Measurement-Based Characterization and Classification of QoS-Enhanced Systems

Aleksandar Kuzmanovic and Edward W. Knightly, *Member*, *IEEE*

Abstract—Quality-of-Service mechanisms and differentiated service classes are increasingly available in networks and Web servers. While network and Web server clients can assess their service by measuring basic performance parameters such as packet loss and delay, such measurements do not expose the system's core QoS functionality such as multiclass service discipline. In this paper, we develop a framework and methodology for enabling network and Web server clients to assess system's multiclass mechanisms and parameters. Using hypothesis testing, maximum likelihood estimation, and empirical arrival and service rates measured across multiple time scales, we devise techniques for clients to 1) determine the most likely service discipline among Earliest Deadline First (EDF), class-based Weighted Fair Queuing (WFQ), and Strict Priority (SP), 2) estimate the system's parameters with high confidence, and (3) detect and parameterize non work-conserving elements such as rate limiters. We describe the important role of time scales in such a framework and identify the conditions necessary for obtaining accurate and high confidence inferences.

Index Terms—QoS, measurement, multiclass, statistical envelopes.

1 Introduction

Both research and commercial networks and Web servers are increasingly able to provide minimum quality-of-service levels to traffic and application classes, e.g., [1]. Example components of such networks include QoS schedulers [2], [3], diffserv-style service level agreements [4], [5], [6], [7], edge-based traffic shaping and prioritizing devices, and novel architectures and algorithms for scalable QoS management [8], [9], [10]. Similar resource management mechanisms, request scheduling policies, and algorithms are also developed for quality-of-service Web servers [11], [12], [13], [14]. However, even as both the network's and Web server's infrastructure and services become increasingly sophisticated, the network's and Web server's *clients* lack reciprocal tools for validation and monitoring of the system's QoS capabilities, and the available tools allow only the inferences of parameters such as bottleneck link speeds or available bandwidth [15], [16], [17], [18]. Clients of Service Level Agreements (SLAs) will have monitoring requirements ranging from basic validation of the SLA's raw bandwidth to more sophisticated inference of multiclass functionalities. For example, is a class rate limited (policed)? If so, what are the rate limiter's parameters and what is necessary to detect this? In a multiclass environment with multiple classes within or among SLAs, what is the interclass relationship? Fair, weighted fair, strict priority, and with what parameters? Is resource "borrowing" across classes fully allowed or only allowed within certain limits?

Similar issues occur in a Web server scenario. The requirements of a client of a Web hosting service range from the ability to track, assess, and *quantify* basic service capabilities, such as minimum rate at which user's requests are serviced, to the ability to assess mechanisms and parameters by which capacity is allocated to various hosted sites. Besides clients, Web hosting *providers* will have similar objectives in a larger scale Web-hosting environment in which a number of front-end servers use resources of back-end servers and when different QoS mechanisms are simultaneously implemented in the system. In such an environment, a need to quantify service and assess the interclass relationship arises.

Obtaining "offline" answers to such questions can be quite trivial. In particular, consider a system with an unknown service (suppose the system is a single router for simplicity). To assess whether classes are rate limited, one could probe each class, one at a time, with a high rate test sequence: the output of the system would yield the policing parameters. Similarly, simultaneously probing at a high rate in all classes would yield the interclass relationships: if one class receives all of the service, the system is strict priority (at least for that class); if weighted service is received, the system performs a variant of weighted fair queuing.

In contrast, the "online" case, in which one cannot force all other traffic classes to remain idle while experiments are performed, is quite different. Even for classes which are under the control of the client, it may be highly undesirable to disrupt the class with experiments such as above. For example, sending at a high rate to detect rate-limiters may cause excessive packet losses for established sessions.

The goal of this paper is to develop a framework for monitoring, validation, and inference of multiclass services for the online case in which existing services cannot be disrupted. In particular, we show how passive monitoring of system arrivals and departures can be used to detect if a class has a minimum guaranteed rate and/or a rate limiter. Moreover, if such elements exist, we will show how to compute their maximum likelihood parameters. Beyond a single class, we will also show how interclass relationships can be assessed. For example, we devise tests which infer not only whether a service discipline is work-conserving or

- *The authors are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005.*
E-mail: {akuzma, knightly}@rice.edu.

Manuscript received 9 Jan. 2002; revised 23 Aug. 2002; accepted 3 Dec. 2002.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 115671.

1045-9219/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

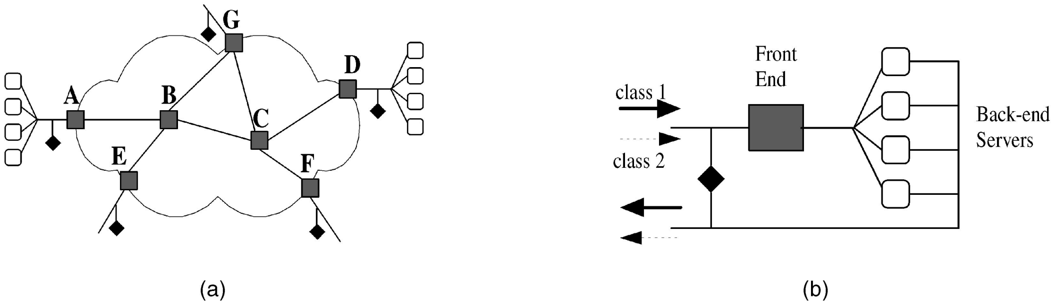


Fig. 1. Targeted systems. (a) Network and (b) Web server.

non work-conserving, but also the relationship among classes, such as weighted fair or strict priority.

Throughout our analysis, it is clear that time scales play a key role. Short time scale measurements are crucial for detecting and analyzing non work-conserving elements such as rate limiters. In contrast, long time scale measurements best reveal "link sharing" rules and weights. Thus, a key aspect of our contribution is that we develop all such measurement tools using a unifying abstraction of envelopes [19], [20], [21], hypothesis testing, and maximum likelihood estimations. In this way, we treat phenomena occurring at different time scales in a uniform and methodical way.

We, therefore, consider a general system model that encompasses a broad class of multiservice elements ranging from routers to Web servers, yet we necessarily forgo modeling of many of the intricacies of realistic systems (e.g., we limit our discussion to a single bottleneck node). For inferences of the system's multiclass characteristics, we consider the case where internal system information, such as buffer size or link capacity, is not available. Moreover, we assume that arrivals and departures of other classes (i.e., "cross traffic") cannot be explicitly observed and measured at the network edge. Thus, an integral part of our technique is to first, assess and statistically characterize the service available to the traffic aggregate that is explicitly measured at the network edge and, then, determine mutual relationships among classes within the aggregate.

We perform a large set of simulation experiments in both networking and Web server scenarios and find that the technique is practically applicable. For example, in our networking experiments with the majority-rule hypothesis test performed across multiple time scales, multiclass EDF scheduling was correctly inferred 100 percent of the time when the class delay bounds were sufficiently differentiated, and class-based fair queuing was correctly inferred 94 percent of the time. Once the service discipline is known, the algorithm estimated class WFQ weights within 1.4 percent of the correct value with 95 percent confidence. In our Web server experiments, we correctly classified the scheduling discipline in more than 90 percent of the cases.

The remainder of this paper is organized as follows: In Section 2, we explain targeted Web server and network scenarios, define the measurement and inference problem, and describe the system model. In Section 3, we provide basic background on envelopes and describe the measurement methodology. In Section 4, we devise the maximum likelihood estimates for the system parameters and hypothesis tests for inference of the service discipline. Next, in Section 5, we present a set of simulations to evaluate the effectiveness of the scheme in both Web server and network scenarios and under a number of different system functionalities. Finally, in Section 6, we conclude.

2 Targeted Systems and Problem Statement

In this section, we describe networking and Web server scenarios in which our framework can be applied and formally define the problem and system model.

2.1 Targeted Systems

2.1.1 Network Scenario

Fig. 1a depicts the targeted networking scenario. In this case, measurement modules are placed at the periphery of the network. The goal is to use passive edge-based client measurements to infer the multiclass QoS mechanisms and parameters employed by the network operator. With an improved understanding of the way traffic is internally serviced, clients can better manage their use of multiclass networks. Also, network clients can use the framework to *quantatively* estimate their service when only relative performance guarantees are provided or when end-to-end service is provided through more than one ISP. For example, if the provider guarantees that class X will have higher priority than class Y, our framework can determine maximum likelihood lower and upper service bounds of both classes and infer actual interclass relationship. Such inferences can be used by network clients to better utilize their available bandwidth, i.e., for capacity planning. Similarly, operators or third parties can employ the methodology to test and validate the performance and potential performance of multiple service classes.

2.1.2 QoS Web Servers

Fig. 1b depicts a two-class distributed Web server, where a passive measurement module is depicted by a diamond. QoS functionalities in the server may include prioritized scheduling of incoming requests at the front-end, prioritized distribution of jobs to back-end nodes, and operating-system mechanisms such as prioritized scheduling of CPU, memory, and disk access [13]. In any case, our goal is to provide an application-layer characterization of the system's multiclass QoS mechanisms. For example, weighted share of CPU resources does not guarantee the same level of differentiation for the application, since the actual response times also depend on the file type (static or dynamic), file size, and its caching state. Also, if several QoS mechanisms are simultaneously employed with the goal of providing weighted fair service among different classes, our technique will estimate a class' net "guaranteed rate," i.e., it's minimum serviced request throughput. Such inferences have important implications for both performance monitoring and resource management.

2.2 Problem Formulation

For inferences of the system's multiclass characteristics, we consider the case where internal system information is *not* available, i.e., neither static configuration information (such as the scheduler's parameters) nor empirical information (such as mean buffer length). Instead, the available information consists of the external observations from passive monitoring of requests, namely, request arrival and departure times along with request class labels and sequence numbers. In the case of Web servers (both single node and distributed), both arrivals and departures are directly observable from the system's front end (see [11], [13] for a detailed description of such an architecture). In the case of networks, packet time stamping at ingress nodes provides a mechanism to observe both arrival and departure times at the departure node [22]. In particular, for low speed links (e.g., up to 100 Mb/sec), *tcpdump* can capture and record header information at line rate [22]. For higher speed implementations, this functionality would best be achieved with hardware support.

Otherwise, the measurement modules can communicate their collected information offline. Below, we formally define the multiclass service inference problem.

