, we propose a probing method called a *minimally backlogging method* and propose two statistics. The first statistic is based on the delay of each probing packet and the second statistic is based on the amount of probing packets served in a specific time interval. We first show that an $M/G/1$ queueing system is stable when probing packets are sent to the system according to the minimally backlogging method. We also show that the available bandwidth can be estimated by using either of the two statistics if the probing packets are sent to the queueing system by the minimally backlogging method. Especially, the second statistic can be used to estimate the available bandwidth of a $G/G/1$ queueing system. We apply the theory developed for a single server in order to estimate the available bandwidth for a local server as an application. The accuracy of the two proposed statistics is evaluated numerically under Poisson and

self-similar traffic loads.

Second, a new mechanism which estimates the available bandwidth for multiple hop routes is proposed by extending the approach for a single server, especially with the second statistic, and introducing a simplified path model which simplifies a multiple hop path into a combination of a fixed delay component and a virtual server. Since the proposed mechanism can estimate the available bandwidth quickly and track it adaptively and continuously, a reasonable range of available bandwidth for a short time interval can be obtained using the mean and variance of the estimated available bandwidth. The performance of the proposed available bandwidth estimation mechanism is evaluated by simulation in a multiple hop network topology.

Finally, a scalable architecture and an admission control algorithm for real-time flows are proposed. Since individually managing each traffic flow on each of its traversed routers causes a fundamental scalability problem in both data plane and control plane, we consider that each flow is classified at an ingress router and data traffic is aggregated according to the class inside the core network in our proposed resource management architecture as shown in a DiffServ framework. In our approach, admission decision is made for each flow at the edge (ingress or egress) routers, but it is scalable because the algorithm consists of simple arithmetic computations and a single comparison logic. In the proposed admission control scheme, an admissible bandwidth, which is defined as the maximum rate of a flow that can be accommodated additionally while satisfying the delay performance requirements for both existing and new flows, is calculated based on the available bandwidth which is estimated by edge routers through monitoring minimally backlogging probing packets. The admissible bandwidth is a threshold for admission control, and thus, it is very important to accurately estimate the admissible bandwidth. The performance of the proposed scheme is evaluated with a set of simulation experiments using highly bursty traffic flows.

Contents

List of Figures

1. Introduction

1.1 Background

The Internet was originally designed to provide best-effort services to all users. Thus far the Internet has not provided resource reservation mechanisms for Quality of Service (QoS) control, and all packets are treated equally. In the past this approach worked well, because applications did not require strict QoS in terms of delay. However, a drastic increase in the capacity of IP core networks due to high-speed optical fiber and high-speed router/ethernet switches, and the development of powerful compression techniques, led to creating new types of applications such as Internet telephony, multimedia streaming, and web casting. These applications generally require low end-to-end delay and low delay variation. To address this problem, the Internet Engineering Task Force (IETF) has proposed two different service models: Integrated Services (IntServ) [1] and Differentiated Services (DiffServ) [2].

In the IntServ architecture admission control is performed for each flow. Request from each flow is accepted (or rejected) depending on the level of available resources. A signalling protocol, RSVP [3] is used to reserve resources for each router along the path between a source and a destination. While this architecture can guarantee QoS, it may cause a significant scalability problem. Routers need to process per-flow reservation requests, and maintain per-flow forwarding and QoS states to guarantee QoS for each flow. If the number of flows is very large, the implementation may not be possible. Even though there are some attempts to make such designs more scalable through aggregation and hierarchy [4], this scalability of the IntServ architecture is still questionable.

DiffServ is another service model developed to provide QoS while avoiding the scalability problem of IntServ. DiffServ requires neither per-flow admission control nor signalling. Routers are divided into two groups, edge and core routers, in the DiffServ architecture. Only the edge routers process traffic on a per-flow basis. Core routers merely forward packets based on the DS field in the packet headers and does not maintain per-flow state. Thus, DiffServ scales well with an increasing number of flows, but it has some disadvantages compared to the IntServ model. Since admission control has not been defined yet, QoS can not be guaranteed through DiffServ.

Thus, scalability in DiffServ and QoS support capability in IntServ need to be combined to provide a scalable solution to QoS guarantee problems for real-time traffic. In order to guarantee QoS for real-time flows that are sensitive to the end-to-end delay, the network should provide some functionality both in user plane and control planes. In the user plane routers should have a capability to distinguish real-time traffic from best effort traffic and schedule them differently depending on the class of traffic in order to satisfy delay constraint and provide minimum bandwidth for high priority class traffic. Many scheduling algorithms such as Weighted Fair Queueing (WFQ) [5], Self-Clocked Fair Queueing (SCFQ) [6], Virtual Clock (VC) [7], Worst-case Fair Weighted Fair Queueing (WF^2Q) [8], and Rate-Controlled Earliest Deadline First (RC-EDF) [9] have been proposed to support QoS in the user plane thus far. Core routers can identify the class of each flow based on the DS field.

In order to guarantee QoS in terms of packet loss, a proper buffer management mechanism should be used. For buffer management schemes with priority, a push-out mechanism and a partial buffer sharing mechanism were proposed using the threshold or resume level control [10–15].

In the control plane, admission control, resource reservation and QoS state management are very important to support QoS. We can avoid maintaining per-flow states in core routers if we classify the flow according to the value of DS field and manage aggregate traffic for each class. Even though we serve several classes of traffic according to a strict priority scheduling policy, if high priority class is overloaded, all flows in that class suffer from a degradation in service and QoS may not be guaranteed for that class. Thus, in order to guarantee QoS for real-time flows, admission control is indispensable to limit the amount of offered traffic for the premium class.

1.2 Problem Statements

It is very important to reliably estimate available bandwidth of a path for high utilization of network resources as well as QoS guarantee for real-time flows. If the *available bandwidth* for a specific network path is known to a traffic source node, the source node can avoid paths in congestion in advance [16] or the information about the available bandwidth can be used for traffic engineering (TE) in IP/MPLS networks [17–19]. Thus, it is very important to monitor available bandwidth in order to exploit network resources efficiently. The first issue of this dissertation is to propose a quick and reliable mechanism for estimating the end-to-end available bandwidth without incurring overload to the network.

IntServ can not provide QoS support for real-time traffic over the global scale due to a serious scalability problem. DiffServ can not also satisfy delay constraints for real-time flows without an admission control scheme. Thus, we propose a new flow admission control scheme for DiffServ-like service models where no per-flow state is managed in core routers, there are only a small number of classes defined in the backbone domain, and thus, flows are treated on the aggregate traffic level based on the classes.

We develop a scalable architecture and an admission control scheme for real-time flows. We consider end-to-end delay as a QoS requirement because real-time flows are more sensitive to delay than loss. In our approach, admission control decisions are made at ingress routers, without maintaining per-flow state in either network core nodes or egress nodes, and without coordination of states with core nodes. Conventional admission control schemes usually send probing packets or signalling packets to the destination node or egress router upon receiving a request from a new flow [20, 21]. Thus, admission decision can be made after at least round-trip time from the request time. For real-time flows with a very small delay constraint or networks with long round-trip time, this can be a problem. In our scheme, admission decision is made promptly upon receiving a request from a new flow while estimating the available bandwidth for a specific path periodically independent of request arrivals. Thus, fast admission decision is possible.

The second issue of this paper is to develop an architecture and an admission control scheme by taking into account the following:

- guarantee of delay QoS;

- high utilization through statistical sharing among flows;
- scalability (no per-flow state management in core routers);
- fast admission decision

1.3 Chapter Organization

This dissertation is organized as follows: In Chapter 2, we review related works. In Chapter 3, we propose a probing method called a minimally backlogging method and propose two statistics in order to estimate the available bandwidth of a queueing system. In Chapter 4, we introduce a simplified network path model and propose a mechanism to estimate the available bandwidth for an end-to-end path based on the simplified path model and the minimally backlogging method. The proposed estimation mechanism is evaluated numerically. In Chapter 5, we propose an admission control scheme, and develop a mechanism to calculate admissible bandwidth, a threshold for admission control, based on the estimated available bandwidth. We evaluate the performance of the proposed admission control scheme numerically. Finally, conclusions and further studies are presented in Chapter 6.

2. Review of Related Works

2.1 Bandwidth-Related Metrics

We first introduce three bandwidth metrics: capacity, available bandwidth, and bulk transfer capacity (BTC). The first two can be defined for both individual links and end-to-end paths, while BTC is usually defined only for an end-to-end path. We only consider links at the IP layer (layer 3), which is also called *hops* in this chapter.

2.1.1 Capacity

The capacity C_i of a hop i is defined as the maximum possible IP layer transfer rate at that hop. The capacity C of an end-to-end path is defined as

$$C = \min_{1 \leq i \leq H} C_i,$$

where C_i is the capacity of the i-th hop, and H is the number of hops in the path. Thus, the end-to-end capacity C is determined by the minimum link capacity in the path. The link with the minimum capacity is the *bottleneck link* on the path.

2.1.2 Available Bandwidth

We first define the available bandwidth of hop i over a certain time interval. If C_i is the capacity of hop i and u_i is the average utilization of that hop in the given time interval, the available bandwidth A_i of hop i in the given time interval is defined as

$$A_i = (1 - u_i)C_i,$$

that is, the unused portion of the capacity.

The available bandwidth for an end-to-end path is defined as [22]

$$A = \min_{1 \leq i \leq H} A_i,$$

where H is the number of hops in the path. The link with the minimum available bandwidth is called the *tight link* of the path.

Since the available bandwidth can change dynamically over time, it is very difficult to estimate the available bandwidth quickly and accurately. However, quick and reliable estimation is required especially for applications that use available bandwidth measurements to adapt their transmission rates. In contrast, the capacity of a path typically remains constant for long time intervals if route changes or link failures do not occur. Therefore, the capacity of a path does not need to be measured as quickly as the available bandwidth.

2.1.3 Bulk Transfer Capacity

Another key bandwidth-related metric in TCP/IP networks is the throughput of a TCP connection. It is not easy to define the expected throughput of a TCP connection since several factors may influence TCP throughput, including transfer size, type and load of cross traffic (UDP or TCP), number of competing TCP connections, TCP socket buffer sizes at both sender and receiver sides, congestion along the reverse path, as well as the size of router buffers and capacity. Furthermore, the throughput of a large TCP transfer over a certain network path can vary significantly when different versions of TCP are used even if the available bandwidth is the same [22].

The BTC [23] is a metric that represents the achievable throughput by a TCP connection. BTC is defined as *the maximum throughput obtainable by a single TCP connection*. The connection must implement all TCP congestion control algorithms as specified in RFC 2581 [24].

We need to note that the BTC and available bandwidth are fundamentally different metrics. Different from BTC depending on TCP, the available bandwidth metric does not depend on a specific transport protocol. The BTC depends on how TCP shares bandwidth with other TCP flows, while the available bandwidth is the additional bandwidth a path can support before the tight link of the path is saturated. As an example to illustrate the difference, we consider a single-link path with capacity C that is saturated by a single TCP connection. The available bandwidth for this path would be zero due to path saturation, but the BTC would be about $C/2$ if the BTC connection has the same round-trip time (RTT) as the competing TCP connection.

2.2 Estimation of Available Bandwidth

The concept of *available bandwidth* has been important throughout the history of packet networks, from the aspects of both research and practice. In the context of transport protocols, robust and efficient use of available bandwidth has always been a major issue, including Jacobson's TCP [25]. The available bandwidth is also a crucial parameter in capacity provisioning, traffic engineering [17–19], optimal route selection in overlay networks [26], QoS management, streaming applications [27], server selection [28], and in several other areas.

While studies on characterizing bottleneck link capacity have received a lot of attention [29–36], how to estimate available bandwidth on an end-to-end Internet path is also becoming an important issue and has been studied recently. The first attempt to measure available bandwidth was C-probe [36]. The C-probe is to estimate the available bandwidth from the dispersion of trains of eight packets. A similar approach was taken in *pipechar* [37]. They assumed that the dispersion of long packet trains is inversely proportional to the available bandwidth. However, it was shown that this is not true by Dovrolis et al. [38]. The dispersion of long packet trains does not measure the available bandwidth in a path, but measures a different throughput metric that is referred to as *Asymptotic Dispersion Rate* (ADR).

Another available bandwidth measurement technique, called *Delphi*, was proposed in [39]. The main idea in Delphi is that the spacing of two probing packets at the receiver can provide an estimate of the amount of traffic at a link, provided that the queue of that link is not empty between the arrival times of the two packets. Delphi assumes that the bottleneck link bandwidth is known. Since Delphi assumes that the tight link is the same as the bottleneck link, this model is not applicable when the tight link is different from the bottleneck link.

Melander et al. [40] proposed a TOPP probing method which is an extension to the packet pair probing technique. TOPP uses sequences of packet pairs sent to the path at an increasing rate. They estimate the available bandwidth and the capacity of the link with the smallest link rate from the relation between the input and output rates of different packet pairs. The relation between the sending rate and the receiving rate is analyzed based on a segmented regression method. The regression method works well when the breakpoint of each segment is known, but in the case that the network is highly congested, it is usually difficult to obtain these breakpoints and apply the regression method. In addition, TOPP is computationally intensive to implement.

Jain and Dovrolis [41, 42] proposed a tool called *pathload*. Pathload is to estimate the range of available bandwidth iteratively, not the exact value of available bandwidth. Since the pathload tries to find the available bandwidth for a network path iteratively based on a binary-search algorithm, it has a rather long convergence time and may fail to accurately estimate the available bandwidth especially when the available bandwidth varies significantly before the iteration ends.

Ribeiro et al. [43] proposed a tool called *pathChirp*. PathChirp is based on the concept of self-induced congestion. PathChirp uses an exponentially spaced chirp probing train in order to rapidly increase the probing rate within each chirp and estimates the available bandwidth based on the queueing delay signature [43]. The optimal choice for the pachChirp-related parameters including the busy period threshold L and decrease factor F may depend on the cross-traffic statistics at queues on the path. Although pathChirp needs lighter probing load than for pathload, pathChirp's estimates usually have a negative bias yielding conservative results.

Hu and Steenkiste [44] proposed two available bandwidth measurement techniques: an initial gap increasing (IGI) method and a packet transmission rate (PTR) method. The IGI and PTR algorithms send a sequence of packet trains with increasing initial gap from the source to the destination host. Different from pachChirp, inter-packet spacing is fixed during a packet train and the probing rate varies for different packet trains. The IGI algorithm uses the information about changes in gap values of a packet train to estimate the competing bandwidth on the tight link of the path. The available bandwidth is obtained by subtracting the estimated competing traffic throughput from an estimate of the bottleneck link capacity. Since IGI is developed under the assumption that a tight link is also a bottleneck link and uses the bottleneck link capacity in the estimation of the available bandwidth, the accuracy degrades if there are errors in the bottleneck link capacity measurement or the tight link is not the bottleneck link. The PTR method uses the average rate of the probing packet train as an estimate of the available bandwidth. Although IGI and PTR yield the estimation results faster than pathload [42], their accuracy degrades when the tight link is different from the bottleneck link.

In order to estimate the available bandwidth for a network path quickly and accurately by overcoming the drawbacks of existing schemes, we propose a new available bandwidth estimation mechanism based on a simplified path model and a minimally backlogging concept.

2.3 Admission Control

Admission control algorithms for internet flows can be classified into two categories. The first is a model-based approach and the second is a measurement-based approach. In the model-based approach input traffic is usually mathematically modeled and admission is determined based on the mathematical model and the parameters characterizing input traffic. There were some approaches calculating effective bandwidth for a fluid input model or leaky-bucket regulated input traffic [45, 46]. Guerin et al. [45] evaluated the equivalent capacity of a set of connections multiplexed on a link defined as the amount of bandwidth required to achieve a desired QoS in terms of buffer overflow probability. They use a two-state fluid-flow model for input traffic. Assuming the burst and idle periods are exponentially distributed, they use a connection metric vector (R_{peak}, ρ, b), where R_{peak} is the peak rate of a connection, ρ is utilization of a connection, i.e., fraction of time the source is active, and b is the mean of the burst period. They obtain the smallest value $\hat{C}$ of the service rate c that ensures a buffer overflow probability smaller than ε for a given buffer size x as follows:

$$\hat{C} = \min \left\{ m + \alpha' \sigma, \sum_{i=1}^{N} \hat{c}_i \right\},$$

where m is the mean aggregate bit rate, σ is the standard deviation of the aggregate bit rate, $\alpha' = \sqrt{-2\ln(\varepsilon) - \ln(2\pi)}$, and $\hat{c}_i$ is the equivalent capacity of a single connection i. The detailed form of $\hat{c}_i$ is given in [45].

Elwalid et al. [46] considered admission control for leaky-bucket regulated input traffic. The token rate r bounds the long-term average rate of the regulated traffic, B_T is the token buffer size, and P is the burst size which bound the peak rate. They consider two cases of lossless multiplexing and statistical multiplexing. For lossless multiplexing, e_0 is referred to as the effective bandwidth for lossless performance. If $e_{0,i}$ is the effective bandwidth of the i-th virtual circuit, then the set of circuits $\{1, 2, \cdots, I\}$ is admissible if

$$\sum_{i=1}^{I} e_{0,i} \leq C,$$

where C is the transmission bandwidth of the multiplexer. The detailed from of $e_{0,i}$ is given in [46]. For statistical multiplexing, they consider loss ratio as a QoS requirement. Let J denote the number of classes, where each class is associated with a particular set of parameters for the regulator (r, B_T, P), and K_j denotes the number of virtual circuits of class j. The conservative bound of admissible set is obtained as

$$\tilde{A}_{L,\tilde{K}} = \left\{ \mathbf{K} : \sum_{j=1}^{J} K_j e_j \leq C \right\},$$

where L is the loss performance target, e_j is the effective bandwidth of class j traffic sources. e_j is obtained using Chernoff's bound [47] and the detailed form of e_j is given in [46].

The reliability of source models is a matter of concern in these model-based approaches. However, both a two-state fluid model of [45] and an on-off model of [46] developed for ATM network do not consider a long-range dependence property which is an important characteristic of the current internet traffic [48, 49]. It is possible to define effective bandwidth for input traffic modeled by fractional Brownian motion which has self-similarity and long range dependence by using a large deviation theory [50, 51]. However, even effective bandwidth based on a large deviation theory is not fully compatible with the realistic internet traffic according to [52]. A Fractional Stable Motion process proposed in [53] can capture not only the self-similarity of the traffic, but they also match its level of burstiness. The marginal behavior of Fractional Stable Motion processes is given by alpha-stable (long-tailed) distribution, of which the Gaussian distribution is a particular case. However, a meaningful definition of effective bandwidth for the general alpha-stable self-similar processes has not been proposed yet [52].

In addition, if we calculate the effective bandwidth just based on the parameters of long-range dependent traffic considering QoS requirements such as loss probability, the utilization of the bandwidth may be very low. Since the rate fluctuation of long-range dependent traffic is very large, if we allocate bandwidth conservatively considering the worst case, a large amount of bandwidth may be wasted for the duration of the flow. However, if we monitor the network status periodically, we can increase the bandwidth utilization by capturing the dynamic network status and allocating the resource accordingly. Measurement-based admission control algorithms (MBACs) can achieve a much higher utilization than parameter-based algorithms while providing somewhat relaxed QoS [54].

We can classify MBACs into three categories depending on the location of admission decision. First, admission decision is made at ingress end hosts. The end host probes the network by sending probe packets at the data rate it wants to reserve and recording the resulting level of packet losses (or ECN congestion marks [55]). The host then admits the flow only if the loss (or marking) percentage is below a threshold value. This type of admission control is called the *endpoint admission control* [21, 56]. The endpoint admission control requires no explicit support from routers; routers do not keep per-flow state information and do not process reservation requests, and routers drop or mark packets in a normal manner. Thus, the endpoint admission control does not have a scalability problem. However, probing inherently involves a significant set-up delay, on the order of seconds, and thus, not all real-time applications can tolerate such a long set-up delay. In addition, since it is not possible to police the amount of traffic offered by a host in this case, strict or statistical QoS can not be guaranteed by this endpoint admission control.

Thus, a source node determines whether to send a flow into the network by itself in the first type of MBAC. In the second type of MBAC, an egress end host performs admission control. As an example, admission control is needed for the overloaded web server in order to protect a severe degradation in throughput and to improve QoS [57]. Evaluation of web server performance generally focuses on achievable throughput and latency for a request-based type of workload as a function of traffic load. In case of commercial web servers, it is very important to process the entire sequence of requests needed to complete a transaction. Thus, Cherkasova et al. [57] considered the following requirement as a crucial web QoS:

- A fair chance of completion for any accepted session, independent of session length.

This type of admission control is appropriate for point-to-multipoint or multipoint-to-point services.

Third, admission decision is made at network nodes. Several measurement-based admission control algorithms belonging to the third type have been proposed [54, 58–67]. Each algorithm has two key components: a measurement process that produces an estimate of network load, and a decision algorithm that uses this load estimate to make admission control decisions. In each algorithm measurements are taken on the aggregate traffic without managing per-flow state and admission control decisions are made for each flow. Since it is difficult to predict future behavior accurately with traffic measurements, MBAC may result in occasional violation of the contracted QoS. It is reported that the admission control algorithms in [54, 58–66] can not meet statistical QoS targets in terms of loss ratio[68]. Each of these algorithms makes the admission decision on a link-by-link basis. Thus, these algorithms require the cooperation of intermediate nodes in the admission control process. Grossglauser and Tse [67] analytically investigated measurement-based admission control using a time-scale decomposition approach. However, their study is limited to a bufferless single link and the considered QoS is the overflow probability at a node, not an end-to-end QoS.

Cetinkaya et al.[69] proposed a scalable admission control algorithm called the *Egress Admission Control*. It achieves scalability by making admission control decisions only at egress routers without maintaining per-flow state. Admission decisions are made based solely on aggregate measurements obtained at a flow's egress router. A *service envelope* is introduced as a new concept to adaptively describe the end-to-end service available to a traffic class. The service envelope effectively exploits the features of backbone nodes' schedulers and the effects of statistical resource sharing at both the flow level and the class level when there are multiple service classes. Before the definition of statistical service envelope is given, the concepts of *essential traffic* and *available service* is defined as follows[69]:

Definition 2.1 (Essential Traffic) *The essential traffic of class n with respect to class i is defined as*

$$A^n_{D_i}(s,t) = A^n(s, t + D_i) \cap Y^n(s, t + D_i),$$

where $A^n(s,t)$ and $Y^n(s,t)$ denote the total class-n traffic arriving and served in time interval $[s,t]$, respectively, and D_i is the delay bound for class i. The essential traffic can be interpreted in the following way: if we suppose a class-i packet arrives at time t and is served exactly at its delay bound $t+D_i$, then $A^n_{D_i}(s,t)$ is the class-n traffic which will be served before the class-i packet. The essential traffic is a function of the particular service discipline.

Definition 2.2 (Available Service) *Let* $\tilde{A}^i(s,t)$ *denote the minimal class* i *input such that class* i *is continuously backlogged in* $[s,t]$ [1]. *The available service of class* i *in* $[s, t + D_i]$ *is defined as the class* i *output* $\tilde{Y}^i_{D_i}(s,t)$ *given this minimally backlogging input traffic* $\tilde{A}^i(s,t)$*, and other classes' input traffic as their essential traffic* $A^n_{D_i}(s,t), n \neq i$.

The available service $\tilde{Y}^i_{D_i}(s,t)$ is a function of the scheduling mechanism and the essential traffic $A^n_{D_i}(s,t), n \neq i$ and is independent of the input traffic of class i. Thus, the available service $\tilde{Y}^i_{D_i}(s,t)$ is decoupled from the input traffic of class i $A^i(s,t)$, while the actual output process $Y^i(s, t + D_i)$ is decided by the inputs of all classes.

Definition 2.3 (Statistical Service Envelope) *A sequence of random variables* $S^i_{D_i}(t)$ *is a statistical service envelope of class* i*'s traffic, if for any interval* $[s, s+t]$*, the available service* $\tilde{Y}^i_{D_i}(s, s+t)$ *satisfies*

$$\tilde{Y}^i_{D_i}(s, s+t) \geq S^i_{D_i}(t).$$

As we can know from Definition 2.2, the available service $\tilde{Y}^i_{D_i}(s,t)$ can be obtained only when there are existing flows $A^i(s,t)$ for the selected path, and consequently service envelope $S^i_{D_i}(t)$ can not be obtained for the first flow for the path. This is a problem of the Cetinkaya's admission control algorithm. In addition, if the load of the existing flows $A^i(s,t)$ are very low, it is difficult to obtain a reliable service envelope. Even if the load of the existing flows is not so low, the service envelope may not be obtained for the full observation window of the length of T.

[1] The concept of continuous backlogging is described in Chapter 4 in detail.

We circumvent the problem of Cetinkaya's admission control algorithm by introducing a different definition of available service, which can be obtained by monitoring probing packets sent through a specific path. Since the probing packets are offered in each window, the problem of low traffic load does not occur in the proposed admission control scheme.

3. Estimation of Available Bandwidth for an Unidentified Queueing System

3.1 Introduction

In this chapter, we investigate how to estimate the unused processing capacity, here called the available bandwidth, of a queueing system with an unknown service rate. Fig. 3.1 shows a queueing system of interest. C and λ denote the service rate and the arrival rate of packets except probing packets, respectively. Let L denote the average size of packets except probing packets. Then, for the queueing system, available bandwidth C_a is defined as

$$C_a = C(1 - \rho),$$

where $\rho = \lambda L/C$. If the parameters C, λ and L representing a queueing system are unknown, this system is said to be *unidentified* in this paper. We propose a new method to estimate the available bandwidth $C(1 - \rho)$ of an unidentified queueing system, in other words, we will show that the unused bandwidth of an unidentified queueing system can be estimated by sending minimally backlogging probing packets and measuring only the probing packets. The definition of minimal backlogging is given in the next section. The concept of minimally backlogging input packet sequence was introduced by Knightly [69, 70] in order to define *available service* between a specific node pair in the communication networks. The available service is also defined in this paper, but it is different from that defined in [69] or [70]. The

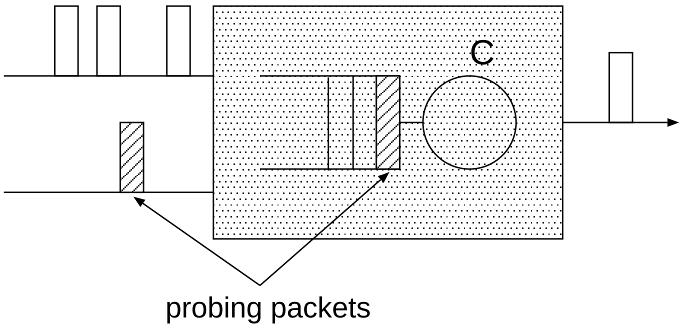


Figure 3.1: Unidentified Queueing System

difference will be described in Section 3.4 along with a new definition of available service.

In the area of communication networks, there have been several attempts to estimate the available bandwidth of a network path [36, 38–41, 44]. However, there has been no approach based on the minimally backlogging concept. Since a network path between a node pair usually consists of multiple hops, a tandem queueing system is required to accurately model a network path in communication networks. It is not a simple problem to estimate the available bandwidth of a tandem queueing system. In this paper we develop a theory to estimate the available bandwidth of a single queueing system. This work will be extended to a more complicated problem of estimating the available bandwidth for tandem queueing systems in the next Chapter.

Supposing that it is possible to send minimally backlogging probing packets, we consider two estimation schemes. The first scheme is to estimate the available bandwidth by measuring the delay of each probing packet, and the second scheme is to estimate the available bandwidth by measuring the total amount of probing packets served during a specific time period. The first estimation scheme is analyzed for an $M/G/1$ queueing system. Furthermore, the second scheme can be used to estimate the available bandwidth of a $G/G/1$ queueing system.

The rest of this chapter is organized as follows. In Section 3.2, we propose a probing method called a minimally backlogging method and investigate the stability of an $M/G/1$ queueing system when the minimally backlogging method is used. In Section 3.3, we propose a statistic based on the delay of each probing packet to estimate the available bandwidth of the $M/G/1$ queueing system. We show that the statistic becomes an unbiased estimator of the available bandwidth in case of probing the queueing system for an infinite duration and the mean square error converges to zero. In Section 3.4, we propose another statistic based on the amount of probing packets served in a specific time interval to estimate the available bandwidth of a $G/G/1$ queueing system. The second statistic is also an unbiased estimator of the available bandwidth with an infinite probing time and the mean square error converges to zero. In Section 3.5, as an application we consider the problem of estimating the available bandwidth of a local server. In Section 3.6, we evaluate the accuracy of two statistics numerically for a finite probing time under Poisson and self-similar traffic loads and evaluate the performance of the available bandwidth estimation scheme for a local server. Finally, conclusions are presented in Section 3.7.

3.2 Minimally Backlogging Method

In this section we propose a probing method to estimate the available bandwidth of a queueing system. This method is based on a minimally backlogging concept. We also investigate the stability of the queueing system when the probing packets are sent to the queueing system by a minimally backlogging method.

We consider an $M/G/1$ queueing system with a First-Come-First-Served (FCFS) service policy. λ denotes the arrival rate of packets and L is the average packet size. Suppose that the service time of a packet is given by the packet size divided by the service rate C of the system. Let G be the service time distribution of the packets and let S be a random variable corresponding to G. Then, the traffic load to the system is $\rho = \lambda E[S]$, which has the same value as $\lambda L/C$. We assume that $\rho < 1$ for the stability of the system. To consider the problem generally, we allow G_p, the service time distribution of probing packets, to be different from G. We let S_p denote a random variable corresponding to G_p. We define two terminologies as follows:

Definition 3.1 *A session is a sequence of packets sent to a queueing system by a user. A session is said to be in a backlogging state if there is at least one packet belonging to the session in the queueing system.*

Definition 3.2 *Suppose that probing packets are sent to a queueing system so that there exists one and only one probing packet in the system. This probing method is called a minimally backlogging method.*

If we send a new probing packet to a queueing system just at the departure time of the previous probing packet, then there exists one and only one probing packet in the system. Let X_i, $i = 1, 2, \ldots$ be the number of non-probing packets in the system seen by the i-th probing packet on arrival. Suppose that we start the probing for the $M/G/1$ queueing system in a stationary state. Then, X_1, the number of packets in the system seen by the first probing packet, is equal to the stationary queue length in number of packets, whose moment generating function is given in [71] as

$$\Pi(z) = \frac{(1-\rho)(1-z)\tilde{G}[\lambda(1-z)]}{\tilde{G}[\lambda(1-z)] - z}, \tag{3.1}$$

where $\tilde{G}(s) = \int_0^\infty e^{-sx}\, dG(x)$ is the Laplace transform of G.

