Differentiated Services with Statistical QoS Guarantees in Static-Priority Scheduling Networks

Shengquan Wang, Dong Xuan, Riccardo Bettati, and Wei Zhao

Abstract— **In this paper, we propose and analyze a methodology for providing statistical guarantees within the diffserv model in a network, that uses static-priority schedulers. We extend the previous work on statistical delay analysis and develop a method that can be used to derive delay bounds without specific information on flow population. With this new method, we are able to successfully employ a utilization-based admission control approach for flow admission. This approach does not require explicit delay computation at admission time and hence is scalable to large systems. To the best of our knowledge, this is the first work in the area to provide statistical end-to-end guarantees within the diffserv model in the multi-node network. We systematically analyze the performance of our approaches in terms of system utilization. As expected, our experimental data show that statistical services can achieve much higher utilization than deterministic services.**

Keywords—**differentiated services, statistical service, real-time application, static-priority scheduling, utilization-based admission control.**

I. Introduction

In this paper, we propose and analyze a methodology for providing differentiated services with statistical real-time guarantees in networks that use static-priority schedulers. The problem we address has three dimensions as shown in Fig. 1: *Quality of Service*, *service model*, and *system complexity*, and solutions which have been proposed for different vertices within this space:

A. Quality of Service: Deterministic guarantees vs statistical guarantees

Traditionally, *best-effort* service has been the main type of service available over internetworks. While this type of service has contributed much towards the rapid growth of the Internet, it cannot effectively support applications that have real-time requirements. Many real-time applications, e.g., Voice over IP, DoD's C4I, and industrial control systems, are delay sensitive. In this context, by *real time* we mean that a packet is required to be delivered from its source to the destination within a predefined end-to-end deadline.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: {swang, dxuan, bettati, zhao}@cs.tamu.edu .

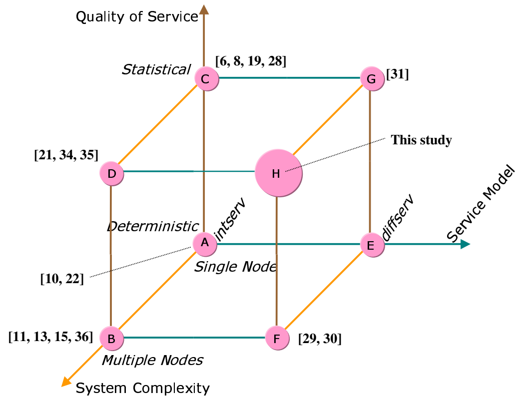

Fig. 1. Problem Space and Related Work

While deterministic services provide a simple model to the applications, they tend to heavily over-commit resources because they account for the worst-case scenario. In practical systems, this likely results in significant portions of network resources (bandwidth, etc.) being wasted [32]. *Statistical services*, on the other hand, significantly increase the efficiency of network use by allowing increased statistical multiplexing of the underlying network resources. This comes at the expense of packets occasionally being dropped or excessively delayed.

B. Service model: Integrated Services vs Differentiated Services

One way to provide guaranteed services is to control input traffic by admission control and policing, and to schedule packets based on the requirements of the connection they belong to. A systematic approach based on connection admission control and packet scheduling was proposed within the Integrated Services (`intserv`) architecture of the IETF [7]. In Integrated Services, connections are controlled both by admission control at connection establishment time and packet scheduling during the lifetime of the connections.

Integrated services are difficult to deploy in large-scale high-speed internetworks, as they do not scale well, for two reasons: first, routers are required to maintain status information of flows and schedule packets for large numbers of flows. Second, as the number of flows increases, the run-time overhead incurred for flow establishment and tear-down increases as well.

This lack of scalability is, to a large extent, being addressed within the Differentiated Services (`diffserv`) architecture [4], [5], [25]. From the user's point of view, the `diffserv` model partitions user-level *flows*[1] into a set of predefined *classes*. Packets of each class are served in the network according to a class-based scheduling policy. The result is that routers are aware only of *aggregations* of flows [2], which are formed from the individual flows at the edge routers. In this fashion, the `diffserv` model makes the network scalable regardless of the number of flows.

C. System complexity: single-node network vs multi-node network

Many projects usually start investigation with single-node networks. This helps to develop the basic concept model. In reality, most of networks are multi-node networks. Study on a multi-node network is more difficult than single-node network, as concepts and methodologies developed in the single-node case usually cannot be easily extended to that of the multi-node case.

Studies on providing guaranteed services illustrate this. Utilization-Based Admission Control (UBAC) is a scalable mechanism to provide guaranteed services. This mechanism defines a utilization bound below which all the workload using the resource is guaranteed to meet its deadline. UBAC was first proposed in [22] for preemptive scheduling of periodic tasks on a single server. Then the concept was applied to FDDI networks for scheduling synchronous traffic [1]. Most recently, the idea was used in general multi-node networks in the `diffserv` setting for providing deterministic guarantees [30]. In these studies, although the idea of UBAC remains the same, the approaches taken are different. It is much more difficult to derive the utilization bounds in the multi-node network than in the single-node network (the single server).

Extensive studies have been carried out for various problems in the problem space shown in Fig. 1. For example, for *Vertex A* which represents the case of providing deterministic services under the `intserv` model in single-node networks, various solutions have been reported early on [10] and [22]. The case of deterministic services within the `intserv` model in multi-node networks, represented by *Vertex B* in Fig. 1, has been covered in [11], [13], [15], [36]. A number of projects ([6], [8], [19], [28]) address the problem associated with *Vertex C* in Fig. 1, while studies of [21], [34], [35] are for the case of *Vertex D*. For the cases related to *Vertices F*, solutions have been discussed in [29], [30]. The problems associated with statistical services within the `diffserv` model seem to be particularly challenging. However, a preliminary study for single-node networks has been previously reported in [31]. The work reported in this paper is associated with *Vertices H*.

In this paper, we propose and analyze a methodology for providing statistical guarantees within the `diffserv` model in multi-node networks. We use Utilization-Based Admission Control (UBAC) to perform scalable admission control. The challenge in using the UBAC method is how to verify the correctness of the utilization bound at the configuration time. The verification has to rely in a delay analysis method. We derive flow-unaware statistical delay formula, which allows us to analyze delays without depending on the dynamic information about flows. We systematically analyze the system performance of our approaches in terms of utilization. As expected, our experimental data show that statistical services can achieve much higher utilization than deterministic services.