Problem statement: Consider a multiclass system with an unknown scheduling discipline fed by requests from N classes. Denote with G the number of observable or explicitly measured classes and assume that classes $1, \cdots, G$ are observable, while classes $G+1, \cdots, N$ are not. Denote the arrival and departure times of request j from class i as a_j^i and d_j^i, respectively. Given $a_1^i, a_2^i, \cdots$ and $d_1^i, d_2^i, \cdots$ for $i = 1, \cdots, G$,

1. Estimate the available service of the aggregate consisting of G classes.
2. Assess the most likely service discipline among SP, WFQ [23], and EDF.
3. Estimate the maximum likelihood values of the class parameters for each of G measured traffic classes: "guaranteed rate" (ϕ_i) in WFQ, delay bound (δ_i) in EDF, and rate limiters (r_i) in non work-conserving servers.

2.3 System Model

The general system model considered in this paper is depicted in Fig. 2. As in the basic abstraction of service disciplines described in [24], it consists of two stages: non work-conserving elements which limit a class' rate and a work-conserving packet or request scheduler. For rate limiters, we consider single-level leaky bucket regulators, and for the packet scheduler, we consider SP, WFQ, and EDF. An SP scheduler consists of one queue per traffic class with packets from the highest priority nonempty class serviced first. For example, a packet in level i is serviced only if no packets are backlogged in levels $1, \cdots, i-1$. For WFQ, each traffic class i is allocated a guaranteed capacity $\phi_i C$ such that, whenever packets from class i are backlogged, the class receives service at a rate of at least $\phi_i C$. Unused capacity of nonbacklogged classes is distributed in a weighted fair manner among backlogged classes. For EDF, each class has an associated delay bound so that packet j of class i arriving at time a_j^i has deadline a_j^i plus its delay bound, and the scheduler selects the packet with the smallest (earliest) deadline for service.

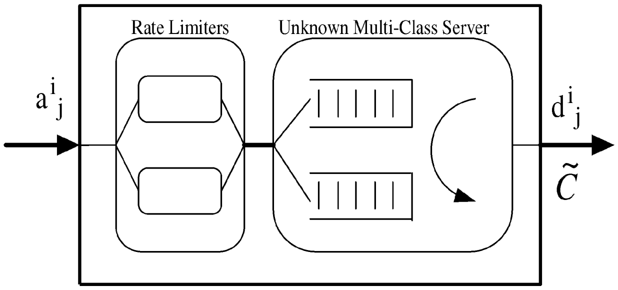


Fig. 2. System model for multiservice measurement.

This formulation covers a broad set of class-based scheduling elements, including minimum guaranteed rates, maximum policed rates, weighted fairness, sorted priority, and strict priority. While necessarily not comprehensive, it incorporates both work-conserving and non work-conserving service disciplines and a number of mechanisms for interclass resource sharing and quality-of-service differentiation. The choice of SP, WFQ, and EDF, which belong to rate and delay-based classes of schedulers, is made since these schedulers are both well-studied and implemented in practice. Also, it should be noted that the multiclass inference framework developed in this paper can be applied to any other scheduler for which one can derive a statistical service envelope, the key inference tool that we explain in the following section.

We consider that the capacity of a multiclass system is not known and can vary over time. In the networking scenario, this formulation covers the problem of unknown cross-traffic, while it applies equivavalently to the Web server inference problem, where the capacity is nonconstant as the service times for different requests vary due to different CPU service times, disk service times, and variable file sizes. The first step in our inference methodology is to assess and statistically characterize the service available to an aggregate of all measured classes and, then, determine interclass relationships within the aggregate.

A special case of our general system model that is considered throughout the paper is a single bottleneck multiclass networking router with fixed service capacity C, in which all classes' arrivals and departures are known. We will study this special case service model for two reasons: first, for simplicity of presentation and, second, as a reasonable and intuitive checkpoint of our inference methodologies applicable to a general system model with variable capacity.

3 Service Measurements and Concept of Envelopes

As described above, our goal is to infer the elements and parameters of the multiclass system. In such a system, the request service discipline defines the interclass relationships or the service received when different classes compete for resources. For example, with an SP scheduler, the highest priority class receives all demanded service up to the available link capacity and, in that way, is completely isolated from other classes' demands. In contrast, lower priority classes utilize only *remaining* capacity from higher priority classes and their performance is strongly dependent on these classes' demands.

In Section 3.1, we provide a theoretical description of such interclass relationships via statistical service envelopes. Next, in Sections 3.2 and 3.3, we describe empirical arrival and service models, i.e., we explain how the theoretical concepts described in Section 3.1 can be applied in practice.

3.1 Theoretical Envelopes

Here, we review general traffic and service characterizations that can be applied to our multiclass inference problem. The framework is based on statistical envelopes which provide a unifying abstraction for both arrivals and services and incorporate a system's behavior across time scales.

To characterize a flow's rate, an associated interval length must also be specified. However, the arrival workload (expressed in number of bits) varies in time over intervals of the same length, simply due to variable source behavior. Thus, to accurately characterize randomness of flow arrivals, we use the concept of *statistical* arrival envelopes to capture a flow's variability over intervals of different length. We denote class is statistical arrival envelope as $B^i(t)$, which is a sequence of random variables that characterizes arrivals from class i over time intervals of duration t.[1] It is assumed that each class arrival process is stationary and that $B^i(t)$ and $B^j(t)$ are statistically independent when $i \neq j$.

In [21], statistical admission control tests are developed for several multiclass schedulers. The key technique for exploiting interclass resource sharing is to characterize a class' available service beyond its worst-case allocation. For example, in a WFQ server, a class with weight ϕ_i receives service at rate no less than $\phi_i C$ whenever it is backlogged ($\sum_j \phi_j = 1$). However, due to statistically varying demands of other classes, the service received can be far greater than this lower bound. A statistical *service* envelope $S^i(t)$ is therefore a general characterization of the service received by class i over intervals of length t for which the class is continually backlogged.

Equations (1), (2), and (3) show the statistical service envelopes for SP, WFQ, and EDF schedulers, respectively.

$$S^i_{SP}(t) = \left(Ct - \sum_{n=1}^{i-1} B^n(t) \right)^+, \tag{1}$$

$$S^i_{WFQ}(t) = \phi_i Ct + \left((1-\phi_i)Ct - \sum_{n \neq i} B^n(t) \right)^+, \tag{2}$$

$$S^i_{EDF}(t) = \left(Ct + CD_i - \sum_{n \neq i} B^n(t - \delta_n + \delta_i) \right)^+. \tag{3}$$

The envelopes are a function of the link capacity C and, as described above, the other class' input traffic, described by the arrival envelope $B^i(t)$. For SP, observe that class i's service is only a function of the workload in classes $1, 2, \cdots, i-1$. In contrast, for WFQ, class i's service is a function of all other classes' traffic, but is upper bounded by Ct if all other classes are always idle and lower bounded by $\phi_i C$ if all other classes are continuously backlogged. Finally, with EDF class, i's service again depends on all other class' inputs as well as the delay bound of class i denoted by δ_i.

3.2 Empirical Arrival Model

Here, we show how statistical arrival envelopes $B^i(t)$ can be measured over multiple time scales using class i's arrival request sequence. Measurement at multiple time scales is important in this context as different system components are most accurately detected at different time scales.

Focusing on a single class for illustration, denote the total arrivals in the interval $[s, s+t]$ by $A[s, s+t]$. A traffic envelope refers to a time invariant characterization of the arrivals as a function of interval length t (see [25] for examples of deterministic envelopes). For a measurement window $[s, s+T]$ and a particular interval length I_k beginning at time $s + (j-1)I_k$, class i's arrival *rate* is given by

$$R^{i,A}_{k,j} = \frac{A^i[s + (j-1)I_k,\ s + jI_k]}{I_k},$$

for $j = 1, \cdots, N_k$, where $N_k = \lfloor T/I_k \rfloor$ is the number of successive intervals of length I_k in the measurement window $[s, s+T]$.

Using measured rates over different subintervals within the window T, the mean and variance of the empirical rate envelope of class i for intervals of length I_k can be computed as

$$\bar{R}^{i,A}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} R^{i,A}_{k,j}, \tag{4}$$

and

$$RV^{i,A}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} (R^{i,A}_{k,j} - \bar{R}^{i,A}_k)^2. \tag{5}$$

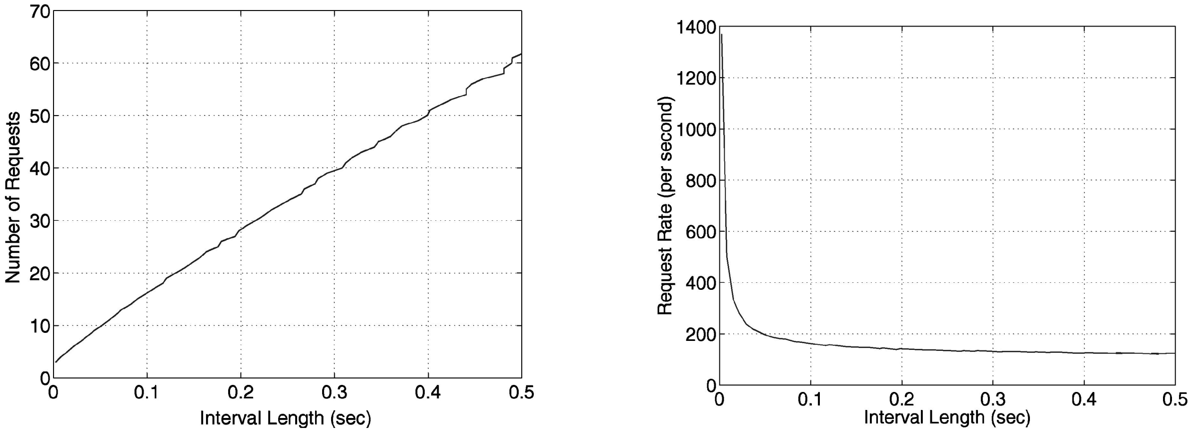
Observe that the first two moments of the *rate* arrival envelope (e.g., $\bar{R}^{i,A}_k$ and $RV^{i,A}_k$) are simply empirical and *normalized* versions of the first two moments of class i's arrival envelope $B^i(I_k)$ at time scale I_k. As an example envelope, Fig. 3a shows the representation of the arrival envelope $B^i(t)$ for the Rice University CS Department trace described in Section 5, while Fig. 3b shows the reciprocate *rate* envelope normalized to the interval length I_k, so that the y-axis is rate. Specifically, Fig. 3b depicts

$$\bar{R}^{i,A}_k + 1.6\sqrt{RV^{i,A}_k},$$

for 50 time scales, $I_k = 0.01, 0.02 \cdots, 0.5$. It is clear that, over short interval lengths, significantly more requests than the mean 100 per second (as can be seen from the rate to which the curve in Fig. 3b converges) can arrive. It will be shown that such characteristics of the request workload, i.e., its variability over time scales, is the key input for obtaining accurate scheduler inferences.