Clearly, X_{i+1} is the number of packets arriving during the total service time of the X_i packets and the i-th probing packet. Let N_k^i be the number of non-probing packets arriving during the service time of the k-th non-probing packet among the X_i packets and let N_p^i be the number of non-probing packets arriving during the service time of the i-th probing packet. Since the arrival process of non-probing packets is a Poisson process, N_k^i depends only on the service time of the k-th packet. Thus, for all i and k, N_k^i's are independent and identically distributed. By the same reason, for all i, N_p^i are also independent and identically distributed. Now, we obtain the following relation:

$$X_{i+1} = \sum_{k=1}^{X_i} N_k + N_p, \tag{3.2}$$

where for all k, N_k is a random variable with the same distribution as N_1^1 and N_p with the same distribution as N_p^1, and each random variable is independent of the others. For simplicity, we will use N instead of N_1^1.

The probing based on the minimally backlogging method keeps the queueing server continuously busy. Thus, the probing may make the queueing system unstable. Theorem 3.1 answers this question.

Theorem 3.1 *Let X_i be the number of packets in the system upon arrival of the i-th probing packet. Then, $\{X_i,\ i = 1, 2, \ldots\}$ is an aperiodic and irreducible Markov Chain and it is positive recurrent.*

proof. By Eqn. (3.2), we can see that $\{X_i,\, i = 1, 2, \ldots\}$ is a Markov chain. Since N_k, $k = 1, 2, \ldots$ and N_p can have any nonnegative integers with a positive probability, $\{X_i,\, i = 1, 2, \ldots\}$ is irreducible and aperiodic. By Pakes [72], in order to show the positive recurrence, it suffices to show that

$$\text{i) } |E[X_{i+1} - X_i | X_i = n]| < \infty, n = 0, 1, 2, \ldots.$$

$$\text{ii) } \limsup_{n\to\infty} E[X_{i+1} - X_i | X_i = n] < 0.$$

By conditioning on X_i in Eqn. (3.2), we have that

$$E[X_{i+1} | X_i = n] = nE[N] + E[N_p]. \tag{3.3}$$

Since N is the number of Poisson arrivals during a random time of mean $E[S]$, it can be easily shown that $E[N] = \lambda E[S]$. By the similar reason, $E[N_p] = \lambda E[S_p]$. Then, Eqn. (3.3) is rewritten as

$$E[X_{i+1} | X_i = n] = n\rho + \lambda E[S_p]. \tag{3.4}$$

By subtracting n from the both sides of the above equation, we have that

$$E[X_{i+1} - X_i | X_i = n] = n(\rho - 1) + \lambda E[S_p].$$

Thus, for any n, $E[X_{i+1} - X_i | X_i = n]$ is finite. From the assumption that $\rho < 1$, it follows that $\lim_{n\to\infty} E[X_{i+1} - X_i | X_i = n] = -\infty$. □

By taking expectation on X_i in Eqn. (3.4), we derive that

$$E[X_{i+1}] = \lambda E[S_p] + \rho E[X_i], \quad i = 1, 2, \ldots.$$

The solution of the above recurrence relation is given by

$$E[X_i] = \frac{\lambda E[S_p]}{1-\rho} + \rho^{i-1}\left(E[X_1] - \frac{\lambda E[S_p]}{1-\rho}\right), \quad i = 1, 2, \ldots, \tag{3.5}$$

where $E[X_1]$ has a value of $\lambda^2 E[S^2]/[2(1-\rho)] + \rho$, the expected queue length of a stationary $M/G/1$ queueing system.

Let W_i be the waiting time of the i-th probing packet. By conditioning on X_i, we derive the Laplace transform of W_i as follows:

$$
\begin{aligned}
E[e^{-sW_i}] &= \sum_{n=0}^{\infty} E[e^{-sW_i} | X_i = n] \Pr\{X_i = n\} \\
&= \sum_{n=0}^{\infty} \tilde{G}_p(s) \tilde{G}(s)^n \Pr\{X_i = n\} \\
&= \tilde{G}_p(s) \Pi_i(\tilde{G}(s)),
\end{aligned}
\tag{3.6}
$$

where $\tilde{G}_p$ is the Laplace transform of G_p and $\Pi_i(z)$ is the moment-generating function of X_i. Differentiating the above equation and substituting $s = 0$, we obtain that

$$
E[W_i] = E[S_p] + E[S]E[X_i], \quad i = 1, 2, \ldots. \tag{3.7}
$$

From Theorem 3.1 we can see that the embedded Markov chain $\{X_i\}$ has a limiting distribution. To extend this result to the queue length process of an $M/G/1$ queueing system probed by the minimally backlogging method, we obtain the following theorem:

Theorem 3.2 *Suppose that we start probing an $M/G/1$ queueing system according to the minimally backlogging method. Let $\{X(t), t \in [0, \infty)\}$ be the queue length process of the queueing system. Then, $\{X(t)\}$ is a stable process, i.e. $\{X(t)\}$ converges to a stationary process. Moreover, $E[X(\infty)] < \infty$.*

proof. We assume that the first probing packet is sent to the queueing system at time 0 without loss of generality. Consider the epochs $\{\tau_1, \tau_2, \tau_3, \ldots\}$ such that there is no non-probing packet upon arrival of probing packets. Then, $\{X(t)\}$ is a regenerative process with regeneration points of $\{\tau_1, \tau_2, \tau_3, \ldots\}$. In order to show that $\{X(t)\}$ is stable, it is sufficient to show that the expectation of the length of a regeneration cycle is finite [71, Theorem 17 of Chapter 2].

Let a_t^p be the number of probing packets arriving until time t, i.e.,

$$a_t^p = 1 + \max\{n | \sum_{i=1}^{n} W_i \leq t\}. \tag{3.8}$$

Let $Z(t) = X_{a_t^p}$, where $\{X_n\}$ is the Markov chain defined in Theorem 3.1. Since the sojourn time of $Z(t)$ in state k is the total sum of service times of the number of k non-probing packets and a probing packet, the sojourn time only depends on k. This implies that $\{Z(t)\}$ is a semi-Markov process with embedded Markov chain $\{X_n\}$. Let μ_k be the expectation of the sojourn time of $Z(t)$ in state k, and π_k be the stationary distribution of $\{X_n\}$. Then,

$$\begin{aligned}\sum_{k=0}^{\infty} \pi_k \mu_k &= \sum_{k=0}^{\infty} \pi_k (kE[S] + E[S_p]) \\ &= E[S]E[X_\infty] + E[S_p].\end{aligned}$$

Since $E[X_\infty]$ is finite by Eqn. (3.5), $\sum_{k=0}^{\infty} \pi_k \mu_k$ is also finite. By [71, Theorem 9 of Chapter 4], we can see that $\{Z(t)\}$ is positive recurrent. Thus, the expectation of $\tau_{i+1} - \tau_i$ is finite. Now, we have shown that $\{X(t)\}$ is a stable process.

Since $\{X(t)\}$ is a stable regenerative process, $E[X(\infty)]$ is equal to $\lim_{t\to\infty} E[X(t)]$. In order to show the finiteness of $E[X(\infty)]$, it suffices to show that $\lim_{t\to\infty} E[X(t)] < \infty$. Let T_i be the time at which the i-th probing packet arrives. Then, $T_{a_t^p}$ is the latest arrival time of the probing packets until time t. Thus, $T_{a_t^p} \leq t < T_{a_t^p+1}$. Then, it follows that

$$X(t) \leq X(T_{a_t^p}) + a^n(T_{a_t^p}, T_{a_t^p+1}), \tag{3.9}$$

where $a^n(s,t)$ is the number of non-probing packets arriving during a time interval $[s,t]$. Since there is always one probing packet in the queueing system, $X(T_{a_t^p}) = X_{a_t^p} + 1$. Since the arrival process of non-probing packets is a Poisson process with rate λ and $T_{a_t^p+1} - T_{a_t^p}$ is equal to $W_{a_t^p}$, the random variable $a^n(T_{a_t^p}, T_{a_t^p+1})$ is a Poisson with a parameter of $\lambda W_{a_t^p}$. Then, from Eqn. (3.9), it follows that

$$E[X(t)] \leq E[X_{a_t^p}] + 1 + \lambda E[W_{a_t^p}]. \tag{3.10}$$

Depending on the value of $E[X_1] - \lambda E[S_p]/(1-\rho)$ in Eqn. (3.5), $E[X_i]$ is monotonically increasing or monotonically decreasing. Then, Eqns. (3.5) and (3.7) imply that for any i,

$$E[W_i] \leq \max\{E[S_p] + E[S]E[X_1], E[S_p]/(1-\rho)\}.$$

Thus, it can be deduced from Eqn. (3.8) that a_t^p goes to infinity almost surely. Then, Eqn. (3.10) gives

$$\begin{aligned} \lim_{t\to\infty} E[X(t)] &\leq 1 + \lim_{n\to\infty}(E[X_n] + \lambda E[W_n]) \\ &= 1 + \frac{2\lambda E[S_p]}{1-\rho}, \end{aligned}$$

where the last equality is obtained from Eqns. (3.5) and (3.7). □

3.3 Estimation based on Delay

In this section, we investigate how to estimate the available bandwidth of an $M/G/1$ queueing system by measuring the delay of each probing packet sent according to the minimally backlogging method.

Theorem 3.3 *Let W_i be the waiting time of the i-th probing packet. If we fix the size of the probing packets to a constant of L_p and let $\bar{W}_n = (W_1+W_2+\ldots+W_n)/n$, then*

$$\lim_{n\to\infty} E\left[\frac{\bar{W}_n}{L_p}\right] = [(1-\rho)C]^{-1}.$$

Thus, the statistic $\bar{W}_n/L_p$ is an asymptotically unbiased estimator of $[(1-\rho)C]^{-1}$.

proof. It follows from Eqn. (3.7) that

$$E\left[\sum_{i=1}^{n} W_i\right] = nE[S_p] + E[S]\sum_{i=1}^{n} E[X_i].$$

Since $\lim_{i\to\infty} E[X_i] = \lambda E[S_p]/(1-\rho)$ by Eqn. (3.5), we obtain that

$$\begin{aligned}\lim_{n\to\infty} E\left[\frac{\sum_{i=1}^{n} W_i}{n}\right] &= E[S_p] + E[S]E[X_\infty] \\ &= \frac{E[S_p]}{1-\rho}.\end{aligned}$$

Since the size of the probing packets is fixed to L_p, S_p is equal to L_p/C, which completes the proof. □

Theorem 3.3 says that $\bar{W}_n/L_p$ can be a candidate for an estimator of the available bandwidth. By the following theorem and corollary, we can observe that $\bar{W}_n/L_p$ is a good candidate.

Theorem 3.4 *Let W_i be the waiting time of the i-th probing packet and let $\bar{W}_n = (W_1+W_2+\ldots+W_n)/n$. Then, the variance of $\bar{W}_n$ converges to zero with order of $1/n$, moreover, for a constant c not depending on n,*

$$Var[\bar{W}_n] \le \frac{c}{n}.$$

proof. The proof is given in Appendix. □

Corollary 3.5 *Let W_i be the waiting time of the i-th probing packet. If we fix the size of the probing packets to a constant of L_p and let $\bar{W}_n = (W_1+W_2+\ldots+W_n)/n$, then*

$$\lim_{n\to\infty} E\left[\left|\frac{\bar{W}_n}{L_p} - [C(1-\rho)]^{-1}\right|^2\right] = 0.$$

proof. Let $Z_n = \bar{W}_n/L_p$. Then, by Minkowski's inequality, we can obtain

$$E\left[\left|Z_n - [C(1-\rho)]^{-1}\right|^2\right]^{\frac{1}{2}} \le E\left[|Z_n - E[Z_n]|^2\right]^{\frac{1}{2}} + \left|E[Z_n] - [C(1-\rho)]^{-1}\right|.$$

By Theorems 3.3 and 3.4, the right hand side of the above inequality converges to zero. This completes the proof. □

3.4 Estimation based on Packet Amount

In Section 3.3, we proposed a statistic to estimate the available bandwidth of an unidentified queueing system when the arrival process of non-probing packets is a Poisson process. We can estimate the available bandwidth by measuring the delay of each probing packet. In this section, we propose another statistic to estimate the available bandwidth of a queueing system when the arrival process of non-probing packets is a general process. The available bandwidth can be estimated by measuring the total amount of minimally backlogging probing packets that are served during a specific time period. We define the concept of *Available Service*, which is defined in a different way from that in [69, 70].

Definition 3.3 *The available service $\hat{Y}_{[s,t]}$ for a queueing system is the amount of probing packets served in interval $[s,t]$ when probing packets are sent to the queueing system according to the minimally backlogging method.*

Before we investigate the characteristics of the available service analytically, we briefly explain why the term of *available service* is used for $\hat{Y}_{[s,t]}$. In case that the minimally backlogging method is not used, an *idle period*, i.e. a time interval when the server is not busy, can exist if the load of non-probing packets are less than 1. In case that the probing packets are sent to the queueing system according to the minimally backlogging method, there always exists at least one probing packet in the queueing system, and thus, there is no idle period during the probing time. If there is no non-probing packet in the system, probing packets will be served continuously until a new non-probing packet arrives. Thus, we can know that the amount of probing packets served in a given time interval will be at least the maximum amount of service that the server can additionally support while serving all arriving non-probing packets according to an FCFS policy. On the other hand, the available service defined in [69, 70] represents the maximum amount of service that the server can do in a given time interval.

The size of each probing packet is fixed to a constant of L_p in this section. We assume that the first probing packet is sent to the system at time 0 without loss of generality. For simplicity, we will use $\hat{Y}_t$ instead of $\hat{Y}_{[0,t]}$. Then, the available service $\hat{Y}_t$ is expressed as

$$\hat{Y}_t = L_p \cdot \max\{n | \sum_{i=1}^{n} W_i \leq t\}.$$

Let Q_t denote the amount of packets in the queueing system at time t. Then,

$$Q_t = L_p + \sum_{k=1}^{X_t^n} L_k,$$

where X_t^n is the number of non-probing packets in the system at time t and L_k is the size of the k-th non-probing packet in the system. Let A_t be the amount of packets arriving during $[0,t]$ and let Y_t be the amount of packets served during $[0,t]$. Note that A_t consists of probing packets, A_t^p, and non-probing packets, A_t^n. Then,

$$Q_t = Q_0 + A_t - Y_t = Q_0 + A_t^n + A_t^p - Y_t. \tag{3.11}$$

The following lemma and theorem say that $\hat{Y}_t/t$ converges to $C(1-\rho)$ in L^q.

Lemma 3.6 *Let a_t^p be the number of probing packets arriving until time t. If probing packets are sent according to the minimally backlogging method, then $\lim_{t\to\infty} a_t^p = \infty$ almost surely ($a.s.$).*

proof. Eqn. (3.11) is rewritten as

$$A_t^p = Q_t - Q_0 + Y_t - A_t^n.$$

Since $Q_t \geq 0$, we have that $A_t^p \geq Y_t - A_t^n - Q_0$. Thus,

$$\liminf_{t\to\infty} \frac{A_t^p}{t} \geq \liminf_{t\to\infty} \frac{Y_t - A_t^n - Q_0}{t}. \tag{3.12}$$

By the assumption that the input load of non-probing packets is ρ, $\lim_{t\to\infty} A_t^n/t = \rho C$ $a.s.$ Since the server is continuously busy during the period of probing, $\lim_{t\to\infty} Y_t/t = C$ $a.s.$ Thus, it follows from Eqn. (3.12) that

$$\liminf_{t\to\infty} \frac{A_t^p}{t} \geq (1-\rho)C, \quad a.s. \tag{3.13}$$

Since $A_t^p = L_p a_t^p$, $\liminf_{t\to\infty} a_t^p/t \geq (1-\rho)C/L_p$ $a.s.$ Then, $\liminf_{t\to\infty} a_t^p = \infty$ $a.s.$ because $\rho < 1$. □

Theorem 3.7 *Let $\hat{Y}_t$ be the available service for a $G/G/1$ queueing system. The size of each probing packet is fixed to a constant of L_p. Then, for $0 < q < \infty$,*

$$\lim_{t\to\infty} E\left[\left|\frac{\hat{Y}_t}{t} - C(1-\rho)\right|^q\right] = 0.$$

proof. We first show that the input load is equal to 1, i.e., $\lim_{t\to\infty} A_t/(Ct) = 1$, a.s. In order to show this, it is sufficient to show that $\lim_{t\to\infty} Q_t < \infty$ a.s. because $Q_t = Q_0 + A_t - Y_t$ and $\lim_{t\to\infty} Y_t/(Ct) = 1$, a.s.

We define a set of sample paths Ω as $\Omega = \{\omega : \lim_{t\to\infty} Q_t(\omega) = \infty\}$. Choose a sample path $\omega \in \Omega$. For a real number $M > 0$, there exists $t_0(\omega)$ such that

$$t > t_0(\omega) \Rightarrow Q_t(\omega) > M. \tag{3.14}$$

Let Q_m be the amount of packets in the system upon arrival of the m-th probing packet. Then Eqn. (3.14) implies that $Q_m(\omega) > M$, for $m > a^p_{t_0}(\omega)$. Thus, we obtain that

$$W_m(\omega) = \frac{Q_m(\omega) + L_p}{C} > \frac{M + L_p}{C}, \quad \text{for } m > a^p_{t_0}(\omega). \tag{3.15}$$

Let M be an integer lager than ϵL_p, where $\epsilon > \rho/(1-\rho)$. Then, it follows from the above inequality that

$$W_m(\omega) > \frac{(1+\epsilon)L_p}{C}, \quad \text{for } m > a^p_{t_0}(\omega). \tag{3.16}$$

Since $A^p_t(\omega) = L_p a^p_t(\omega)$ and $\sum_{i=1}^{a^p_t(\omega)-1} W_i(\omega) \le t$, we obtain that for $t > t_0(\omega)$,

$$\begin{aligned} A^p_t(\omega) &\le \frac{a^p_t(\omega)\, L_p\, t}{\sum_{i=1}^{a^p_t(\omega)-1} W_i(\omega)} \\ &\le \frac{a^p_t(\omega)\, L_p\, t}{\sum_{i=a^p_{t_0}(\omega)+1}^{a^p_t(\omega)-1} W_i(\omega)} \\ &< \frac{a^p_t(\omega)}{a^p_t(\omega) - a^p_{t_0}(\omega) - 1} \frac{C}{1+\epsilon}\, t, \end{aligned}$$

where the last inequality is obtained by Eqn. (3.16). Then,

$$\frac{A^n_t(\omega) + A^p_t(\omega)}{t} < \frac{A^n_t(\omega)}{t} + \frac{a^p_t(\omega)}{a^p_t(\omega') - a^p_{t_0}(\omega) - 1} \frac{C}{1+\epsilon}. \tag{3.17}$$

We define Ω_1 and Ω_2 as follows:

$$\Omega_1 = \{\omega : \lim_{t\to\infty} A_t^n(\omega)/t = \rho C\},$$

$$\Omega_2 = \{\omega : \lim_{t\to\infty} a_t^p(\omega) = \infty\}.$$

We have shown that $\Pr\{\Omega_1\} = \Pr\{\Omega_2\} = 1$ in the proof of Lemma 3.6. For each $\omega' \in \Omega_1 \cap \Omega_2 \cap \Omega$, $a_t^p(\omega') \to \infty$ as t goes to infinity, and we can obtain from Eqn. (3.17) that

$$\limsup_{t\to\infty} \frac{A_t^n(\omega') + A_t^p(\omega')}{t} \leq \rho C + \frac{C}{1+\epsilon}.$$

Since $\epsilon > \rho/(1-\rho)$, we have that

$$\limsup_{t\to\infty} \frac{A_t^n(\omega') + A_t^p(\omega')}{t} < C.$$

The above equation implies that

$$\Pr\{\Omega_1 \cap \Omega_2 \cap \Omega\} \leq \Pr\left\{\limsup_{t\to\infty} \frac{A_t^n + A_t^p}{Ct} < 1\right\}. \tag{3.18}$$

If we write Eqn. (3.13) again,

$$\liminf_{t\to\infty} \frac{A_t^p}{t} \geq (1-\rho)C, \quad a.s.$$

Moreover, $\liminf_{t\to\infty} A_t^n/t = \lim_{t\to\infty} A_t^n/t = \rho C$ $a.s.$ Since $\liminf_{t\to\infty}(A_t^p + A_t^n)/t \geq \liminf_{t\to\infty} A_t^p/t + \liminf_{t\to\infty} A_t^n/t$, we have that

$$\liminf_{t\to\infty} \frac{A_t^p + A_t^n}{Ct} \geq 1, \quad a.s.$$

Applying the above inequality to Eqn. (3.18), we have that

$$\Pr\{\Omega_1 \cap \Omega_2 \cap \Omega\} = 0.$$

Since $\Pr\{\Omega_1\} = \Pr\{\Omega_2\} = 1$, we can easily show that $\Pr\{\Omega_1 \cap \Omega_2\} = 1$. Then, $\Pr\{\Omega\} = \Pr\{\Omega \cap (\Omega_1 \cap \Omega_2)\} + \Pr\{\Omega \cap (\Omega_1 \cap \Omega_2)^c\} \leq \Pr\{(\Omega_1 \cap \Omega_2)^c\} = 0$. Since $\Pr\{\Omega\}$ is non-negative, $\Pr\{\Omega\} = 0$. Thus, we have that

$$\lim_{t\to\infty} \frac{A_t}{Ct} = 1, \quad a.s.$$

Since $A_t = A_t^n + A_t^p$ and $\lim_{t\to\infty} A_t^n/t = \rho C,\ a.s.$, we have that

$$\lim_{t\to\infty} \frac{A_t^p}{t} = C(1-\rho), \quad a.s.$$

The difference of the values between A_t^p and $\hat{Y}_t$ is exactly the size of one probing packet. This means that $\lim_{t\to\infty} A_t^p/t = \lim_{t\to\infty} \hat{Y}_t/t, a.s.$ Thus, $\hat{Y}_t/t - C(1-\rho)$ converges to 0 in probability. By [73, Theorem 4.1.4.], in order to complete the proof, it is sufficient to show that there is a random variable Z in L^q such that for all $t > 0$, $|\hat{Y}_t/t - C(1-\rho)| \leq Z, a.s.$ Since the service rate of the system is C, the total amount of packets served during the time interval $[0, t]$ is less than Ct. Thus, $\hat{Y}_t$ is less than Ct. This implies that for $t > 0$,

$$\left| \frac{\hat{Y}_t}{t} - C(1-\rho) \right| < C(2-\rho).$$

□

3.5 Application to Estimation of Available Bandwidth of a Local Server

Thus far, we considered a problem of estimating the available bandwidth of a queueing system which is directly accessible with no access delay. However, in real situation, an unidentified queueing system may be physically separated from the probing site such that the access time delay is not zero due to a minimal propagation delay or minimal pre-processing time. Thus, we consider the problem of estimating the available bandwidth when there exists access time delay between the target queueing system and a probing site. The target queueing system is called a local server in this section.

Fig. 3.2 shows a simplified path model for a local server. Although the local server is a multi-stage switching system, if significant queueing delay occurs only at one stage and the variations of the queueing delay at other stages are negligibly small, the packet transmission time from the measurement point to the major queueing stage can be modeled as a constant. Then, we can obtain a simplified path model consisting of a fixed delay component D_f and a single server as shown in Fig. 3.2. Even in case of estimating the available bandwidth for a single stage switch or router, there may exist a processing delay or a propagation delay, which corresponds to D_f. Since most routers have a processing delay component such as routing table look-up time, the path model of Fig. 3.2 is more realistic than a simple queue. Extending the approach developed in this chapter, we propose a new method to estimate the available bandwidth of a local server. The proposed method is also based on the minimally backlogging concept. The performance of the proposed method is evaluated by simulation.

Fig. 3.3 illustrates a measurement process for estimation of the available bandwidth of a local server. The application or machine at a measurement point A sends probing packets to the local server and receives feedback information. The feedback information includes the departure time of a probing packet from the server, and the delay of each probing packet. Based on the feedback information, Node A measures the amount of probing packets arriving during a specific time period. We assume that the feedback information is available to Node A after a negligibly small delay. Probing packets sent from Node A experience a fixed delay of D_f, which is unknown in advance. Thus, the minimally backlogging method described in Section 3.3 or 3.4 can not be directly applied to this case. It is very important to reliably estimate the fixed delay component D_f in order to use the minimally backlogging concept to estimate the available bandwidth of a local server.

The estimation procedure consists of three steps. In the first step, we estimate the service rate C of the queueing system. In the second step, we estimate the fixed delay D_f. Finally, we estimate the available bandwidth based on C, D_f, and the minimally backlogging concept. The length of each probing packet is fixed to a constant of L_p. We send n consecutive probing packets back-to-back to the queueing system in order to estimate the service rate C. Let p_i denote the i-th probing packet among n probing packets. Suppose that the aggregate arrival rate of the input traffic is higher than C during the back-to-back probing time. If n is sufficiently large, there may exist a probing packet pair (p_j, p_{j+1}), between which there is no other packet. If we let e_i be the time when the i-th probing packet departs from the server, then $e_{j+1} - e_j = L_p/C$. Thus, we estimate C by

$$\hat{C} = \frac{L_p}{\min\limits_{1 \le i \le n-1} (e_{i+1} - e_i)}.$$

We now investigate how to estimate the fixed delay D_f. Let d_i, s_i, and w_i be the one-way delay from the departure time from the measurement point to the departure time from the server, the service time, and the queueing delay of the i-th probing

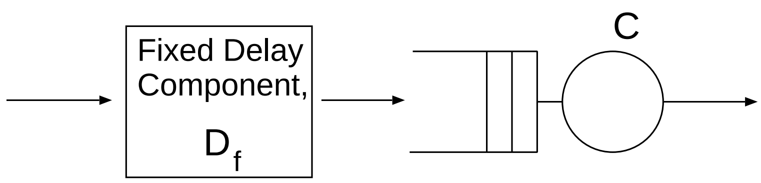


Figure 3.2: A simplified path model for a local server

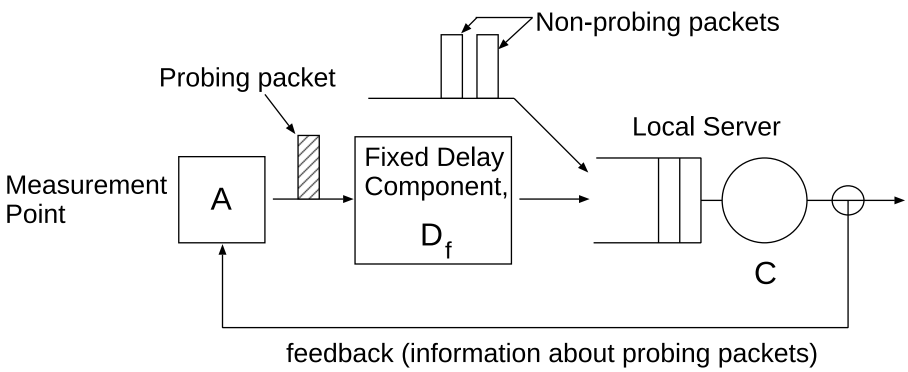


Figure 3.3: A measurement setup for estimation of the available bandwidth at a local server

packet, respectively. Since each probing packet experiences a fixed delay of D_f, the one-way delay d_i can be expressed as

$$d_i = D_f + w_i + s_i, \quad \text{for } i = 1, 2, \cdots, n.$$

If there exists a number j such that $w_j = 0$, then $d_j = D_f + s_j$. Since the length of each probing packet is fixed to L_p, $s_j = L_p/C$. Thus, the value of D_f can be obtained from $D_f = d_j - L_p/C$. If the arrival process of the non-probing packets is a Poisson process, we can obtain the following result:

Theorem 3.8 *Suppose that non-probing packets arrive at a local server according to a Poisson process with an arrival rate of* λ. *Let* w_i *be the time that* i*-th probing packet waits in the server before service. If we send probing packets to the simplified path model for the local server, which consists of a fixed delay component* D_f *and a single server, according to the minimally backlogging method, then* $\lim_{n\to\infty} \min_{1\le i\le n} w_i = 0$.

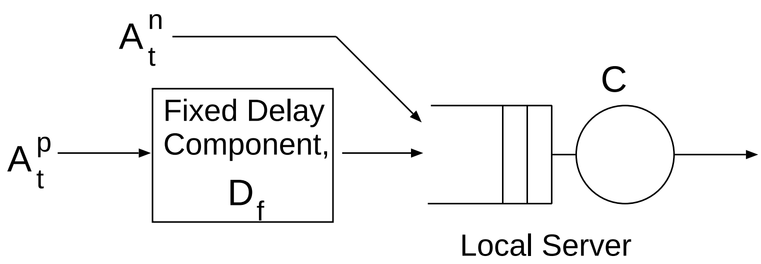
Proof: In order to prove the theorem easily, we introduce an equivalent path model for a local server. Fig. 3.4(a) shows the original path model for a local server, and Fig. 3.4(b) shows an equivalent path model corresponding to Fig. 3.4(a). We assume that the first probing packet is sent to the server at time 0 in Fig. 3.4(a). In Fig. 3.4(a), A_t^n is the amount of non-probing packets arriving at the server during $[0, t]$, and A_t^p is the amount of probing packets sent toward the server during $[0, t]$. We assume that the first probing packet is sent to the server at time D_f and the same input process of non-probing packets A_t^n is applied to the server from time 0 in Fig. 3.4(b). Since probing packets are sent according to the minimally backlogging method, i.e., the next probing packet is sent from the measurement node upon departure of the previous probing packet from the path, the queueing behavior at the server in Fig. 3.4(a) is the same as that in Fig. 3.4(b). Especially, if X_i denotes the number of packets in the server seen by the i-th probing packet on arrival at the server, the distribution of X_i in Fig. 3.4(a) is the same as that of X_i in Fig. 3.4(b).

The path model in Fig. 3.4(b) can be modeled as an $M/G/1$ queueing system if we incorporate D_f into the service time of the server as a minimum fixed service time in case of no probing packets. Then, by Theorem 3.1 of Section 3.2 $\{X_i, i = 1, 2, \ldots\}$ becomes a positive recurrent Markov chain. By the positive recurrent property of $\{X_i\}$, there exists a number k such that $X_k = 0$ with probability of one. Thus, the proof is completed. □

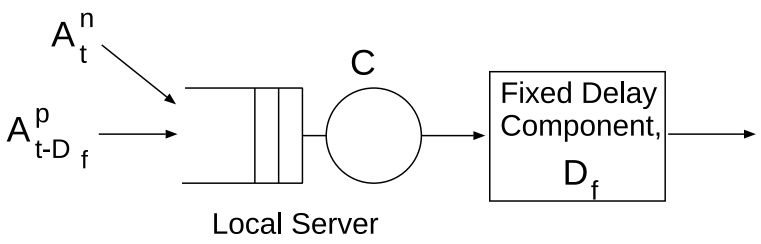
Thus, in general case we estimate D_f by

$$\hat{D}_f = \min_i \{d_i\} - L_p/C. \tag{3.19}$$

As a final step, we propose a heuristic method to estimate the available bandwidth of a local server based on the second statistic of Section 3.4. According to the proposed minimally backlogging method described in Section 3.2, probing packets should be sent to the server while maintaining one and only one probing packet in the local server. However, even if we know the exact value of D_f, it is not easy to send probing packets while maintaining one and only one probing packet in the server. Thus, we attempt to maintain a minimally backlogging condition with the following heuristic method assuming that the values of C and D_f are estimated in advance. The proposed method is based on the idea that if probing packets are sent to the server according to the minimally backlogging method, the inter-packet



(a) Original path model for a local server



(b) Equivalent path model for a local server

Figure 3.4: Path models for a local server

spacing between two consecutive probing packets is equal to the sojourn time of the former probing packet. The proposed available bandwidth estimation method is described as follows:

1. The measurement node sends a probing packet to the local server and obtains the delay d_0 of the probing packet from the feedback information.