The rest of the paper is organized as follows. In Section II, we discuss previous related work. The underlying network and traffic models for this study are introduced in Section III. In Section IV, we introduce our proposed architecture for providing statistical services in networks with static-priority scheduling. In Section V, we develop a delay analysis methodology that is insensitive to flow information. In Section VI, we provide extensive experimental data to illustrate that the utilization achieved within a statistical model is much higher than that of a deterministic model. A summary of the paper is given in Section VII.

II. Previous Works

In this section, we will briefly review the previous work on the problems which cover the three dimenssions in Fig. 1.

A. Providing deterministic services within the `intserv` model (Vertex A and Vertex B)

A lot of studies have been conducted on providing deterministic services within the `intserv` model both in single-node [22], [10] and multi-node [11], [15], [13], [36] networks. Under the `IntServ` model, the work on the single-node network can be relatively easily extended to the multi-node scenario. The main approach is to shape traffic flows at each hops in the network [15]. The nature of awareness about per-flow information under `intserv` model allows this kind of per-flow shaping.

[1] In the following, we will use the term flow to indicate a stream of data between a source and a destination, and the term connection to indicate the virtual circuit that needs to be established to carry the flow.

[2] In this paper we use aggregated flow and class interchangeably.

The work in this area mainly focuses on admission control and packet scheduling. With certain traffic models, a number of end-to-end delay calculation methods have been proposed for a large variety of packet scheduling mechanisms [2], [3], [13], [16], [26], [36]. Finally, admission control has also been investigated [12], [14].

One common limitation of all these systems is their scalability. While they provide deterministic services, they are not scalable to a large internetwork due to their run-time overhead in admission control and packet scheduling.

B. Providing deterministic services whithin the `diffserv` model (Vertex E and Vertex F)

Supporting deterministic services within the `diffserv` model has been addressed in [29] and [30].

Stoica and Zhang [29] suggested to move per-flow information from the core routers to the edge of the network by using packets to carry the flow state information. The result is that the core router doesnot need to maintain per flow information anymore but infers it based on the information carried with the packet. The expense is additional scheduling overhead at routers and a degree of incompatibility with the traditional IP protocol.

In [30], we proposed and analyzed a methodology for providing derterministic services within the diffserv model. We developed a method that can be used to derive delay bounds without specific information on flow population. With this new method, we are able to successfully employ a utilization-based admission control approach for flow admission. This approach does not require explicit delay computation at admission time and hence is scalable to large systems.

C. Providing statistical services within the `intserv` model (Vertex C and Vertex D)

Much progress has been made in providing statistical services under the `intserv` model both in the single-node [6], [8], [19], [28] and multiple-node [21], [34], [35] cases. Most of these studies focus on a single-node first, and then extend to the multi-node scenorio [19], [21].

Statistical service has been investigated using envelopes of bounding moment generating functions [8], exponentially bounded envelopes [34], and envelopes consisting of families of bounding distributions [6]. Statistical envelopes have also been applied to resource allocation for inter-class resource sharing [28]. In [19], a *rate-variance envelope* is introduced, which describes the variance of the arrivals of a flow as a function of a time period of length. In [17], arrivals on a flow are assumed to be characterized by the rate-variance envelope and a long-term arrival rate. Then, applying the a central limit theorem argument, a bound for the probability of a delay bound violation is derived for a static priority scheduler. In [19], the same framework is used to address bounds on the rate-variance envelope for regulated, adversarial traffic sources.

Few have studied on providing statistical services within the `diffserv` model. [31] conducts a preliminary study for a single-node case. Up to date, there is no study reported to address the issue in the context of the multi-node scenerio, which this paper aims to address.

III. Network and Traffic Models

Our assumptions about network nodes and network traffic follow the Differentiated Services architecture: In the network, we differentiate flow-aware *edge routers* from *core routers*, which are only aware of aggregations of flows in form of *flow classes*. The network traffic consists of *flows*. We limit our solution to the system with two flow classes. Each flow belongs to one of two flow classes: (1) high-priority class with deadline constraints, i.e., real-time class and (2) a low priority, best-effort class. We proceed below to describe our models in detail.

A. Network Model

The IETF `diffserv` architecture distinguishes two types of routers: *Edge routers* are located at the boundary of the network, and provide support for traffic policing. *Core routers* are within the network, and maintain no flow information. Scheduling decisions within the core are therefore made based on class information only. Routers are connected through *links*, which we assume – for simplicity of argument – to all be of capacity C, in bits per second.

When we compute delays, we follow standard practice and model a router as a set of *servers*, one for each router component where packets can experience delays. Packets are typically queued at some of these servers, typically at the output buffers, where they compete for the output link. Hence, we model a router as a set of output *link servers*. The other servers (input buffers, non-blocking switch fabric, wires, etc.) are typically not congested, and therefore can be eliminated by appropriately subtracting their constant delays from the deadline requirements of the traffic.

We assume that the schedulers in the nodes are not flow aware. Policing and scheduling therefore happen at class level, with class-based static priority schedulers assumed in this paper. Packets belonging to the class with higher priority will be served prior to those with lower priorities.

B. Traffic Model

We call a stream of packets between a sender and receiver a *flow*. Packets of a flow are transmitted along a single path, which we model as a list of link servers. Following the `diffserv` architecture, QoS requirements and traffic specifications of flows are defined on a class-by-class basis [3]. We assume that at each link server, a certain percentage of bandwidth is reserved for each particular traffic class. Let α_k denote the percentage of bandwidth reserved for real-time traffic at Server k. It is the responsibility of the admission control module to ensure that the bandwidth usage of individual classes does not exceed the reserved portion. This is necessary to provide isolation among classes and hence to guarantee end-to-end delays to the flows in each class.

We model the traffic arrival for a flow as a stochastic arrival process $A = \{A(t), t \geq 0\}$, where random variable $A(t)$ denotes the incoming traffic amount of flows for a link server during time interval $[0, t]$. The arrival process A is stationary and ergodic. Assume that the traffic arrivals for any two different input links are stochastically independent. The traffic arrival can be bounded either deterministically or stochastically by the arrival traffic envelopes as follows:

Definition 1 (Arrival Traffic Envelope) Consider a traffic arrival process $A = \{A(t), t \geq 0\}$, where random variable $A(t)$ denotes any incoming flow amount for a link server during time interval $[0, t]$.