In Section 4, we describe how this empirical class-based arrival rate envelope is incorporated into the above multiclass inference problems and, in Section 5, we experimentally investigate applications of this traffic characterization.

[1] For a particular time scale $t = t_1$, $B^i(t_1)$ is a random variable.



(a)

(b)

Fig. 3. Arrival envelopes ($mean + 1.6\ deviation$). (a) Statistical arrival envelope and (b) statistical arrival rate envelope.

3.3 Empirical Service Model

Here, we describe a general mechanism for measuring and characterizing a service rate. Analogous to the traffic envelope, the service rate envelope is not simply a single service bandwidth, but a statistical characterization of service across time scales. We distinguish two types of service rate envelopes: *aggregate* and *class* envelopes.

Aggregate service rate envelopes characterize a service rate available to an aggregate of explicitly measured classes. It captures the effects of non measured cross traffic in networks, or reveals and characterizes nonconstant service capacity of a Web sever. Similarly, a *class* service rate envelope characterizes a service rate available to each traffic class *within* the aggregate and across time scales. In both cases, this multiple-time-scale characterization is critical to inference of diverse service components such as maximum policed bandwidth, minimum service, and analysis of interclass resource sharing relationships. Moreover, its statistical nature reflects the fact that a class' service can fluctuate according to the varying demands of other classes and the mechanism by which the scheduler arbitrates this demand.

3.3.1 Backlogging Condition

The empirical service rate envelope characterizes the service rate received by the flow (either a class' or an aggregate's flow) as a function of the interval length over which the flow is backlogged, where a flow is said to be backlogged whenever it has at least one packet in the system. A traffic flow is continuously backlogged for k packet transmissions in the interval $[a_j, d_{j+k-1}]$ if

$$d_{j+m} > a_{j+m+1}, \text{ for all } 0 \leq m < k-2,$$

for $k \geq 2$. Note that all packet transmissions are backlogged for $k = 1$ in the interval $[a_j, d_j]$ (see reference [26] for an illustration of a backlogging condition).

Thus, denoting $U[s, s+t]$ as number of flow's bits[2] received in $[s, s+t]$ is simply

$$M^S(t) = \frac{U[s, s+t]}{t}. \tag{6}$$

Finally, the measurement for each backlogged interval is included in the measurement $\vec{M}_k^S$ if

$$(k - 0.5)I_1 < t \leq (k + 0.5)I_1. \tag{7}$$

Measured service envelope samples $\vec{M}_k^S$, both per class and aggregate, are used in inferring the scheduling discipline as will be, in detail, explained in the following section.[3]

3.3.2 Empirical Aggregate Service Model

Analogous to arrival traffic envelope, the *aggregate* service rate envelope is determined with the first two moments of its service rates across the time scales. Denote the aggregate service rate measurements in time scale I_k as $\vec{C}_k = \vec{M}_k^S$, where $\vec{M}_k^S$ is measured as explained in Section 3.3.1 for the arrival-departure sequence of the aggregate flow. Then, using these measured rates over different subintervals within the window T, the mean and variance of the empirical aggregate rate for intervals of length I_k can be computed as

$$\bar{C}_k = \frac{1}{M_k} \sum_{j=1}^{M_k} C_{k,j} \tag{8}$$

and

$$CV_k = \frac{1}{M_k} \sum_{j=1}^{M_k} (C_{k,j} - \bar{C}_k)^2, \tag{9}$$

where M_k is the number of measured system rate samples in time scale I_k.

3.3.3 Empirical Class Service Model

In the case of the *class* service rate envelope, denote the service rate measurements for class i and time scale I_k as $\vec{R}_k^{i,S} = \vec{M}_k^S$, where $\vec{M}_k^S$ is measured as explained in Section 3.3.1 for the arrival-departure sequence of class-i flow. It should be noted

[2] The actual units of monitoring are packets or requests. However, since these can be of different size, we represent the workload in bits.

[3] Notice that, for convenience, the arrival envelope is discretized in time and the service envelope is discretized in bits. However, to perform the comparative computations of Section 4, both are expressed in discrete time rates with service interpolated.

that $\vec{R}_k^{i,S}$ contains normalized (on intervals of length I_k, i.e., $\left(\frac{\vec{R}_k^{i,S} I_k}{I_k}\right)$) samples of the service envelope $S^i(I_k)$.

Furthermore, note that, according to Section 3.3.1, the measured class must be backlogged in order to infer its service rate. However, the measured class does not require *other* classes to be backlogged when monitoring its service, as this information is indirectly revealed by fluctuations in its own measurements.

Finally, observe that, in the case of the empirical *aggregate* service model, we compute the first two moments of the system rate, while in the case of the *class* service model, we retain the measurement vector $\vec{R}_k^{i,S}$. This is due to specific inference methodology, as the first two moments of the available aggregate service rate, together with the first two moments of each class arrival rates are used for obtaining expected class service rate distributions for different schedulers. On the other hand, empirical *class* service rate measurements are used for detecting the scheduling discipline itself, as will be explained in detail in the following section.

4 Service Inference

Here, we explain how to use both theoretically ideal and measured envelopes as described above to characterize elements and parameters of the multiclass system. In Section 4.1, we explain concept of empirical service rate distributions, while in Sections 4.2 and 4.3, we present parameter estimation and scheduler inference methodologies. Finally, in Section 4.4, we summarize and discuss the proposed methodology.

Under a particular scheduler hypothesis, we perform Maximum Likelihood Estimations (MLEs) of the scheduler's parameters, such as guaranteed rates in WFQ and deadlines in EDF. Using the envelope's ideal description of a class' service, we then develop hypothesis tests to infer which service discipline is employed by the system via statistical analysis of the empirical interclass sharing relationships. Finally, we select the MLEs of the unknown parameters under the inferred scheduler.

Throughout the analysis, we emphasize the important role of time scales, both in system parameter estimation and scheduling inference procedures.

4.1 Empirical Service Distributions

Here, we describe the expected distributions of service for a given arrival distribution under different service disciplines. For simplicity, we consider a two-class system and aggregate traffic $A^i[s, s+t]$ with a Gaussian distribution.[4] Notice that, even under Gaussian arrivals, the service envelopes will be non-Gaussian due to the nonlinearities of the multiclass server. For simplicity, we first describe the expected service distributions for constant aggregate service rate and, then, generalize the analysis for variable service rate.

Denote X_k^i as a Gaussian random variable with mean $CI_k - \sum_{n=1}^{i-1} \bar{R}_k^{n,A} I_k$, variance $\sum_{n=1}^{i-1} RV_k^{n,A} {I_k}^2$, and probability density function $p_{X_k^i}(x)$.

$$X_k^i \sim N\left(CI_k - \sum_{n=1}^{i-1} \bar{R}_k^{n,A} I_k, \sum_{n=1}^{i-1} RV_k^{n,A} {I_k}^2\right).$$

From (1), the probability density function of the service envelope $S_k^i = S^i(I_k)$ under the hypothesis that the server is SP, is given by

$$\begin{aligned} p_{S_k^i}^{SP}(x) =& P(X_k^i \leq \phi_i CI_k)\delta(x - \phi_i CI_k) + p_{X_k^i}(x) \\ & I(\phi_i CI_k \leq x \leq CI_k) + P(X_k^i \geq CI_k)\delta(x - CI_k), \end{aligned} \tag{10}$$

where $I(\cdot)$ is an indicator function and $\delta(\cdot)$ is a delta function.

Similarly, denote Y_k^i as a Gaussian random variable with mean $CI_k - \sum_{n\neq i} \bar{R}_k^{n,A} I_k$, variance $\sum_{n\neq i} RV_k^{n,A} {I_k}^2$, and probability density function $p_{Y_k^i}(y)$

$$Y_k^i \sim N\left(CI_k - \sum_{n\neq i} \bar{R}_k^{n,A} I_k, \sum_{n\neq i} RV_k^{n,A} {I_k}^2\right).$$

From (2), the probability density function of the service envelope $S_k^i = S^i(I_k)$ under the hypothesis that the server is WFQ, is given by

$$\begin{aligned} p_{S_k^i}^{WFQ}(y) =& P(Y_k^i \leq \phi_i CI_k)\delta(y - \phi_i CI_k) + p_{Y_k^i}(y) \\ & I(\phi_i CI_k \leq y \leq CI_k) + P(Y_k^i \geq CI_k)\delta(y - CI_k). \end{aligned} \tag{11}$$

Finally, define the random variable Z_k^i such that

$$Z_k^i \sim N\left(CI_k + C\bar{D}_i - \sum_{n\neq i} \bar{R}_{l_n}^{n,A} I_{l_n}, \sum_{n\neq i} RV_{l_n}^{n,A} I_{l_n}^2\right).$$