2. The measurement node sends the first probing packet p_1 for estimation of the available bandwidth after acquiring d_0.

3. Let p_j be the last probing packet that is sent toward the server and let v_j be the time when p_j is sent to the server. If the last delay value available to the measurement node is d_i, we estimate the sojourn time of p_j in the server as $d_i - D_f$, and thus, the next probing packet is sent at time $v_j + d_i - D_f$. Exceptionally, if the last probing packet p_j arrives before $v_j + d_i - D_f$, there is no probing packet in the path. Thus, the next probing packet is sent upon arrival of p_j in order to maintain at least one probing packet in the server.

4. The measurement node measures the available service $\hat{Y}_t$ whenever the feedback information arrives, and estimates the available bandwidth by $\hat{Y}_t/t$ using the second statistic of Section 3.4.

Thus far, we assumed that the feedback information is available to the measurement node without delay. If the feedback delay is not zero, the fixed delay estimation step needs to be modified a little. Assume that the feedback information is available to the measurement node after a constant delay of D_b. In that case, Theorem 3.8

is also valid if we consider a path model whose fixed delay component is $D_f + D_b$ instead of D_f. Thus, the fixed delay component can be estimated by using the minimally backlogging method for the modified path model. If d_i is the one-way delay excluding the feedback delay, D_f can be estimated using Eqn. (3.19). If d_i is the round-trip delay from departure to arrival at the measurement node, $D_f + D_b$ can be estimated by

$$\hat{D}_{fix} = \min_i \{d_i\} - L_p/C.$$

For the available bandwidth estimation procedure, if d_i is the one-way delay, the above procedure can be used without modification. If d_i is the round-trip delay, the available bandwidth can be estimated from the above procedure if $D_f + D_b$ is used instead of D_f in the third stage of the procedure.

3.6 Numerical Results

We showed that the first statistic based on packet delay is an unbiased estimator of the reciprocal of the available bandwidth and the second statistic based on packet amount is an unbiased estimator of the available bandwidth if the queueing system is probed by the minimally backlogging method for an infinite time duration. However, it is not possible to probe a queueing system for an infinite time period. Thus, we evaluate the accuracy of the two statistics numerically in case of probing a queueing system during a finite time duration.

Fig. 3.5 shows a simulation topology for estimation of the available bandwidth of the unidentified queueing system. The measurement node directly connected to the queueing system sends probing packets to the queueing system by the minimally

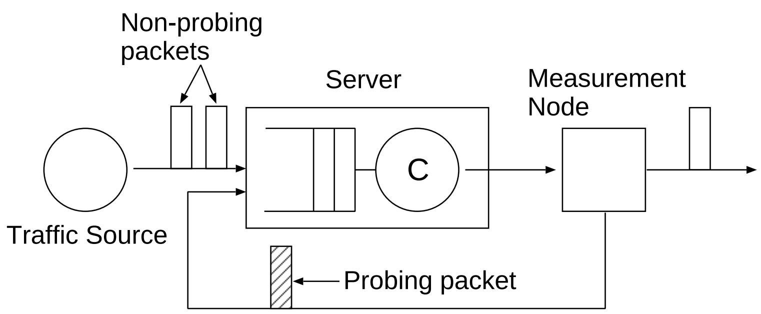


Figure 3.5: A measurement setup for estimation of the available bandwidth

backlogging method, i.e., the node sends a new probing packet upon arrival of the previous probing packet and calculates the values of two statistics. The measurement node bypasses every non-probing packet.

The traffic source generates two types of non-probing packet traffic patterns: Poisson and self-similar traffic. The traffic patterns of today's IP networks have been known to exhibit self-similarity and long-range dependence [74–76]. Neither of them can be modeled using conventional Markovian models. Thus, we use multi-fractal model [77] to generate self-similar traffic. Since it is reported that some internet traffic exhibits Hurst parameters in the range of $0.7 - 0.8$ [74, 75], we use the Hurst parameter of 0.8. The sizes of both probing and non-probing packets are fixed to 500 bytes. The service rate (C) of the unidentified queueing system is 10 Mbps.

Fig. 3.6 compares the estimated available bandwidth (AB) with the measured AB under a Poisson traffic load. The *Statistic #1* and *Statistic #2* denote the AB estimated by the statistic based on the amount of probing packets served in a specific time interval and the AB estimated by the statistic based on the delays of probing packets, respectively. The value of *Measured AB* is obtained in the queueing system by subtracting the service rate of non-probing packets from the service rate C when the probing traffic is not sent. The same traffic patterns are used for both estimation and measurement of the AB at the same load. We can observe that the estimation results obtain by *Statistic #1* and *Statistic #2* agree well with the measured AB for all traffic loads. In addition, the estimation results are accurate even when the observation time is short. The reason can be explained as follows. We know that the estimation result converges to the AB of $C(1-\rho)$ when the observation time goes to infinity by Theorem 3.7. Let us consider a finite time interval $[s,t]$ after start of probing. Then, the server is continuously busy for the interval $[s,t]$ because there is at least one probing packet in the queueing system. When the server does not serve non-probing packets, the server surely serves probing packets. Thus, all unused capacity of the server is used by probing packets in any finite interval. If the probing traffic is greedy like TCP flows, then the throughput of non-probing packets may be degraded. However, since probing traffic tries to prevent from being greedy by maintaining only one probing packet in the queueing system, the AB is estimated reasonably in a finite time interval.

Fig. 3.7 compares the estimated available bandwidth (AB) with the measured AB under a self-similar traffic load. The sigma/mean ratio of self-similar traffic is 0.68, 0.54, and 0.39 for the loads (ρ) of 0.3, 0.5, and 0.7, respectively. First, we can observe that the traffic is even more bursty than the case of Poisson traffic. Thus, it takes longer time for the average rate of the traffic converges to $C\rho$ for all traffic loads compared with the case of Poisson traffic. However, the AB's estimated by statistics #1 and #2 agree well with the measured AB regardless of the length of the observation time for various input loads as shown in Fig. 3.7. Due to the burstiness and the long-range dependence of the traffic, it takes longer time for the estimated AB to converge to $C(1-\rho)$ compared with the case of Poisson traffic.

We now evaluate the accuracy of the proposed estimation method for a local server through simulation. Fig. 3.3 shows a reference interconnection of a local server and a measurement node for estimation of the available bandwidth. We

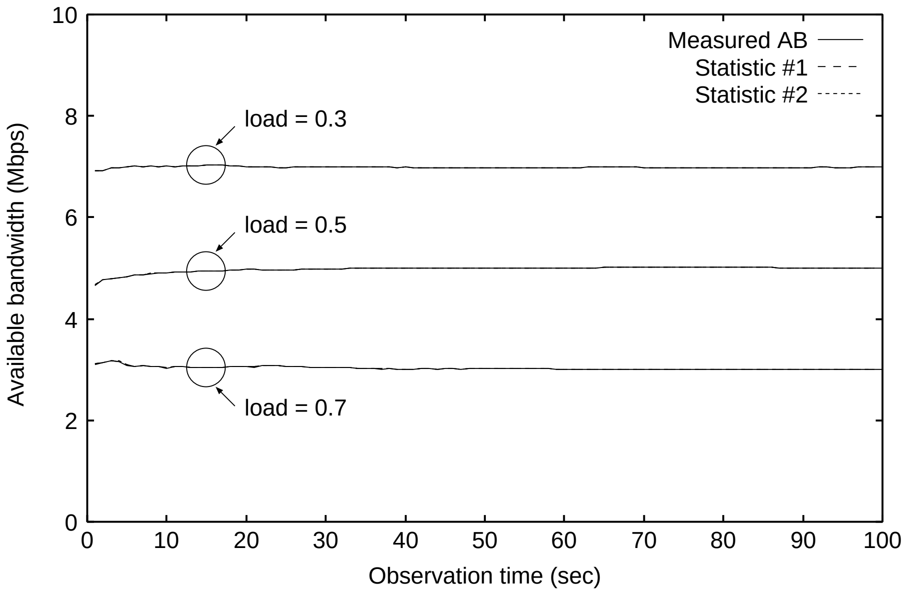


Figure 3.6: Comparison of the estimated AB and the measured AB under a Poisson traffic load

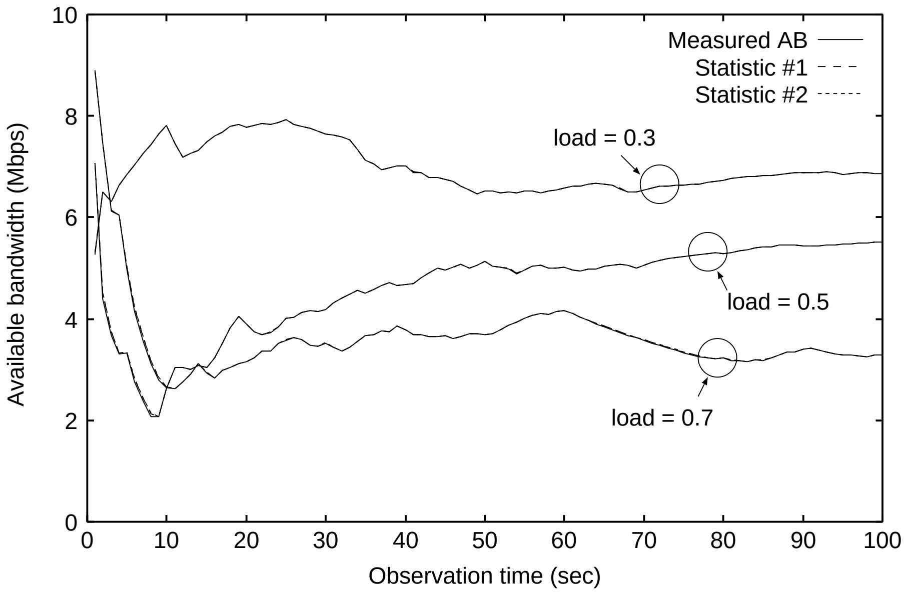


Figure 3.7: Comparison of the estimated AB and the measured AB under a self-similar traffic load

assume that the feedback delay is zero. We also assume that the value of the service rate C is reliably estimated in advance. In the simulation, the local server is a queueing system with an FCFS policy and the service rate C is fixed to 10 Mbps. The size of each data packet is fixed to 500 bytes.

Fig. 3.8 shows the convergence time of the estimator $\hat{D}_f$ in Eqn. (3.19) when D_f is 0.001 sec. The simulation is performed for two types of input traffic with various offered loads. We consider two probing packet sizes: 1 kbits and 12 kbits. In most cases, $\hat{D}_f$ converges to D_f within approximately 2 secs. However, the convergence time is approximately 20 secs when the probing packet size is 12 kbits under a Poisson offered load and the load is 0.9. Thus, a small probing packet size is adequate for D_f estimation because the system can be probed more frequently in a given time. Hereafter, we assume that a reliable value of D_f is obtained.

For estimation of the available bandwidth, a large size of probing packets is preferred. If the fixed delay D_f is large, it is difficult to maintain the minimally backlogging condition for the local server because it takes long for the source to react to the increased or decreased queueing delay at the server. For a given D_f, if we use a large size of probing packets, the queueing delay of a probing packet increases compared with the fixed D_f and the effect of fixed delay can be decreased. Thus, the size of probing packets is fixed to 12 kbits hereafter.

Fig. 3.9 compares the estimated available bandwidth (AB) with the measured AB under a Poisson traffic load for a D_f value of 0.1 msec. We can observe that the proposed method estimates the measured AB very accurately regardless of a short or long length of observation time. The proposed method is accurate for both low

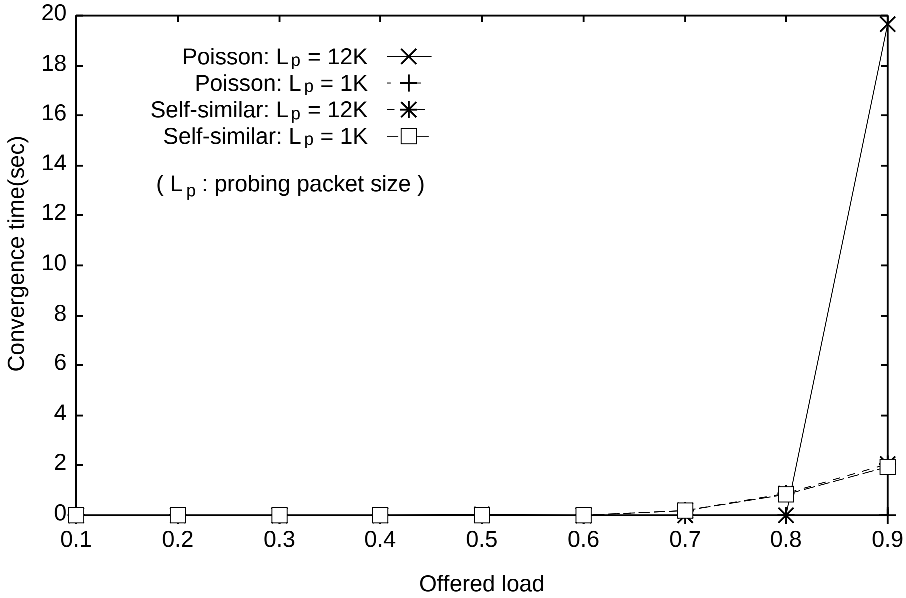


Figure 3.8: Convergence time of $\hat{D}_f$ for various input traffic load

and high traffic loads. Fig. 3.10 compares the estimated AB with the measured AB under a self-similar traffic load. In this case, the estimation results agree well with the measured ones for various traffic loads. Thus, the proposed method estimates the AB accurately if the fixed delay D_f is low compared with the sojourn time. For a packet size of 12 kbits, the sojourn time is at least 1.2 msec.

Fig. 3.11 shows the accuracy of the proposed method for various values of D_f when the offered load is 0.3 and the observation time is 100 sec. We can observe that the accuracy degrades as D_f increases. The reason is that long response time makes it difficult to maintain the minimally backlogging condition for the local server. Especially, as considered in the third stage of the AB estimation procedure, if the last probing packet sent arrives before the next probing packet is sent, the local server remains in a probing-packet-free state for at least D_f. In other words, the next probing packet arrives late at the local server D_f, compared with the case that the probing packets are sent ideally according to the minimally backlogging method. Thus, the amount of probing packets sent to the local server in a given time is always less than that of the ideal case due to D_f. Thus, the estimation result of the proposed method is conservative if D_f is significantly large. However, if the value of D_f is not so large for a local server or router, the proposed estimation method can work reliably.

3.7 Summary

A new estimation method of the available bandwidth for an unidentified queueing system is proposed using a minimally backlogging concept. Two statistics are also

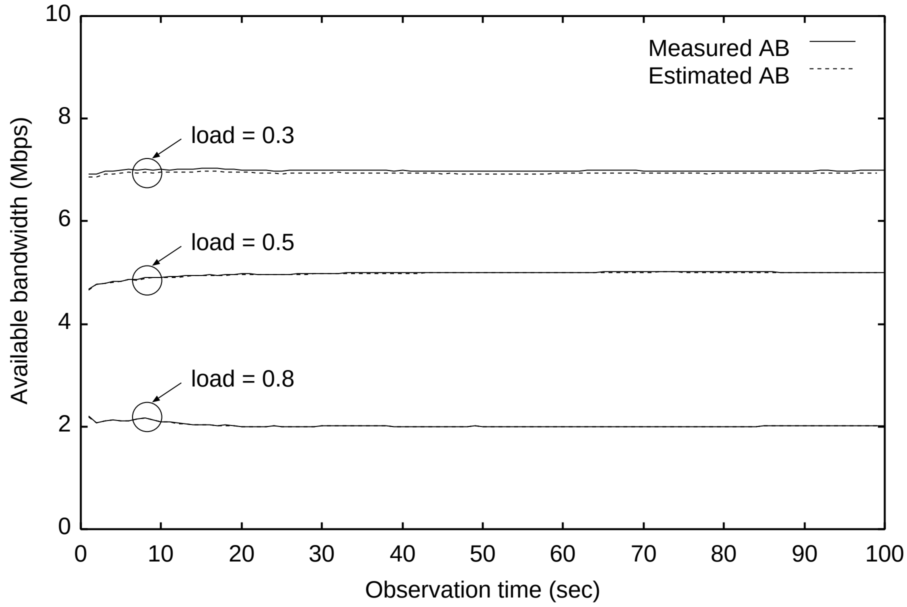


Figure 3.9: Comparison of the estimated AB and the measured AB under a Poisson traffic load

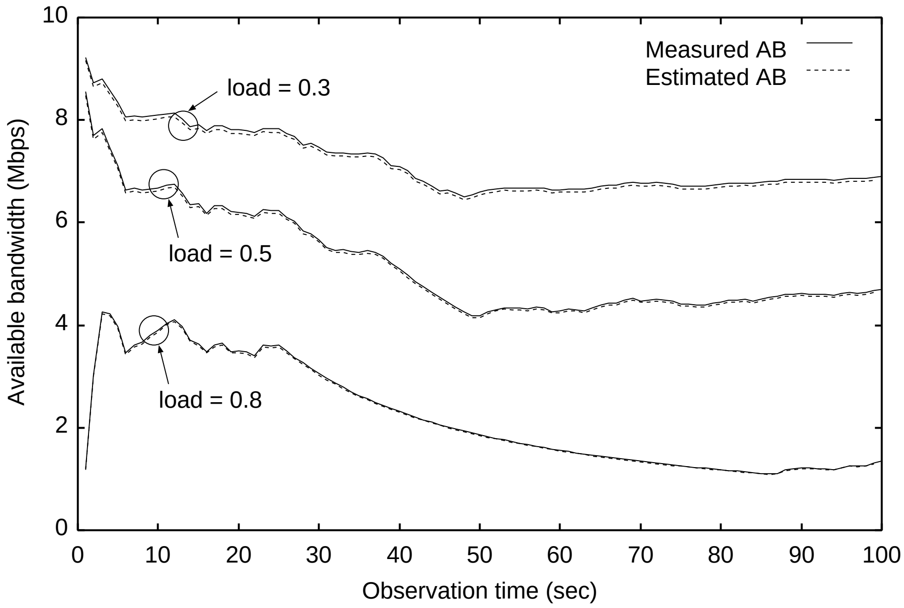


Figure 3.10: Comparison of the estimated AB and the measured AB under a self-similar traffic load

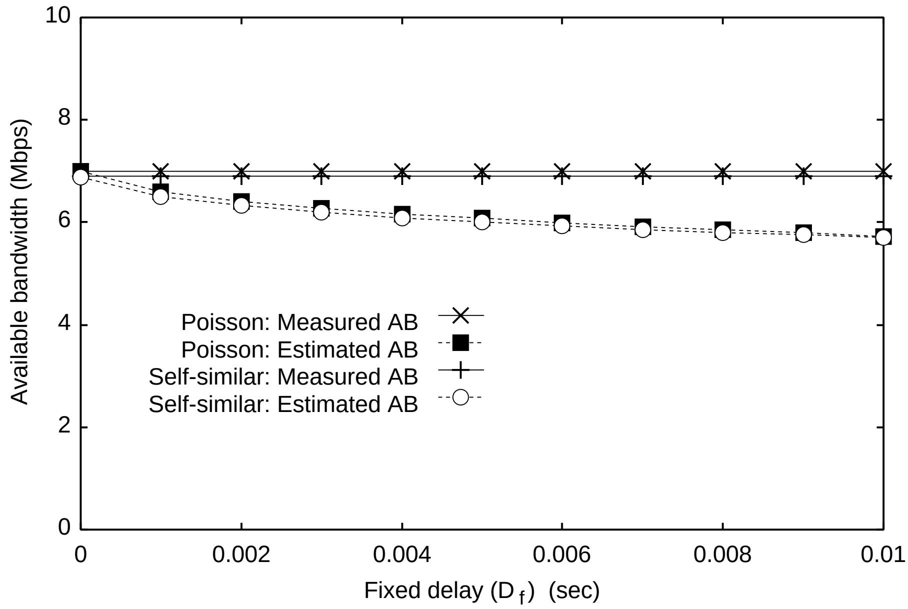


Figure 3.11: Comparison of the estimated AB and the measured AB for various D_f values

proposed to estimate the available bandwidth: the first one is based on the delay of each probing packet and the second one is based on the amount of probing packets served during a specific time period. If the probing packets are sent to the queueing system according to the minimally backlogging method, the available bandwidth of the system can be estimated by either of two statistics. If the load of input traffic for an $M/G/1$ queueing system is less than 1, the queueing system is still stable when the minimally backlogging method is used. The first statistic is an asymptotically unbiased estimator of the reciprocal of the available bandwidth and the mean square error converges to zero. The second statistic is an asymptotically unbiased estimator of the available bandwidth with a mean square error converging to zero. The second statistic can be used to estimate the available bandwidth of a $G/G/1$ queueing system. Though the two statistics are unbiased estimators of the available bandwidth or its reciprocal in case of an infinite probing time, since infinite probing time can not be realized, we evaluated the accuracy of two statistics by simulation and observed that two statistics agree well with the measured available bandwidth even for a finite probing time.

We also proposed a scheme to estimate the available bandwidth of a local server by extending the theory for a single server. The proposed scheme yields an accurate estimation result for various traffic loads when the fixed delay is relatively small compared with the queueing delay at the local server.

4. Estimation of End-to-End Available Bandwidth

4.1 Introduction

It is very important to allocate and manage resources for multimedia traffic flows with real-time performance requirements in order to guarantee quality of service (QoS). If *available bandwidth* for a specific network path is known to an application, congestion can be avoided in advance at the application level [16] or the information about available bandwidth can be used for traffic engineering (TE) in IP or MPLS networks [17–19]. Thus, monitoring the available bandwidth is very important to exploit network resources efficiently.

For a path $\mathcal{P}$ consisting of H serially connected links, the available bandwidth C_a for the path in a given time interval is defined as

$$C_a = \min_{1 \leq i \leq H} C_i(1 - u_i),$$

where C_i and u_i denote the link rate and the utilization of the i-th link in the given time interval, respectively. The link with the least unused bandwidth of C_a is referred to as *tight link* and the link with the minimum link rate is referred to as *bottleneck link.*

There are three requirements for good available bandwidth estimation mechanisms: accuracy, speed, and non-intrusiveness. In other words, it is required to provide an accurate estimate of the available bandwidth for a path quickly, without generating significant traffic load and without affecting the throughput of other traffic in the path.

Several methods have been proposed to estimate this available bandwidth. Carter and Crovella[36] developed a tool called the C-probe to estimate the available bandwidth from the dispersion of trains of eight packets. They assumed that the dispersion of long packet trains is inversely proportional to the available bandwidth. However, it was shown that this is not true by Dovrolis et al.[38]. Melander et al.[40] proposed a TOPP probing method which is an extension to the packet pair probing technique. TOPP is computationally intensive to implement. Jain and Dovrolis[41, 42] proposed a tool called *pathload*. Pathload is to estimate the range of available bandwidth iteratively, not the value of available bandwidth. Since the pathload tries to find the available bandwidth for a network path iteratively based on a binary-search algorithm, it has a rather long convergence time. Ribeiro et al.[43] proposed a tool called *pathChirp*. Although pathChirp needs lighter probing load than for pathload, pathChirp's estimates usually have a negative bias yielding conservative results. Hu and Steenkiste[44] proposed two available bandwidth measurement techniques: the initial gap increasing (IGI) method and the packet transmission rate (PTR) method. Although IGI and PTR yield the estimation results faster than pathload[42], their accuracies degrade when the tight link is different from the bottleneck link.

In order to estimate the available bandwidth for a network path quickly and accurately overcoming the drawbacks of existing schemes, we propose a new available bandwidth estimation mechanism by introducing a simplified path model and using a minimally backlogging concept. We extend the theory developed for a single server in Chapter 3 to available bandwidth estimation for a network path. Basically, we assume that intermediate nodes except edge (i.e. ingress/egress) nodes do not discriminate between probing packets and data packets because discriminating them according to the packet types and treating them differently in scheduling or buffer management can burden intermediate nodes with high implementation costs. In addition, since a first-in first-out (FIFO) policy is used in many routers, we assume no discrimination between probing and data packets in the intermediate nodes. However, ingress/egress nodes should discriminate probing packets from data packets because the ingress node should send a probing packet stream at an adaptive rate and the egress node estimates the available bandwidth by monitoring each probing packet.

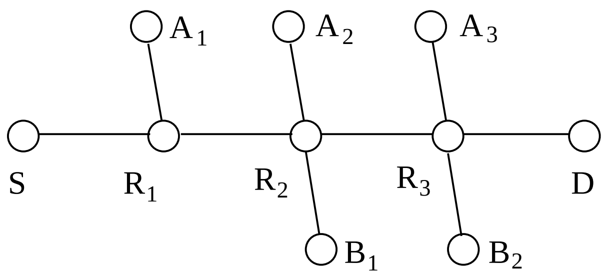
In the proposed mechanism, only a small and fixed number of probing packets are sent at the rate adapted to the condition of the network. Thus, the available bandwidth can be estimated quickly without incurring overload. The performance of the proposed mechanism is evaluated by simulation.

4.2 Network Path Model

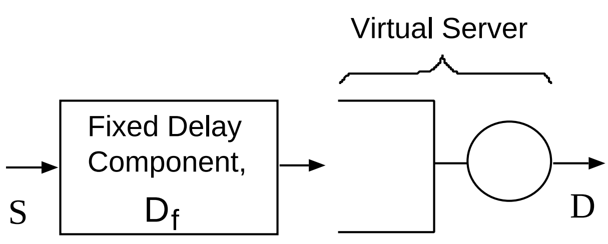
The available bandwidth estimation mechanism for a single server developed in the previous chapter can not be directly applied to available bandwidth estimation for a network path between a specific node pair because a network path usually consists of multiple hops. A tandem queueing system is required to accurately model a network path in communication networks. However, it is difficult to analyze a tandem queueing system. Thus, we introduce a simplified path model for multiple hop paths in order to simplify the estimation problem. We consider a single tight link along a multiple hop path because multiple tight links are not likely to occur frequently in real networks due to variation of the available bandwidth at each link. However, the proposed mechanism can be applied to multiple tight link environments.

Fig. 4.1(a) shows a network topology under consideration. We estimate an end-to-end available bandwidth using probing packets. An end system in Fig. 4.1(a) can be either an end host or an edge router. In the proposed mechanism, available bandwidth is calculated based on the end-to-end delay of each probing packet. Thus, if we consider a specific route $S - R_1 - R_2 - R_3 - D$ in Fig. 4.1(a), Source Node S should record the packet sending time at the *timestamp* field of each probing packet. The available bandwidth for the path can be estimated at Destination Node D, or it can be estimated at the Source Node S if Destination Node D gathers the sending and arrival times of every probing packet and returns the information to the Source Node.

For a path consisting of multiple ($H \geq 1$) hops, each of which represents a combination of a router and the connected outgoing links, let w_h, s_h, and g_h denote the waiting time, the service time, and the propagation delay of a packet at the h-th hop, respectively. Then, the end-to-end delay is $d = \sum_{h=1}^{H}(w_h + s_h + g_h)$. A tight link is assumed to occur at the z-th link. Let d_R denote the summation of every w_h and s_h except those for the z-th link, i.e, $d_R = \sum_{1 \leq h \leq H, h \neq z}(w_h + s_h)$. If we let the expectation of d_R be $\bar{d}_R$ and put $d_R - \bar{d}_R = \tilde{d}_R$, then the end-to-end delay can be



(a) Network topology



(b) Simplified path model

Figure 4.1: An end-to-end network path model

expressed as:

$$d = \sum_{h=1}^{H} g_h + \bar{d}_R + \tilde{d}_R + [w_z + s_z]. \tag{4.1}$$

Since the propagation delay g_h's have fixed values and $\bar{d}_R$ is the expectation of d_R, the values of the first and second terms of (4.1) are constant and their sum is denoted as D_f. $E[\tilde{d}_R] = 0$ and if we neglect the term of $\tilde{d}_R$, the remaining term is the queueing delay of $w_z + s_z$ at the tight link. Then, we can obtain a path model consisting of a fixed delay component (D_f) and a virtual server $\mathcal{S}$ for the tight link as shown in Fig. 4.1.

Suppose that a probing packet p arrives at the path at time a_p and departs from the path at time d_p. Then, the packet p arrives at the virtual server $\mathcal{S}$ at time $a_p^s = a_p + D_f$. When the packet arrives at the destination node, it departs from both the path and the virtual server $\mathcal{S}$. The virtual server is continuously backlogged for $k\,(k \geq 2)$ probing packet transmissions from the j-th probing packet in the interval $[a_j^s, d_{j+k-1}]$ if

$$d_{j+m} \;\geq\; a_{j+m+1}^s, \quad \text{for all } 0 \leq m \leq k-2. \tag{4.2}$$

In addition, the interval $[a_j^s, d_{j+k-1}]$ is called a busy period of probing packets.

4.3 Estimation of End-to-End Available Bandwidth

Sending minimally backlogging probing packets to the virtual server $\mathcal{S}$, we can estimate the available bandwidth for the path. We send a packet train of N probing packets for a path and the time interval of $[a_1, d_N]$ is called a *probing period.* Then, available bandwidth for the path is estimated as follows.

If N probing packets are sent to the virtual server by the minimally backlogging method, then, by Theorem 3.7 of Chapter 3, the available bandwidth for the virtual server in the interval $[a_1^s, d_N]$ can be estimated by $NL/(d_N - a_1^s)$, where $a_1^s = a_1 + D_f$ and D_f is the fixed delay for the current probing period. However, since the inter-probing-packet spacing is fixed during a probing period by the corresponding probing rate in real applications, several busy periods of probing packets may exist during a probing period. Consider the i-th busy period containing k continuously backlogged probing packets[1]. Probing packets arriving during the busy period are indexed from 1 to k. Fig. 4.2(a) illustrates a sample service curve for the busy period showing the amount of probing packets served for $[a_1, t]$. The *Measured Probing Rate* (*MPR*) for the i-th busy period is defined as:

$$MPR(i) = \frac{kL}{d_k - a_1^s} = \frac{kL}{(d_k - a_1) - D_f}.$$

The *MPR* for the longest busy period during a probing period is used to reliably estimate the available bandwidth.