- The function $b(t)$ is called the deterministic arrival traffic envelope [10] of the random variable $A(t)$ if

$$A(\tau + t) - A(\tau) \leq b(t), \tag{1}$$

for any $\tau > 0$ and $t > 0$;

- The distribution $B(t)$ forms the stochastic arrival traffic envelope [20] of the random variable $A(t)$ if [4]

$$A(\tau + t) - A(\tau) \leq_{st} B(t), \tag{2}$$

for any $\tau > 0$ and $t > 0$.

A possesses independent and stationary increments, i.e., for any τ, t, the random variables $A(\tau + t) - A(\tau)$ for different τ are mutually independent and possesses the same probability distributions. We use $\mu(t)$ and $RV(t)$ to describe the mean rate value and rate-variance of $\frac{A(\tau+t)-A(\tau)}{t}$, respectively.

[3] While at network level all flows within the real-time class have identical traffic specifications and QoS requirements, user-level connections with varying bandwidth and burstiness requirements can be established by using more than one flow.

[4] $X \leq_{st} Y$ means $P\{X > Z\} \leq P\{Y > Z\}$.

We assume that a real-time traffic flow is controlled at its source router by a leaky bucket with burst size σ and average rate ρ.

IV. A QoS Architecture

In this section, we introduce an architecture which is used to provide statistical guaranteed services under the diffserv model in static priority scheduling networks. This architecture consists of three major modules:

1. *Statistical flow-population-insensitive delay computation and utilization bound verification:* This module is invoked at configuration time. An ingenious flow-population-insensitive delay method is used to estimate the delay upper bound for every class at each router. This module verifies whether the end-to-end delay bound in each feasible path of the network satisfies the deadline requirement as long as the bandwidth usage on the path is within a pre-defined limit, i.e., the utilization bound.
2. *Efficient admission control:* Utilization-based admission control is adopted. As the delay has been verified at configuration time, this module checks only if the bandwidth is available along the path of the new flow.
3. *Packet forwarding control:* In a router, packets are transmitted according to their priorities which are marked in their header. Within the same priority, packets are served in FIFO order.

Among the above three modules, the first module, statistical flow-population-insensitive delay computation and utilization bound verification, is of most importance. Utilization-based admission control works around a utilization bound. The correctness of this utilization bound need to be verified at configuration time. We propose to realize this critical function in three steps:

1. We first obtain a general delay formula which depends on dynamic flow information.
2. We then remove the dependence in the delay formula to get a statistical flow-population-insensitive delay formula.
3. Finally, we verify the correctness of the utilization bound by applying the statistical flow-population-insensitive delay formula,.

Among the above three steps, Step 3 is relatively straightforward once we have the statistical flow-population-insensitive delay formula. We can apply the delay formula to check, under the given the utilization bound, whether the end-to-end delay bound in each feasible path satisfies the deadline requirement by the given probability.

In many cases, application system designers would like to know the *Maximum Usage Utilization* (MUU). By the *Maximum Usage Utilization*, we mean the maximum value of the utilization level, under which the end-to-end delay bound in each feasible path of the network satisfies the deadline requirement by the given probability. The value of MUU will be used to evaluate the efficiency of the system which employs the UBAC method. The MUU can be obtained by simply applying the delay formula and a binary search algorithm.

Accordingly, in the rest of this paper, we will focus on Step 1 and Step 2. Before we proceed, we would like to address a related issue. While utilization-based admission control significantly reduces the admission control overhead, excessive connection establishment activity can still add substantial strain to the admission control components. In [9], we describe ways to distribute the load for admission control by appropriately pre-allocating resources. For sake of space limitation, this paper will not address this issue further.

V. Statistical Flow-Population-Insensitive Delay Computation

A statistical delay guarantee can be defined as a bound on the probability of the suffered delay by a packet exceeding a deadline as follows:

$$P\{D > d\} \leq \epsilon, \tag{3}$$

where the delay D suffered by a packet is a random variable, d is the given deadline, and ϵ is the given violation probability, which is generally small.

We take two steps to perform statistical delay analysis: *Per-hop delay analysis* and *End-to-end delay analysis*. The per-hop delay analysis cover the servers at edge routers and core routers. The per-hop delay will then be used to compute to the end-to-end delay.

Recall that we limit our solution to systems with two flow classes. We conveniently omit the class index in the following description. Thus, unless otherwise pointed out, all the traffic parameters are referred to real-time traffic class.

A. Per-hop Delay Analysis

In this section, we will present a new delay computation formula, which is insensitive to flow population. We then provide the details of the approach with which the delay formula is derived [5].

A.1 Main Result

Since static priority scheduling does not provide flow separation, the local delay at a server depends on information (number and traffic characteristics) of other flows both at the server under consideration and at servers upstream. Therefore, all the flows *currently* established in the network must be known in order to compute delays. While some formulas could be used (at quite some expense) for flow establishment at system run time, they are not applicable for delay computation during configuration time, as they rely on information about flow population.

Fortunately, the following theorem gives a delay violation probability formula without information about flow population.

Theorem 1: Consider a static-priority scheduler with link capacity C such that the real-time traffic at a given server has an associated deadline d, the delay violation probability for a random real-time packet at this server is bounded by

$$P\{D > d\} \leq 1 - \Phi\left(\min_{s_0 \leq s \leq s_1} \frac{\frac{d}{\alpha}\frac{r-1}{\frac{\sigma}{\rho}+Y} + \frac{1-\alpha}{\alpha}}{\sqrt{\frac{1}{s}(r^2 - 2\,r) + r - 1}}\right), \tag{4}$$

where

$$\Phi(a) = \int_{-\infty}^{a} e^{-\frac{x^2}{2}} dx, \tag{5}$$

$$r = \frac{1}{\frac{1}{1-e^{-s}} - \frac{1}{s}}, \tag{6}$$

and Y is the maximum worst-case delay suffered by the traffic before arriving at the current server. It can be computed by a method in [30]. $s_0 = s(\tau)$, $s_1 = s(\beta)$, where the function $s = s(t)$ can be determined by $\frac{1}{\frac{1}{1-e^{-s}} - \frac{1}{s}} = \frac{1}{t}(\frac{\sigma}{\rho} + Y)$, $\tau = \frac{\frac{\sigma}{\rho}+Y}{\frac{C}{\rho}-1}$ and $\beta = \frac{\alpha}{1-\alpha}(\frac{\sigma}{\rho} + Y)$.