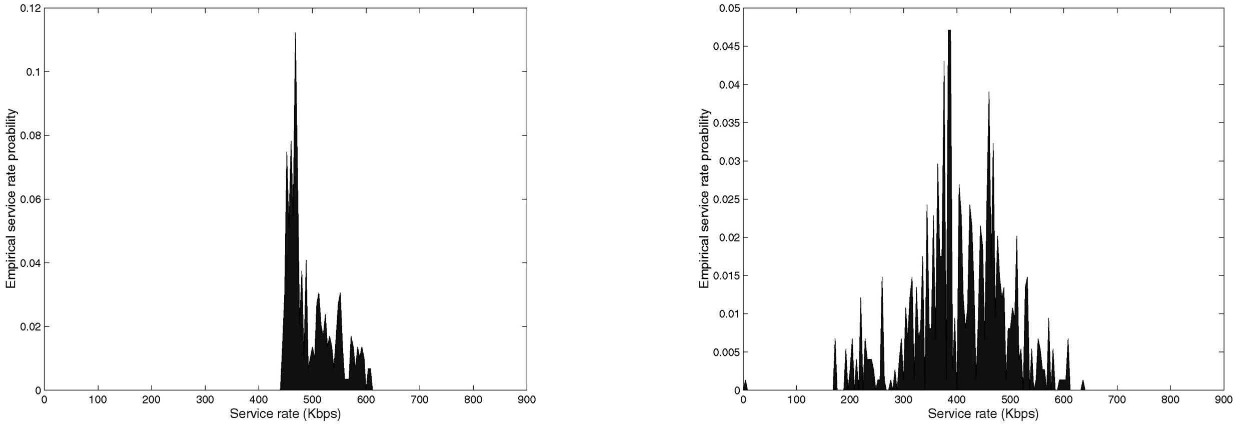
Furthermore, denote the probability density function of Z_k^i by $p_{Z_k^i}(z)$, where $l_n = k - \lfloor \delta_n - \delta_i \rfloor$ and $\bar{D}_i$ is empirical *mean* delay. From the EDF service envelope of (3), we have that the probability density function of S_k^i under the EDF hypothesis, is given by

$$\begin{aligned} p_{S_k^i}^{EDF}(z) =& P(Z_k^i \leq 0)\delta(z) + p_{Z_k^i}(z) I(0 \leq z \leq CI_k) + \\ & P(Z_k^i \geq CI_k)\delta(z - CI_k). \end{aligned} \tag{12}$$

Examples of empirical class service rate distributions for WFQ and SP servers are presented in Figs. 4a and 4b. The interval length I_k is 400 ms and additional parameters such as traffic load and statistical workload characterization are given in Section 5.

We make several observations about the figures. First, the service distribution of WFQ visibly exhibits the truncated behavior defined by (11): This is due to WFQ's guaranteed rate which lower bounds the service. Second, observe that no such "hard" lower border exists for SP without strict rate limiters on all higher priority traffic classes. Finally, notice that upper limits on the density functions are not evident here, as in this case, neither class reached its upper limits due to statistical fluctuations in the

[4] The motivation behind the Gaussian traffic characterization is that it is very simple and accurate when a large number of sources are multiplexed (via the Central Limit Theorem). In fact, it has been shown in [26] that aggregation of even a fairly small number of traffic streams is usually sufficient for the Gaussian characterization of the input process to accurately predict queue performance. However, note that the Gaussian assumption is not necessary for traffic envelopes; see [19] for example. Regardless, we make the assumption in this paper as it makes our solution more computationally efficient while also retaining a high degree of accuracy.



(a)

(b)

Fig. 4. (a) WFQ Service Rate Histogram and (b) SP Service Rate Histogram. Service Rate Histograms for WFQ and SP.

demand of the other class. Also, it should be noted that the variance of arrival traffic plays a key role in revealing the scheduler type. For example, as the variance of arrivals becomes larger, according to (11), the probability of clipping lower service bound increases. Likewise, the probability of detecting scheduler correctly increases since the service distributions for WFQ and EDF schedulers become statistically more differentiated.

Next, we describe the expected distributions of a class' service for a given arrival distribution and for a given aggregate distribution. Without loss of generality, we assume Gaussian distribution[5] for C_k.

Denote the probability density function of the aggregate service envelope in time scale I_k with $p_{C_k I_k}(y)$. Next, denote the probability density function of a class service envelope in the same scale, but, for a given aggregate service (e.g., (11) for WFQ), with $p^{SCH}_{S_k^i}(x|C_k I_k = y)$, where SCH denotes scheduling discipline which can be SP, WFQ, or EDF. Then, the probability density function of the class service envelope is given by

$$\tilde{p}^{SCH}_{S_k^i}(x) = \int_0^{\infty} p_{C_k I_k}(y) p^{SCH}_{S_k^i}(x|C_k I_k = y) dy. \qquad (13)$$

Observe that, when the aggregate service rate is constant, i.e., when $p_{C_k I_k}(y) = \delta(y - C I_k)$, then $\tilde{p}^{SCH}_{S_k^i}(x) = p^{SCH}_{S_k^i}(x|y = C I_k)$, which is a special case we treated above.

4.2 Parameter Estimation under Scheduler Hypothesis

Here, we describe how a scheduler's parameters such as weights in WFQ and deadlines in EDF can be estimated under the hypothesis of a particular scheduler EDF, WFQ, or SP. We employ the Generalized Likelihood Ratio Test by first, obtaining Maximum Likelihood Estimates of unknown parameters under each hypothesis and, then, using the likelihood ratio test. We then show how the scheduling mechanism itself can be inferred by choosing the more likely hypothesis as the true one. Finally, the MLEs of unknown parameters under the chosen hypothesis become the final estimates.

4.2.1 SP Relative Priority Estimation

The first problem is to determine unknown class' priorities under the hypothesis that server is SP. Given G classes, there are $G!$ combinations of relative class' prioritizations and our goal is to find the most probable one. Thus, for $j = 1, \cdots, G!$, denote $\vec{\epsilon}_j$ as a $j\mathrm{th}$ priority vector corresponding to the $j\mathrm{th}$ priority combination, e.g., $\vec{\epsilon}_1 = (1, \cdots, G)$. Given the observations of each class' service in intervals of length I_k, we use MLE to determine the most likely priority vector $\vec{\epsilon}_j$ as

$$\hat{\vec{\epsilon}}_{j,k} = \underset{\vec{\epsilon}_{j,k}}{\operatorname{argmax}} \, \tilde{p}^{SP}(\vec{R}_k^{1,S} I_k, \vec{R}_k^{2,S} I_k, \cdots, \vec{R}_k^{G,S} I_k | \vec{\epsilon}_{j,k}), \qquad (14)$$

where

$$\tilde{p}^{SP}(\vec{R}_k^{1,S} I_k, \vec{R}_k^{2,S} I_k, \cdots, \vec{R}_k^{G,S} I_k | \vec{\epsilon}_{j,k}) = \\ \prod_{m=1}^{M} \tilde{p}^{SP}_{S_k^1}(x = \vec{R}_{k,m}^{1,S} I_k) \prod_{n=1}^{N} \tilde{p}^{SP}_{S_k^2}(y = \vec{R}_{k,n}^{2,S} I_k) \cdots \\ \prod_{l=1}^{L} \tilde{p}^{SP}_{S_k^G}(z = \vec{R}_{k,l}^{G,S} I_k),$$

and M, N, and L denote the respective sizes of $\vec{R}_k^{1,S}$, $\vec{R}_k^{2,S}$, and $\vec{R}_k^{G,S}$. Thus, we employ a numerical search over all possible priority combinations, and find the most likely one for each time scale I_k. The final solution $\vec{\epsilon}_j$ is obtained by using the majority rule over all time scales.

4.2.2 WFQ Relative Weight Estimation

The next problem is to determine each class' unknown weight parameter under the hypothesis that the server is WFQ. Given the observations of each class' service in intervals of length I_k, we use the MLE to estimate the unknown parameters ϕ_i as

[5] Observe that, if the distribution is non-Gaussian, we can simply estimate the pdf of the aggregate service and use it in (13).

$$\begin{aligned}(\hat{\phi}_{1,k},\hat{\phi}_{2,k},\cdots,\hat{\phi}_{G,k}) = &\underset{(\phi_{1,k},\phi_{2,k},\cdots,\phi_{G,k})}{\operatorname{argmax}}\ \tilde{p}^{WFQ} \\ &(\vec{R}_k^{1,S}I_k,\vec{R}_k^{2,S}I_k,\cdots\vec{R}_k^{G,S}I_k,|\phi_{1,k},\phi_{2,k},\cdots,\phi_{G,k}),\end{aligned} \qquad (15)$$

where $\tilde{p}^{WFQ}(\vec{R}_k^{1,S}I_k,\vec{R}_k^{2,S}I_k,\cdots,\vec{R}_k^{G,S}I_k|\phi_{1,k})$ is computed similarly, as for the SP scenario explained above. Since a closed form expression cannot be found for the MLE in (15), we employ a numerical grid search by maximizing the likelihood function with respect to the unknown parameters $\phi_{i,k}$ in the interval $[0,1]$, such that $\sum_{i=1}^{G}\phi_{i,k}=1$. (Notice that the unknown values have known and closed borders so that the grid numerical search is justified.) The estimate is obtained for each interval I_k independently, and the final estimate of $\hat{\phi}_i$ is computed by averaging the estimates for different time scales.

The physical interpretation of (15) is as follows: The relative class weight estimation can be performed only over time intervals when all classes are backlogged since it is only during such intervals that all classes incur their lower bounds in service. Such intervals cause peaks at the lower clipping of the service rate distribution and also maximize the joint distribution of (15).

For EDF, similar expressions can be derived by applying the same methodology of using the EDF service envelopes to compute the MLE expressions for the class delay bounds, and performing a grid search to estimate $\hat{\delta}_i$.

4.2.3 Rate-Limiter Parameter Estimation

Thus far, we have considered work-conserving service disciplines. Here, we develop a measurement methodology applicable to rate-limiters, i.e., non work-conserving elements which limit a flow's arrivals to within a prespecified constraint. For a single token bucket with a bucket depth of one packet, the rate limiter for class i is characterized by an unknown rate r_i. The key problem is to distinguish such a limit on class is service from throughput limits due to the workloads of other traffic classes and other mechanisms in the multiclass scheduler.

Thus, the goal is to find the maximum likelihood estimation of r_i under the hypothesis of a particular scheduler (inferred as above). With rate limiters, the service envelopes of (1), (2), and (3) have r^i in place of C as the maximum service rate. Thus considering the EDF hypothesis as an example, the maximum likelihood estimation of r_i can be computed as

$$(\hat{r}_k^i,\hat{\delta}_i) = \underset{r_k^i,\delta_i}{\operatorname{argmax}}\,\tilde{p}^{EDF}(\vec{R}_k^{1,S}I_k,\vec{R}_k^{2,S}I_k,\cdots,\vec{R}_k^{G,S}I_k|r_k^i,\delta_i). \qquad (16)$$

Estimation of rate limiter parameters highlights the importance of time scales. This is illustrated in Fig. 5, which depicts the probability that a class transmits at the rate limiter's bound as a function of interval length. The scenario is a two-class, class-based fair queuing scheduler with class weights of 0.5. The classes have 60 and 40 exponential on-off flows with peak rate 32 kb/s. The figure shows the empirical probability that the aggregate traffic of class 1 transmits at its rate limit of 1 Mb/s as a function of interval length. As shown, for short time scales, this occurs quite frequently whereas it is increasingly rare over longer time scales. While this property is an inherent characteristic of any variable rate flow, the key point is that inference of rate limiter parameters at long time scales is inhibited by flows becoming less and less likely to send at peak rates for sustained periods. As a consequence, measurement of multilevel leaky buckets, which *require* longer time scale measurements due to traffic constraint functions which shape the traffic differently at different time scales (see [27], for example), will incur higher measurement errors.