We use the value of D_f calculated in the previous probing period to estimate the value of D_f for the current probing period. If we assume that the service rate for the first packet and the average service rate for the other $k-1$ packets are the same, then we can estimate D_f as

$$\tilde{D}_f = (d_1 - a_1) - \frac{d_k - d_1}{k-1}.$$

D_f is estimated at the longest busy period. However, $\tilde{D}_f$ may not be reliable if it is directly used in the next window. $\tilde{D}_f$ may be so large that $a_p^s = a_p + \tilde{D}_f > d_p$

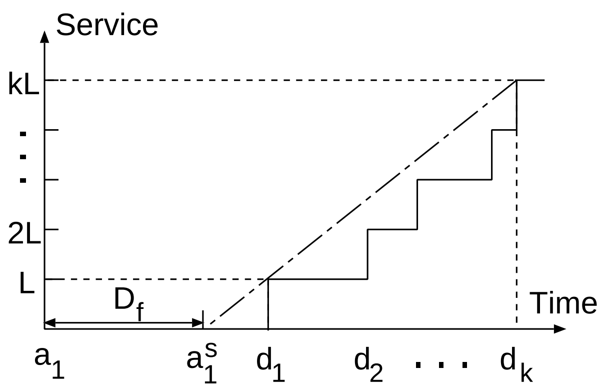
[1] When we check the continuous backlogging condition described in (4.2) for the current probing period, the D_f value calculated in the previous probing period is used.

for some probing packet p. In that case, $\tilde{D}_f$ need to be decreased to satisfy $a_p^s \leq d_p$ for any packet p. This modification is required to make the estimation mechanism reliable.

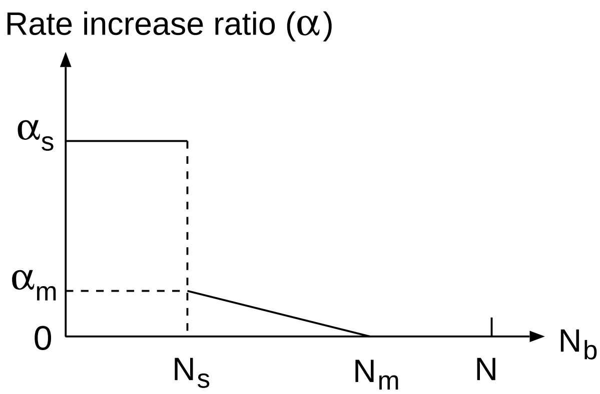
We now consider a probing rate adaptation scheme. Let $N_b = \max_i N_b(i)$, where $N_b(i)$ is the number of probing packets belonging to the i-th busy period in a probing period. If N probing packets are sent to the virtual server according to the minimally backlogging method, then there will be only a single busy period containing N probing packets during a probing period, and thus, $N_b = N$. However, since the inter-packet-spacing is fixed during a probing period, even if probing packets are sent at the rate which is the average rate of minimally backlogging probing packets, N_b may be less than N. We try to maintain N_b within a reasonable range by an adaptive probing scheme. A small value of N_b is due to a lower probing rate than for minimal backlogging and a large value of N_b is due to a higher rate. If N_b is in the reasonable range, we may assume that the minimal backlogging occurs. Thus, *MPR* is a reliable estimate of the available bandwidth. Let $(N_s, N_m]$ be the reasonable range of N_b. Fig. 4.2(b) shows the proposed probing rate adaptation scheme, which is explained as follows:

Case 1: If $N_b > N_m$, then *MPR* is considered to be larger than the available bandwidth (AB) due to a higher probing rate than for minimal backlogging, and the next input rate is set to *MPR*. The AB is estimated by *MPR* since *MPR* quickly approaches to the AB.

To give a reason for the use of *MPR* as the next probing rate, we consider an example. For a First-Come First-Served (FCFS) server with a link rate of C and an



(a) Example of a service curve



(b) Rate increase ratio curve

Figure 4.2: Probing rate adaptation scheme

AB of C_a, if the probing packets arrive at a rate of $r\,(\geq C_a)$, they are served at a rate of

$$m(r) = \frac{r}{C - C_a + r}\,C.$$

If we adjust the $(n{+}1)$-th probing rate by $r_{n+1} = m(r_n)$, then we have that

$$r_{n+1} = \frac{r_n}{C - C_a + r_n}\,C.$$

We can easily know that if $r_1 \geq C_a$, then $r_n \geq C_a$ for $n \geq 2$, and r_n can be expressed as

$$r_n = \frac{1}{\dfrac{1}{C_a} + \left(\dfrac{C - C_a}{C}\right)^{n-1}\left(\dfrac{1}{r_1} - \dfrac{1}{C_a}\right)}.$$

Thus, $\lim_{n\to\infty} r_n = C_a$ if $C_a > 0$, that is, *MPR* converges to the AB.

Case 2: If $N_b \leq N_s$, *MPR* for this short busy period may be inaccurate because the minimally backlogging condition is not satisfied. Thus, the current AB is estimated by the AB at the last probing period for $N_b > N_s$. If $N_b \leq N_s$ consecutively i times since the last probing period with $N_b > N_s$, then the next input rate is set to $\mathrm{AB} \cdot (1+\alpha_s)^i$. α_s determines the tracking speed of the proposed algorithm when the probing rate is lower than the AB. When the current probing rate is lower than the AB, if α_s is large, then *MPR* quickly approaches to the AB, but large values of α_s may cause temporary ripples.

Case 3: If $N_s < N_b \leq N_m$, then *MPR* is a reliable estimate of the AB. However, it is necessary to maintain the probing rate slightly higher than AB in order to obtain a reliable value of *MPR*. Thus, the next input rate is increased to $MPR \cdot (1+\alpha(N_b))$, where $\alpha(N_b) = \alpha_m(N_m - N_b)/(N_m - N_s)$, and α_m is the maximum rate increase ratio in the medium busy period range. If the value of *MPR* is close to that of AB, then the next probing rate is higher than AB by a ratio of $\alpha(N_b)$. For a given value of N_b, if α_m or N_m increases, $\alpha(N_b)$ also increases.

As explained above, the proposed probing scheme attempts to send probing packets at a slightly higher rate than the AB. Thus, the load offered to the tight link may slightly exceed one during a probing period. In order to prevent degradation of the throughput of data traffic at the tight link due to overload, consecutive probing periods are separated by at least one probing period length of $d_N - a_1$. Then, the average load offered by the probing traffic is approximately equal to or lower than half of the AB in a longer time interval than the duration of one probing period, and thus, the tight link is not overloaded in a long time scale.

4.4 Numerical Results

In this section, we compare the performance of the proposed available bandwidth estimation mechanism for a multiple hop path with that of pathload [42] through OPNET simulation. A multiple hop topology is illustrated in Fig. 4.1(a). Each node is modeled as an output queued router with a FIFO queue. We estimate the available bandwidth for the path $S - R_1 - R_2 - R_3 - D$. Every link except $R_2 - R_3$ has a link rate of 20 Mbps and a propagation delay of 5 ms. Link $R_2 - R_3$ with a link rate of 10 Mbps is the bottleneck link. The sizes of both probing packets and data packets are 4000 bits. For the proposed mechanism, the number of probing packets sent in one probing period (N) is 100. The values of the rate adaptation related parameters are set to $N_m = 0.95 \times N = 95$, $N_s = 0.30 \times N = 30$, $\alpha_m = 0.10$, and $\alpha_s = 1.0$. For the pathload [42], the user-specified resolution of available bandwidth ω is set to 0.2 Mbps and the grey-region resolution χ is 0.3 Mbps.

Two types of traffic patterns are used for non-probing packet sequence: constant bit rate (CBR) and self-similar traffic generated using a multi-fractal model [77]. The Hurst parameter is 0.8 and the sigma/mean ratio of a flow is approximately 0.5. The mean rate of each flow is 4 Mbps except a flow which is sent from A_2 to B_2 and has a rate of 2 Mbps. During a simulation time of 200 seconds, 4 flows with a lifetime of 70 seconds are sent on route $A_1 - R_1 - R_2 - B_1$ sequentially at an interval of 10 seconds from time 0. 4 flows with a lifetime of 70 seconds are sent on route $A_3 - R_3 - D$ sequentially at an interval of 10 seconds from time 100. Thus, link $R_1 - R_2$ is a tight link in the interval $[20, 80]$. Link $R_2 - R_3$ is a tight link in the intervals of $[0, 30]$, $[70, 130]$, and $[170, 200]$. Link $R_3 - D$ is a tight link in the interval $[120, 180]$. Thus, two tight links exist in the intervals of $[20, 30]$, $[70, 80]$, $[120, 130]$, and $[170, 180]$.

Fig. 4.3 compares the available bandwidth of the proposed mechanism with that of the pathload under a CBR traffic load. The pathload iteratively estimates the range $[R^{min}, R^{max}]$ of the available bandwidth. The trace of $(R^{min} + R^{max})/2$ is plotted in Fig. 4.3 and the range of $[R^{min}, R^{max}]$ is also shown at the instant of termination. The pathload is restarted just after it terminates. We can observe that it takes about 8 seconds for the pathload to terminate. The pathload sometimes yields a significant error in the estimation of the available bandwidth, especially at time 153.8 as shown in Fig. 4.3. However, the proposed mechanism closely tracks the available bandwidth even if the available bandwidth changes abruptly, there exist two tight links or the tight link is different from the bottleneck link. The error observed in the intervals of $[30, 70]$ and $[130, 170]$ is due to the fact that the proposed probing scheme tries to maintain the probing rate slightly higher than the available bandwidth to obtain a reliable value of *MPR*. If N_m or α_m is decreased, this error can also be decreased, but more ripples may occur due to unreliable values of *MPR*.

The reason why the proposed scheme can estimate the available bandwidth faster than the pathload can be explained as follows. The pathload changes the probing rate using a binary search to find the available bandwidth, while our scheme directly tries to find the available bandwidth based on the previous estimation value of the available bandwidth and the observed value of N_b. Thus, the pathload yields a reliable range of available bandwidth only at the instant of termination after several iterations of binary search. However, our scheme estimates the available bandwidth quickly and tracks it adaptively without restart of iteration and yields the value of estimated available bandwidth every probing period. Thus, it takes longer time for the pathload to converge than for the proposed scheme.

Fig. 4.4 compares the mean (μ) and the standard deviation (σ) of the available bandwidth estimated by the proposed mechanism with those of the measured available bandwidth under a self-similar traffic load. The range of $[\mu-\sigma, \mu+\sigma]$ is plotted based on the measurement at an interval of 10 seconds. We can observe that the mean of the measured available bandwidth lies within σ from μ of the estimated available bandwidth for every estimation time.

Fig. 4.5 compares the available bandwidth of the proposed mechanism with that of the pathload under the same traffic trace as Fig. 4.4. The curve for the measured AB is the value of measured available bandwidth averaged for every 5 seconds. The

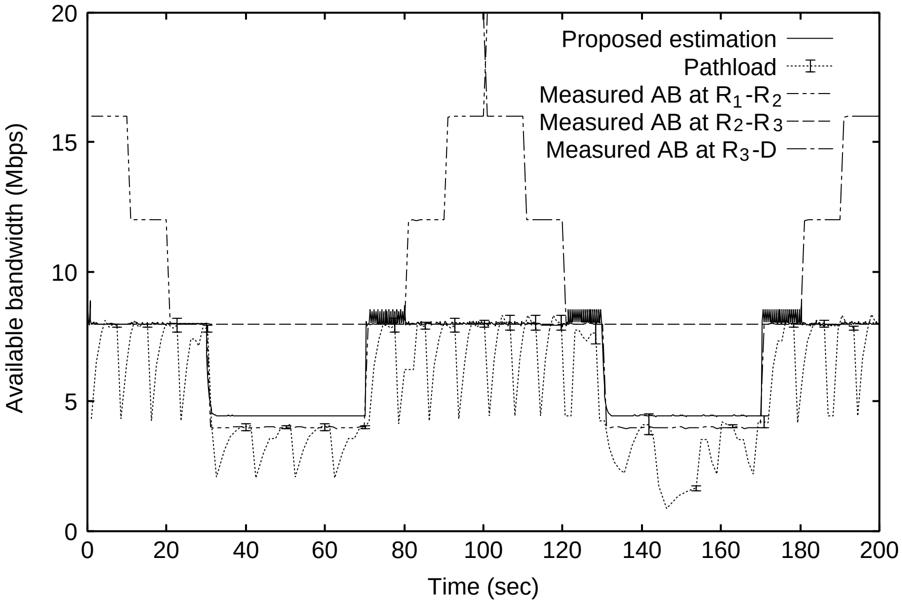


Figure 4.3: Comparison of available bandwidth estimated by the proposed mechanism and by the pathload under a CBR traffic load

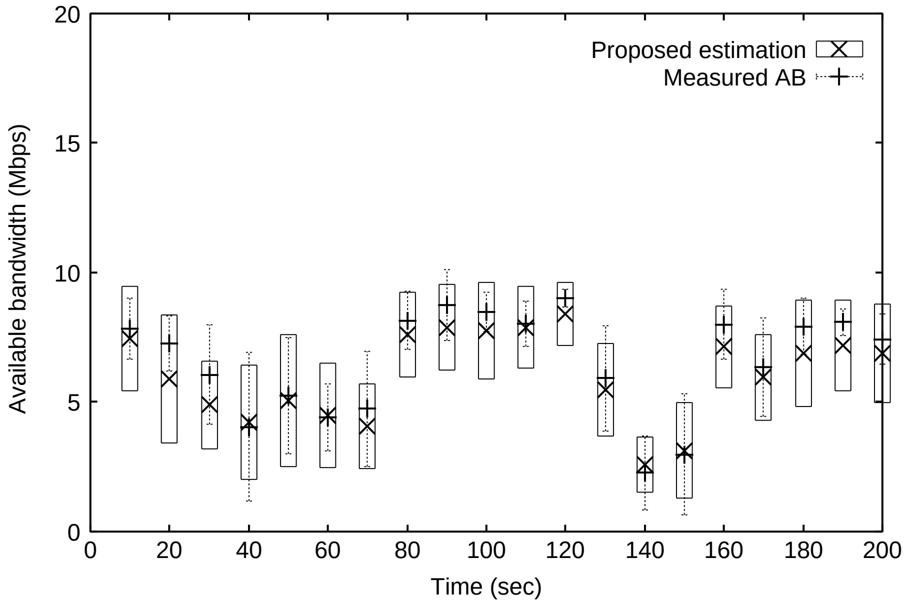


Figure 4.4: Available bandwidth estimated by the proposed mechanism under a self-similar traffic load

curve for the proposed estimation mechanism is the same as that in Fig. 4.4. We can observe many problems of pathload from the curve for the pathload. First, the convergence time increases for bursty traffic. We can observe that the average convergence time is longer than 10 seconds in this case. Second, pathload frequently fail to give a converged range of available bandwidth when the traffic load changes dynamically. Third, even the estimation range of the available bandwidth sometimes deviates from the average value of the measured available bandwidth, especially at time 51, 79, 147, and 193 seconds. Thus, it is difficult to obtain a reasonable range of available bandwidth for a short period of 10 seconds by the pathload because of a long convergence time. On the other hand, the proposed mechanism gives a reasonable range of the available bandwidth even when the traffic load significantly changes.

Thus far, consecutive probing periods are separated by about the duration of the previous probing period. In this case, probing traffic uses approximately half of the available bandwidth. We now consider the case that only a packet train of N packets are sent once per time window of 1 second. N is fixed to 100 again. Since the probing packet size is 4000 bits, the long-term average rate of probing traffic is limited to 400 kbps in this case. Fig. 4.6 compares the mean (μ) and the standard deviation (σ) of the available bandwidth estimated by the proposed mechanism with those of the measured available bandwidth under a self-similar traffic load. The accuracy is rather lower than the case of Fig. 4.4, since the mean of the measured available bandwidth sometimes lies out of σ from μ of the estimated available bandwidth. However, the performance is not worse than pathload since the deviation is not

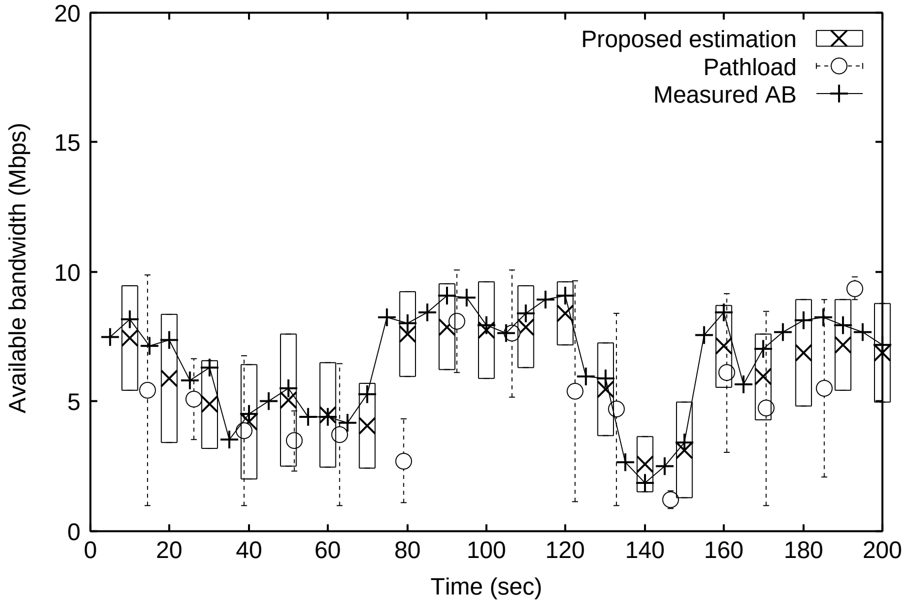


Figure 4.5: Comparison of available bandwidth estimated by the proposed mechanism and by the pathload under a self-similar traffic load

significant as pathload even if the mean of the measured available bandwidth lies out of σ from μ of the estimated available bandwidth. If the available bandwidth is 4 Mbps, then the probing traffic uses only about 10 % of the available bandwidth. As the available bandwidth increases, the relative portion of the probing traffic decreases. Although the probing traffic uses the total available bandwidth during probing time, the duration of one probing period is not long since the number of probing packets sent per probing period is limited to a rather small number of 100. Since the proposed probing scheme can operate at a rather low average probing rate compared with the available bandwidth, the proposed probing scheme can be used non-intrusively.

4.5 Summary

A new available bandwidth estimation mechanism for a network path is proposed by introducing a simplified path model and extending the approach for a single server. Since it is not possible to send probing packets by minimally backlogging method for a network path, we proposed a new probing scheme to maintain the minimally backlogging condition approximately. In a multiple hop topology, it is observed that the proposed available bandwidth mechanism tracks the available bandwidth rather accurately even when the available bandwidth changes abruptly. Thus, the proposed mechanism can be used to obtain a reasonable range of dynamic available bandwidth for a network path in a short time interval. Since the proposed probing scheme can operate at a much lower rate than the available bandwidth, the proposed probing scheme can be used non-intrusively.

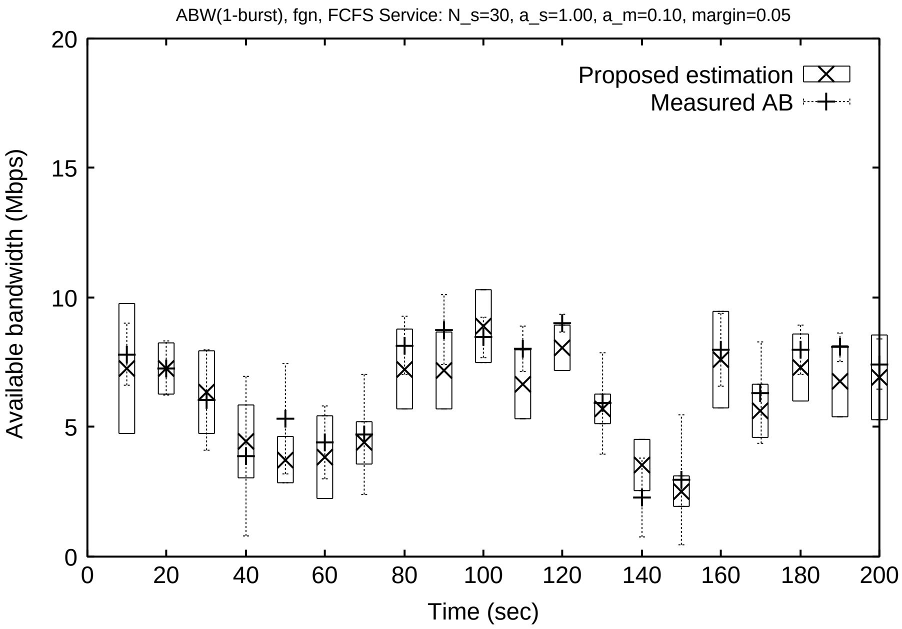


Figure 4.6: Comparison of the estimated available bandwidth and measured available bandwidth under a self-similar traffic load

5. Measurement-Based Admission Control

5.1 Introduction

Although the capacity of core networks has increased tremendously due to high-speed optical transmission techniques, current IP networks can not guarantee strict or statistical quality-of-service (QoS) requirements for real-time traffic flows. Thus, it is very important to allocate and manage resources for multimedia traffic flows with real-time performance requirements in order to guarantee QoS. Specifically, admission control is of interest in this chapter. There are two important goals in conventional admission control algorithms. The first one is to guarantee the QoS contracted for real-time flows, and the other one is to achieve high network utilization. Conventional parameter-based admission control algorithms use the worst case bounds derived from the parameters describing the flow. These algorithms typically result in low utilization under the load of bursty input traffic [68]. However, *measurement-based* admission control algorithms (MBACs) can achieve much higher network utilization than parameter-based algorithms, while providing somewhat relaxed QoS [54]. Since it is difficult to predict future behavior accurately with traffic measurements, MBAC may result in occasional violation of the contracted QoS. Several measurement-based admission control algorithms have been proposed [54, 58–66]. However, it is reported that the conventional admission control algorithms can not meet their strict QoS target in terms of loss ratio [68].

We consider delay as a QoS target because real-time flows are more sensitive to delay than loss. Furthermore, packets from real-time flows are rarely dropped if the packets from real-time flows are treated with higher priority than those from best-effort traffic at each network node. Thus, we assume that the packets from the flow admitted by the admission control algorithm are given higher priority than those from the best-effort flow which is initiated regardless of admission control.

In order to overcome the problem of the conventional admission control schemes addressed above, we develop a scalable architecture and an admission control algorithm for real-time flows. Since individual management of each traffic flow on each of its traversed routers can cause a fundamental scalability problem in both data and control planes, we consider that each flow is classified at the ingress router and data traffic is aggregated according to its class inside the core network in the proposed resource management architecture as shown in a DiffServ framework. In the proposed approach, admission decision is made for each flow at edge (ingress or egress) routers. However, it can be scalable because the algorithm can be simply implemented using a single comparison logic. In the proposed admission algorithm, each ingress router manages admissible bandwidth for each possible egress router. The admissible bandwidth is calculated considering delay QoS based on the available bandwidth which is estimated by the egress router through monitoring probing packets. The performance of the proposed algorithm is evaluated using a set of simulation experiments for bursty traffic flows.

5.2 System Architecture

Consider an autonomous system as depicted in Fig. 5.1. Routers A, E, F, G, and I are edge routers, and B, C, D, and H are core routers. Routers which provide interface to access networks are edge routers, and core routers do not operate as an interface. In the proposed architectural solution, an ingress router manages admissible bandwidth for each path to each egress router. For example, Edge Router A manages admissible bandwidths for Egress Nodes E, F, G, and I, individually. Traffic arrivals at ingress routers in DiffServ domain are differentiated by the given QoS requirements. All arriving traffic with the same QoS requirements is treated as an aggregate class.

Admissible bandwidth is managed according to the classified classes. Admissible bandwidth between a specific ingress/egress node pair is defined considering the level of services that can be provided. In this chapter, we consider only delay bound violation probability as a QoS requirement. Let R_j^* denote the admissible bandwidth for the j-th class between Ingress Router A and Egress Router E. Let d_j and ε_j be the delay bound and the threshold for the delay violation probability, respectively. $D_j(0)$ is a random variable representing the current end-to-end delay, and $D_j(R)$ is a random variable representing the end-to-end delay which the total traffic of class j experiences after admitting a flow with a rate of R. Then, the admissible bandwidth R_j^* is defined as:

$$R_j^* = \max\{R : P(D_j(R) > d_j) \ \leq \ \varepsilon\}. \tag{5.1}$$

Thus, R_j^* is the maximum available bandwidth that can be supported additionally satisfying the delay constraint.

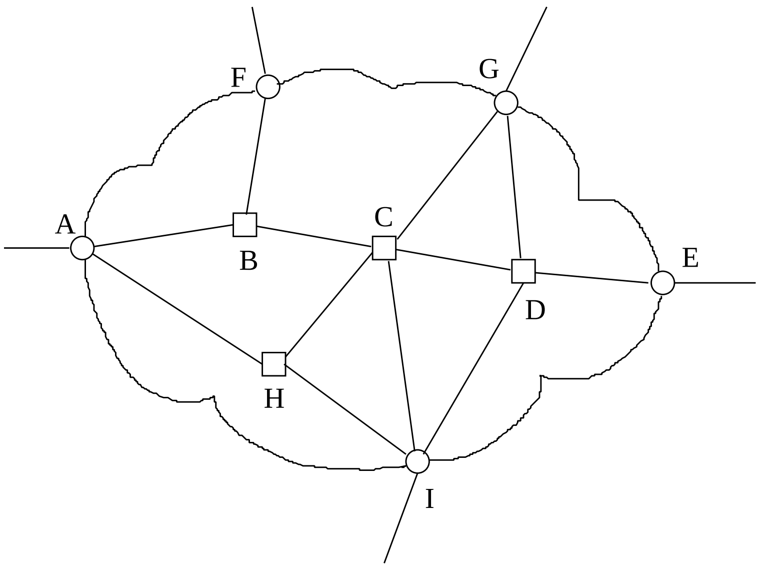


Figure 5.1: Reference network model

In order to support QoS for a new flow while guaranteeing the contracted QoS for the existing flows, a negotiation is needed between a network and a new end-point application. The network determines whether to admit the new flow or not according to an admission control policy/algorithm assuming that the user complies with the contract. The characteristics of the new flow should be included in the contract because the network can not determine whether the required QoS will be satisfied or not if it does not know how much traffic will be offered by the new flow. Thus, we assume that the contract is made just based on the peak rate r_p of a flow. Peak rate r_p is the only significant traffic parameter in our admission algorithm, and we assume that each flow is policed so that the instantaneous traffic rate is kept less than or equal to the peak rate r_p.

If a new flow request with a peak rate of r_p, which is destined to Router E, arrives at Edge Router A, then Router A can accept the flow as the j-th class if the following condition is satisfied:

$$r_p < R_j^*. \tag{5.2}$$

Then, the delay constraint can be satisfied for both the existing and the new traffic. Since the proposed admission control algorithm is rather simple and ingress router determines whether it accepts the new flow or not, admission control can be performed very quickly for real-time flows.

In this scheme, ingress routers do not need to calculate admissible bandwidth whenever a new flow arrives. In order to calculate admissible bandwidth R_j^* between Nodes A and E, we need to know the status of the network. An edge router sends probing packets to every possible egress router to monitor the current condition of the path to each egress router. If the round trip time between Nodes A and D is $RTT(A, D)$, then R_j^* can be calculated once every $RTT(A, D)$. If the calculation time of admissible bandwidth, T_{cal} is longer than the round trip time, then R_j^* can be calculated once every $\max\{RTT(A, D), T_{cal}\}$.

We use the combination of fixed delay component (D_f) and a virtual single server $\mathcal{S}$ proposed in Section 4.2 as an end-to-end network path model in this chapter again. Suppose that a packet p arrives at the path at time a_p and departs from the path at time d_p. Then, the packet p arrives at the virtual server $\mathcal{S}$ at time $a_p^s = a_p + D_f$. When the packet arrives at the destination node, it departs from both the path and the server model $\mathcal{S}$. The single server is continuously backlogged for k packet transmissions from the j-th packet in the interval $[a_j^s, d_{j+k-1}]$ if

$$d_{j+m} \;\geq\; a_{j+m+1}^s, \quad \text{for all } 0 \leq m \leq k-2, \tag{5.3}$$

for $k \geq 2$.

5.3 Admission Control Scheme

As described in the previous Sections, calculation of the admissible bandwidth considering delay QoS is a crucial part of the proposed admission control scheme. If the calculated value is larger than the real capacity, then delay QoS may not be guaranteed. On the other hand, if the calculated value is smaller than the real capacity, then the utilization of the network resource decreases. In this section, we investigate how to evaluate the admissible bandwidth considering delay QoS.

We assume that there are only two classes of flows in a core network. The first is the premium class in which all flows follow their peak rate constraints and have a delay QoS requirement. This is the only class that is subject to admission control. The second is the best-effort class. Intermediate routers are assumed to give a strict priority to the premium class in managing two classes so that the delay of the premium class traffic is not affected by the best-effort traffic.

We will derive a relation which predicts a new delay distribution in case of accepting a new flow of rate R in order to calculate the admissible bandwidth between a specific ingress/egress router pair according to (5.1). In order to solve this problem analytically, we consider a fluid model on a continuous time scale. As noted in Section 5.2, we structurally model a network path from a specific ingress router to an egress router as a simple path consisting of a fixed delay component (D_f) and a virtual server $\mathcal{S}$. If we let D_e and D denote the end-to-end delay of a packet and the delay that the packet experiences at the virtual server, respectively, then we can obtain the following relation:

$$D_e = D_f + D.$$

Since D_f is fixed, we focus on the variable component of D. Let $X_{u,v}$ denote the amount of traffic arriving at the virtual server $\mathcal{S}$ in time interval $[u, v]$, and denote $Y_{u,v}$ as the amount of traffic served by the virtual server $\mathcal{S}$ in time interval $[u, v]$. If we let a'_j, d'_j, and l'_j denote the arrival time, departure time, and size in bits of the j-th packet, respectively, then

$$X_{u,v} = \sum_{a'_j \in [u,v]} l'_j\,, \qquad Y_{u,v} = \sum_{d'_j \in [u,v]} l'_j.$$

Letting Q_t denote the backlog of traffic in the virtual server $\mathcal{S}$ at time t, Q_t can be expressed as [70]:

$$Q_t = \sup_{s \leq t}\{X_{s,\,t} - Y_{s,\,t}\}. \tag{5.4}$$

An interval $[u,\, v]$ is referred to as the *backlogged interval* of the virtual server $\mathcal{S}$ if $Q_t > 0$, for $u \leq t \leq v$. Let D_t denote the virtual delay experienced by a bit arriving at $\mathcal{S}$ at time t. If the virtual server $\mathcal{S}$ is empty at time 0, then D_t is expressed as [79]:

$$D_t = \min\left\{\Delta : \Delta \geq 0 \text{ and } X_{0,\,t} \leq Y_{0,\,t+\Delta}\right\}. \tag{5.5}$$

Let $X^e_{u,v}$ and $X^n_{u,v}$ be the amount of traffic arriving at $\mathcal{S}$ from existing flows in time interval $[u, v]$ and that of arriving traffic from the new flows in time interval $[u, v]$, respectively. The aggregate arriving traffic $X_{u,v}$ consists of $X^e_{u,v}$ and $X^n_{u,v}$. Since our objective is to evaluate the maximum admissible capacity R^*_j of (5.1), we consider only constant rate flow as $X^n_{u,v}$. Let $Y^e_{0,t}$ and $Y^n_{0,t}$ be the amounts of existing traffic and new traffic served by $\mathcal{S}$ during time interval $[0, t]$ in case of accepting new flows, respectively. We assume that

- Aggregate traffic is served according to the first-come-first-service (FCFS) policy in the same class.
- $X^n_{\tau,\,t} = R(t - \tau)$, where τ is the starting time of a new flow X^n and R is a constant denoting the rate of the new flow.
- $Y_{0,\,t}$ is continuous.
- $Y^n_{0,\,t}$ is continuous and it is increasing after the starting time of the new flow.