Derivation of (4) will be discussed in the following subsection. At this point, we would like to make the following observations on *Theorem 1*:

- Usually a delay violation probability for a server would depend on the state of the server, i.e., the number of flows that are admitted and pass through the server. We note that (4) is independent from this kind of information, and just depends on σ, ρ, α, C, and Y. The values of these parameters are available at the time when the system is (re-)configured. Hence, the delay computation formula is insensitive to the flow population information.
- The right-hand side of (4) can be computed numerically. It is a *minimum* optimization problem for an explicit function.
- We define $\frac{\sigma}{\rho}$ as *relative burstiness*. The delay formula depends on the relative burstiness $\frac{\sigma}{\rho}$.

A.2 Deriving Formula (4)

In this subsection, we discuss how to derive the formula given in (4). We will start with a formula for delay that de-

[5] Since the per-hop delay analysis focuses on one server, without any confusion, in this subsection, we'll omit server index k.

pends on flow population, which we call *the general delay violation probability formula*. We will then describe how to remove its dependency on information of flow population.

The *Central Limit Theorem* states that the summation of a set of independent random variables converges in distribution to a random variable that has a *Normal Distribution*. The following lemma is based on this theorem [19]. Ihe general delay violation probability formula is given out by this lemma.

Lemma 1: Consider a static priority scheduler with link capacity C such that the real-time traffic at a given server has an associated deadline d. Suppose that the traffic process into the server consists of independent traffic processes A_j, which have stochastic envelope $B_j(t)$ with mean rate $\mu_j(t)$ and rate-variance envelope $RV_j(t)$. The delay violation probability for a random real-time packet is approximately bounded by

$$\begin{aligned} \Pr\{D > d\} &\leq \max_{t<\beta} \Pr\{\frac{1}{C}\sum_j B_j(t) - t \geq d\}, \quad (7) \\ &\leq 1 - \Phi(\min_{t<\beta} \frac{C(t+d) - t\sum_j \mu_j(t)}{\sqrt{t^2 \sum_j RV_j(t)}}) \quad (8) \end{aligned}$$

where β is the busy period bound for the real-time traffic at this server and is given by

$$\beta = \min\{t : \sum_j b_j(t) \leq Ct,\ t > 0\}, \tag{9}$$

and $\Phi(a)$ is defined in (5).

For this lemma, we make the following observations:

- All A_j's need to be independent. In this paper, we consider the group of flows from the same input link as one traffic process. In the following, we will use the notation G_j to denote a group of flows from Input Link j and use $b_j(t)$, $B_j(t)$, $\mu_j(t)$, and $RV_j(t)$ to specify the deterministic arrival traffic envelope, the stochastic arrival traffic envelope, and the rate-variance envelope applied to the group of flows G_j respectively. Since these traffic processes from different input links, they can be assumed to be independent.
- The general delay violation probability formula depends on the flow information, i.e., the individual mean rate $\mu_j(t)$, rate-variance $RV_j(t)$, and the deterministic envelope $b_j(t)$.

In the following, we will describe methods how to compute these parameters and remove the dependency of the delay formula on the flow information.

A.2.a Computation of Mean Rate, Rate-Variance, and Deterministic Envelope.

- *Mean Rate $\mu_j(t)$:* With the ergodicity of traffic processes, the mean rate can be computed by the following lemma:

Lemma 2: Given the deterministic arrival traffic envelope $b_j(t)$ of the traffic process A_j, if A_j is ergodic and its long-term average rate $\rho_j = \lim_{t\to\infty} \frac{b_j(t)}{t}$, then the mean rate $\mu_j(t)$ for the random variable $A_j(t)$ can be described as follows:

$$\mu_j(t) = \rho_j. \tag{10}$$

The proof of *Lemma 2* is given in APPENDIX A.

- *Rate Variance $RV_j(t)$:* Based on the maximum entropy principle, we have the following result:

Lemma 3: Given the deterministic arrival traffic envelope $b_j(t)$ of the traffic process A_j with mean rate ρ_j, the maximum entropy estimate of rate-variance is

$$RV_j(t) = \omega_j(t)\rho_j^2, \tag{11}$$

where

$$\omega_j(t) = \frac{1}{s_j}(r_j^2 - 2\,r_j) + r_j - 1, \tag{12}$$

s_j can be obtained from the following equation (inverse function) [6]

$$r_j = \frac{1}{\frac{1}{1-e^{-s_j}} - \frac{1}{s_j}}, \tag{13}$$

and

$$r_j = \frac{b_j(t)}{\rho_j t}. \tag{14}$$

The proof of *Lemma 3* is given in APPENDIX B.

- *Deterministic Envelope $b_j(t)$:* From the above lemmas, we note that $\mu_j(t)$ and $RV_j(t)$ are determined by the deterministic envelope $b_j(t)$. Recall that we assume the source traffic of a flow is controlled by a leaky bucket with burst size σ and average rate ρ. Hence at the first server (usually in some edge router), $b_j(t)$ can be obtained. At an intermediate server (usually in some core router), the traffic will be distorted after traffic aggregation, we need to redetermine the deterministic envelope $b_j(t)$. Our approach is to use the worst-case delay to bound the deterministic envelope at intermediate servers. For group of flows G_j with n_j flows, let Y_j be the maximum worst-case delay suffered by traffic through Input Link j before arriving at the current server. Then the deterministic envelope $b_j(t)$ is given as follows:

$$b_j(t) = n_j \min\{Ct, \sigma + \rho(t + Y_j)\}. \tag{15}$$

[6] Note that s_j depends on t.

A.2.b Removing the Dependency on the Flow Information.

The per-hop delay violation probability formula depends on the flow population n_j and Y_j. In the following, we want to remove the specific input link j and the flow population from this formula.