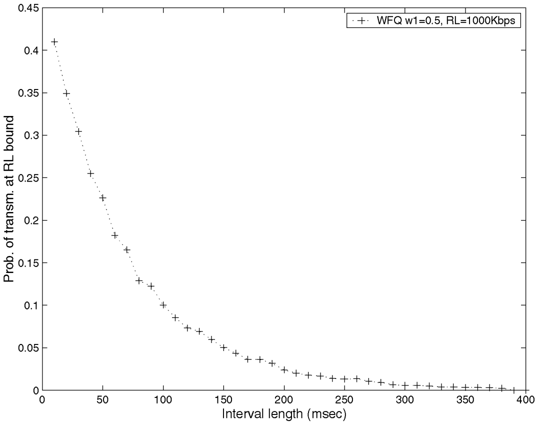


Fig. 5. Probability of transmitting at rate limiter bound.

4.3 Scheduler Inference

The above technique allows estimation of a scheduler's parameters under the hypothesis of a particular scheduler. Here, we show how the scheduling policy itself can be inferred. The key technique is to choose the hypothesis that makes the measured service observation most likely. We highlight the importance of variability of both class arrival and aggregate service envelopes and the crucial role of time scales.

To infer which service discipline is the most likely under the observations, we apply the Generalized Likelihood Ratio Test (GLRT), a detection method in which estimated unknown parameters are used in the likelihood ratio test. Thus, for each time scale I_k, we have the scheduler hypothesis test given by

$$\frac{\tilde{p}^{EDF}(\vec{R}_k^{1,S}I_k)\tilde{p}^{EDF}(\vec{R}_k^{2,S}I_k)\cdots\tilde{p}^{EDF}(\vec{R}_k^{G,S}I_k)}{\tilde{p}^{WFQ}(\vec{R}_k^{1,S}I_k)\tilde{p}^{WFQ}(\vec{R}_k^{2,S}I_k)\cdots\tilde{p}^{WFQ}(\vec{R}_k^{G,S}I_k)} \overset{EDF}{\underset{WFQ}{\gtrless}} 1, \qquad (17)$$

for EDF and WFQ hypothesis. If there are more than two hypothesis, then similar tests are used for finding the most likely one. Since we apply GLRT for *all* time scales I_k, and should provide only one final decision about the scheduler hypothesis, our next problem is to determine which time scales to consider in determining the most likely scheduling policy. As explained above, increased variability of arrivals makes the service rate distributions more statistically differentiated. In contrast, increased variability of the aggregate available service has opposite effect. An example is given in Fig. 6, which depicts the service rate distributions in a QoS enabled Web server with FCFS and WFQ scheduling policies implemented in the listen queue. The curves shown in Fig. 6 are numerically computed using (11) and (12). The interval length is 200 ms and additional

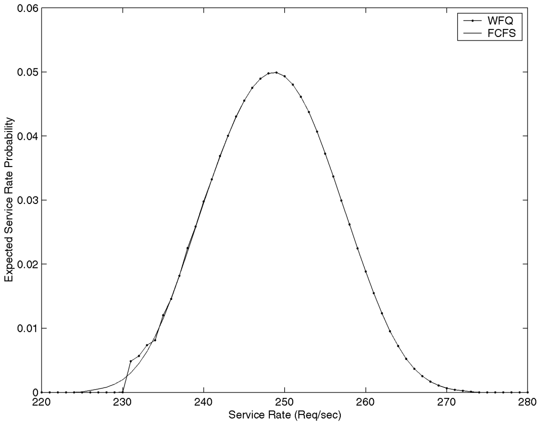


Fig. 6. WFQ and FCFS expected probability density functions.

simulation parameters are given in Section 5. Observe that, due to large variance of the aggregate service rate compared to the variance of the class arrival rate (the actual ratio in the experiment is 2.1 for 200 ms time scale), no hard lower bound is observable in Fig. 6 for WFQ. This is because the high variability of the *aggregate* service envelope directly affects the variability of *class* service envelopes according to (13). Also, observe that the curves in Fig. 6 are almost the same, except for the slight difference in the rates around 230 req/sec. Thus, increased variability of the *aggregate* service rate makes the inference problem harder, as the service rate distributions become statistically closer.

To include this effect in our inference procedure, we define a rate variance ratio

$$\gamma_{k,i} = \frac{\sum_{n \neq i} RV_k^{n,A}}{CV_k}$$

for each class i and time scale I_k as the measure of detection accuracy. A decision from the particular time scale I_k is included in determening the final decision if the following rate-variance condition

$$\gamma_{k,i} > \gamma^* \tag{18}$$

is met for a certain threshold γ^*. Thus, we choose only those time scales that have larger probability of correct service inference, i.e., time scales for which service rate distributions are statistically more differentiated. The final decision is obtained using majority rule over time scales and classes that satisfied the rate variance condition. While analytical calculation of a threshold γ^* for $\gamma_{k,i}$ that guarantees a desired probability of correct detection is intractable (because of nonlinearities in expected class service distributions), we experimentally find the relationship between probability of correct decision and threshold using trace driven simulation in Section 5. This relationship can serve as a guideline for setting the threshold γ^* in practice.

The physical interpretation of the rate variance condition is as follows: In the network case, CV_k is the measure of the variability of unknown cross traffic. If the variability of the cross traffic increases, the probability to correctly detect the scheduling policy decreases. Likewise, in the Web server case, if the variability of application layer service increases (e.g., due to file size distribution or caching), the probability to correctly detect the differentiated policy implemented either in the listen queue or CPU decreases because class service measurement will be "blurred" due to this effect. Furthermore, this illustrates challenges in providing strong capacity guarantees in systems in which it is not possible to control all the elements of the system that influence service times (i.e., file sizes in this particular case).

Observe that when $CV_k = 0$ for all k, all time-scales are included in measurements, which is exactly the case when aggregate capacity is constant. Another extreme case is when $RV_k^{n,A} = 0$, i.e., it is not possible to infer the scheduling discipline when there is no variability in arrivals.

4.4 The Algorithm Summary and Discussion

Fig. 7 summarizes the proposed methodology, which is divided into measurement, parameter estimation, and scheduler-inference procedures.

Since most of the statistical inference techniques proposed in the parameter-estimation procedure are iterative, it may become a computational botteleneck when the number of classes G increases. However, note that the overall algorithm can be implemented in a computationally efficient way by decoupling measurement procedure on one side from parameter-estimation and scheduler-inference procedures on the other. For example, data can be collected within measurement windows of the length of several seconds (see [22] for implementation details), followed by a duration of several tens of seconds to allow computationally intensive parameter-estimation procedure to converge.

Also, note that the Gaussian traffic characterization substantially contributes to computational efficiency. This is because Gaussian processes are completely specified by their first two moments, which makes the Gaussian traffic characterization ideal from a measurement point of view since measuring statistics beyond the second moment is often impractical.

5 Experimental Investigations

In this section, we perform a set of simulation experiments to evaluate the effectiveness of the multiclass inference tools described above. We study WFQ weight estimation, inference of the service discipline for EDF, SP, and WFQ as well as "measurable regions," the conditions necessary to obtain accurate estimates of WFQ weights. Experiments are performed for both QoS network routers and QoS enabled Web servers.

All networking simulations are performed with the *ns*-2 simulator with a single router and various numbers of hosts in the topology of Fig. 2. The link capacity is 1.5 Mb/s and packet sizes are 500 and 100 Bytes, as specified in the various experiments. The minimum interval length for measuring arrival and service envelopes is I_1 = 10 msec and the maximum interval-length for measurement is 0.5 sec for a 50-point arrival envelope. For these experiments, the measurement window T is varied in the experiments from two to 10 sec as indicated. We consider two traffic classes and EDF, WFQ, and SP scheduling.

For the Web server simulations, we modified the simulator described in [11], that was developed to closely approximate the behavior of OS management or CPU, memory, and caching/disk storage. A simplified model of a distributed Web server is depicted in Fig. 8. The simulated

A. Measurement:

Denote θ as the set of all measurement time scales. For $I_k \in \theta$ compute:

1. $\bar{R}_k^{i,A}$ *and* $RV_k^{i,A}$ *for* $i = 1, \cdots, G$, *(Eq. (4) and Eq. (5)).*

2. $\vec{R}_k^{i,S}$ *for* $i = 1, \cdots, G$, *(Eq. (6) and Eq. (7)).*

3. $\bar{C}_k$ *and* CV_k *(Eq. (8) and Eq. (9)).*

B. Parameter Estimation:

4. *Determine a subset of time scales* $\psi_i \subseteq \theta$ *for which the rate-variance condition (Eq. (18)) holds, for* $i = 1, \cdots, G$.