The third assumption that $Y_{0,\,t}$ is continuous is natural because if the output link rate of egress node is R_o, the maximum amount of traffic that can be served for the interval of length Δt can not exceed $R_o \Delta t$.

Let d_0 be the delay bound which should be guaranteed for the class subject to admission control. Let λ and C denote the average arrival rate of existing flows and the service rate of the virtual server, respectively. If we assume that $R + \lambda < C$ for the stability of the system and $X_{0,t}$ is stationary and ergodic, then D_t has a limiting distribution and D^* is defined as

$$D^* = \lim_{t \to \infty} D_t.$$

If we estimate the delay of a packet arriving at $\mathcal{S}$, D, by D^*, then the delay bound violation probability $P(D_e > d_0)$ can be expressed as:

$$\begin{aligned} P(D_e > d_0) &= P(D + D_f > d_0) \\ &= P(D^* > d_0'), \end{aligned} \tag{5.6}$$

where $d_0' = d_0 - D_f$. We need to investigate the virtual delay D_t in more detail in order to obtain information about the delay bound violation probability. Under the above assumptions regarding a fluid model, we can obtain the following relation:

Proposition 5.1 *Suppose that a new flow starts at time τ, $\tau \geq 0$. If we define $D_t^n = \min \left\{ \Delta : \Delta \geq 0,\ X_{0,t}^n \leq Y_{0,t+\Delta}^n \right\}$, then we have*

$$D_t^n = D_t,\ t > \tau.$$

Proof) Suppose that $t > \tau$. Since the service policy is FCFS, any traffic which arrives at $\mathcal{S}$ after time t cannot be served before time $t + D_t$. Thus, the served traffic until time $t + D_t$ consists of the traffic from existing flow during $[0,\ t]$ and the traffic from the new flow during $[0,\ t]$. $Y_{0,\,t+D_t} = X_{0,\,t}$ since $Y_{0,\,t}$ is continuous, and this implies that

$$Y^e_{0,\,t+D_t} = X^e_{0,\,t} \quad \text{and} \quad Y^n_{0,\,t+D_t} = X^n_{0,\,t}. \tag{5.7}$$

Thus, $D^n_t \le D_t$. To prove that $D_t \le D^n_t$, choose a positive real number δ arbitrarily. Since $Y^n_{0,\,t}$ is increasing for $t \ge \tau$, we have that $Y^n_{t+D^n_t,t+D^n_t+\delta} > 0$ and this served traffic belongs to the traffic which arrives after time t. From the fact that the service policy is FCFS, it is clear that $D_t \le D^n_t + \delta$. Since δ is arbitrary, we have $D_t \le D^n_t$.

□

By Proposition 5.1, D_t can be evaluated if $Y^n_{0,t}$ can be obtained. However, $Y^n_{0,t}$ can be measured when a new flow is really offered, but we want to estimate D_t before the new flow is offered into the network. Thus, we introduce the available service $\tilde{Y}^n_{0,t}$, which can be estimated by the probing scheme developed in Chapter 4. We consider a virtual backlogging process $\tilde{Q}^n_t$ which is defined as

$$\tilde{Q}^n_t = \sup_{s \le t} \{X^n_{s,\,t} - \tilde{Y}^n_{s,\,t}\}. \tag{5.8}$$

$\tilde{Q}^n_t$ denotes the amount of backlogging traffic in the queueing system for which the arrival process is $X^n_{s,\,t}$ and the service process is $\tilde{Y}^n_{s,\,t}$. We define $b(t)$ as

$$b(t) = \sup\{s : s \le t, \tilde{Q}^n_s \le 0\}. \tag{5.9}$$

$b(t)$ is the start time of the current busy period if $\tilde{Q}^n_t > 0$ and is the current time if $\tilde{Q}^n_t = 0$. Then, we can define a virtual delay $\tilde{D}^n_t$ as

$$\tilde{D}^n_t = \min\{\Delta : \Delta \ge 0,\ X^n_{b(t),\,t} \le \tilde{Y}^n_{b(t),\,t+\Delta}\}. \tag{5.10}$$

$\tilde{D}^n_t$ is defined in a different way from the definition of D_t, (5.5) or that of D^n_t because $\tilde{Y}^n_{s,\,t}$ can be larger than $X^n_{s,\,t}$ when the amount of arriving traffic $X^n_{s,\,t}$ is rather small. However, since (5.10) is a generalized description of virtual delay, D_t and D^n_t can be defined in a similar way. We can obtain the following relation between D^n_t and $\tilde{D}^n_t$.

Proposition 5.2 *Suppose that traffic from a new flow is offered into a queueing system from time τ and probing is started at the same time τ, then we have*

$$D^n_t \leq \tilde{D}^n_t, \qquad \text{for } t \geq \tau.$$

Proof) We first show that the above relation holds at a packet level for an arbitrary size of probing packets. We assume that the minimally backlogging traffic at a fluid level can be realized by decreasing the size of a probing packet infinitely small. Then, proving the above relation at a packet level for an arbitrary packet size implies the relation is also valid at the fluid level.

We consider busy periods for the traffic of the new flow. For simple notation, a packet belonging to the new flow is called an n-packet. A busy period is an interval that begins when an arriving n-packet finds no n-packet in the queueing system and ends when a departing n-packet sees no n-packet in the system for the first time after the beginning time. Idle periods are intervals between successive busy periods. Then, busy and idle periods alternate.

Let β_i and ζ_i be the durations of the i-th busy period and the i-th idle period, respectively. The i-th busy period and the i-th idle period constitutes the i-th cycle. If we let U_i denote the duration of the i-th cycle, then $U_i = \beta_i + \zeta_i$. If we let

$$\tau_1 = \tau,$$
$$\tau_n = \tau + \sum_{i=1}^{n-1} U_i, \quad n \geq 2,$$

then τ_n is the time when the n-th cycle begins. We will show that $D_t^n \le \tilde{D}_t^n$ for an arbitrary cycle j. We assume that $\tau_j = 0$ without loss of generality. The j-th busy period β_j can be expressed as

$$\beta_j = \min\{s : s \ge 0, X_{0,s}^n \le Y_{0,s}^n\}. \tag{5.11}$$

We first show that $Y_{0,t}^n \ge \tilde{Y}_{0,t}^n$ for $t \in [0, \beta_j]$. At time $t = 0$, the arriving n-packet observes only packets from existing flows or no packet. Let X_0 denote the amount of packets in bits in the system at time 0 except the arriving n-packet. We assume that the packet size of each n-packet is L_p, and it is the same as the size of each probing packet. We estimate $\tilde{Y}_{0,t}^n$ by the amount of served probing traffic during the time interval $[0, t]$ when probing packets are sent to the queueing system with an initial backlog of X_0 according to the minimally backlogging method from time 0. Let m denote the number of n-packets arriving during the j-th busy period.

a_i and $\tilde{a}_i\,(i = 1, 2, \cdots, m)$ denote the arrival times of the i-th n-packet and the i-th probing packet, respectively. Since the j-th busy period starts at time 0, $a_1 = 0$. Due to the assumption that the probing starts at time 0, $\tilde{a}_1 = 0$. d_i and $\tilde{d}_i\,(i = 1, 2, \cdots, m)$ denote the departure times of the i-th n-packet and the i-th probing packet, respectively.

Since the i-th probing packet can depart when the service amount reaches to the amount of traffic arriving until the arrival time of the i-th probing packet, $\tilde{a}_i$, we can obtain the following relation:

$$\tilde{X}_{[0,\tilde{a}_i]}^n + X_0 + X_{(0,\tilde{a}_i]}^e = C\tilde{d}_i, \quad i \ge 1,$$

where C is the service rate of the virtual server. Since $\tilde{a}_{i+1} = \tilde{d}_i$ when probing is done in a minimally backlogging manner and $\tilde{X}^n_{[0,\tilde{a}_i]} = iL_p$, $\tilde{a}_{i+1}$ can be expressed as

$$\tilde{a}_{i+1} = \tilde{d}_i = \frac{X_0 + iL_p + X^e_{(0,\tilde{a}_i]}}{C}, \quad i \geq 1. \tag{5.12}$$

In a similar way, we can obtain the following relation for the n-traffic:

$$X^n_{[0,a_i]} + X_0 + X^e_{(0,a_i]} = Cd_i, \quad i \geq 1.$$

Since $X^n_{[0,a_i]} = iL_p$, d_i can be expressed as

$$d_i = \frac{X_0 + iL_p + X^e_{(0,a_i]}}{C}, \quad i \geq 1. \tag{5.13}$$

We now show that $a_i \leq \tilde{a}_i,\ i \geq 1$ by induction. For $i = 1$, $a_i = \tilde{a}_i = 0$.

We assume that $a_i \leq \tilde{a}_i$ for $i = 1, \cdots, k\,(k \leq m-1)$. Since $a_k \leq \tilde{a}_k$, $X^e_{(0,a_k]} \leq X^e_{(0,\tilde{a}_k]}$. Thus, $\tilde{a}_{k+1} \geq d_k$ by (5.12) and (5.13). Since $k+1 \leq m$, the busy period should continue until the arrival time of the $(k+1)$-th n-packet. Thus, a_{k+1} should be less than or equal to d_k and we have that $\tilde{a}_{k+1} \geq a_{k+1}$.

Since $a_i \leq \tilde{a}_i$ $(1 \leq i \leq m)$, $X^e_{(0,a_i]} \leq X^e_{(0,\tilde{a}_i]}$ and the following relation can be obtained from (5.12) and (5.13):

$$\tilde{d}_i \geq d_i, \quad i \geq 1. \tag{5.14}$$

Since we assume that the sizes of all n-packets and probing packets are fixed to L_p, $Y^n_{0,t}$ and $\tilde{Y}^n_{0,t}$ can be expressed as

$$Y^n_{0,t} = L_p \sup\{i : d_i \leq t\},$$

$$\tilde{Y}^n_{0,t} = L_p \sup\{i : \tilde{d}_i \leq t\}.$$

From (5.14) and the definitions of $Y^n_{0,t}$ and $\tilde{Y}^n_{0,t}$, we can obtain that

$$\tilde{Y}^n_{0,t} \leq Y^n_{0,t}, \quad 0 \leq t \leq \beta_j. \tag{5.15}$$

Since we assume that the first probing packet of the j-th busy period is sent at time 0, there is no backlogging of probing traffic at time 0-, and thus, $\tilde{Q}^n_{0-} = 0$. (5.11) implies that $X^n_{0,t} > Y^n_{0,t}$ for $0 \le t < \beta_j$. Thus, $X^n_{0,t} > \tilde{Y}^n_{0,t}$ for $0 \le t < \beta_j$ by (5.15) and $b(t) = 0,\ 0 \le t < \beta_j$ by (5.8) and (5.9).

We now show that $D^n_t \le \tilde{D}^n_t$ for $0 \le t \le U_j$ by partitioning an interval $[0, U_j]$ for the j-th cycle into $[0, \beta_j)$ and $[\beta_j, U_j]$.

Case 1: We consider the case of $t < \beta_j$. By (5.15), $\tilde{Y}^n_{0,t} \le Y^n_{0,t}$ for $t < \beta_j$ and we can obtain that

$$\begin{aligned} D^n_t &= \min\{\Delta : \Delta \ge 0,\ X^n_{0,\,t} \le Y^n_{0,\,t+\Delta}\} \\ &\le \min\{\Delta : \Delta \ge 0,\ X^n_{0,\,t} \le \tilde{Y}^n_{0,\,t+\Delta}\} = \tilde{D}^n_t, \qquad \text{for } t < \beta_j. \end{aligned} \tag{5.16}$$

Case 2: We consider the case of $\beta_j \le t \le U_j$. Since $Y^n_{0,t} = X^n_{0,t}$ for $t \ge \beta_j$, $D^n_t = \min\{\Delta : \Delta \ge 0,\ X^n_{0,\,t} \le Y^n_{0,\,t+\Delta}\} = 0$ for $t \ge \beta_j$. Thus, we have

$$\tilde{D}^n_t = \min\{\Delta : \Delta \ge 0,\ X^n_{0,\,t} \le \tilde{Y}^n_{0,\,t+\Delta}\} \ge 0 = D^n_t, \quad \text{for } t \ge \beta_j. \tag{5.17}$$

By (5.16) and (5.17), $\tilde{D}^n_t \le D^n_t$ for $t \in [0, U_j]$. □

By Propositions 5.1 and 5.2, we can use $\tilde{D}^n_t$ as an upper bound of D_t. From (5.10), we know that $\tilde{D}^n_t$ is expressed in terms of $X^n_{s,t}$, $\tilde{Y}^n_{s,t}$, and $b(t)$. Though the shape of $X^n_{s,t}$ is fixed to a constant-rate flow and $\tilde{Y}^n_{s,t}$ can be estimated by a probing scheme, whose detailed explanation will be given later in this section, $b(t)$ is difficult to manipulate. Thus, we derive a relation about $\tilde{D}^n_t$ which excludes the use of $b(t)$ and is easier to manipulate.

Theorem 5.3 *Suppose that a new flow starts at time τ, $\tau \ge 0$. Then, we have*

$$P(\tilde{D}^n_t > d'_0) \le P(\tilde{Q}^n_t > \tilde{Y}^n_{t,t+d'_0}), \quad t \ge \tau.$$

Proof) We first show that the relation is valid when $\tilde{Q}_t^n = 0$. If $\tilde{Q}_t^n = 0$, then $b(t) = t$ by the definition of $b(t)$, (5.9), and $\sup_{s \le t}\{X_{s,t}^n - Y_{s,t}^n\} = 0$, i.e., $X_{s,t}^n \le Y_{s,t}^n$ for $s \le t$. Since $b(t) = t$ and $X_{s,t}^n \le Y_{s,t}^n$ for $s \le t$, $X_{b(t),t}^n \le Y_{b(t),t}^n$ and $\tilde{D}_t^n = 0$ by the definition of $\tilde{D}_t^n$, (5.10). Since $P(\tilde{D}_t^n > d_0') = 0$, the the proof is done when $\tilde{Q}_t^n = 0$.

We now consider the case that $\tilde{Q}_t^n > 0$. Since the condition that $\min\{\Delta : \Delta \ge 0,\ X_{b(t),\,t}^n \le \tilde{Y}_{b(t),\,t+\Delta}^n\} > d_0'$ is equivalent to the condition that $X_{b(t),\,t}^n > \tilde{Y}_{b(t),\,t+d_0'}^n$, we have that

$$\begin{aligned} P(\tilde{D}_t^n > d_0') &= P(\min\{\Delta : \Delta \ge 0,\ X_{b(t),t}^n \le \tilde{Y}_{b(t),t+\Delta}^n\} > d_0') \\ &= P(X_{b(t),t}^n > \tilde{Y}_{b(t),t+d_0'}^n). \end{aligned}$$

Since $b(t) \le t$ by the definition of $b(t)$, if $X_{b(t),\,t}^n > \tilde{Y}_{b(t),\,t+d_0'}^n$, then the following relation holds:

$$\sup_{s \le t}\{X_{s,t}^n - \tilde{Y}_{s,t+d_0'}^n\} > 0.$$

Thus, the following inequality is obtained:

$$\begin{aligned} P(\tilde{D}_t^n > d_0') &= P(X_{b(t),t}^n > \tilde{Y}_{b(t),t+d_0'}^n) \\ &\le P(\sup_{s \le t}\{X_{s,t}^n - \tilde{Y}_{s,t+d_0'}^n\} > 0). \end{aligned}$$

Since $\tilde{Y}_{s,t+d_0'}^n = \tilde{Y}_{s,t}^n + \tilde{Y}_{t,t+d_0'}^n$ and $\tilde{Y}_{t,t+d_0'}^n$ is independent of s, the right hand term of the above inequality can be changed into

$$\begin{aligned} P(\tilde{D}_t^n > d_0') &\le P(\sup_{s \le t}\{X_{s,t}^n - \tilde{Y}_{s,t+d_0'}^n\} > 0) \\ &= P(\sup_{s \le t}\{X_{s,t}^n - \tilde{Y}_{s,t}^n\} > \tilde{Y}_{t,t+d_0'}^n) \\ &= P(\tilde{Q}_t^n > \tilde{Y}_{t,t+d_0'}^n), \end{aligned}$$

where the last equality is obtained by the definition of $\tilde{Q}_t^n$. $\square$

Using Propositions 5.1 and 5.2, Theorem 5.3, and (5.6), we can obtain an upper bound of delay violation probability as follows:

$$\begin{aligned} P(D_e > d_0) &= P(D^* > d_0') \\ &= P(\lim_{t\to\infty} D_t > d_0') \\ &= P(\lim_{t\to\infty} D_t^n > d_0') \\ &\le P(\lim_{t\to\infty} \tilde{D}_t^n > d_0') \\ &\le P(\lim_{t\to\infty} \tilde{Q}_t^n > \lim_{t\to\infty} \tilde{Y}_{t,t+d_0'}^n), \end{aligned} \tag{5.18}$$

where the second equality comes from the definition of D^*, the third equality is valid by Proposition 5.1, the first inequality holds by Proposition 5.2, and the final inequality is obtained from Theorem 5.3. In order to express the final expression on the right hand side of (5.18) in a more explicit form, we make one more assumption on $\tilde{Y}_{0,t}^n$.

Assuming $\{\tilde{Y}_{0,t}^n, t \ge 0\}$ has independent increments and the increments have a Gaussian distribution, we model $\tilde{Y}_{0,t}^n$ as

$$\tilde{Y}_{0,t}^n = at + \sigma B_t, \tag{5.19}$$

where $\{B_t, t \ge 0\}$ is a standard Gaussian process with independent increments, a is the mean of $\tilde{Y}_{t-1,t}^n$, and σ^2 is the variance of $\tilde{Y}_{t-1,t}^n$. Then, $\tilde{Q}_t^n$ can be expressed as

$$\begin{aligned} \tilde{Q}_t^n &= \sup_{s\le t}\{X_{s,t}^n - \tilde{Y}_{s,t}^n\} \\ &= \sup_{s\le t}\{-\sigma B_{t-s} - (a - R)(t - s)\}. \end{aligned}$$

If we define $\tilde{Q}^n$ as $\tilde{Q}^n = \lim_{t\to\infty} \tilde{Q}_t^n$, then $\tilde{Q}^n$ has the following distribution [80, p.361]:

$$P(\tilde{Q}^n > x) = e^{-\mu x},$$

where $\mu = 2(a - R)/\sigma^2$.

Real-time applications or services will require a small value of d_0, usually less than 1 second. According to Recommendation G.114 [81] of Telecommunication standardization sector of International Telecommunication Union (ITU-T), one-way transmission time of up to 150 msec is acceptable for most user applications. Thus, $d'_0 = d_0 - D_f$ may be smaller than 1. If $d'_0 \ll 1$, it is not appropriate to model $\tilde{Y}^n_{t,t+d'_0}$ as Gaussian. Since d'_0 usually has a small value, we approximate $\tilde{Y}^n_{t,t+d'_0}$ by ad'_0, and we have

$$\begin{aligned} P(\lim_{t\to\infty} \tilde{Q}^n_t > \lim_{t\to\infty} \tilde{Y}^n_{t,t+d'_0}) &= P(\tilde{Q}^n > ad'_0) \\ &= e^{-\mu a d'_0}. \end{aligned}$$

Then, from the above equation and (5.18), we can obtain the following upper bound of delay bound violation probability:

$$P(D_e > d_0) \leq \exp\left(-\frac{2(a-R)a(d_0 - D_f)}{\sigma^2}\right) \tag{5.20}$$

Let $g(R)$ denote the right hand term of (5.20). If we evaluate R^* by

$$R^* = \max\left\{R : R \geq 0, g(R) \leq \varepsilon\right\}, \tag{5.21}$$

then R^* becomes a lower bound of the admissible bandwidth for the class between the selected ingress/egress node pair. The explicit form of R^* can obtained from (5.20) and (5.21) as

$$R^* = a + \frac{\log(\varepsilon)\sigma^2}{2(d_0 - D_f)a}. \tag{5.22}$$

From the above equation, we can obtain some insights about the behavior of the admissible bandwidth. First, we can observe that R^* increases as the average of the available bandwidth a increases because the first term on the right hand side of (5.22) is dominant when a increases. Second, if the variance of the available bandwidth σ^2 increases, R^* decreases because $\log(\varepsilon)$ is negative for $\varepsilon < 1$. Thus, the second term accommodates the burstiness of traffic by decreasing R^* for large variance. Third, as the constraint is becoming more strict, that is, as the value of ε decreases, R^* also decreases. This is natural because in order to satisfy a more rigorous requirement, less traffic has to be admitted. Fourth, as the delay bound increases, R^* increases. This is also reasonable if we consider the limiting case that d_0 goes to infinity. Thus, we can know the behavior of the admissible bandwidth through the explicit form of R^*, (5.22), and the calculation complexity of R^* is very low since the value of R^* can be evaluated directly from the simple equation of (5.22) if the mean and variance of the available service are obtained through measurements.

5.3.1 Estimation of Available Service

From (5.19), we can obtain that

$$E[\tilde{Y}^n_{0,t}] = at, \quad Var[\tilde{Y}^n_{0,t}] = \sigma^2 t.$$

In this subsection, we describe how to estimate the parameters a and σ of the available service $\tilde{Y}^n_{0,t}$ by using probing packets. If we can provide the minimally backlogging probing traffic exactly, we can obtain the value of $\tilde{Y}^n_{0,t}$ exactly. However, it is not possible to send the minimally backlogging probing traffic in real networks. Thus, we send the probing traffic while trying to satisfy the minimally backlogging condition as closely as possible. The detailed probing scheme is described in Section 4.3, and we assume that the probing traffic is offered to the virtual server of the network path satisfying the minimally backlogging condition approximately in this subsection.

As the probing scheme is window-based, the calculation of the mean and variance of $\tilde{Y}^n_{0,t}$ is also window-based. Let T denote the duration of one window. Before considering the mean and variance, we first need to estimate $\tilde{Y}^n_{0,t}$ for the current window. Since the probing traffic can not satisfy the minimally backlogging condition fully during the probing period, several busy periods of probing packets may exist during a window. Let a^s_p and d_p be the times when a probing packet p arrives at the virtual server $\mathcal{S}$ corresponding to the specific path and p departs from the server, respectively. The virtual server is continuously backlogged for k probing packet transmissions from the j-th probing packet in the interval $[a^s_j, d_{j+k-1}]$ if

$$d_{j+m} \ \geq \ a^s_{j+m+1}, \quad \text{for all } 0 \leq m \leq k-2, \tag{5.23}$$

for $k \geq 2$.

Consider the i-th busy period made up by k continuously backlogging probing packets. Probing packets arriving during the busy period are indexed from 1 to k. If L is the size of a probing packet, the amount of traffic served from the time a^s_1 can be expressed as

$$\tilde{Y}^i_{a^s_1,t} = L \cdot \sup\{n : d_n \leq t\}.$$

Then, $\tilde{Y}^i_{0,t}$ can be used to estimate $\tilde{Y}^n_{0,t}$. Considering reliability, we use $\tilde{Y}^i_{0,t}$ obtained from the longest busy period to estimate $\tilde{Y}^n_{0,t}$ when there are multiple busy periods in one window.

From Theorem 3.7 of Chapter 3, we know that the service rate of minimally backlogging probing traffic $\tilde{Y}^n_{s,t}/(t-s)$ is an asymptotically unbiased estimator of the available bandwidth. Thus, if there are k probing packets in the longest busy period of the current window, we approximate the available bandwidth $\tilde{Y}^n_{t-1,t}$ for an interval of 1 second in the current window as

$$\hat{R} = \frac{kL}{d_k - a^s_1}.$$

We let R_j denote the value of $\hat{R}$ for the j-th previous window from the current window, and $j = 0$ corresponds to the current window. Then, the mean of $\tilde{Y}^n_{t-1,t}$, a, is estimated by

$$a' = \frac{1}{M_a} \sum_{m=0}^{M_a-1} R_m.$$

Thus, we use the most recent M_a windows for the estimation of the mean of $\tilde{Y}^n_{t-1,t}$. The variance of $\tilde{Y}^n_{t-1,t}$, σ^2, is estimated by

$$\sigma'^2 = \gamma \cdot \frac{1}{M_v - 1} \sum_{m=0}^{M_v-1} (R_m - \bar{R})^2,$$

where $\bar{R} = \sum_{m=0}^{M_v-1} R_m/M_v$ and γ is the variance multiplication factor (VMF). We use the most recent M_v windows for the estimation of the variance of $\tilde{Y}^n_{t-1,t}$. VMF plays an important role in guaranteeing QoS. Since we estimate the available service by sending probing packets for a relatively short period, the accuracy may not be guaranteed. Especially, the estimated variance tends to be lower than the variance of the real additionally available service because a long busy period of probing packets usually occurs when the available bandwidth is low during one window. In addition, the change of available bandwidth can be so frequent and significant that the available bandwidth may not be estimated accurately by the probing scheme in some cases. Thus, the estimated variance may be less than the actual variance and the VMF γ plays an important role in such cases. We will investigate the effect of VMF through simulation in the next section.

5.3.2 Admission Control Algorithm

We finally summarize the proposed admission control algorithm as follows. Let's consider an admission control algorithm for a specific ingress/egress router pair. The egress router calculates the lower bound of the admissible bandwidth R^* using (5.22) once every T seconds. The egress router sends the value of R^* back to the ingress router when it is calculated.

If the ingress router has not given admission to any flow in the previous window, the ingress router uses this new value R^* to determine whether to allow a new flow or not. The ingress router admits the request of the new flow with a peak rate of r_p if the following condition is satisfied:

$$r_p < R^*. \tag{5.24}$$

However, if some flow has been admitted after the last probing time, then the rate of the newly admitted flow can not be reflected in the calculation of R^*. If we use R^* directly in that case, the delay QoS may not be guaranteed. In order to overcome this problem, the ingress router need to memorize the sum of the peak rates of the flows that might not have been considered in the calculation of R^*. Let the sum of the peak rates of those flows be r_s. When the ingress receives R^* from the egress router, if we use $R^* - r_s$ as the admissible bandwidth instead of R^*, then the delay QoS will not be violated because we estimate the admissible bandwidth conservatively. Thus, the admission decision criterion of (5.24) is changed into

$$r_p < R^* - r_s = \tilde{R}^*. \tag{5.25}$$

In addition, if a flow of rate r_p is accepted in the current window, r_p also should be included in r_s.

If the probability of accepting new flows between the last probing time and the receiving time of R^* is negligible, then it is sufficient to consider only accepted flows of current window in r_s. However, if the probability is not negligible or accepted flows start to send packets after a silence period, the period considered for r_s, T_r need to be extended.

5.4 Numerical Results

In this section, we evaluate the performance of the proposed admission control scheme in terms of delay QoS, i.e., delay violation probability and utilization through OPNET simulation. We consider a network topology as shown in Fig. 5.2 for simulation. Nodes IR_1 and IR_2 are ingress routers and Node ER is an egress router. Nodes R_1, R_2, and R_3 are core routers. Nodes S_1, S_2, S_a, and S_b are source nodes where data traffic is generated. Nodes S_1 and S_2 generate only real-time flows that are subject to admission control, and Nodes S_a and S_b generates only background traffic that is not subject to admission control. Flows from source nodes S_1 and S_2 are always destined to the destination node D_1. Background traffic streams generated at Nodes S_a and S_b are directed to Nodes D_a and D_b, respectively. Data packets sent from Source Nodes S_1 and S_2 traverse six and five hops to reach Destination Node D_1, respectively. The probing traffic sent from the Ingress Routers IR_1

and IR_2 to the Egress Router ER traverses four and three hops, respectively. The proposed admission control scheme does not guarantee the end-to-end delay QoS from Source Nodes S_1 or S_2 to Destination Node D_1, but guarantees the delay QoS from the Ingress Routers IR_1 or IR_2 to the Egress Router ER.

Each node is modeled as an output queued router with a strict-priority(SP) scheduling policy. Premium class traffic that is allowed by admission control is given a strictly higher priority than best-effort traffic that is not subject to admission control. Each link has a link rate of 10 Mbps and a propagation delay of 5 ms. The sizes of all probing packets and data packets are fixed to 4000 bits. The duration of one time window T is 1 sec, and the number of probing packets sent per time window, N is 100 packets. Thus, the average probing traffic rate is 400 kbps. The values of the rate adaptation related parameters are set: $N_m = 0.70 \times N = 70, N_s = 0.05 \times N = 5, \alpha_m = 0.1,$ and $\alpha_s = 0.6$. If probing traffic from the two ingress routers passes a *tight* link with a least available bandwidth concurrently, then both ingress routers may significantly underestimate the available bandwidth for each path. In order to avoid such a situation in advance, we put an interval of 0.5 second between the probing start times of IR_1 and IR_2 in every window. Thus, IR_1 starts probing at the beginning of each window and IR_1 starts probing 0.5 second later.