• *Removing the Dependency on Individual Input Link:* Recall that Y is the maximum worst-case delay suffered by traffic before arriving at the current server, then the deterministic envelope $b_j(t)$ is given as follows:

$$\begin{aligned} b_j(t) &= n_j \min\{Ct, \sigma + \rho(t + Y_j)\} \\ &\leq n_j \min\{Ct, \sigma + \rho(t + Y)\}. \end{aligned} \tag{16}$$

• *Removing the Dependency on the Number of Flows on Individual Input Link:* As we described earlier, admission control at run-time makes sure that the link utilization allocated to each class of flows is not exceeded. Recall that α is the ratio of the link bandwidth allocated to the real-time traffic. The total number of the real-time flows on all input links are therefore subject to the following constraint:

$$\sum_{j=1}^{L} n_j \leq \alpha \frac{C}{\rho}. \tag{17}$$

Therefore, we have

$$\sum_{j=1}^{L} n_j^2 \leq (\alpha \frac{C}{\rho})^2. \tag{18}$$

Applying (17) and (18) into (10) and (11), we have the following results [7]:

$$\sum_{j=1}^{L} \mu_j(t) \leq \sum_{j=1}^{L} n_j \rho \leq \alpha C, \tag{19}$$

$$\sum_{j=1}^{L} RV_j(t) \leq \sum_{j=1}^{L} \omega(t)(n_j \rho)^2 \leq \omega(t)(\alpha C)^2, \tag{20}$$

where

$$r = \min\{\frac{C}{\rho}, \frac{1}{t}(\frac{\sigma}{\rho} + Y) + 1\}. \tag{21}$$

Substituting (19) and (20) into (8) in *Lemma 1*, we have the delay violation probability upper-bound as follows:

$$P\{D > d\} \leq 1 - \Phi(\min_{t<\beta} \xi(t)), \tag{22}$$

where

$$\xi(t) = \frac{\frac{d}{\alpha}\frac{1}{t} + \frac{1-\alpha}{\alpha}}{\sqrt{\omega(t)}}, \tag{23}$$

$$\beta = \frac{\alpha}{1-\alpha}(\frac{\sigma}{\rho} + Y). \tag{24}$$

[7] In (20), $\omega(t) = \frac{1}{s}(r^2 - 2\,r) + r - 1$, and s can be solved from $r = \frac{1}{\frac{1}{1-e^{-s}} - \frac{1}{s}}$. Since r is independent of j, so is $\omega(t)$.

We note that there is an implicit function $s = s(t)$ (with respect to t) in the above formula. If we use s as the parameter, we can obtain an explicit formula as shown in *Theorem 1*, which makes the computation easy. The details of derivation is given in APPENDIX C.

B. End-to-end Delay Analysis

Having derived the delay violation probability formula for the individual hops (servers), in this section, we will study how to perform end-to-end delay analysis. Because of the fact that delays in servers along the route may be correlated to each other, it is difficult to synthesize per-hop delays into the end-to-end delay. One way is to explicitly separate the correlation among the per-hop delays by using policers. With policers, the end-to-end delay will be partitioned into each server along the route, therefore the delays suffered in all servers will be independent. However, heavy overhead will be introduced by policers. Hence, this strategy is not applicable in the `diffserv` model.

Fortunately, in [35], the authors test the hypothesis that per-hop delay distributions are independent of each other. The delay distributions they obtained suggest strong hop-to-hop independence at high link utilization, as evidenced by the fact that the convolution of the delay distributions (which assumes independence of these distributions) is so close to the observed end-to-end distribution.

Our objective of the delay analysis is to get the maximum usable utilization. In general, with the maximum usable utilization, the network will have relatively heavy traffic. Hence, the above hypothesis can be applied into our derivation and the end-to-end delay distribution can be obtained by the convolution of per-hop delay distributions.

Suppose that $D_{\mathcal{R}}$ is the suffered delay for a random real-time packet along the route $\mathcal{R}$, and D_k is the per-hop suffered delay at Server k, for any $k \in \mathcal{R}$. By (4), we can get their density functions $f_{D_k}(x)$ of D_k respectively. By convolution, we can obtain the density function of the end-to-end delay as follows:

$$f_{D_{\mathcal{R}}}(x) = (\bigotimes_{k \in \mathcal{R}} f_{D_k})(x), \tag{25}$$

where the right-hand side is a $|\mathcal{R}|$-fold convolution. The convolution can be solved numerically by Fast-Fourier-Transform.

Therefore, for a random real-time packet going through route $\mathcal{R}$ with end-to-end deadline requirement d, the end-to-end delay violation probability is

$$\Pr\{D_{\mathcal{R}} > d\} = \int_{d}^{\infty} f_{D_{\mathcal{R}}}(x) dx. \tag{26}$$

VI. Experimental Evaluation

A. Experimental Model

• *Network and Traffic Model:* The network evaluated consists of 74 nodes as shown in Fig. 2. All output link capacity $C = 100M$ bps. We choose Nodes $0, 1, \ldots$, and 63 to be the sources of traffic and Node 73 to be the destination. We assume that the traffic belongs to a single real-time class. We consider voice traffic, with bursts varying from $1,280$ bits to $12,800$ bits, an average rate of $64K$ bps. We vary the deadline from 3 ms to 24 ms. We assume that requests for flow establishment form a Poisson process with rate λ, and that flow lifetimes are exponentially distributed with an average of 180 seconds. The system uses the remaining capacity to support best-effort traffic, which would not affect the results of this evaluation, and is therefore omitted in these experiments.

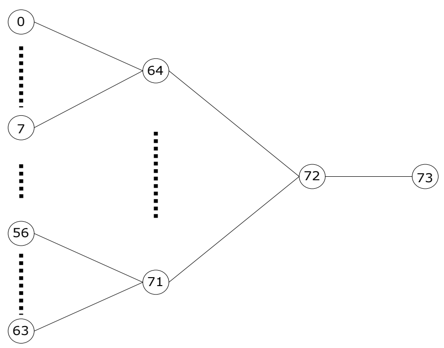

Fig. 2. The network topology for the simulation

• *Performance Metrics:* We are interested in two metrics:
– MUU: As mentioned in Section IV, the *maximum usable utilization* (MUU) is the bandwidth that can be allocated to real-time traffic.
– Admission Probability: This is the probability that a flow is admitted in a stable system. The higher the admission probability, the better the network resources are being used.
• *Evaluation Methods:* The MUU can be computed by (4) and (26) using the simple binary search. The admission probability of systems we consider can be analyzed by queuing theory and fixed point method [24]. The run-time delay violation probability can be measured by our simulation.

B. Numerical Results and Observations

In this sub-section, we report performance results and make observations. Due to the limited space, we only present a limited number of cases here. However, we find that the conclusions we draw here generally hold for many other cases we have evaluated.