5. SP server: *for* $I_k \in \psi_i, i = 1, \cdots, G$,

(a) *Determine* $\hat{\vec{\epsilon}}_{j,k}$ *(Eq. (14), using Eq. (10) and Eq. (13)).*

(b) *Compute* $\tilde{p}^{SP}(\vec{R}_k^{1,S} I_k, \vec{R}_k^{2,S} I_k, \cdots, \vec{R}_k^{G,S} I_k | \vec{\epsilon}_{j,k} = \hat{\vec{\epsilon}}_{j,k})$

6. WFQ server: *for* $I_k \in \psi_i, i = 1, \cdots, G$,

(a) *Determine* $\hat{\phi}_{1,k}, \cdots, \hat{\phi}_{G,k}$ *(Eq. (15), using Eq. (11) and Eq. (13)).*

(b) *Compute* $\tilde{p}^{WFQ}(\vec{R}_k^{1,S} I_k, \vec{R}_k^{2,S} I_k, \cdots, \vec{R}_k^{G,S} I_k | (\phi_{1,k}, \cdots, \phi_{G,k}) = (\hat{\phi}_{1,k}, \cdots, \hat{\phi}_{G,k}))$

7. EDF server: *for* $I_k \in \psi_i, i = 1, \cdots, G$,

(a) *Determine* $\hat{\delta}_{1,k}, \cdots, \hat{\delta}_{G,k}$ *(using Eq. (12) and Eq. (13)).*

(b) *Compute* $\tilde{p}^{EDF}(\vec{R}_k^{1,S} I_k, \vec{R}_k^{2,S} I_k, \cdots, \vec{R}_k^{G,S} I_k | (\delta_{1,k}, \cdots, \delta_{G,k}) = (\hat{\delta}_{1,k}, \cdots, \hat{\delta}_{G,k}))$

C. Scheduler Inference:

8. *Determine the most likely scheduler by finding* $max(\tilde{p}^{SP}(.), \tilde{p}^{WFQ}(.), \tilde{p}^{EDF}(.))$, *where* $\tilde{p}^{SP}(.)$, $\tilde{p}^{WFQ}(.)$ *and* $\tilde{p}^{EDF}(.)$ *are computed in 5(b), 6(b) and 7(b), respectively.*

9. *For the most likely hypothesis, determine final parameter estimates, e.g., for WFQ server* $\hat{\phi}_i = \frac{\sum_{I_k \in \psi_i} \hat{\phi}_{i,k}}{\sum_{I_k \in \psi_i} 1}$, *for* $i = 1, \cdots, G$.

Fig. 7. Summary of the measurement/inference algorithm.

server has a listen queue in which all incoming requests are queued before being serviced. Upon arrival, each request is queued onto the listen queue or dropped if the listen queue is full. Processing a request requires the following steps: dequeuing from the listen queue, connection establishment, disk reads (if needed), data transmission and, finally, connection tear down. We set a transfer time of 0.41 ms per 4 KB (resulting in the peak transfer rate of 10 MB/s). We implemented WFQ scheduling in the server listen queue and CPU scheduling algorithm. CPU differentiation is implemented such that each traffic class is guaranteed its fixed share of CPU time as long as it is backlogged, while each request from the same class is given a fair share of the CPU time within that class. For example, if there are two requests from class 1 and two other requests from class 2, and WFQ weights are 0.7 and 0.3, then each request from class 1 is given 35 percent of CPU time, while each request from class 2 is given 15 percent of CPU time. Maximum CPU time per request is 100 ms. The trace used in our simulation is generated from the CS department server log at Rice University. We simulate interarrival times as exponential.

5.1 WFQ Weight Estimation

5.1.1 Network Router Experiment

Here, we experimentally investigate the statistical properties of the WFQ weight estimation algorithm. In this scenario, the system has from 65 to 68 exponential on-off sources with on-rate 32 kb/s and on and off periods of 0.36 sec. Moreover, there are from 25 to 28 sources of the same type for class 2. The number of flows in the system is varied to simulate flow-level arrivals and departures which are common in a real system. The true WFQ weights are $\phi_1 = 0.7$ and $\phi_2 = 0.3$. The packet size is 500 Bytes.

In the experiments, 50 simulation runs are performed corresponding to each data point in Fig. 9a. For a particular simulation, the measurement window T is set to two, five, or 10 sec as reported on the horizontal axis. Each point on the plot indicates the maximum likelihood estimation of ϕ_1, $\hat{\phi}_1$, using the methodology of Section 4.

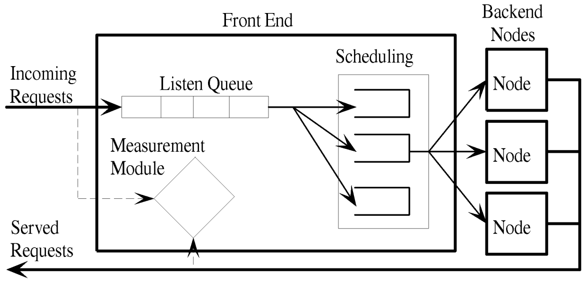
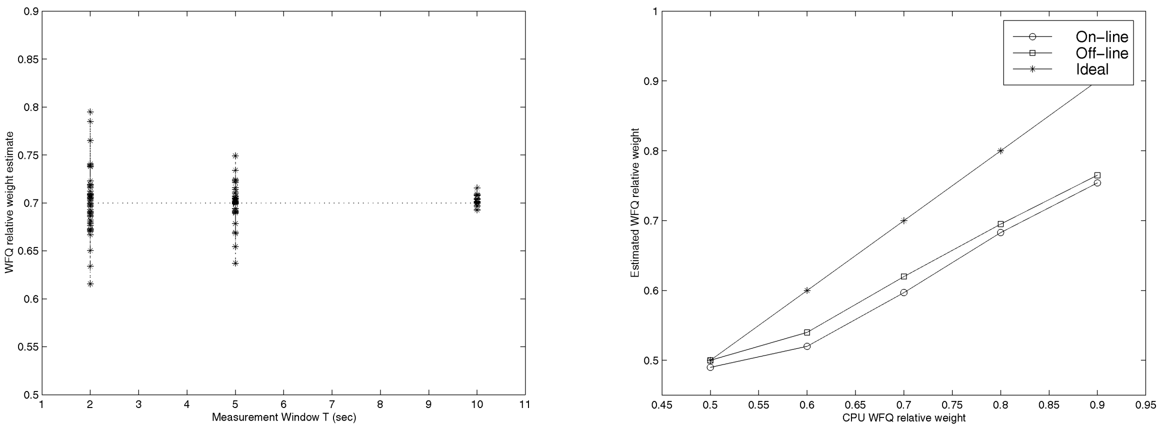


Fig. 8. Distributed QoS Web server.



(a)

(b)

Fig. 9. WFQ weight estimation in a router and a Web server scenarios. (a) WFQ weight estimation versus measurement window T. (b) WFQ weight estimation versus CPU weight.

First, observe that the variance of the estimator reduces with increasing measurement period T, due simply to the fact that more sample points are available with larger T. This is because $\frac{T}{I_k}$ increases with T, where I_k is the length of a particular interval. For example, with T = 2 sec, 95 percent of the weight estimations are within 11 percent of the true value, whereas with T = 10 sec, 95 percent of the weight estimations are within 1.4 percent of the true value. However, T should not be set arbitrarily large, as longer-time-scale fluctuations due to flow arrivals and departures may introduce nonstationarities which would bias the tests. While the number of flows in the system did vary in these simulations, as defined above it is within a range of 5 to 10 percent of the system load.

5.1.2 QoS Web Server Experiment

Here, we experimentally investigate the WFQ weight estimates in the QoS enabled Web server. We estimate relative class weights and the simulation setup consists of two traffic classes for which CPU WFQ weights are varied from 0.5 to 0.9. We perform two types of experiments. In the first one, called online, we passively monitor the requests from two traffic classes entering and leaving the system. The total arrival rate is 1800 req/s, and the mean arrival rate of each class is proportional to its relative CPU weight. For the relative classes weight estimates, we use inference methodology presented in this paper. On the other side, we perform another set of experiments, called offline, where each class is artificially probed with the arrival rate of 2500 req/s, thus making the total request arrival rate as high as 5000 req/s and saturating the Web server. Recall that the experiments probing at a high rate in all classes yield a true interclass relationships—lower service bounds in this particular case. In the offline experiments, we measure *mean* service rates for both classes (m_1 and m_2), and the appropriate estimator for relative class-1 weight is $\hat{\phi}_1 = \frac{m_1}{m_1+m_2}$.

In the experiments, 10 simulation runs are performed for each WFQ weight shown on the x-axis of Fig. 9b and the averaged WFQ estimates are computed using both online and offline estimation procedures. First, observe that the results for the online experiments are just slightly biased when compared to the offline case. However, the overall results confirm the accuracy of the passive monitoring estimation methodology developed in this paper.

Second, note that the ideal class' CPU relative weights are not reached in the online nor offline experiments. For example, a CPU weighted share of 0.7 is revealed as a weighted share of 0.59 in the online case (consider a point with coordinates (0.7; 0.59) in the x-y plane of Fig. 9b). This effect is due to a preemptive nature of CPU scheduling, i.e., the fact that the request service time does not depend only on CPU scheduling weight, but also on the number of requests present in the system and the CPU time required by the request. This example emphasizes an important feature of our inference methodology—it estimates net service class parameters, as seen by users from the system edge.

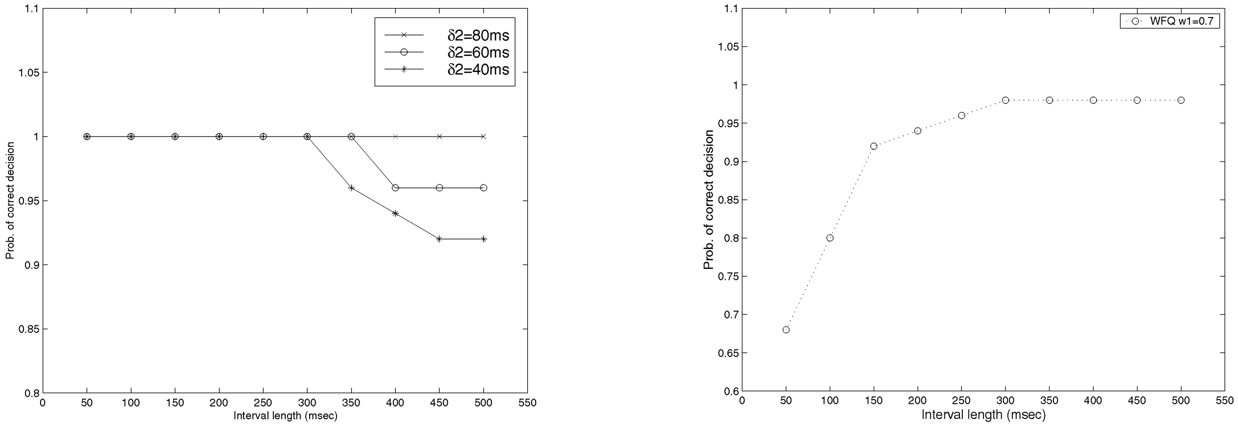
5.2 Scheduler Inference

As described in Section 4, the above WFQ weight estimations can only be performed under the hypothesis that the server is WFQ. Thus, statistical tests are necessary to infer the scheduling mechanism itself.