We consider two types of traffic patterns for the flows that are subject to the admission control. The average lifetime of each flow is 200 seconds and the lifetime is exponentially distributed. The first one is simple on-off traffic whose *on* and *off* period lengths are exponentially distributed. The average lengths of *on* and *off* periods are both 0.5 second. No traffic is generated during *off* period and packets

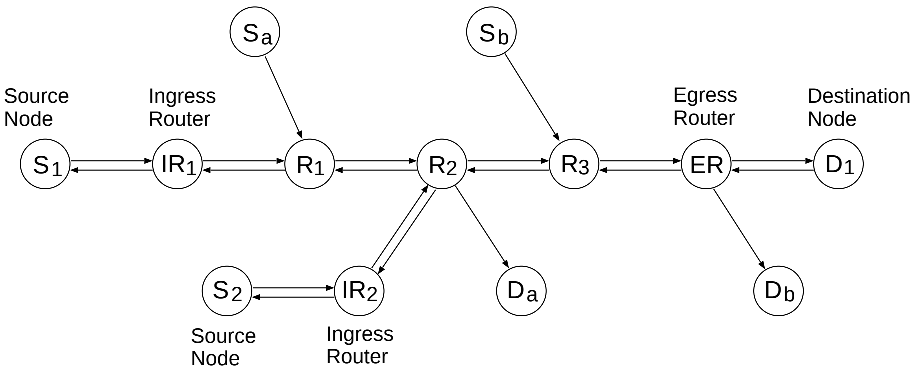


Figure 5.2: Network topology for simulation

are generated at the peak rate of r during *on* period. Thus, the average rate is $r/2$. The peak rate of each flow is fixed to 512 kbps. The flow inter-arrival time is exponentially distributed with an average of 1 second.

The second one is also on-off traffic, but the lengths of *on* and *off* periods have a Pareto distribution. If X has a Pareto distribution with shape parameter α and scale parameter β, then X has a density function $f(x)$ and a distribution function $F(x)$ of

$$f(x) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}, \quad F(x) = 1 - \left(\frac{\beta}{x}\right)^{\alpha}, \quad \text{for } x \geq \beta.$$

If the shape parameter α is less than 2, X has an infinite variance. If α is less than 1, X has infinite mean and variance. We fix the value of α to 1.9 for both *on* and *off* periods, and the values of β's are 0.2 and 2 for *on* and *off* periods, respectively. Since $E[X] = \beta\alpha/(\alpha - 1)$ if $\alpha > 1$, the length of *off* period is 10 times longer than that of *off* period on an average. Thus, the Pareto on-off traffic is very bursty. The peak rate of each flow is fixed to 512 kbps. Since the average rate of a Pareto flow is approximately one fifth of that of an exponential on-off flow. The average inter-arrival time of Pareto flows is decreased to 0.2 second in order to increase the flow arrival rate.

According to ITU-T Recommendation G.114 [81], one-way delay of up to 150 ms is usually tolerable for most user applications. In case of Voice over Internet Protocol (VoIP) services the perceived quality degrades with increased end-to-end delivery time. However, subjective tests have shown little perceived degradation until the end-to-end delay reaches 150 ms [82]. Thus, we consider 150 ms as the maximum allowable end-to-end delay in this paper. In case of 3GPP, end-to-end delay of 150 ms is preferred and transfer delay is defined as 95-percentile of the distribution of delay for all delivered data packets [83, 84]. We set the threshold for the delay bound violation probability (ε) to 0.01 rather conservatively.

First, we investigate the effect of T_r, a period that is considered to complement the flows that is omitted in the calculation of the admissible bandwidth by summing their peak rates to r_s and subtracting them from the calculated value of the admissible bandwidth. Fig. 5.3 shows the delay violation probability for various values of complement window T_r under exponential on-off traffic loads. Each probability value is obtained from 10 simulations with different seeds in the random number generator. The zero value of T_r implies that only the current window is considered to calculate r_s. If $T_r = i,\ i \geq 1$, then T_r includes up to i-th previous window from the current window. The delay bound d_0 is set to 150 ms for both Ingress Routers 1 and 2. We can observe that when $T_r = 0$, the delay performance requirement is significantly violated. The measured delay bound violation probability is over 0.1 compared with the target value of 0.01. This is because the rates of flows that are accepted after the last probing in the previous window are not reflected on the calculation of the admissible bandwidth. However, as the value of T_r increases to 1, the measured probability improves significantly since the rates of flows that are accepted after the last probing in the previous window are now reflected on the calculation of the admissible bandwidth through the term of r_s. As the value T_r increases, the measured delay violation probability improves. However, since every accepted flow starts from busy period in the simulations considered here, the T_r values that are larger than 1 are not needed. The value of T_r is fixed to 1 hereafter. We need to find a different reason for the violated delay QoS. There is no significant difference between the performances experienced by the traffic streams from IR_1 and from IR_2.

The following simulation result shows that the violation of delay QoS comes from the estimation error of the probing mechanism. We fix the value of T_r to 1. Fig. 5.4 compares the measured available bandwidth and the available bandwidth estimated by the probing mechanism proposed in Chapter 4 for the path between IR_1 and ER. The egress router obtains the available bandwidth for the path from the measured available bandwidths of intermediate nodes every time window. Fig. 5.4 also shows the admissible bandwidth calculated by (5.22). We can observe that the available bandwidth estimated by the probing scheme is closely tracking the measured available bandwidth. The calculated admissible bandwidth is much lower than the available bandwidth. This is natural if we consider the following situation. Let us consider a peak rate allocation as an example. If flows with a peak rate of 512 kbps and an average lifetime of 200 seconds arrive at an interval of 1 second on average from IR_1, then the link $R_3 - ER$ just before the egress router ER will be occupied by 19 flows in about 19 seconds. If flows also arrive from IR_2, then $R_3 - ER$ will be occupied by 19 flows in about 10 seconds in case of peak rate allocation. In that case, no more flows can be admitted from either IR_1 or IR_2 until one flow ends around 200 seconds. Thus, the admissible bandwidth will remain very low compared with a peak rate of a flow during full simulation time. In case of measurement-based admission control, the situation is a little better, but the admissible bandwidth should be low enough to guarantee the required delay QoS. Although the estimation by the probing mechanism seems rather accurate, we can

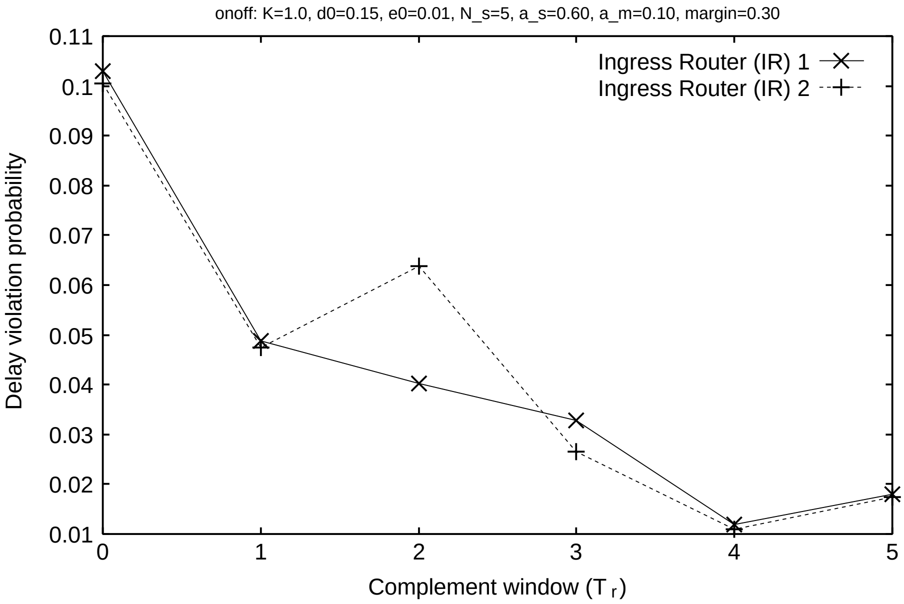


Figure 5.3: Delay violation probability for various values of complement window T_r under exponential on-off traffic loads

know the difference between the measurement and estimation of available bandwidth from the following figure.

Fig. 5.5 shows the ratio of the measured variance to the estimated variance over time when admission control is performed under exponential on-off traffic loads. The measured variance of available bandwidth is obtained from $M_v = 30$ windows. The estimated variance is also obtained from $M_v = 30$ windows. The length of window for variance estimation M_v needs to be sufficiently long for robust estimation of variance. However, we set the window for estimation of the average of available bandwidth M_a to 10 in order to follow the changing average value of available bandwidth quickly. As we can observe in the figure, the ratio of the measured variance to the estimated variance is in the range of $[0.5, 3.0]$ for most of the simulation time. We know that the admissible bandwidth decreases as the variance σ^2 increases from (5.22). Thus, if the ratio is less than 1, that is, the estimated variance is larger than the measured variance, then the admissible bandwidth R^* will be estimated conservatively. Since a conservative value of R^* allows less flows than the possible maximum number, delay QoS will not be violated. However, smaller values of estimated variance can yield optimistic value of R^* resulting violation of delay QoS. Thus, variance multiplication factor VMF γ may be needed in order to complement this error in the variance of the available bandwidth and guarantee delay QoS for real-time flows.

Fig. 5.6 shows the delay violation probability for various values of VMF γ when the delay bound d_0 is 150 ms and the delay violation probability threshold ε is 0.01. The peak rate of each exponential on-off traffic is 512 kbps. Simulation is performed for 500 seconds and the lifetime of each on-off traffic is exponentially distributed with

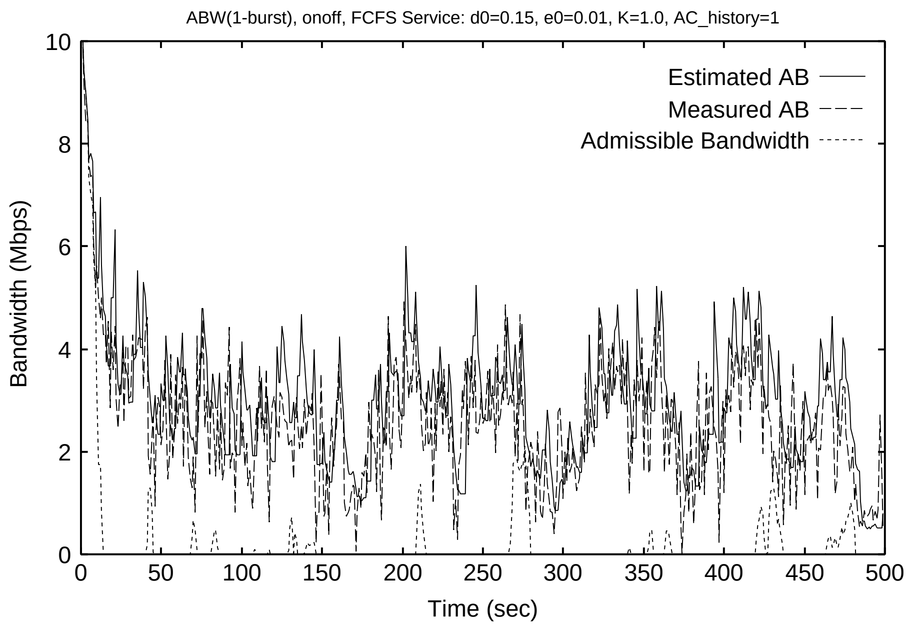


Figure 5.4: Measured/Estimated available bandwidths and admissible bandwidth over time

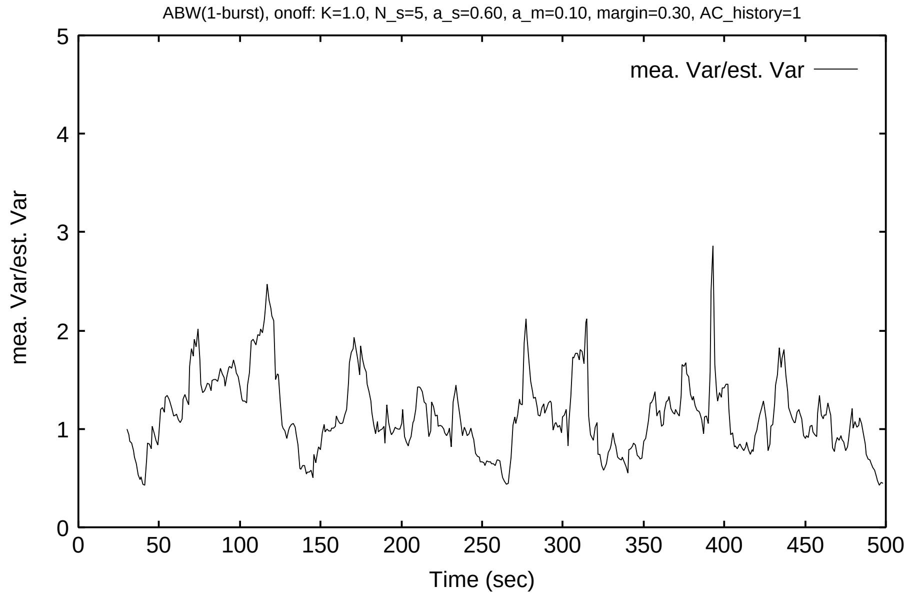


Figure 5.5: Ratio of the measured variance to the estimated variance of available bandwidth under exponential on-off traffic loads

an average of 200 seconds. When $\gamma = 1.5$, the measured delay violation probability is approximately 0.013 for both IR_1 and IR_2. Thus, the delay QoS is slightly violated for $\gamma = 1.5$. The delay QoS is satisfied very well for $\gamma \geq 2.0$.

Fig. 5.7 shows the utilization of the link between Egress Router ER and Destination Node D_1 for various values of γ under the same environment as Fig. 5.6. The utilization is measured during the last half of the simulation time. The utilization is kept over 60% for all values of γ. If we perform admission control based on only the peak rate of each flow, then maximum 19 flows can be admitted to ER concurrently. Since the ratio of *on* period length to *off* period length is 1, the utilization of 19 $\times$ 5.12/2 = 48.6% can be obtained from the peak rate allocation scheme. For the proposed scheme, the utilization tends to decrease as γ values increase because σ^2 is overestimated. Since the objective of the admission control is to increase the utilization while guaranteeing the delay QoS, the optimal value of γ can be determined around 1.5.

Fig. 5.8 compares the delay violation probability for $\gamma = 1.0$ with that for $\gamma = 1.5$ when delay bound (d_0) has various values from 50 msec to 150 msec under exponential on-off traffic loads. We can observe that the delay QoS is satisfied for almost all values of d_0 when $\gamma = 1.5$ since the variance σ^2 is rather conservatively estimated. On the other hand, the delay QoS is satisfied for $d_0 \leq 0.08$ when $\gamma = 1.0$. We need to note that the measured delay violation probability does not significantly deviate from the target value of 0.01 even when $d_0 \geq 0.10$ and $\gamma = 1.0$. The VMF of the value of 1.5 is better for guaranteeing the delay requirements. We can observe that the measured delay violation probability increases as the delay bound

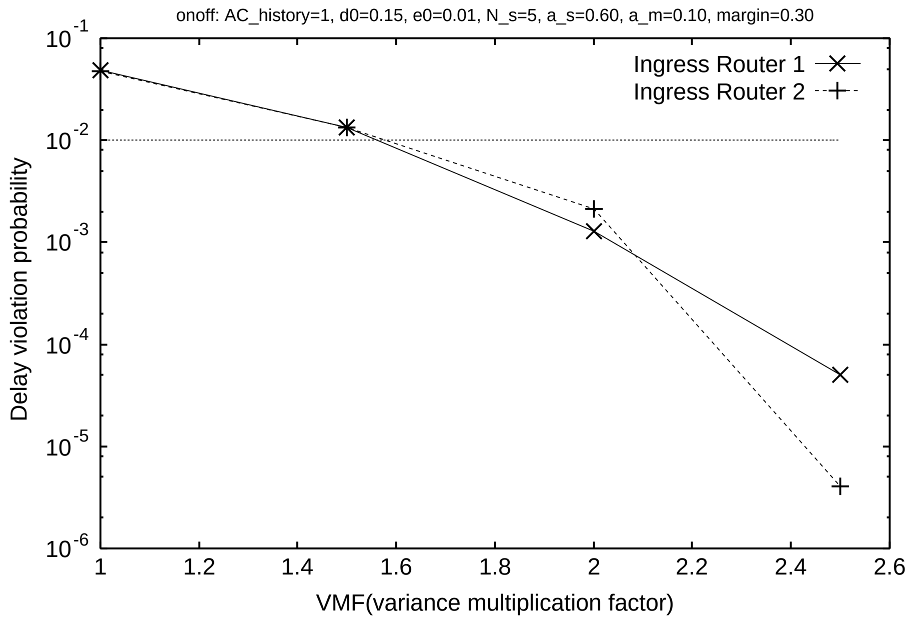


Figure 5.6: Delay violation probability for various VMF values under exponential on-off traffic loads

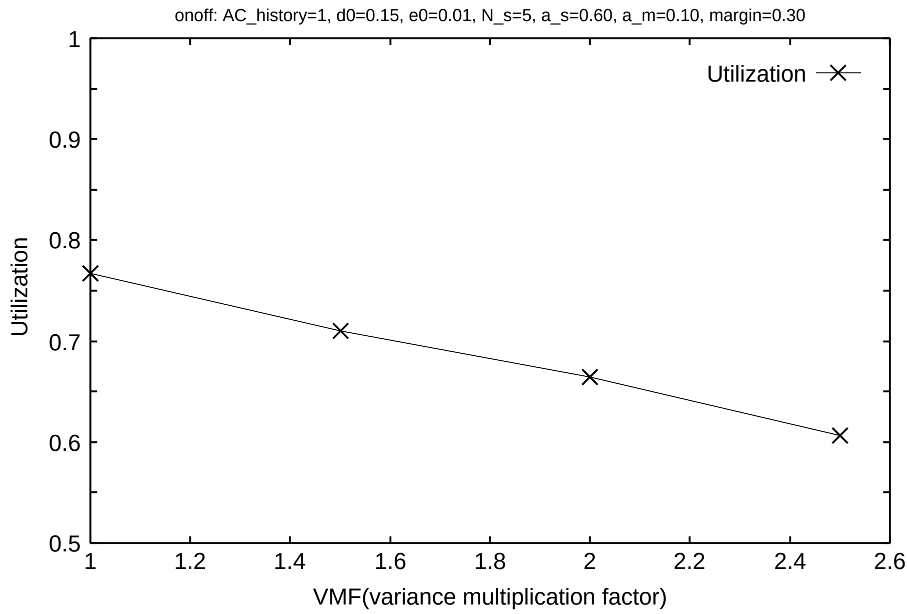


Figure 5.7: Measured utilization of link $\mathrm{ER} - \mathrm{D}_1$ for various VMF values under exponential on-off traffic loads

d_0 increases. This tendency can be explained as follows. Let us consider the case of $d_0 = 0.05$. If we consider the path $\mathrm{IR}_1 - \mathrm{R}_1 - \mathrm{R}_2 - \mathrm{R}_3 - \mathrm{ER}$, Node ER is 4 hops away from IR_1. Thus, the fixed delay component for the path is at least 4×5 msec $= 20$ msec due to four link propagation delays. Then, according to the requirement that $P(D_e > d_0) \leq 0.01$, the admission control algorithm tries to maintain the queueing delay at the virtual server less than 50-20=30 msec with a high probability of about 99%. In this case, the average delay at the virtual server will be much lower than 30 msec. We assume that the average delay is one tenth of 30 msec, i.e., 3 msec. If the available bandwidth at the virtual server is 10 Mbps, i.e., the link rate, then the service time of a packet of 4 kbits is 0.4 msec and the average delay of 3 msec implies that there are about 7.5 packets in the virtual server. However, in the scenario considered in this simulation, the available bandwidth at the virtual server is much lower than 10 Mbps, and there will be fewer than 7.5 packets in the virtual server on average. Since fluid assumption is not good for a queueing system where the number of packets is very small, the gap between the target and the measured probability is rather large for small values of d_0 and the gap decreases as d_0 increases as shown in Fig. 5.8.

Fig. 5.9 compares the utilization of the link between ER and D_1 for $\gamma = 1.0$ with that for $\gamma = 1.5$ when delay bound (d_0) has various values from 50 msec to 150 msec. We can observe that the utilization is better for $\gamma = 1.0$ than for $\gamma = 1.5$ since overestimation of the variance σ^2 tends to admit less flows than the possible maximum amount. Thus, $\gamma = 1.0$ is good for high utilization of resources and $\gamma = 1.5$ is good for guaranteeing the delay QoS. The appropriate value of γ can be

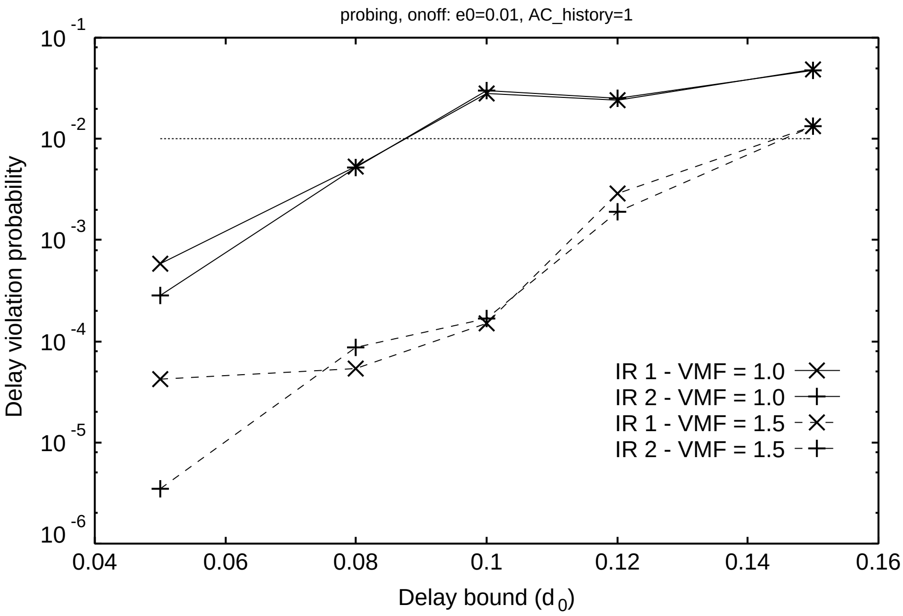


Figure 5.8: Delay violation probability for various delay bounds under exponential on-off traffic loads

chosen considering the relative importance between the two targets.

Thus far, simulations were performed for only exponential on-off traffic loads. The following results show the performance of the proposed admission control scheme for Pareto on-off traffic loads. Fig. 5.10 shows the measured delay violation probability obtained under Pareto on-off traffic loads. The value of VMF γ is fixed to 1. We can observe that the delay QoS is satisfied for most delay bounds even if γ is 1. Compared with the case of Fig. 5.8, we can observe that smaller delay violation probabilities are obtained for Pareto on-off traffic loads than for exponential on-off traffic loads. The reason can be explained as follows. While the average inter-arrival time of exponential on-off flow is 1 second, the average inter-arrival time of Pareto on-off flow is 0.2 second. The average arrival time of Pareto on-off flow is approximately one fifth of that of exponential on-off flow in order to match the arrival rate of the aggregate traffic of each traffic pattern at the same level. We need to note that the bandwidth resources are allocated conservatively in a time window, that is, admission control is performed conservatively due to r_s. In other words, since the rates of accepted flows in the current time window can not be reflected on the admissible bandwidth obtained from the last measurement of the previous window, bandwidth resources are reserved according to the peak rate of the accepted flow in the current time window by a term of r_s in (5.25). Since higher flow arrival rates in a window can cause conservative reservation of bandwidth resources more frequently, the delay QoS is well satisfied for Pareto on-off traffic with a VMF value of 1.0.

Fig. 5.11 shows the utilization of the link between ER and D_1 under the same condition as Fig. 5.10. If we compare Figs. 5.9 and 5.11, we can observe that the

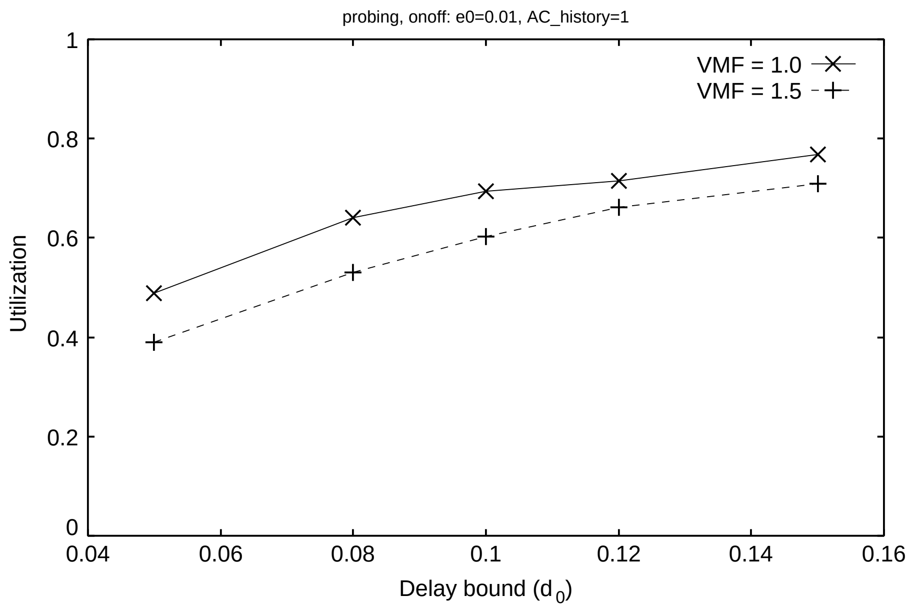


Figure 5.9: Measured utilization of link $\mathrm{ER} - \mathrm{D}_1$ for various delay bounds under exponential on-off traffic loads

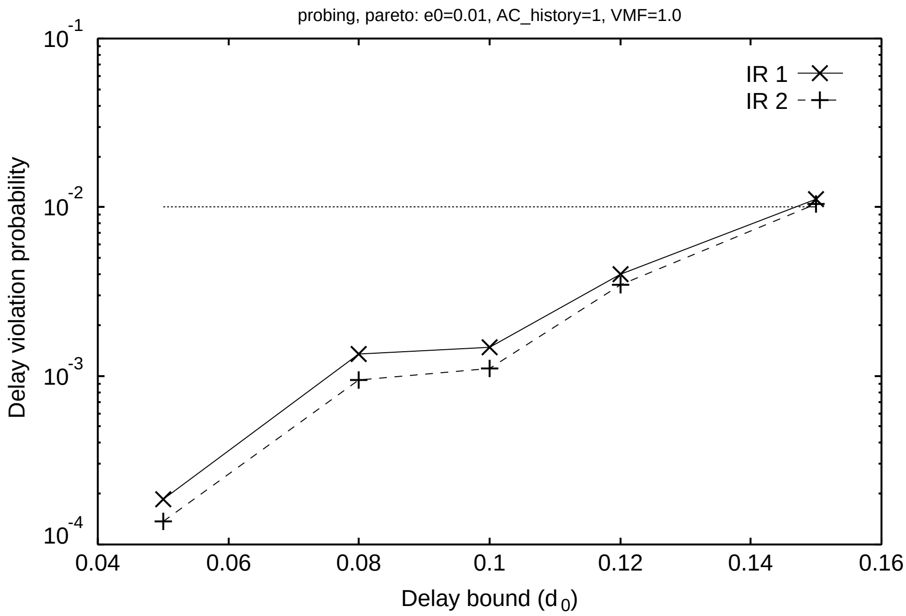


Figure 5.10: Delay violation probability for various delay bounds under Pareto on-off traffic loads

utilization is rather low compared with the case of exponential on-off traffic loads. Since the Pareto on-off traffic is very bursty, we can easily expect that the variance σ^2 will be larger for Pareto on-off traffic loads than for exponential on-off traffic loads. Then, from (5.22) the admissible bandwidth decreases due to a relatively large value of σ^2. Thus, utilization decreases for bursty Pareto on-off traffic patterns compared with the case of exponential on-off traffic. However, the utilization is kept above 36% for all delay bounds. If we perform admission control and reserve bandwidth based on only the peak rate of each flow, then maximum 19 flows can be admitted concurrently because the link rate of each link is 10 Mbps and the peak rate of each flow 512 kbps. Since the ratio of *on* period length to *off* period length is 0.1. Then, the utilization of only $97.28/11 = 8.8\%$ can be obtained from the peak rate allocation scheme. Thus, the benefit of the proposed measurement-based admission control scheme is significant especially with regard to utilization when the offered traffic is highly bursty. Since we observed the performance of the proposed admission control scheme under the Pareto on-off traffic loads is better than or similar to the case of exponential on-off traffic loads, we consider only exponential on-off traffic loads hereafter.

Fig. 5.12 compares the delay violation probability measured under no cross traffic with that measured under cross traffic. Node $\mathrm{S_a}$ sends a self-similar traffic flow to Node D_a through the path $\mathrm{S_a} - \mathrm{R_1} - \mathrm{R_2} - \mathrm{D_a}$ as a cross traffic and Node $\mathrm{S_b}$ sends another self-similar traffic flow to Node $\mathrm{D_b}$ through the path $\mathrm{S_b} - \mathrm{R_3} - \mathrm{ER} - \mathrm{D_b}$ in Fig. 5.2. Self-similar traffic patterns are generated using a multi-fractal model [77]. The average rate of each self-similar traffic flow is 1.5 Mbps. The Hurst parameter

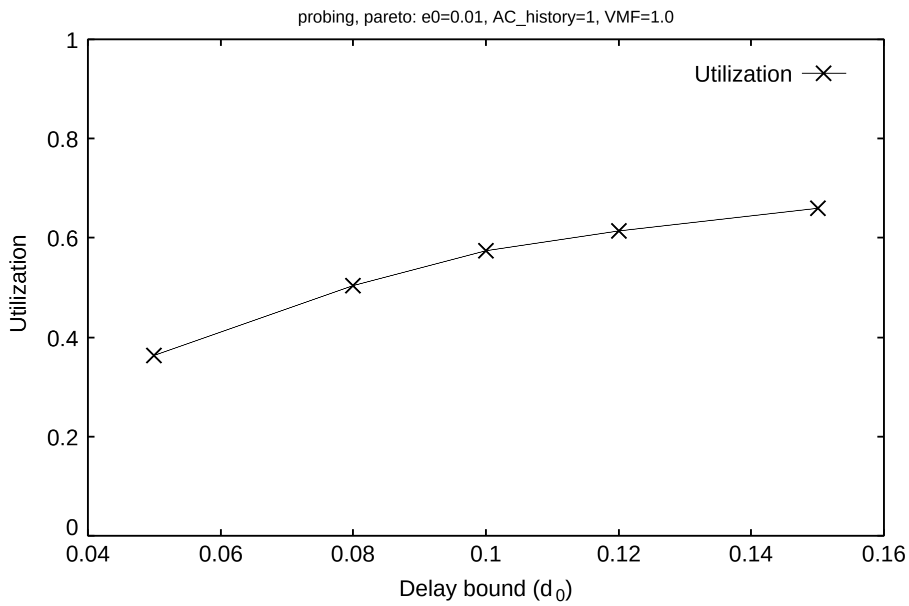


Figure 5.11: Measured utilization of link $\mathrm{ER} - \mathrm{D}_1$ for various delay bounds under Pareto on-off traffic loads

of each flow is 0.8. We can observe that the delay performance is not very different from the case of no cross traffic for both IR_1 and IR_2. This is because the cross traffic is best-effort traffic that is not subject to admission control and core routers gives strict priority to the premium class that is admitted by admission control. Thus, the cross traffic does not affect the delay performance of the higher priority class traffic significantly.