B.1 Sensitivity of MUU to Deadline

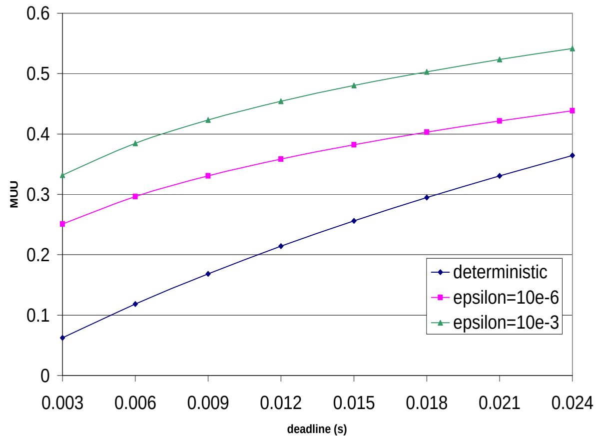

Fig. 3. Sensitivity of MUU to Deadline ($\sigma = 1280$ bits, $\rho = 64K$ bps)

Data on sensitivity of utilization to deadline are given in Fig. 3. From Fig. 3, we have the following observations:
• As expected, the utilization for both deterministic and statistical models increases as the deadline increases. This means that the higher the deadline, the higher utilization we can get for both deterministic and statistical models. A longer deadline results in more packets meeting deadline and, therefore, higher values for the MUU. Both statistical cases ($\epsilon = 10^{-3}$ and $\epsilon = 10^{-6}$) can achieve higher value for MUU than the deterministic model. Since the deterministic model does not allow delay violations, more resources need to be reserved, which decreases MUU. The statistical model better exploit available statistical multiplexing gain.
• The utilization for statistical model always decreases as ϵ changes from $\epsilon = 10^{-3}$ to $\epsilon = 10^{-6}$. This means that the higher the delay violation probability, the higher utilization we can get. A higher delay violation probability allows for larger bandwidth allocations and, therefore, higher values for the MUU.

B.2 Sensitivity of MUU to Relative Burstiness

From Fig. 4, we see that the relative burstiness of traffic heavily impacts on the MUU. The MUU decreases as the value of $\frac{\sigma}{\rho}$ increases. In fact, for very bursty traffic the MUU for deterministic model can be quite low. However MUU for both statistical cases remains relatively high.

B.3 Sensitivity of Admission Probability to Flow Arrival Rate

Data on sensitivity of admission probability are given in Fig. 5. From this figure, we make the following observations:

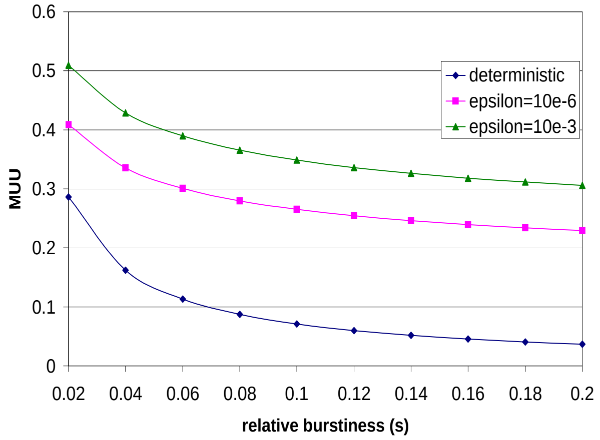

Fig. 4. Sensitivity of MUU to Relative Burstiness ($d = 20$ ms)

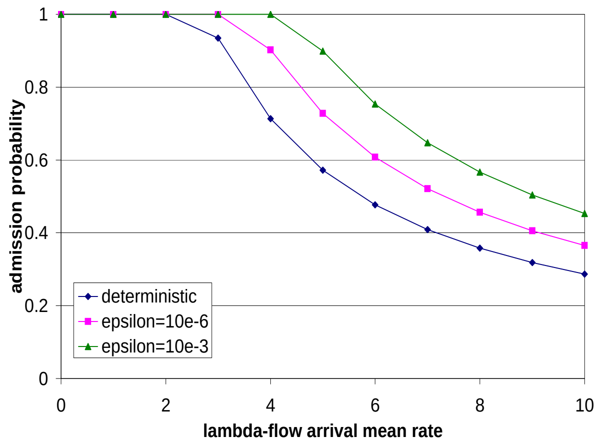

Fig. 5. Sensitivity of Admission Probability to Flow Arrival Rate ($d = 20$ ms, $\sigma = 1280$ bits, $\rho = 64K$ bps, $C = 100M$ bps)

- Admission probability is sensitive to the flow arrival rate λ for all models. Admission probability decreases as λ increases in all models. The reason is obvious: A large λ value implies a large number of flow arrivals in the system. Since the bandwidth is limited, some flows are not allowed to enter the network. Therefore, the admission probability decreases.
- Different models have different sensitivities to λ in terms of admission probability. The statistical models always achieve higher admission probabilities than the deterministic model. Larger delay violation probability (i.e., ϵ) always achieve higher admission probability. This is because of the achievable utilization is proportional to ϵ.

VII. Conclusions

In this paper, we have proposed a methodology for providing differentiated services with *statistical* performance guarantees in networks that use static priority schedulers. Given that static priority schedulers are widely supported by current routers, we believe that our approach is practical and effective to support real-time applications in existing networks.

We have used Utilization-based Admission Control (UBAC) approach which requires a configuration-time verification to determine a safe utilization bound of servers. Admission control at runtime then is reduced to simple utilization tests on the servers along the path of the new flow. Hence, the approach is scalable.

Obviously, the verification of the utilization bound will have to be involved with delay analysis. The general delay derivation depends on flow population information, which is not available at the system configuration time. We have extended the general approach and developed a method that allows us to analyze the delays without depending on the dynamic status of flow population.

References

[1] G. Agrawal, B. Chen, W. Zhao, and S. Davari, *Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol*, IEEE International Conference on Distributed Computing Systems, 1992, pp.468-475.

[2] A. Banerjea and S. Keshav, *Queueing delays in rate controlled ATM networks*, Proceedinds of IEEE INFOCOM'93, 1993.

[3] J. Bennett and H. Zhang, *WF^2Q: worst-case fair weighted fair queueing*, Proceedinds of IEEE INFOCOM'96, 1996.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Service*, RFC 2474, Dec. 1998.

[5] Y. Bernet et al., *A Framework for Differentiated Services*, Internet-Draft, IETF, Feb. 1999.

[6] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, *Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks*, Proceedings of IEEE INFOCOM'00, March 2000.

[7] R. Braden and D. Clark and S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, Internet RFC 1633, June 1994.