5.2.1 Constant System Capacity

Here, we describe simulation experiments for scheduler inference using the same number of sources for each class and the same packet size as in the previous network router experiment. Fig. 10 depicts the experimental probability of correct decision versus time scale for the respective correct hypothesis of EDF and WFQ. In both cases, 50 simulations are performed and the probability of correct decision is computed as the number of correct decisions versus total number of tests for each time scale (recall the final decision is performed by majority rule).

For the experiments of Fig. 10a, the correct hypothesis is EDF with delay bounds of δ_1 = 20 ms for class 1 and δ_2 = 40, 60, and 80 ms for the three curves for class 2. As indicated in the figure, EDF is correctly inferred 100 percent of the time at short time scales (I_k up to 300 ms), while less frequently for longer interval lengths, especially as $\delta_2 - \delta_1$ decreases.



(a)

(b)

Fig. 10. Probability of correct decision versus time scale. (a) EDF and (b) WFQ.

Yet, in all cases, the probability of correct decision is no less than 92 percent. The reason that the probability of correct decision decreases as $\delta_2 - \delta_1$ decreases is that there is less and less differentiation provided by the scheduler, making the service envelopes statistically closer and the inference problem more difficult. Indeed, if $\delta_2 = \delta_1$, the scheduler is actually performing FCFS, as is also evident from the service envelope in (3).

Regardless, in all cases, the correct *final* decision is made as majority rule is performed over different time scales, and incorrect decisions at a particular time scale are never frequent enough to form a majority. Also, observe that *all* time scales are included in determining final decision, as $CV_k = 0$, i.e., rate variance condition is fulfilled for all k.

Fig. 10b depicts the experimental results for WFQ. Observe that, in this case, the correctness ratio is quite poor on shorter time scales. This is due to the mismatch between the fluid approximation used in the analytical model and the packet-layer simulations. In particular, over short time intervals, the fluid approximation does not hold and not every packet gets serviced at rate $\phi_i C$ (indeed, see [28] for a detailed discussion of such short-time-scale unfairness). In this case, such errors impact the final decision and the overall correctness probability is 0.94 (less than the correctness of one achieved in the EDF case) as the short-time-scale errors form a majority in 6 percent of the cases.

Finally, notice that the relationship of the probability of correct decision and time scale are reversed for WFQ as compared to EDF. The reason for this is that over longer time scales, WFQ overcomes packet level unfairness and, when flows are backlogged for long durations, it can become quite clear (statistically) that there is a minimum guaranteed service rate clipping the distribution of the service envelope. In contrast, for EDF, the differences are most pronounced for small interval lengths where the shifts in the arrival envelopes (compare (3)) are more prominent.

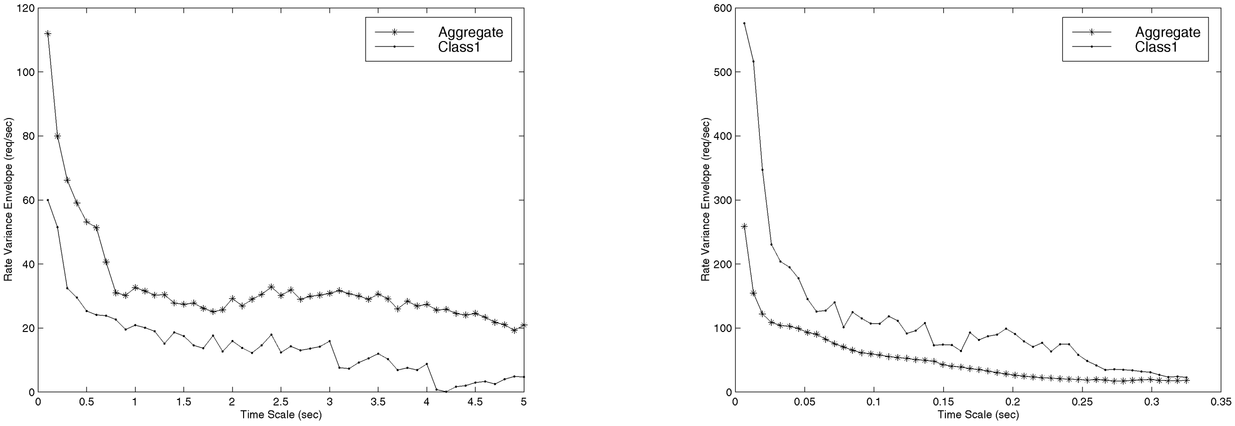
5.2.2 Variable System Capacity

Our goal here is to explore and quantify the extent to which the system variability influences the probability of correct scheduler inference. We perform experiments with WFQ and FCFS scheduling policies[6] in a listen queue of a QoS enabled Web server. In the simulation, we use traces generated from the Computer Science department server log at Rice University. The Poisson arrival rate was 125 req/s, with arrival rate of 87 req/s and 38 req/s for classes 1 and 2, respectively, i.e., the ratio of means of arrival rates for traffic classes was 7:3. WFQ scheduler weights in the listen queue were set to 0.7 and 0.3.

Fig. 11a depicts the class 1 arrival rate variance envelope and aggregate rate variance envelope for a single simulation run with the Rice CS trace. Observe that the aggregate rate variance is larger than the class arrival rate variance for all measured time scales. This is not surprising since the arrival process is Poisson, while heavy tailed file size distribution cause increased variability of serviced requests over longer time scales.

Recall that we have defined a rate variance ratio as a measure of detection accuracy. High values of this ratio indicate high detection probability and vice versa. To validate this technique, we run the simulation 10 times for different random seeds, five times for each scheduler. Thus, the total number of GLRT tests is 500 (in each simulation run we performe 50 GLRT tests corresponding to 50 time scales). The percentage of correct scheduler detection averaged over all 500 tests is 0.53. Namely, it is 1.0 for FCFS (all 250 tests for FCFS scheduler were correct), while it is 0.06 for WFQ (only 15 out of 250 decisions were correct). This is because the aggregate rate variability is too high compared to class arrival variability (i.e., the rate variance ratio is too small, less than 0.6 in all cases). Thus, variations of service times due to variability of file sizes and caching dominate the inference tests, thereby overwhelming the scheduling policy implemented in the Web server's listen queue and making the system to statistically appear closer to FCFS than WFQ when observed from the edge. An analogous networking example would be the one when highly bursty cross-traffic flow, which we cannot measure from the edge, interferes with the edge-measured traffic in the bottleneck router. In this case again, high variance of the aggregate service envelope would overwhelm the inference of bottleneck node's scheduling policy.

[6] Recall that EDF scheduler with $\delta_i = 0$ performs FCFS.



(a)

(b)

Fig. 11. Rate variance envelope versus measurement interval. (a) Rice CS trace. (b) 4K Rice CS trace.

Next, to further explore and *quantify* the influence of the aggregate rate envelope variability on correct scheduler inference probability, and to be able to determine and experiment with different values for threshold γ^*, we change the distribution of file sizes of our trace by replacing all files larger than 10 KBytes with files of 4 KBytes.[7]

Fig. 11b depicts the class 1 arrival rate and aggregate rate variance envelopes for a single simulation run with this changed trace. Observe that the rate arrival variance is now larger than the aggregate rate variance for all time scales, which is a direct consequence of the change of the file size distribution. For this setup, we again run 10 simulations, five for each scheduler.

Recall from Section 4 that if the rate variance ratio $\gamma_{k,i}$ is greater than the threshold γ^*, the decision from particular time scale I_k is included in determining final decision. For example, for threshold $\gamma^* = 1.0$ total number of GLRT tests is 500. Out of these 500 tests, 492 fulfilled the condition that $\gamma_{k,i} > 1.0$ for both classes $i = 1, 2$. Further, in 305 out of these 492 tests, the correct scheduler is detected. Thus, the probability of correct scheduler detection averaged over *all time scales* for which the rate variance condition is fulfilled is 0.62. However, the majority rule over those time scales that fulfilled the rate variance ratio condition gives final correctness detection probability of 0.9 (only once failed for WFQ scheduler).

The probability of correct detection increases when threshold γ^* increases. For example, when $\gamma^* = 2.0$, the per-time-scale correctness probability increases to 0.94, while the majority rule over time gives final correctness probability of 1.0. However, one should not use arbitrarily large threshold values γ^*, as the number of time scales for which the condition from (18) is fullfiled decreases when γ^* increases. Finally, note that reduced variability of aggregate service envelope, as compared to the example from Fig. 11a, increases the probability of correct scheduler detection.

5.3 Measurable Region

The methodology presented in this paper is based on *passive* measurements, i.e., no probing packets are transmitted to modify the system's workload. However, with passive monitoring, it is possible that other classes' particular workloads *prohibit* inference of certain network elements. For example, in the extreme case that all other classes are idle, it is impossible to detect a guaranteed minimum rate. Similarly, the multiclass nature of the scheduler itself would not be measurable, and only rate-limiter parameters could be obtained. We refer to the required workload to measure a particular network behavior as the *measurable region*.

Here, we address the issue of the conditions necessary to infer lower and upper service limits for WFQ schedulers. For the simulations, each flow has on and off periods of 0.36 sec and on-rate 32 kb/s. The packet size is 100 Bytes.

Fig. 12 depicts the resulting measurable regions for WFQ weight estimates. Each point represents the minimum number of class 1 and class 2 flows needed such that the relative weights can be estimated within 5 percent of their correct value. In other words, these curves represent the borders between measurable and nonmeasurable regions. That is, if either class has fewer flows than indicated by this measurable region, then estimation is not possible, as the conditions required for weight estimation occur too rarely. Similarly, the scheduler inference correctness probability (not shown in the figure) sharply decreases when the

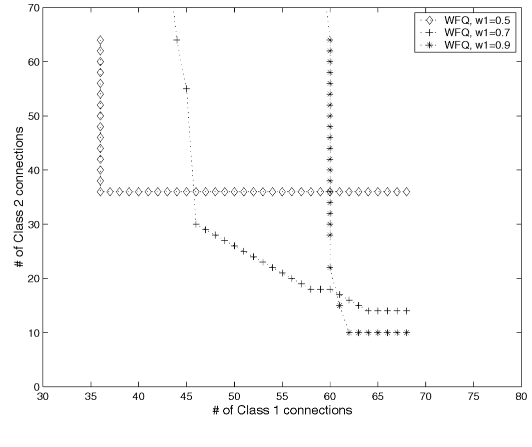


Fig. 12. Measurable region for lower service bounds.