Fig. 5.13 compares the utilization of the link $\mathrm{ER} - \mathrm{D}_1$ measured under no cross traffic with that measured under cross traffic. The utilizations of the two cases are very similar as shown in the figure. Thus, we can know that if core routers use a strict priority policy, then the utilization of the premium class traffic is not affected by the best-effort traffic significantly. In addition, Figs. 5.12 and 5.13 show the proposed admission control scheme operates in a normal way when there is a lower priority cross traffic.

We now investigate the effect of measurement time window T. Thus far, the value of T is fixed to 1 second. Fig. 5.14 shows the measured delay violation probability for various values of T and Fig. 5.15 shows the measured utilization of the link $\mathrm{ER} - \mathrm{D}_1$ for various values of T. No cross traffic is offered and the values of VMF γ and T_r are 1.0 and 1, respectively. We can observe that the delay performance requirements are well satisfied as the the value of T increases. Especially, when $T = 5.0$, the delay QoS is satisfied for all delay bounds. During one window of interval length of T, bandwidth resources are reserved according to the peak rate of a flow. Thus, admission control is performed very conservatively during a window according to the peak rate allocation policy. Therefore, we can expect the band-

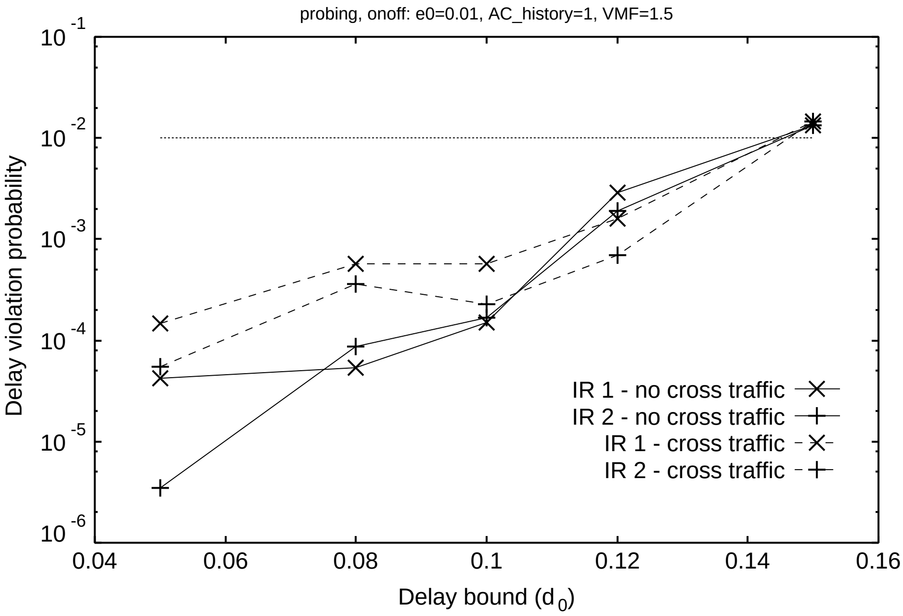


Figure 5.12: Comparison of the delay violation probabilities with or without cross traffic

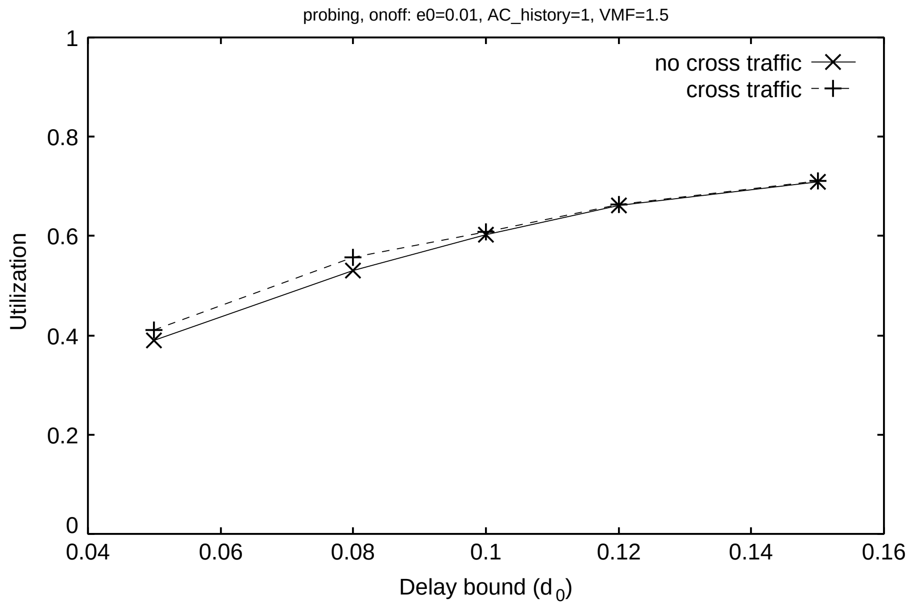


Figure 5.13: Comparison of the measured utilizations of link $ER-D_1$ with or without cross traffic

width resource will be allocated conservatively as T increases. As a consequence of conservative resource allocation, the delay QoS is well satisfied for large values of T. However, we can observe that the utilization decreases significantly as the value of T increases in Fig. 5.15. Thus, the value of T needs to selected considering the tradeoff between the delay QoS and the resource utilization. When $T = 2.0$, the delay QoS is approximately satisfied for $d_0 \leq 0.12$. If the objective of admission control is to increase the utilization while guaranteeing the delay QoS up to a delay bound of 0.12 second, then 2.0 will be a good value for T under the condition of $\gamma = 1.0$.

Thus far, we have fixed the link rate of each link to 10 Mbps. We now investigate the performance of the proposed admission control scheme for different link rates. Fig. 5.16 compares the delay violation probability obtained when the link rate of every link is 50 Mbps with that obtained when each link rate is 10 Mbps. The values of the measurement window T and T_r is fixed to 1 second and 1, respectively. The value of VMF γ is 1, that is, there is no recovery of the estimation error of σ^2 by γ. When the link rate is 10 Mbps, the delay QoS is not guaranteed for $d_0 \geq 0.1$ since the estimation error of σ^2 is not compensated by VMF. On the other hand, the delay QoS is guaranteed for all values of d_0 when the link rate is 50 Mbps even though VMF is not used. This is because the resources are used more conservatively when the link rate is 50 Mbps compared with the case of the link rate of 10 Mbps. Under the same admission request arrival patterns, the net amounts of available and admissible bandwidths are usually higher for a link rate of 50 Mbps than for 10 Mbps. Consequently, the effective arrival rate of accepted flows for a

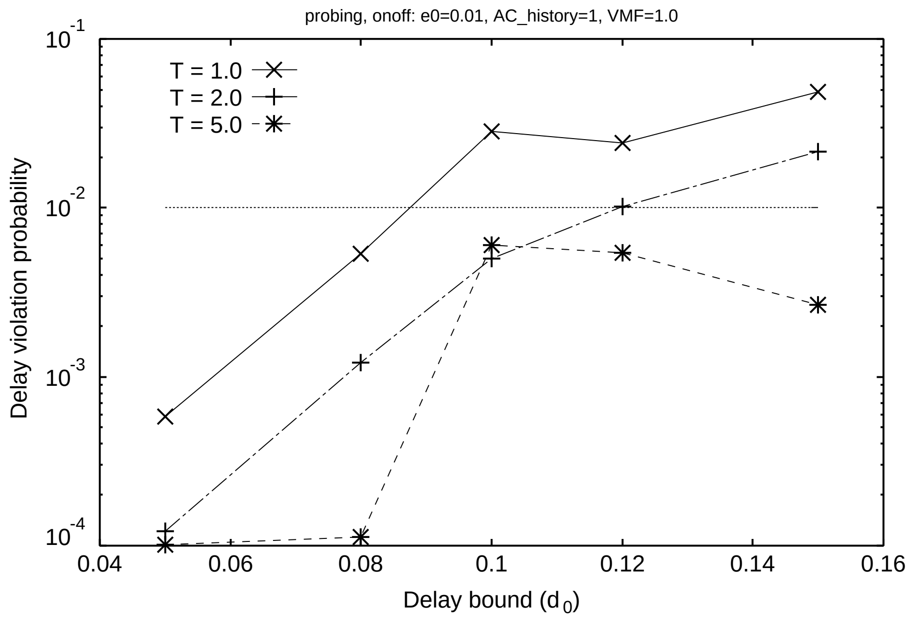


Figure 5.14: Delay violation probabilities for various measurement window T

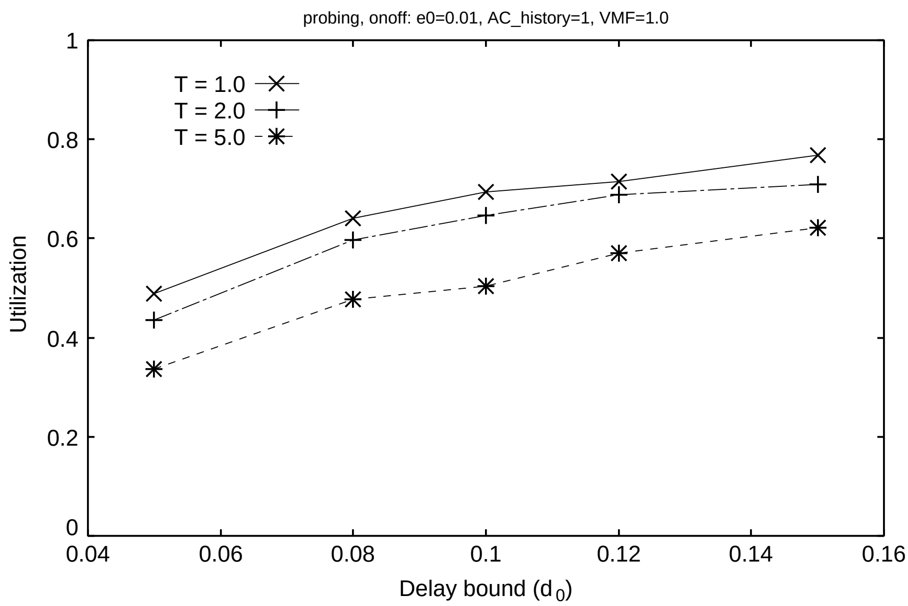


Figure 5.15: Utilizations of link $\mathrm{ER} - \mathrm{D}_1$ for various measurement window T

time window of length T is higher for a link rate of 50 Mbps than for 10 Mbps. Since bandwidth resources are allocated conservatively according to the peak rate of each flow during a window, a higher acceptance rate during a window implies more frequent conservative allocation of resources. Due to this conservative allocation of resources, the delay QoS is well satisfied for a link rate of 50 Mbps.

Fig. 5.17 compares the utilization of link $\mathrm{ER} - \mathrm{D}_1$ measured when the link rate is 50 Mbps with that obtained when the link rate is 10 Mbps. The environment is the same as the case of Fig. 5.16. We can observe that the utilization also improves significantly as the link rate increases. We can find a reason for the improved utilization from (5.22). If $\sigma = 0$, then the admissible bandwidth R^* will be equal to the available bandwidth a. If we admit flows up to the rate of R^* according to the admission control algorithm of Subsection 5.3.2, then the utilization of the tight link will be 1, since $a = C - \lambda$, where C is the link rate of the tight link and λ is the arrival rate of cross traffic that passes the tight link in a given time interval. If bandwidth resources are allocated up to R^* when $\sigma \neq 0$, the the utilization u can be expressed as

$$u = \frac{\lambda + R^*}{C} = 1 + \frac{\log(\varepsilon)\sigma^2}{2(d_0 - D_f)aC}.$$

We need to note that the second term on the right hand side of the above equation is negative due to $\log(\varepsilon)$. If we assume that the available bandwidth a is proportional to C approximately, then the second term is proportional to σ^2/a^2. The sigma/mean ratio, σ/a of the available bandwidth tends to decrease as the link rate C increases and more traffic flows are multiplexed. Thus, a decrease in the absolute value of the second term yields higher utilization as the link rate C increases.

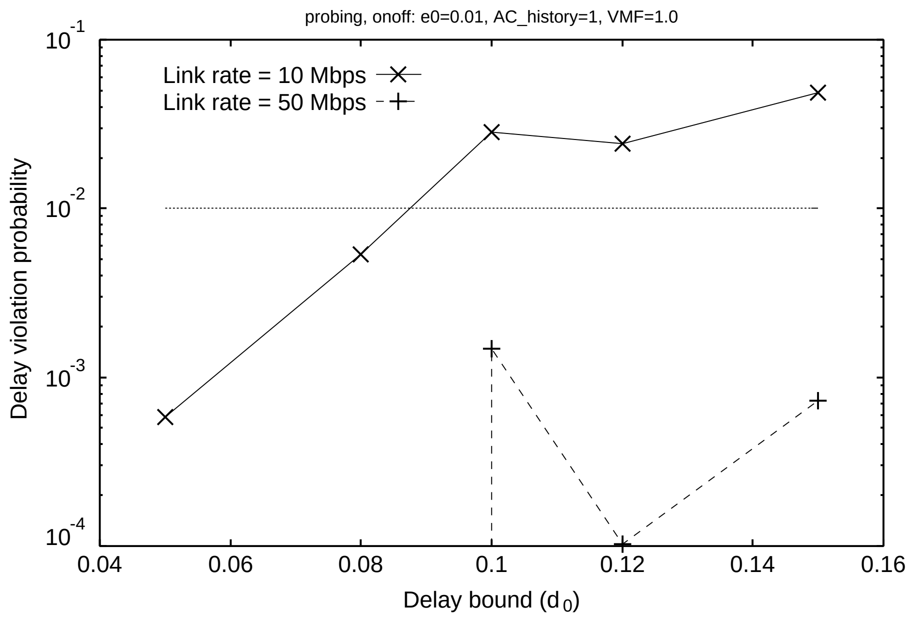


Figure 5.16: Comparison of the delay violation probabilities for different link rates

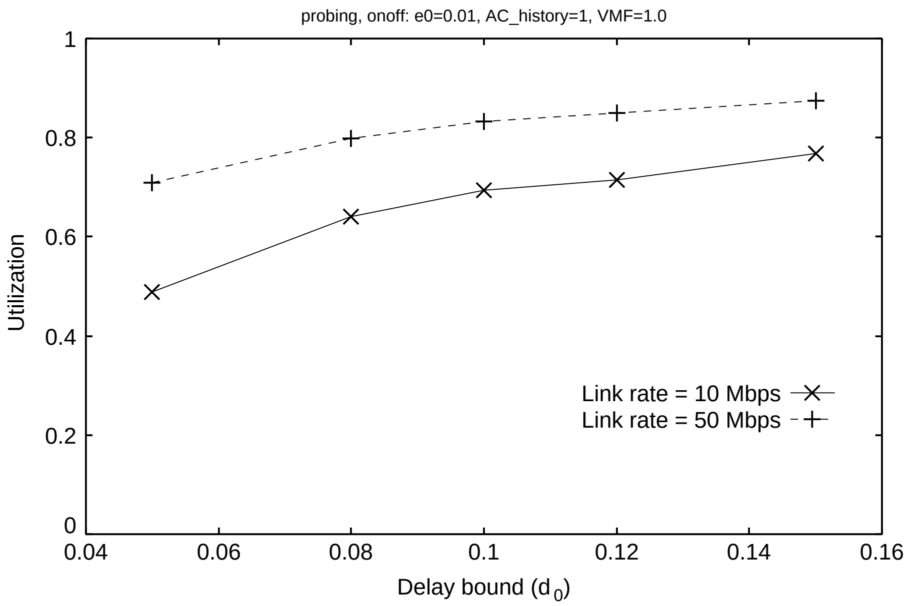


Figure 5.17: Comparison of the utilizations of link $\mathrm{ER} - \mathrm{D}_1$ for different link rates

5.5 Summary

In this chapter, we proposed a new admission control scheme. In the proposed scheme, an ingress router manages the admissible bandwidth for the path to each possible egress router. Since the admissible bandwidth is calculated considering the delay QoS, it is possible to guarantee the delay performance of the aggregate traffic for a specific path if the proposed admission control scheme is used. We derived an expression for a lower bound of the admissible bandwidth. We use the lower bound as an estimate for the admissible bandwidth. Since the bound is explicitly expressed in terms of delay bound (d_0), threshold for the delay violation probability (ε), fixed delay component (D_f), and mean a and variance σ^2 of the available bandwidth, we can understand the effect of each factor on the admissible bandwidth intuitively. Using the probing scheme developed in Chapter 4, we can estimate the available bandwidth and can obtain the mean and variance from the history of the available bandwidth. In case that the probing scheme can not accurately track the available bandwidth due to too frequent and large-scale changes, a variance multiplication factor (VMF) can be used in order to compensate the variance (σ^2) estimation error.

Through simulations, we investigated the effect of VMF γ, measurement time window T, and link rate C on the performance of the proposed admission control scheme. Even without VMF, it was possible to obtain a delay violation probability on the order of the required threshold for most delay bounds. If a small value of VMF such as 1.5 is used, then the delay QoS can be satisfied for most delay bounds. As the measurement window T increases, the delay QoS is satisfied very well, but the utilization is lowered since bandwidth resources are conservatively allocated during a time window of T. Thus, too large values of T are not good in terms of utilization. Since large values of T lead to conservative allocation of bandwidth resources, a proper value of T can yield high utilization while guaranteeing delay QoS for all delay bounds without need of VMF. Finally, as the link rate increases, the admission control yields better performance guaranteeing delay QoS for most delay bounds and resulting in higher utilization compared with the case of lower link rate.

Thus, the proposed admission control scheme can yield high utilization while guaranteeing the required delay QoS. However, we need to consider the scalability of the probing scheme used to estimate the available bandwidth for a given path. In order to explain this problem in more detail, we give an example. When Ingress Router 1 IR_1 and 2 IR_2 send probing traffic to the same Egress Router ER, if the probing traffic streams from each ingress routers pass a common tight link concurrently, then both IR_1 and IR_2 tend to detect the available bandwidth of $a/2$, where a is the real available bandwidth. This is because the probing traffic from IR_1 looks like a cross traffic to that from IR_2, and vice versa. As the number of probing traffic flows increases, it is likely that such a situation occurs more frequently at a tight link. One possible solution to this problem is to increase the interval between successive probing times, T. However, as shown in the simulation result of the previous section, large values of T may lead to a low utilization of bandwidth resources. Even though T is fixed to 1 second, if the link rate increases, which has been thus far and is likely to be in the future, then the time required for the probing packet stream to pass a tight link can be decreased. Let us assume that 100 packets of the same size of 4 kbits are sent once during a time window of $T = 1$ second. If the available bandwidth of a tight link is 10 Mbps, then it takes about 100 $\times$ 4 kbits / 10 Mbps = 40 msec for the full probing packet sequence to pass the node corresponding to the tight link. Thus, one probing sequence occupies a tight link during 4 % of 1 second. However, as the link rate increases this ratio will decrease. For example, if the available bandwidth increases to 100 Mbps, then the ratio decreases to 0.4 % of 1 second, and consequently, the collision probability of different probing packet sequences will be lowered. More fundamental solutions to this problem need to investigated further in the future.

6. Conclusions and Further Studies

We studied monitoring of available bandwidth on a network path and proposed a preventive traffic control scheme based on the network monitoring in order to satisfy quality-of-service (QoS) requirements for real-time traffic flows. We especially consider end-to-end delay as the QoS target of the real-time applications.

First, a new methodology is proposed to estimate the available bandwidth of a queueing system, whose service rate and the load of input traffic are not known in advance. In order to estimate the available bandwidth, we propose a probing method called a *minimally backlogging method* and propose two statistics. The first statistic is based on the delay of each probing packet and the second statistic is based on the amount of probing packets served in a specific time interval. We first show that an $M/G/1$ queueing system is stable when probing packets are sent to the system according to the minimally backlogging method. We also show that the available bandwidth can be estimated by using either of the two statistics if the probing packets are sent to the queueing system by the minimally backlogging method. Especially, the second statistic can be used to estimate the available bandwidth of a $G/G/1$ queueing system. We use the theory developed for a single server to estimate the available bandwidth for a local server as an application. The simulation results show that the two proposed statistics yield very accurate estimates under a Poisson and a self-similar traffic loads even for a finite probing duration.

Second, a new mechanism to estimate the available bandwidth for multiple hop routes is proposed by extending the approach for a single server, especially with the second statistic, and introducing a simplified path model which simplifies a multiple hop path into a combination of a fixed delay component and a virtual server. Since the proposed mechanism can estimate the available bandwidth quickly and track it adaptively and continuously, a reasonable range of available bandwidth for a short time interval can be obtained using the mean and variance of the estimated available bandwidth. It is observed that the proposed available bandwidth estimation mechanism yields more accurate estimates than pathload especially when the available bandwidth changes dynamically. Since the proposed probing scheme can operate at a much lower rate than the available bandwidth, the proposed probing scheme can be used non-intrusively.

Finally, a scalable architecture and an admission control algorithm for real-time flows are proposed. In our approach, admission decision is made for each flow at the edge (ingress or egress) routers, but it is scalable because the algorithm is very simple as a single comparison logic. In the proposed admission control scheme, an estimate of the admissible bandwidth, which is defined as the maximum rate of a flow that can be accommodated additionally while satisfying the delay performance requirements for both existing and new flows, is calculated based on the available bandwidth which is estimated by edge routers through monitoring minimally backlogging probing packets. Since a lower bound of the admissible bandwidth is derived and expressed explicitly in terms of delay bound (d_0), threshold for the delay violation probability (ε), fixed delay component (D_f), and mean a and variance σ^2 of the available bandwidth, we can investigate the effect of each factor on the admissible bandwidth intuitively. Since the available bandwidth used for calculation of the admissible bandwidth is estimated by the proposed probing scheme, an estimation error may exist. In order to complement the error in the estimation of variance σ^2 of the available bandwidth, a variance multiplication factor (VMF) γ can be used.

Through simulations, we investigated the effect of VMF γ, measurement time window T, and link rate C on the performance of the proposed admission control scheme. Even without VMF, it was possible to obtain a delay violation probability on the order of the required threshold for most delay bounds. A proper value of measurement window T can yield high utilization while guaranteeing delay QoS for all delay bounds without need of VMF. Finally, as the link rate C increases, the admission control yields better performance guaranteeing delay QoS for most delay bounds and resulting in higher utilization compared with the case of lower link rate.

The proposed available bandwidth estimation mechanism yields better performance than existing schemes in terms of speed and accuracy, and the proposed admission control scheme can also yield high utilization while guaranteeing the required delay QoS. However, we need to consider the scalability problem of the probing scheme used to estimate the available bandwidth for a given path. In order to explain this problem in more detail, we give an example. When Ingress Router 1 IR_1 and 2 IR_2 send probing traffic to the same Egress Router ER, if the probing traffic streams from each ingress routers pass a common tight link concurrently, then both IR_1 and IR_2 tends to detect the available bandwidth of $a/2$, where a is the real available bandwidth. This is because the probing traffic from IR_1 looks like a cross traffic to that from IR_2, and vice versa. As the number of probing traffic flows increases, it is likely that such a situation occurs more frequently at a tight link. One possible solution to this problem is to increase the interval between successive probing times, T. However, as shown in the simulation result of Chapter 5, large values of T may lead to a low utilization of bandwidth resources. Even though T is fixed to 1 second, if the link rate increases, then the time required for the probing packet stream to pass a tight link can be decreased. Let us assume that 100 packets of the same size of 4 kbits are sent once during a time window of $T = 1$ second. If the available bandwidth of a tight link is 10 Mbps, then it takes about 100×4 kbits / 10 Mbps = 40 msec for the full probing packet sequence to pass the node corresponding to the tight link. Thus, one probing sequence occupies a tight link during 4 % of 1 second. As the link rate increases, this ratio will be decreased. For example, if the available bandwidth increases to 100 Mbps, then the ratio decreases to 0.4 % of 1 second, and consequently the collision probability of different probing packet sequences will be lowered. However, more fundamental solutions to this problem need to investigated further in the future.

There are additional further study issues as follows:

- Enhancement of available bandwidth estimation mechanism in terms of stability and transient performance
- Measurement-based admission control in an environment where most flows including the flows that are subject to admission control are characterized as responsive traffic such as TCP traffic

- Available bandwidth estimation of each class when there are multiple classes of traffic in a network
- Multi-class admission control for each possible scheduling policy such as GPS and strict priority scheduling

A. Appendix: Proof of Theorem 3.4

Since the arrival process is a Poisson process with rate λ, the conditional distribution of $X_i|W_{i-1}$ is a Poisson with mean λW_{i-1}. Using this, we obtain the Laplace transform of the random variable $W_i|W_{i-1}$.

$$\begin{aligned}
E[e^{-sW_i}|W_{i-1}] &= \sum_{k=0}^{\infty} E[e^{-sW_i}|X_i = k, W_{i-1}] \Pr\{X_i = k|W_{i-1}\} \\
&= \sum_{k=0}^{\infty} E[e^{-sW_i}|X_i = k] \Pr\{X_i = k|W_{i-1}\}.
\end{aligned}$$

In the above equation, $E[e^{-sW_i}|X_i = k, W_{i-1}]$ is replaced by $E[e^{-sW_i}|X_i = k]$ because W_i depends only on X_i. If we observe that $E[e^{-sW_i}|X_i = k]$ is calculated to be $\tilde{G}_p(s)\tilde{G}(s)^k$, then the above equation is rewritten as

$$E[e^{-sW_i}|W_{i-1}] = \tilde{G}_p(s)e^{-\lambda W_{i-1}(1-\tilde{G}(s))}. \tag{A.1}$$

Let $\varphi_n(s) = E[e^{-s(W_1+W_2+\ldots+W_n)}]$. Then, by the Markov property, we have that

$$\begin{aligned}
\varphi_n(s) &= E[E[e^{-s(W_1+W_2+\ldots+W_n)}|W_1, W_2, \ldots, W_{n-1}]] \\
&= E[e^{-s(W_1+W_2+\ldots+W_{n-1})}E[e^{-sW_n}|W_{n-1}]].
\end{aligned}$$

Applying Eqn. (A.1) to the above equation, we obtain that

$$\varphi_n(s) = \tilde{G}_p(s)E[e^{-s(W_1+W_2+\ldots+W_{n-2})}e^{-(s+\lambda(1-\tilde{G}(s))W_{n-1}}]. \tag{A.2}$$

We define a sequence $(s_1, s_2, s_3, \ldots)$ recursively as follows:

$$\begin{aligned}
&s_1 = s, \\
&s_i = s + \lambda(1 - \tilde{G}(s_{i-1})), \quad i = 2, 3, \ldots
\end{aligned} \tag{A.3}$$

Note that each s_i in the sequence $(s_1, s_2, s_3, \ldots)$ is a function of s. Applying Eqn. (A.1) to Eqn. (A.2) iteratively, we obtain that

$$\varphi_n(s) = \prod_{k=1}^{n-1} \tilde{G}_p(s_k) E[e^{-s_n W_1}].$$

Then, by Eqn. (3.6), we obtain a formula of $\varphi_n(s)$ given by

$$\varphi_n(s) = \prod_{k=1}^{n} \tilde{G}_p(s_k) \Pi(\tilde{G}(s_n)).$$

Let $\beta_k(s) = \tilde{G}(s_k)$ and $\beta_{p,k}(s) = \tilde{G}_p(s_k)$. Then,

$$\log \varphi_n(s) = \log \Pi(\beta_n(s)) + \sum_{k=1}^{n} \log \beta_{p,k}(s).$$

Since the variance of $\sum_{i=1}^{n} W_i$ is equal to $d^2 \log \varphi_n(0)/ds^2$, we have that

$$\begin{aligned} V\left[\sum_{i=1}^{n} W_i\right] &= E[X_1]\{\beta_n''(0) - \beta_n'(0)^2\} + V[X_1]\beta_n'(0)^2 \\ &\quad + \sum_{i=1}^{n} \left\{\beta_{p,i}''(0) - \beta_{p,i}'(0)^2\right\}. \end{aligned} \tag{A.4}$$

In order to obtain an upper bound of $V[\sum_{i=1}^{n} W_i]$, we derive the formulas of $\beta_i'(0)$, $\beta_i''(0)$, $\beta_{p,i}'(0)$, and $\beta_{p,i}''(0)$ for $i = 1, 2, \cdots$. From Eqn. (A.3), it follows that

$$\beta_{n+1}(s) = \tilde{G}(s + \lambda - \lambda\beta_n(s)), \quad n = 1, 2, 3, \cdots. \tag{A.5}$$

By differentiating the above equation and substituting $s = 0$, we obtain that

$$\beta_{n+1}'(0) = \rho\beta_n'(0) - E[S].$$

Since $\beta_1(s) = \tilde{G}(s)$, $\beta_1'(0) = -E[S]$. Then, $\beta_n'(0)$ is obtained as

$$\beta_n'(0) = \frac{\rho^n - 1}{1 - \rho} E[S]. \tag{A.6}$$

By the same manner as the above and using Eqn. (A.6), we derive a recursive equation given by

$$\beta''_{n+1}(0) = E[S^2]\left(\frac{1-\rho^{n+1}}{1-\rho}\right)^2 + \rho\beta''_n(0), \quad n = 1, 2, 3, \cdots.$$

Clearly, $\beta''_1(0) = E[S^2]$. Solving the above recursive equation yields

$$\beta''_n(0) = \frac{1-(2n+1)(1-\rho)\rho^n - \rho^{2n+1}}{(1-\rho)^3}\, E[S^2]. \tag{A.7}$$

We now evaluate $\beta'_{p,i}(0)$ and $\beta''_{p,i}(0)$. Eqn. (A.3) implies that

$$\beta_{p,n+1}(s) = \tilde{G}_p(s + \lambda - \lambda\beta_n(s)), \quad n = 1, 2, 3, \cdots.$$

Then, by the similar method used to obtain $\beta'_n(0)$ and $\beta''_n(0)$, we obtain that

$$\beta'_{p,n}(0) = \frac{\rho^n - 1}{1-\rho} E[S_p], \quad n = 1, 2, 3, \ldots \tag{A.8}$$

$$\beta''_{p,n}(0) = E[S_p^2]\left(\frac{1-\rho^n}{1-\rho}\right)^2 + \lambda E[S_p]\beta''_{n-1}(0), \quad n = 2, 3, 4, \ldots, \tag{A.9}$$

where the explicit formula of $\beta''_n(0)$ is given by Eqn. (A.7). Since $\beta_{p,1}(s) = \tilde{G}_p(s)$, $\beta''_{p,1}(0) = E[S_p^2]$. If we recall that $\rho < 1$, then an upper bound of $\beta''_{p,n}(0)$ is obtained from Eqn. (A.9), which is given by, for all n,

$$\beta''_{p,n}(0) \le \frac{E[S_p^2]}{(1-\rho)^2} + \frac{\lambda E[S_p]E[S^2]}{(1-\rho)^3}.$$

Moreover, from Eqns. (A.6) and (A.7), we obtain that

$$\beta''_n(0) \le \frac{E[S^2]}{(1-\rho)^3},$$

$$\{\beta'_n(0)\}^2 \le \left(\frac{E[S]}{1-\rho}\right)^2.$$

From the above equations and Eqn. (A.4), we have that $V\left[\sum_{i=1}^n W_i\right] \le A\,n$, where A is a constant not depending on n. Since $V\left[\sum_{i=1}^n W_i/n\right] = V\left[\sum_{i=1}^n W_i\right]/n^2$, the proof is completed. □

요 약 문

인터넷 망에서의 가용대역 예측과 측정 기반의 수락 제어

광전송 기술과 고속 라우터 기술의 발달로 통신망의 전송 능력이 크게 향상 되었음에도 불구하고, 자원이 예약되지 않고 모든 패킷이 동등하게 처리되는 특성으로 인해 인터넷 망에서 엄격하거나 통계적인 서비스 품질(QoS: Quality of Service)이 아직 보장되지 않고 있다. 실시간 성 흐름(flow)들에 대해 서비스 품질을 보장해 주기 위해서는 망의 관측과 자원의 관리가 반드시 필요하다. 실시간 성 흐름들은 지연에 민감하기 때문에, 본 학위 논문에서는 지연을 주된 서비스 품질로써 다룬다. 지연은 많은 자원들 가운데 가용 대역에 크게 의존하기 때문에, 본 학위 논문에서는 먼저 가용 대역을 추정하는 문제를 다루고, 다음으로 추정된 가용 대역에 기반해 실시간 성 흐름들을 수락 제어하는 새로운 방법을 제안한다.