[8] C. Chang, *Stability, queue length, and delay of deterministic and stochastic queueing networks*, IEEE Transactions on Automatic Control, 39(5):913-931, May 1994.

[9] B. Choi and R. Bettati, *Endpoint Admission Control: Network Based Approach*, IEEE Proceedings of ICDCS, April 2001.

[10] R. L. Cruz, *A Calculus for Network Delay, Part I*, IEEE Transactions on Information Theory, Vol. 37. Jan. 1991.

[11] R. L. Cruz, *A Calculus for Network Delay, Part II*, IEEE Transactions on Information Theory, Vol. 37. Jan. 1991.

[12] A. Dailianas and A. Bovopoulis, *Real-time admission control algorithms with delay and loss guarantees in ATM networks*, Proceedings of INFOCOM'94, pages 1065–1072, 1994.

[13] Norival R. Figueira, Joseph Pasquale, *An Upper Bound on Delay for the VirtualClock Service Discipline*, IEEE/ACM Transactions on Networking, Vol. 3. Aug. 1995.

[14] V. Firoiu, J. Kurose, and D. Towsley, *Efficient admission control for EDF schedulers*, Proceedings of INFOCOM'97, 1997.

[15] L. Georgiadis, R. Gurin, V. Peris, K.N. Sivarajan, *Efficient Network QoS Provisioning Based on per Node Traffic Shaping*, IEEE/ACM Transactions on Networking, 4(4):482-501, August 1996.

[16] S. J. Golestani, *Network delay analysis of a class of fair queueing algorithms*, IEEE J. on Seleted Areas in Communications, vol. 13, no. 6, 1995.

[17] E. Knightly, *H-BIND: A new approach to providing statistical performance guarantees to VBR traffic*, Proceedings of IEEE INFOCOM'96, pp.1091-1099, San Francisco, CA, March 1996.

[18] E. Knightly, *Second Moment Resource Allocation in Multi-Service Networks*, Proceedings of ACM SIGMETRICS '97, Seattle, WA, June 1997.

[19] E. Knightly, *Enforceable quality of service guarantees for bursty traffic streams*, Proceedings of Infocom'98, pp.635-642, San Francisco, March 1998.

[20] J. Kurose, *On computing per-session performance bounds in high-speed multi-hop computer networks*, ACM Sigmetrics'92, pages 128-139, 1992.

[21] J. Liebeherr, S. D. Patek and E. Yilmaz, *Tradeoffs in Designing Networks with End-to-End Statistical QoS Guarantees*, Proceedings of IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS '2000), pp. 221-230. June 2000.

[22] C. L. Liu and J. W. Layland, *Scheduling algorithms for multiprogramming in a hard real time environment*, J. ACM, Vol. 20, No. 1, 1973, pp.46-61.

[23] J. Liebeherr, D.E. Wrege, and D. Ferrari, *Exact admission control in networks with bounded delay services*, IEEE/ACM Transactions on Networking, 1996.

[24] D. Mitra and J. A. Morrison, *Erlang capacity and uniform approximations for shared unbuffered resources* IEEE/ACM Transactions on Networking, 1994.

[25] K. Nicols, V. Jacobson, L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, Internet-Draft, Nov. 1997.

[26] A. K. J. Parekh and R. G. Gallager, *A generalized processor sharing aApproach to flow control in integrated services networks: the single-node case*, IEEE/ACM Trans. Networking, vol. 1, no. 3, 1993.

[27] A. K. J. Parekh and R. G. Gallager, *A generalized processor sharing aApproach to flow control in integrated services networks: the multiple-node case*, IEEE/ACM Trans. Networking, vol. 2, no. 2, 1994.

[28] J. Qiu and E. Knightly, *Inter-class resource sharing using statistical service envelopes*, Proceedings of IEEE INFOCOM '99, New York, NY, March 1999.

[29] I. Stoica, H. Zhang, *Providing Guaranteed Services Without Per Flow Management*, ACM SIGCOMM 1999.

[30] S. Wang, D. Xuan, R. Bettati and W. Zhao, *Providing Absolute Differentiated Services for Real-Time Applications in Static Priority Scheduling Networks*, Proceedings of IEEE INFOCOM'01, April, 2001.

[31] S. Wang, D. Xuan, R. Bettati and W. Zhao, *Differentiated Services with Statistical Real-Time Guarantees in Static-Priority Scheduling Networks*, to appear in Proceedings of IEEE RTSS'01, December, 2001.

[32] D. Wrege, E. Knightly, H. Zhang, and J. Liebeherr, *Deterministic delay bounds for VBR video in packet-switching networks: Fundamental limits and practical tradeoffs*, IEEE/ACM Transactions on Networking, 4(3):352-362, June 1996.

[33] Tao Wu and E. W. Knightly, *Enforceable and Efficient Service Provisioning*, Computer Communications Journal: Special Issue on Multimedia Communications over the Internet, 23:(14-15):1377-1388, August 2000.

[34] O. Yaron and M. Sidi, *Performance and Stability of Communication Networks via Robust Exponential Bounds*, IEEE/ACM Transactions on Networking, 1(3):372-385, June 1993.

[35] D. Yates, J. Kurose, D. Towsley, and M.G.Hluchyj, *On Per-Session End-To-End Delay and the Call Admission Problem for Real Time Applications with QoS Requirements*, Journal of High Speed Networks, volume Vol 3, pp 429-458, December 1994.

[36] L. Zhang, *Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks*, ACM Transactions on Comput. Syst., Vol. 9. May. 1991.