[7] Alternative approach in changing rate variance ratio was to increase variability of arrivals.

TABLE 1
Notation

a_j^i	arrival time of request j in class i
d_j^i	departure time of request j in class i
ϕ_i	relative weight of class i in WFQ scheduler
r_i	rate limit bound of class i
$S^i(t)$	class i's theoretical service envelope over intervals of length t
$B^i(t)$	class i's theoretical arrival envelope over intervals of length t
$A^i[s, s+t]$	total class i's arrivals in the interval [s,s+t]
δ_i	class i's delay bound for EDF scheduler
I_k	interval length ($I_k = kI_1$)
$R_{k,j}^{i,A}$	class i's empirical arrival rate in the $[s+(j-1)I_k, s+jI_k]$ interval
N_k	number of successive intervals of length I_k in the measurement window T
$\bar{R}_k^{i,A}$	mean of the empirical arrival *rate* envelope of class i for intervals of length I_k
$RV_k^{i,A}$	variance of the empirical arrival *rate* envelope of class i for intervals of length I_k
$U[s, s+t]$	service received during *backlogging* interval $[s, s+t]$
$M^S(t)$	service *rate* received in the backlogging interval $[s, s+t]$
$\vec{M}_k^S$	vector of empirical service rates over backlogging intervals of length I_k
$\vec{C}_k$	vector of empirical *aggregate* service rates over backlogging intervals of length I_k
C_k	aggregate rate service envelope over intervals of length I_k
$\vec{R}_k^{i,S}$	vector of empirical class i's service rates over backlogging intervals of length I_k
$p_{S_k^i}^{SCH}(\cdot)$	pdf of $S^i(I_k)$ for scheduler SCH and for constant aggregate service capacity
$\bar{D}_i$	mean delay of class i requests
$p_{C_k I_k}(\cdot)$	pdf of the aggregate service envelope in time scale I_k
$\tilde{p}_{S_k^i}^{SCH}(\cdot)$	pdf of $S^i(I_k)$ for scheduler SCH and for variable aggregate service capacity
$\hat{\phi}_{i,k}$	MLE of ϕ_i in time scale I_k
$\hat{\phi}_i$	final estimate of ϕ_i
$\hat{r}_k^i$	MLE of class i's rate limit bound r_i in time scale I_k
$\gamma_{k,i}$	rate variance ratio for class i and time scale I_k
γ^*	threshold for rate variance ratio

number of flows in the system drops below the measurable region minimum.

Observe that as the weight of class 1 increases from ϕ_1 of 0.5 to 0.7 and 0.9 (corresponding to the three curves), the curves shift to the lower right indicating that a higher number of class 1 flows and lower number of class 2 flows are needed to infer ϕ_1. The reason for this is that as ϕ_1 becomes larger, a higher traffic load in class 1 is required to backlog class 1 sufficiently to estimate the guaranteed rate.

Finally, observe that a typical point on the curve refers to a relatively modest resource utilization. For example, under ϕ_1 = 0.7, at least 30 class 2 flows are required when 46 class 1 flows are present. This corresponds to an average system utilization of 62 percent, i.e., the mean utilization must be at least 62 percent to perform the measurements passively, otherwise active probing is required.

6 CONCLUSIONS

Networks and Web servers are increasingly providing quality-of-service functionalities. The goal of this paper is to provide a framework for clients of multiclass services to assess a system's core QoS mechanisms. We developed a scheme for clients to perform a series of hypothesis tests across multiple time scales in order to infer the request service discipline among class-based weighted fair queuing, earliest deadline first, and strict priority. The scheme can be applied to any other scheduler for which a statistical service envelope is derived. For a particular scheduler, we devised techniques for clients to obtain maximum likelihood estimations of the system's class differentiation parameters such as WFQ weights and EDF delay bounds. Finally, we showed how parameters of non work-conserving elements such as rate limiters can be estimated.

We evaluated the methodology in a two-class setting in both networking and QoS Web server scenarios. For networks, the results show high accuracy in both scheduler inference and unknown parameter estimation. For Web servers, we also achieve high accuracy provided that the variability of service times due to factors such as different CPU processing times, disk service times, and variable file sizes is not significantly larger than the service variability due to the other class' workload. In both cases, we utilized a general multiple-time-scale traffic and service model to characterize a broad set of behaviors within a unified framework. The inference techniques developed in this paper are generally applicable and computationally feasible up to a moderate number of service classes.

Appendix

Please see Table 1.

Acknowledgments

A subset of this work appears in the Proceedings of IEEE Infocom '01 [26]. This research is supported by US National Science Foundation Grants ANI-0085842 and ANI-0099148, the Department of Energy, and by a Sloan Fellowship.

References

[1] B. Teitelbaum et al. "Internet2 QBone: Building a Testbed for Differentiated Services," *IEEE Network,* vol. 13, no. 5, pp. 8-17, Sept. 1999.

[2] C. Dovrolis and P. Ramanathan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model," *IEEE Network,* vol. 13, no. 5, pp. 26-35, Sept. 1999.

[3] I. Stoica, S. Shenker, and H. Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks," *Proc. ACM SIGCOMM '98,* Sept. 1998.

[4] S. Blake et al. "An Architecture for Differentiated Services," Internet RFC 2475, 1998.

[5] C. Chuah, L. Subramanian, R. Katz, and A. Joseph, "QoS Provisioning Using a Clearing House Architecture," *Proc. Int'l Workshop Quality of Service '00,* June 2000.

[6] K. Nichols, V. Jacobson, and L. Zhang, "Two-Bit Differentiated Services Architecture for the Internet," Internet RFC 2638, 1999.

[7] A. Terzis, L. Wang, J. Ogawa, and L. Zhang, "A Two-Tier Resource Management Model for the Internet," *Proc. Global Internet Symp. '99,* Dec. 1999.

[8] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," *Proc. ACM SIGCOMM '00,* Aug. 2000.

[9] C. Cetinkaya and E. Knightly, "Scalable Services via Egress Admission Control," *Proc. IEEE INFOCOM '00,* Mar. 2000.

[10] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management," *Proc. ACM SIGCOMM '99,* Aug. 1999.

[11] M. Aron, P. Druschel, and W. Zwaenepoel, "Cluster Reserves: A Mechanism for Resource Management in Cluster-Based Network Servers," *Proc. ACM SIGMETRICS '00,* June 2000.

[12] N. Bhatti and R. Friedrich, "Web Server Support for Tiered Services," *IEEE Network,* vol. 13, no. 5, pp. 64-71, Sept. 1999.

[13] V. Kanodia and E. Knightly, "Multi-Class Latency-Bounded Web Services," *Proc. IEEE/IFIP Int'l Workshop Quality of Service '00,* June 2000.

[14] K. Li and S. Jamin, "A Measurement-Based Admission Controlled Web Server," *Proc. IEEE INFOCOM '00,* Mar. 2000.

[15] R.L. Carter and M.E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Performance Evaluation,* vol. 27, no. 28, pp. 297-318, 1996.

[16] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Trans. Networking,* vol. 7, no. 3, pp. 277-292, June 1999.

[17] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," *Proc. ACM SIGCOMM '00,* Aug. 2000.

[18] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *Proc. ACM SIGCOMM '02,* Aug. 2002.

[19] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Effective Envelopes: Statistical Bounds on Multiplexed Traffic in Packet Networks," *Proc. IEEE INFOCOM '00,* Mar. 2000.

[20] R. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE J. Selected Areas in Comm.,* vol. 13, no. 6, pp. 1048-1056, Aug. 1995.

[21] J. Qiu and E. Knightly, "Inter-Class Resource Sharing Using Statistical Service Envelopes," *Proc. IEEE INFOCOM '99,* Mar. 1999.

[22] J. Schlembach, A. Skoe, P. Yuan, and E. Knightly, "Design and Implementation of Scalable Admission Control," *Proc. Int'l Workshop QoS in Multiservice IP Networks,* Jan. 2001.

[23] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. Networking,* vol. 1, no. 3, pp. 344-357, June 1993.

[24] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks," *Proc. IEEE,* vol. 83, no. 10, pp. 1374-1399, Oct. 1995.

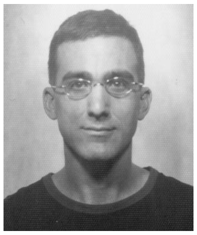
[25] D. Wrege, E. Knightly, H. Zhang, and J. Liebeherr, "Deterministic Delay Bounds for VBR Video in Packet-Switching Networks: Fundamental Limits and Practical Tradeoffs," *IEEE/ACM Trans. Networking,* vol. 4, no. 3, pp. 352-362, June 1996.

[26] A. Kuzmanovic and E. Knightly, "Measuring Service in Multiclass Networks," *Proc. IEEE INFOCOM '01,* Apr. 2001.

[27] D. Wrege and J. Liebeherr, "Video Traffic Characterization for Multimedia Networks with a Deterministic Service," *Proc. IEEE INFOCOM '96,* pp. 537-544, Mar. 1996.

[28] J. Bennett and H. Zhang, "WF^{2}Q: Worst-Case Fair Weighted Fair Queueing," *Proc. IEEE INFOCOM '96,* Mar. 1996.

Aleksandar Kuzmanovic received the BS degree from the University of Belgrade, Serbia, in 1996, and the MS degree from the same university in 1999, both in electrical engineering. Currently, he is a PhD candidate working with the Rice Networks Group in the Department of Electrical and Computer Engineering at Rice University, Houston, Texas. His research interests are in the area of algorithms and architectures for quality of service in wired networks, with an emphasis on scalable endpoint-based control and inference in the Internet.



Edward W. Knightly received the BS degree from Auburn University in 1991, the MS degree from the University of California at Berkeley in 1992, and the PhD degree from the University of California at Berkeley in 1996, all in electrical engineering. Since 1996, he has been an assistant professor in the Department of Electrical and Computer Engineering at Rice University. He currently serves on the editorial board of the *Computer Networks Journal*, *IEEE/ACM Transactions on Networking*, and *IEEE Transactions on Multimedia*. He served as cochair for the 1998 International Workshop on Quality of Service and is the finance chair for ACM MOBICOM 2002. He received the US National Science Foundation CAREER Award in 1997 and the Sloan Fellowship in 2001. His research interests are in the area of theory, algorithms, and architectures for quality of service in wired and wireless network. He is a member of the IEEE.