본 학위 논문은 서비스 율, 입력 트래픽의 부하가 알려지지 않은 대기 시스템의 가용 대역을 추정하는 새로운 방법을 제안한다. 최소 잔량 방법이라는 탐침 방법과 두가지 통계량(statistic)을 제안한다: 한 가지는 탐침 패킷의 지연에 기반한 통계량이고, 다른 하나는 탐침 패킷의 서비스 율에 기반한 통계량이다. 일반적인 서비스 시간을 갖는 시스템에 탐침 패킷을 제외한 데이타 패킷의 부하가 1보다 낮게 포아송 과정으로 입력될 때, 그 시스템에 최소 잔량 방법으로 탐침을 하더라도 그 시스템이 안정하다는 것을 증명하였다. 두 가지 통계량 모두 가용 대역에 대한 추정량이 될 수 있음을 보였으며, 특히 두번째 통계량은 데이타 패킷이 일반적인 과정으

로 도착하는 경우에도 가용 대역의 추정량이 됨을 보였다. 시뮬레이션을 통해 데이타 트래픽이 포아송 과정 또는 자기 유사(self-similar) 과정으로 도착하는 경우에 탐침 시간이 유한하더라도 두 통계량의 정확성이 매우 높음을 관측할 수 있다.

본 학위 논문은 앞서 단일 서버에 대해 개발된 예측 이론을 확장하여, 다중 홉 경로에 대한 가용 대역을 추정하는 새로운 방법을 제안한다. 단일 서버에서 개발된 이론을 사용하기 위해 다중 홉 경로를 고정 지연 성분과 가상 서버로 구성되는 단순화된 경로로 모델링한다. 제안되는 가용 대역 추정 방식은 가용 대역을 빨리 찾아서 가변하는 가용 대역을 계속해서 매우 근사하게 뒤따르기 때문에 짧은 시간에 가변하는 가용 대역의 범위를 알아낼 수 있다. 제안된 방식은 기존의 방식에 비해 데이타 트래픽의 가변성이 높은 환경에서 특히 더 좋은 성능을 내는 것을 시뮬레이션을 통해 확인할 수 있다.

마지막으로 실시간 성 흐름들의 지연 서비스 품질을 보장하기 위해서 새로운 서비스 구조와 수락 제어 방식을 제안한다. 기본적으로 확장성을 위해서 DiffServ 환경에서와 마찬가지로 흐름들은 진입(ingress) 라우터에서 클래스가 구별되어 출구(egress) 라우터에 도달하기까지 망 내부(core) 라우터에서는 클래스 단위로만 처리가 된다고 가정한다. 특히, 클래스는 실시간성의 높은 클래스와 비실시간성의 낮은 클래스 두 가지만 존재하는 것으로 가정하며, 실시간성 클래스에 속하는 흐름들만 수락 제어를 거치게 되고 비실시간성 클래스에 속하는 흐름들은 수락 제어를 거치지 않고 입력이 가능하다. 대신 비실시간성 흐름에 의해 실시간성 클래스의 서비스 품질이 영향을 받지 않도록 망 내부에서는 실시간성 트래픽이 비실시간성 트래픽에 비해 높은 우선 순위로 서비스 받는 것을 가정한다. 제안하는 서비스 구조에서 수락 제어는 진입 라우터에서만 이루어지기 때문에, 수락 요구에 대해 빠른 응답이 가능하고 망 내부 노드에서 각 흐름들에 대한 정보의 관리 혹은 자원의 예약을

요구하지 않기 때문에 확장성 있는 구조라고 할 수 있다. 수락 제어 알고리즘은 단순한 계산과 비교만으로 이루어지기 때문에 확장성에 있어 유리하다. 제안되는 수락 제어 알고리즘에서는 수락 제어를 판단하는 문턱값이 되는 허용 대역(admissible bandwidth: 이 값은 지연 성능을 만족시키는 한도내에서 추가로 받아들일 수 있는 최대의 대역으로 정의된다)의 계산이 가장 중요하다. 허용 대역의 추정치가 명시적으로 유도되기 때문에 각종 관련 변수들(예를 들면, 허용 지연 범위, 지연 위반 확률의 문턱값 등)의 영향을 직관적으로 알 수 있다. 허용 대역은 가용 대역(available bandwidth)의 평균과 분산에 의존해 결정되는데, 가용 대역을 탐침 방식으로 추정하기 때문에 가용 대역의 추정에 오차가 존재할 수 있다. 분산의 오차를 보상하기 위해 VMF(variance multiplication factor)를 도입하지만, 작은 VMF 값의 사용으로도 원하는 서비스 품질의 제공이 가능하며, 기본 측정 구간(T)의 길이를 조금 증가시키면 VMF를 사용하지 않고도 서비스 품질이 보장되는 것을 시뮬레이션을 통해 관측할 수 있다. 제안된 측정에 기반한 수락 제어 방식은 파라메타에 기반한 방식에 비해 훨씬 높은 자원 사용율(utilization)을 얻음과 동시에 실시간성 흐름들의 지연 서비스 품질을 만족시켜 주기 때문에 서비스 품질 제공 가능한 차세대 인터넷 구현에 적용이 될 수 있다.

References

[1] R. Branden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," IETF RFC 1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, Dec. 1998.

[3] R. Branden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) - version 1 functional specification," IETF RFC 2205, Sep. 1997.

[4] S. Berson, S. Vincent, "Aggregation of internet integrated services state," Internet Draft, draft-berson-classy-approach01.txt, 1997.

[5] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control - The single node case," in *Proc. IEEE INFOCOM '92*, pp. 915-924, 1992.

[6] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," in *Proc. IEEE INFOCOM '94*, pp. 636-646, June 1994.

[7] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks," in *Proc. ACM SIGCOMM '90*, pp. 19-29, Sep. 1990.

[8] J. C. R. Bennett and H. Zhang, "WF^2Q: worst-case fair weighted fair queueing," in *Proc. IEEE INFOCOM '96*, pp. 120-128, 1996.

[9] H. Zhang and D. Ferrari, “Rate-controlled static-priority queueing,” in *Proc. IEEE INFOCOM ’93*, pp. 227-236, April 1993.

[10] N.Yin, S.Q.Li, and T.E.Stern, ”Congestion control for packet voice by selective packet discarding,” *IEEE Trans. Commun.,*, Vol.38, No.5, pp.674-683, May 1990.

[11] A.I. Elwalid and D.Mitra, ”Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic,” *Proc. IEEE INFOCOM’92,*, pp.415-425, 1992.

[12] S.W.Min, H.Chung, and C.K. Un, ”Performance of and ATM multiplexer with selective cell discarding for on-off bursty traffics,” *IEICE Trans. Commun.,*, Vol.E78-B, No.9, pp.1253-1261, Sep. 1995,

[13] B.D.Choi and D.I.Choi, ”Discrete-time analysis of the leaky bucket scheme with threshold based token generation intervals,” *IEE Proc. Commun.,* , Vol.143, No.1, pp.5-11, 1996.

[14] S.Floyd and V.Jacobson, ”Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Networking,*, Vol.1, No.4, pp.397-413, Aug. 1993.

[15] S.Q.Li, ”Overload control in a finite message storage buffer,” *IEEE Trans. Commun.,*, Vol.37, No.12, pp.1330-1338, Dec. 1989.

[16] N. Benameur, S. B. Fredj, S. Oueslati-Boulahia, J. W. Roberts, "Quality of service and flow level admission control in the Internet," *Computer Networks*, vol. 40, pp. 57-71, 2002.

[17] A. Bosco, R. Mameli, E. Manconi, F. Ubaldi, "Edge distributed admission control," *IEEE Communications Letters*, vol. 7, no. 2, pp. 88-90, Feb. 2003.

[18] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. Mcmanus, "Requirements for traffic engineering over MPLS," IETF RFC 2702, 1999.

[19] J. Boyle *et al*, "Applicability statement for traffic engineering with MPLS," IETF RFC 3346, 2002.

[20] B. Pang, H. Shao, W. Zhu, W. Gao, "An admission control scheme to provide end-to-end statistical QoS provision in IP networks," in *Proc. IEEE International Performance, Computing, and Communications Conference*, pp. 399-403, 2002.

[21] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, H. Zhang, "Endpoint admission control: architectural issues and performance," in *Proc. SIGCOMM '00*, pp. 57-69, Aug. 2000.

[22] R. Prasad *et al*, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, pp. 27-35, Nov.-Dec. 2003.

[23] M. Mathis and M. Allman, "A Framework for defining empirical bulk transfer capacity metrics," IETF RFC 3148, July 2001.

[24] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," IETF RFC 2581, Apr. 1999.

[25] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SOSP*, 2001.

[26] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Microsoft Research Technical Report MSR-TR-2001-35*, Feb. 2001.

[27] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," in *Microsoft Research Technical Report MSR-TR-2001-35*, Feb. 2001.

[28] K. M. Hanna, N. Natarajan, and B. N. Levine, "Evaluation of a novel two-step server selection metric," in *Proc. IEEE Conference on Network Protocols (ICNP)*, Oct. 2001.

[29] V. Jacobson, "Pathchar: A tool to infer characteristics of internet paths," ftp://ftp.ee.lbl.gov/pathchar.

[30] A. Downey, "Using pathchar to estimate internet link characteristics," in *Proc. SIGCOMM'99*, Boston, MA, Aug. 1999.

[31] B. Mah, "Pchar: A tool for measuring internet path characteristics," http://www.employees.org/bmah/Software/pchar/.

[32] K. Lai and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proc. USENIX Symp. Internet Technologies and Systems*, pp. 123-134, Mar. 2001.

[33] V. Paxson, "Measurement and analysis of end-to-end internet dynamics," *Ph.D thesis*, Computer Science Division, University of California - Berkeley April 1997.

[34] A. Pásztor and D. Veitch, "On the scope of end-to-end probing methods," *IEEE Communications Letters*, vol. 6, no. 11, pp. 509-511, Nov. 2002.

[35] K. Harfoush, A. Bestavros, and J. Byers, "Measuring bottleneck bandwidth of targeted path segments" in *Proc. IEEE Infocom'03*, pp. 2079-2089, March-April 2003.

[36] Robert L. Carter and Mark E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27-28, pp. 297-318, 1996.

[37] G. Jin, G. Yang, B. Crowley, and D. Agarwal, "Network characterization service (NCS)," *Proc. IEEE Symposium on High Performance Distributed Computing*, pp. 289-299, Aug. 2001.

[38] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?," *Proc. IEEE Infocom'01*, vol. 2, pp. 905-914, Apr. 2001.

[39] V. Ribeiro *et al*, "Multifractal cross-traffic estimation," *Proc. ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, Sep. 2000.

[40] B. Melander, M. Bjorkman, and P. Gunningberg, "A new end-to-end probing and analysis method for estimating bandwidth bottlenecks," *Proc. IEEE Globecom'00*, pp. 415-421, Nov. 2000.

[41] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *Proc. ACM SIGCOMM'02*, Aug. 2002.

[42] M. Jain and C. Dovrolis, "Pathload: a measurement tool for end-to-end available bandwidth," *Proc. Passive and Active Measurements (PAM) Workshop*, Mar. 2002.

[43] V. Ribeiro *et al*, "pathChirp: efficient available bandwidth estimation for network paths," *Proc. Passive and Active Measurements Workshop*, Apr. 2003.

[44] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE J. Select. Areas Commun.*, vol. 21, no. 6, pp. 879-894, Aug. 2003.

[45] R. Guerin, H. Ahmadi, and M. Nagshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 968-981, 1991.

[46] A. Elwalid, D. Mitra, R. H. Wentworth, "A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node," *IEEE J. Select. Areas Commun.*, vol. 13, no. 6, pp. 1115-1127, Aug. 1995.

[47] P. Billingsley, *Probability and measure, 2nd Ed.* , New York: Wiley, 1986.

[48] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable bit rate video traffic," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1566-1579, 1995.

[49] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM '94*, London, England, pp. 269-280, Aug. 1994.

[50] F. P. Kelly, "Note on effective bandwidth," in *Stochastic networks: theory and applications.* ser. Royal Statistical Society Lecture Notes, F. P. Kelly, S. Zachary, and I. B. Ziedins, Eds, London, U.K.: Oxford Univ Press, 1996, vol. 4, pp. 141-168.

[51] N. G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single-server queue, with applications," *Math. Proc. Camb. Philos. Soc.*, vol. 118, pp. 363-374, 1995.

[52] J. R. Gallardo, D. Makrakis, and M. Angulo, "Dynamic resource management considering the real behavior of aggregate traffic," *IEEE Transactions on Multimedia*, vol. 3, no. 2, pp. 177-185, June 2001.

[53] J. R. Gallardo, D. Makrakis, and L. Orozco-Barbosa, "Use of alpha-stable self-similar stochastic processes for modeling traffic in broadband networks," *Perform. Eval.*, vol. 40, no. 1-3, pp. 71-98, Mar. 2000.

[54] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 56-70, Feb. 1997.

[55] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP," IETF RFC 2481, 1999.

[56] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed admission control," *IEEE J. Select. Areas Commun.*, vol. 18, no. 12, pp. 2617-2628, Dec. 2000.

[57] L. Cherkasova and P. Phaal, "Session-based admission control: a mechanism for peak load management of commercial web sites," *IEEE Transactions on Computers*, vol. 51, no. 6, pp. 669-685, June 2002.

[58] S. Crosby *et al*, "Statistical properties of a near-optimal measurement-based CAC algorithm," *Proc. IEEE ATM'97*, pp. 103-112, May 1997.

[59] Z. Dziong, M. Juda, and L. Mason, "A framework for bandwidth management in ATM networks - aggregate equivalent bandwidth estimation approach," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 134-147, Feb. 1997.

[60] S. Floyd, "Comments on measurement-based admission control for controlled-load services," Technical report, Lawrence Berkeley Laboratory, July 1997.

[61] R. Gibbens, and F. Kelly, "Measurement-based connection admission control," *15th International Teletraffic Congress*, Jun. 1997.

[62] R. J. Gibbens, F. P. Kelly, and P. B. Key, "A decision-theoretic approach to call admission control in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1101-1113, 1995.

[63] M. Grossglauser, D. Tse, J. Kurose, and D. Towsley, "A new algorithm for measurement-based admission control in integrated services packet networks," *Proc. Fifth International Workshop on Protocols for High-Speed Networks*, Antipolis, France, Oct. 1996.

[64] M. Grossglauser, and D. Tse, "A framework for robust measurement-based admission control," *Computer Communications Review*, vol. 24, pp. 237-248, Oct. 1997.

[65] M. Grossglauser, and D. Tse, "A time-scale decomposition approach to measurement-based admission control," *IEEE INFOCOM'99*, vol. 3, pp. 1539-1547, Mar. 1999.

[66] S. Jamin, S. Shenker, and P. Danzig, "Comparison of measurement-based admission control algorithms for controlled-load service," *INFOCOM'97*, pp. 973-980, Apr. 1997.

[67] M. Grossglauser, and D. Tse, "A time-scale decomposition approach to measurement-based admission control," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, , pp. 550-563, Aug. 2003.

[68] L. Breslau, S. Jamin, "Comments on the performance of measurement-based admission control algorithms," *IEEE INFOCOM'00*, vol. 3, pp. 1233-1242, Mar. 2000.

[69] C. Cetinkaya, V. Kanodia, and E. W. Knightly, "Scalable services via egress admission control," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 69-81, Mar. 2001.

[70] Jing-yu Qiu and E. W. Knightly, "Inter-class resource sharing using statistical service envelopes," in *Proc. IEEE INFOCOM '99*, New York, pp. 1404-1411, Mar. 1999.

[71] R. W. Wolf, *Stochastic modeling and the theory of queues*, Prentice Hall, 1988.

[72] A. G. Pakes, "Some conditions for ergodicity and recurrence of Markov chains," *Oper. Res.*, vol. 17, pp. 1058-1061, 1969.

[73] K. L. Chung, *A course in probability theory, 2nd ed.,* Academic Press, 1974.

[74] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, pp. 835-846, 1997.

[75] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1-15, 1994.

[76] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226-244, 1995.

[77] R. H. Riedi, M. S. Course, V. J. Ribeiro, and R. G. Baranuik, "A multifractal wavelet model with application to network traffic," *IEEE Tran. Information Theory*, vol. 45, pp. 992-1018, 1999.

[78] A. Papoulis, *Probability, random variables, and stochastic processes, 3rd Ed.* , New York: McGraw-Hill, 1991.

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

[80] S. Karlin and H. M. Taylor, *A first course in stochastic processes, 2nd ed.,* Academic Press, 1975.

[81] One-way transmission time, ITU-T Recommendation G.114, May 2000.

[82] A. V. Moffaert *et al*, "Tuning the VoIP gateways to transport international voice calls over a best effort IP backbone," *9th IFIP Conference on Performance Modeling and Evaluation of ATM* & *IP Networks*, Budapest, June 2001.

[83] 3GPP Technical Specification 22.105, Services and service capabilities.

[84] 3GPP Technical Specification 23.107, Quality of Service (QoS) concept and architecture.

감 사 의 글

많은 분들의 관심과 도움으로 오늘의 이 작은 결실이 있게 되었습니다.

이제까지의 오랜 시간동안 저에게 큰 모범이셨으며, 다양한 연구 주제를 접할 수 있는 기회와 통찰력 있는 논문 지도 뿐만 아니라 그 이상으로 많은 것을 가르쳐 주시고 힘이 되어주신 성단근 교수님께 진심으로 감사드립니다. 또한 바쁘신 와중에 귀한 시간을 내셔서 저의 논문 심사를 기꺼이 수락해 주시고 조언을 주신 이황수 교수님, 새로운 문제 제기 뿐만 아니라 새로운 해결책까지 제안해 주신 정송 교수님, 수학적인 전개 부분을 친절하게 검토하고 조언해 주신 황강욱 교수님, 현실적인 관점에서 논문을 평가하고 조언해 주신 이형호 박사님께 진심으로 감사드립니다. 그리고, 학위 논문 계획서를 심사해 주신 조동호 교수님께 감사드립니다.

통신망 연구실에 들어 와서 인연을 맺게 된 선배님들과 후배님들의 실험실 생활과 연구에 관련된 많은 도움 뿐만 아니라 여러분들께서 보여주신 따뜻한 마음이 저에게 큰 힘이 되어서 이렇게 졸업이 가능하게 된 것이라 생각합니다. 다시 한번 모든 분들께 고마운 마음을 전하며 이렇게 맺어진 좋은 인연이 평생을 같이 할 것을 믿습니다.

이 논문이 나오기까지 저의 문제를 함께 깊이 고민해 주시고, 논문의 수준과 완성도를 한차원 높여주실 뿐만 아니라 저의 수학에 대한 갈증을 풀어주신 김성곤 박사님께도 깊은 감사를 드립니다.

어떻게 보면 길다고도 할 수 있는 7년 반동안의 연구실 생활이 전혀 지루하지 않았던 것은 이렇게 좋은 분들을 만나 같이 지낼 수 있었기 때문이라고 생각합니다. 특히 이미 졸업하신 손규식 교수님, 이주용 박사님, 김성곤 박사님, 그리고 권태

정 선배님과 보냈던 좋은 시간은 저에게 많은 힘이 되었었고 오랫동안 잊을 수 없을 것입니다.

학부때부터 줄곧 룸메이트로 동고동락해온 평생지기 친구인 관희와, 지난학기 졸업한 준철이, 이번 학기에 졸업하는 관본이에게도 졸업 축하와 함께 고맙다는 말을 전하고 싶습니다. 고등학교 시절부터 기쁠 때를 함께하고 어려울 때 힘이 되어준 보수잔당 친구들에게도 감사의 마음을 전합니다. 자주 만나지는 못하지만 멀리서 힘이 되어주는 초등학교 시절 친구들에게도 고마움을 전합니다.

제가 이 세상에서 가장 사랑하는 우리 가족, 특히 오늘이 있기까지 옆에서 늘 한결같은 마음으로 지켜봐 주시면서 지원을 아끼지 않으시는 어머니, 늘 힘이 되어주는 누나, 매형, 재현이, 재민이와 이 기쁨을 함께하고 싶습니다. 그리고, 마음으로 늘 응원해 주시던 큰 삼촌, 작은 삼촌, 큰 고모, 작은 고모께도 감사하다는 말을 전하고 싶습니다. 마지막으로 하늘에 계시는, 제가 가장 존경하는 아버지께 감사의 마음을 전하며 이 논문을 바칩니다.

Curriculum Vitae

Name : Seung Yeob Nam

Date of Birth : August 21, 1975

Birthplace : Nam-gu, Daegu

Domicile : Uiseong-gun, Gyeongsangbuk-do

E-mail : synam@cnr.kaist.ac.kr

Educations

1993. 3. – 1997. 2. Department of Electrical Engineering, KAIST (B.S.) (Magna Cum Laude)

1997. 3. – 1999. 2. Department of Electrical Engineering, KAIST (M.S.)

1999. 3. – 2004. 8. Department of Electrical Engineering and Computer Science, KAIST (Ph.D.)

Awards

1. Korea Mathematics Competition for Undergraduates, Fourth Prize, Nov. 1995.
2. Best Paper Award, The 6th Asia-Pacific Conference on Communications (APCC), 2000.
3. Bronze Prize, Samsung Humantech Thesis Prize, Feb. 2004.

Publications

International Journal

1. **Seung Y. Nam** and Dan K. Sung, "Fast Convolution Approximation Scheme for Estimating end-to-end Delay Performance," *Electronics Letters*, vol. 36, no. 16, pp. 1432-1434, Aug. 2000.

2. Jae Yoon Park, **Seung Yeob Nam**, and Dan Keun Sung, "Performance Analysis of an Enhanced DQRUMA/MC-CDMA Protocol for Voice Traffic," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 2, pp. 440-455, Mar. 2003.

3. Sun Jong Kwon, **Seung Yeob Nam**, Ho Young Hwang, and Dan Keun Sung, "Analysis of a Mobility Management Scheme Considering Battery Power Conservation in IP-Based Mobile Networks," accepted for publication in *IEEE Transactions on Vehicular Technology.*

4. Kyu-Seek Sohn, **Seung Yeob Nam**, and Dan Keun Sung, "A Spare Bandwidth Sharing Scheme Based on Network Reliability," accepted for publication in *IEEE Transactions on Reliability.*

5. Kyu-Seek Sohn, **Seung Yeob Nam**, and Dan Keun Sung, "A Distributed LSP Scheme to Reduce Spare Bandwidth Demand in MPLS Networks," submitted in *IEEE Transactions on Communications.*

6. Sunggon Kim, **Seung Yeob Nam**, and Dan Keun Sung, "Effective Bandwidth for a Single Server Queueing System with Fractional Brownian Input," submitted in *Performance Evaluation*, (conditionally accepted).

7. **Seung Yeob Nam**, Sunggon Kim, and Dan Keun Sung, "Probing-Based Estimation of End-to-End Available Bandwidth," accepted for publication in *IEEE Communications Letters.*

8. **Seung Yeob Nam**, Sunggon Kim, and Dan Keun Sung, "Estimation of Available Bandwidth for an Unidentified Queueing System," submitted in *IEEE Transactions on Communications.*

9. **Seung Yeob Nam**, Sunggon Kim, and Dan Keun Sung, "Measurement-Based Edge Router Admission Control," (in preparation).

International Conference

1. **Seung Yeob Nam**, Joo Yong Lee, Dan Keun Sung, and Soo Jong Lee, "Delay Estimation through Monitoring the Address Buffers of a Shared Buffer Type ATM Switch," *in Proc. 13th Int. Conference on Information Networking (ICOIN 13)*, Jeju, Korea, pp. 1c2.1-2.6, Jan. 1999.

2. Jae Yoon Park, **Seung Yeob Nam**, Dan Keun Sung, and Seung Hwan Kim, "Performance Analysis of a Modified DQRUMA/MC-CDMA Protocol for Voice Traffic," *in Proc. APCC 2000*, Seoul, Korea, pp. 744-748, Oct.-Nov. 2000. This paper was selected as **the Best Paper for APCC 2000**.

3. **Seung Yeob Nam** and Dan Keun Sung, "Measurement-Based Delay Performance Estimation in ATM Networks," *in Proc. IEEE Globecom 2000*, San Francisco, USA, pp. 1766-1770, Nov. 2000.

4. **Seung Yeob Nam**, Kyu-Seek Sohn, and Dan Keun Sung, "Quasi-Shared Output Buffer Switch," *in Proc. APCC 2001*, Tokyo, Japan, pp. 85-88, Sep. 2001.

5. **Seung Yeob Nam** and Dan Keun Sung, "Decomposed Crossbar Switches with Multiple Input and Output Buffers," *in Proc. IEEE Globecom 2001*, San Antonio, USA, pp. 2661-2665, Nov. 2001.

6. Jae Yoon Park, **Seung Yeob Nam**, and Dan Keun Sung, "Performance Analysis of an Enhanced DQRUMA/MC-CDMA Protocol with an LPRA Scheme for Voice Traffic," *in Proc. IEEE WCNC 2002*, Orlando, USA, pp. 743-748, Mar. 2002.

7. **Seung Yeob Nam** and Dan Keun Sung, "Desynchronized Two-Dimensional Round-Robin Scheduler for Input Buffered Switches," *in Proc. APCC 2002*, Bandung, Indonesia, pp. 438-441, Sep. 2002.

8. Jong Arm Jun, Sung Hyuk Byun, Byung Jun Ahn, **Seung Yeob Nam**, and Dan Keun Sung, "Two-Dimensional Crossbar Matrix Switch Architecture," *in Proc. APCC 2002*, Indonesia, pp. 411-415, Sep. 2002.

9. Sun Jong Kwon, **Seung Yeob Nam**, Ho Young Hwang, and Dan Keun Sung, "Analysis of a mobility considering battery power conservation in IP-based mobile networks," *in Proc. APCC 2002*, Indonesia, pp. 1-4, Sep. 2002.

10. Jong Arm Jun, Sung Hyuk Byun, Byung Jun Ahn, **Seung Yeob Nam**, and Dan Keun Sung, "A Two-Dimensional Scalable Crossbar Matrix Switch Architecture," *in Proc. IEEE ICC 2003*, pp. 1892-1896, Alaska, USA, May 2003.

Domestic Conference

1. J.W.Yang, S.H.Byun, J.W.Heo, J.Y.Lee, **S.Y.Nam**, and D.K.Sung, "FPGA Implementation of a Multistage Interconnection Network Switch Module," *in*

Proc. JCCI '98, Chonju, Korea, pp. 448-452, Apr. 1998.

2. **Seung Yeob Nam**, Dan Keun Sung, and Yoon-Young An, "Quasi-Shared Output Buffered Switch," *in Proc. IEEK Fall Conference 2000*, pp. 283-286, Nov. 2000.

3. **Seung Yeob Nam** and Dan Keun Sung, "Decomposed Crossbar Switches with Multiple Input and Output Buffers," *in Proc. ITRC Forum 2001*, Seoul, Korea, pp. B1-14 - B1-18, May 2001.

4. Sunggon Kim, **Seung Yeob Nam**, Jun-Su Kim, and Dan Keun Sung, "Effective Bandwidth Theory for Fractional Brownian Input," *The 9th Applied Mathematics Forum*, Busan, Korea, June 2002.

5. Junsu Kim, **Seung Yeob Nam**, Sunggon Kim, and Dan Keun Sung, "Estimation Algorithm of End-to-End Delay Distribution in a DiffServ Network," *in Proc. JCCI 2003*, Anmyeondo, Korea, pp. IV-A-3.1 4, May 2003.

6. Sunggon Kim, **Seung Yeob Nam**, and Dan Keun Sung, "Effective Bandwidth for a Single Server Queueing System with Fractional Brownian Input," *in Proc. KSS (Korean Statistical Society) fall conference 2003*, Oct.-Nov. 2003.

Patents

1. "A Fast Convolution Approximation Scheme for Estimating End-to-End Delay Performance," D.K.Sung, **S.Y.Nam**, no. 342434, Korea, 2002.

2. "A Fast Convolution Approximation Scheme for Estimation of End-to-End Delay," D.K.Sung, **S.Y.Nam**, 10-2000-397873(pending), Japan, 2000.

3. "Quasi-Shared Output Buffer Type Switching Apparatus," Y.Y.An, **S.Y.Nam**, D.K.Sung, K.S.Sohn, 10-2001-0058504(pending), Korea, 2001.

4. "Partitioned Crossbar Switch with Multiple Input/Output Buffers," D.K.Sung, **S.Y.Nam**, no. 404376, Korea, 2003.

5. "Method and Apparatus for Two-Dimensional-Scalable Crossbar Matrix Switch," J.A.Jun, D.K.Sung, **S.Y.Nam**, 10-2003-0098348(pending), Korea, 2003.

6. "Method and Apparatus for Two-Dimensional-Scalable Crossbar Matrix Switch," J.A.Jun, D.K.Sung, **S.Y.Nam**, 10/663476(pending), USA, 2003.