Appendix A: Proof of Lemma 2

Lemma 2: Given the deterministic arrival traffic envelope $b_j(t)$ of the traffic process A_j, if A_j is ergodic and its long-term average rate $\rho_j = \lim_{t\to\infty} \frac{b_j(t)}{t}$, then the mean rate $\mu_j(t)$ for the random variable $A_j(t)$ can be described as follows:

$$\mu_j(t) = \rho_j. \tag{27}$$

Proof: Given an instance $A_j(t)$ of the traffic process A_j, then

$$\begin{aligned}
\mu_j(t) &= \lim_{T\to\infty} \frac{1}{T} \int_0^T \frac{A_j(\tau+t) - A_j(\tau)}{t} d\tau \qquad (28) \\
&= \frac{1}{t} \lim_{T\to\infty} \frac{1}{T} (\int_0^T A_j(\tau+t) d\tau - \int_0^T A_j(\tau) d\tau) \\
&= \frac{1}{t} \lim_{T\to\infty} \frac{1}{T} (\int_t^{T+t} A_j(\tau) d\tau - \int_0^T A_j(\tau) d\tau) \\
&= \frac{1}{t} \lim_{T\to\infty} \frac{1}{T} (\int_T^{T+t} A_j(\tau) d\tau - \int_0^t A_j(\tau) d\tau) \\
&= \frac{1}{t} (\lim_{T\to\infty} \frac{1}{T} \int_T^{T+t} A_j(\tau) d\tau - \lim_{T\to\infty} \frac{1}{T} \int_0^t A_j(\tau) d\tau) \\
&= \frac{1}{t} \lim_{T\to\infty} \frac{1}{T} \int_T^{T+t} A_j(\tau) d\tau \\
&= \lim_{T\to\infty} \frac{1}{T} A_j(T+\xi), \text{ where } \xi = \xi(T) \in [0, t] \\
&= \lim_{T\to\infty} \frac{1}{T} (A_j(T+\xi) - A_j(\xi)) \\
&= \rho
\end{aligned}$$

■

Appendix B: Proof of Lemma 3

Several approaches have been proposed to derive or control *Rate-Variance*. One way [19] is to use policers before each server to regulate the traffic to specify the rate variance. Due to the overhead of policing, it is not applied to the `diffserv` model. The other approach [33] is to use the maximum entropy principle scheme to approximate the rate-variance, which is suitable to the `diffserv` model.

The maximum entropy principle scheme can be viewed as the most likely statistical characterization of a group of flow given only its deterministic bounds. As a measure of uncertainty or randomness, entropy is defined as

$$h(f) = -\int_D f(x) \ln f(x)\, dx, \tag{29}$$

for a continuous random variable with probability density function f(x).

Lemma 3: Given the deterministic arrival traffic envelope $b_j(t)$ of the traffic process A_j with mean rate ρ_j, the maximum entropy estimate of rate-variance is

$$RV_j(t) = \omega_j(t)\rho_j^2, \tag{30}$$

where

$$\omega_j(t) = \frac{1}{s_j}(r_j^2 - 2\, r_j) + r_j - 1, \tag{31}$$

s_j can be got from the following equation (inverse function)

$$r_j = \frac{1}{\frac{1}{1-e^{-s_j}} - \frac{1}{s_j}}, \tag{32}$$

and

$$r_j = \frac{b_j(t)}{\rho_j t}. \tag{33}$$

Proof: In the following proof, we'll omit index j when the context is clear. Assume that $f(x)$ is the density function of $\frac{A(\tau+t)-A(\tau)}{t}$. By the maximum entropy principle, we have the following optimization problem.

$$\textit{Maximize} \quad -\int_0^{\frac{b(t)}{t}} f(x) \ln f(x)\, dx \tag{34}$$

$$\textit{Subject to} \quad \int_0^{\frac{b(t)}{t}} f(x)\, dx = 1 \tag{35}$$

$$\int_0^{\frac{b(t)}{t}} x f(x)\, dx = \rho \tag{36}$$

With the *Lagrange Multilpier Method*, the density function $f(x)$ can be expressed as

$$f(x) = \begin{cases} y, & 0 \le x \le \frac{b(t)}{t} \\ 0, & \text{otherwise} \end{cases}, \tag{37}$$

where

$$y = e^{(s_0 + s_1 \frac{b(t)}{t})}. \tag{38}$$

Substituting (37) into (35) and (36), we have

$$\int_0^{\frac{b(t)}{t}} e^{(s_0+s_1 x)}\, dx = 1, \tag{39}$$

$$\int_0^{\frac{b(t)}{t}} x e^{(s_0+s_1 x)}\, dx = \rho. \tag{40}$$

Simplifying (39) and (40), we have

$$y = s_1 + e^{s_0}, \tag{41}$$

and

$$\frac{b(t)}{t} y = 1 + s_1 \rho. \tag{42}$$

From (38), (41), and (42), we can get

$$e^{s_1 \frac{b(t)}{t}} + \frac{1+s_1\rho}{s_1(\frac{b(t)}{t} - \rho) - 1} = 0. \tag{43}$$

With some algebraic manipulation, (43) can be written as

$$r = \frac{1}{\frac{1}{1-e^{-s_1 r\rho}} - \frac{1}{s_1 r \rho}}, \tag{44}$$

By the definition of rate-variance the above analysis, we have

$$\begin{aligned}
RV(t) &= \int_0^{\frac{b(t)}{t}} x^2 f(x)\, dx - \rho^2 & (45)\\
&= \frac{1}{s_1}(\frac{b(t)}{t})^2 y - \frac{2}{s_1}\int_0^{\frac{b(t)}{t}} x f(x)\, dx - \rho^2 & (46)\\
&= \frac{1}{s_1}(\frac{b(t)}{t})^2 y - \frac{2}{s_1}\rho - \rho^2 & (47)\\
&= \frac{1}{s_1}((1+s_1\rho)\frac{b(t)}{t} - 2\rho) - \rho^2 & (48)\\
&= \frac{1}{s_1}(\frac{b(t)}{t} - 2\rho) + \frac{b(t)}{t}\rho - \rho^2 & (49)\\
&= \frac{1}{s_1}(r-2)\rho + (r-1)\rho^2 & (50)
\end{aligned}$$

Letting $s = s_1 r \rho$ and substituting it into (44) and (50), we can draw the conclusion as claimed. ∎

Appendix C: Making the the Delay Formula Computable

Considering r in (21), the flex point of r is $\tau = \frac{\frac{\sigma}{\rho}+Y}{\frac{C}{\rho}-1}$. As $t \le \tau$, $r = \frac{C}{\rho}$, and $\omega(t)$ is also a constant with respect to t, then $\xi(t)$ approaches the minimum value as $t = \tau$. Therefore we just need to consider the interval $\tau \le t \le \beta$.

In (23), since $\omega(t)$ is an implicit function with respect to t, it's very hard to analyze the property of $\xi(t)$ in term of t. Fortunately, we find that s is an increasing function with respect to t as $t \ge \tau$, then we can use s to explicitly express r, t, and ξ. At the same time, the scope $[s_0, s_1]$ of s corresponding to the scope $[\tau, \beta]$ of t can be evaluated. Therefore, the delay violation probability for a random real-time packet is bounded as shown by (4) in *Theorem 1.*