

Congestion Notification and Probing Mechanisms for Endpoint Admission Control ^{*}

A. J. Ganesh,[†] Peter Key,[‡] Damien Polis[§] and R. Srikant[¶]

January 5, 2005

Abstract

Recently, there has been much interest in admission control schemes that place the burden of admission control decisions on the end users. In these schemes, referred to as Endpoint Admission Control, the decision to join the network is taken by the user, based on the probing of the network using probe packets. Depending on the level of congestion, routers mark the probe packets and thus inform the user of the state of the network. In this paper, we analyze three mechanisms for providing Endpoint Admission Control: virtual-queue marking, random-early marking and tail drop. For each scheme, we analyze the probing duration necessary to guarantee the required QoS and achieve high link utilization. Our main conclusion is that very few probe packets have to be sent when early marking is used, whereas tail drop requires a large number of probe packets.

1 Introduction

Currently, the Internet offers a simple best-effort service where users are expected to adapt their transmission rates in response to congestion signals from the network. Clearly, this is not sufficient to support real-time applications that may require packet loss and delay guarantees, along with a fixed or minimum bandwidth requirement. *Diffserv* addresses this issue by defining a small number of traffic classes, and then designing

^{*}The research of the last two authors was supported by AFOSR URI Grant F49620-01-1-0365 and NSF Grant NCR-9701525.

[†]Microsoft Research Cambridge, UK; Email: ajg@microsoft.com

[‡]Microsoft Research, Cambridge, UK; Email: peterkey@microsoft.com

[§]Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois.

[¶]Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois, 1308 W. Main Street, Urbana, IL 61801; Phone: 217-333-2457; Fax: 217-244-1642; Email: rsrikant@uiuc.edu

the network so that some guarantees are provided to calls according to the class they belong to. However, to provide strict guarantees, some form of admission control is required. Alternatively, the admission of a call to the network could be based on some measurement of the network state by the routers, and some form of communication between the source and the routers to determine if the available resources are sufficient to provide the expected QoS. In a manner similar to the telephone network, *Intserv* relies on signaling. Reservation messages sent by the users are processed by the routers. If enough resources are available, the call is admitted and the routers maintain a state for this call. Otherwise, the call is rejected. While providing very good quality-of-service, *Intserv* obviously suffers from scalability problems, both due to the necessity for the routers to maintain per-call states and due to the amount of signaling required.

In this paper, we study mechanisms for distributed admission control of sources that require a fixed bandwidth from the network. Commonly referred to as Endpoint Admission Control, these schemes rely on the availability of congestion information provided by the routers through dropping or marking (through the ECN bit, for example) [12, 6]. We assume that ECN marking capability is available. When seeking admission, a call sends a certain number of probe packets. Routers mark the probe packets depending on their respective loads, according to a marking strategy. The destination then echoes the marked packets to the source, which takes the appropriate admission decision (join or not) depending on the number of marked probe packets and the QoS requirement. We emphasize that we only consider non-adaptive sources in this paper. Extending the techniques to networks containing both adaptive and non-adaptive sources is a topic for future research.

Related work in [14] looked at the design of optimal probing strategies, specifically on how long to probe for and at what rate to probe for. In contrast to the current work, [14] focused on the probing strategy rather than marking strategies, using a very simple model of marking behavior, ignoring packet-scale effects, and emphasizing the effects of delayed feedback. One of their conclusions was that a critical system parameter was the ratio of call holding time to round trip time — if this is large, then the effects of delayed feedback are small. We shall assume we are operating such a regime, and hence can ignore feedback delay. In this paper, we assume that when accepted a call does not renegotiate, and does not change its rate. The problem of how frequently to reprobe, if a call is allowed to alter its rate, was considered in [3, 4].

There has recently been considerable work on probing schemes for measuring the available or spare capacity on a bottleneck link; see [11], for example. Such schemes can also serve as the basis for admission control. The advantage of our proposed scheme, in comparison, is that it requires fewer probe packets.

In the next section, the main features of an Endpoint Admission Control are described. In Section 3,

we present a modeling framework for studying distributed control schemes. We model the traffic overhead generated by probe packets, and show how the probabilities of system occupancy can be derived. We also introduce a quantity called *service objective*, related to the quality of the service provided to the user and to the network utilization, and propose that the design parameters of the mechanism should be chosen to minimize this quantity. In Section 4 and 5, three marking mechanisms are examined through simulations and through numerical results based on analytical expressions derived using a large system approximation. The three mechanisms considered are *tail drop*, *random-early marking* (REM) and *virtual queue*. Conclusions are provided in the last section.

2 Endpoint Admission Control

A network supporting QoS must guarantee a certain level of performance to the calls sharing the network. Thus, a new call should not be admitted if the traffic it generates is likely to degrade the service offered to the calls already sharing the resources. A router is in the best position to estimate the state of its links. It can thus take accurate decisions regarding call admissions. For example, suppose that no more than n calls can share a link for QoS requirements to be satisfied. An intelligent router, like one implementing Intserv, always knows the number of calls sharing the link. In this case, the $(n + 1)^{th}$ call is always blocked. Such a centralized scheme can rely on an exact measurement of the number of calls. However, besides intelligence in the router, signaling is required for the router and the source to exchange information. This implies additional delays and overhead traffic.

As mentioned in the previous section, an alternative to centralized admission control is endpoint admission control. The performance of any admission control scheme is described by two parameters: the QoS achieved by admitted calls and the fraction of capacity utilized by admitted calls. Clearly, there is a trade-off between these quantities. As noted above, a centralized scheme achieves the maximal capacity utilization compatible with a specified QoS. An end-point admission scheme relies on packet marking to infer the current call occupancy. Since it relies on imperfect information, it has to trade off between QoS and utilization: aggressively blocking calls may achieve the desired QoS but at low utilization whereas a less aggressive scheme would occasionally violate the QoS requirement. In the next section, we define a quantity called service objective which is a weighted combination of utilization and the probability of QoS violation. We use the service objective to evaluate a number of admission control strategies, both analytically (Section IV) and through simulation (Section V).

An endpoint admission control scheme can be broken down into three parts: the probing strategy, which specifies the number of probe packets and the probing rate; the marking strategy, which describes the marking mechanism on the router side, and the decision strategy, which explains how the source relies on packet marking to take its decision. These three strategies have to be designed in such a way that the performance of an endpoint scheme approaches the performance of a centralized scheme.

2.1 Probing strategy

Each source sends a certain number of probe packets into the network. Marked probe packets carry information about the state of the network. The network provides this information by either marking or not marking each probe packet, depending upon the level of congestion. Intuitively, the more probe packets the source sends, the more accurate will be the information it gathers. However, a large number of probe packets can lead to a situation where probing itself may contribute to congestion, in addition to the congestion caused by data traffic [6]. Further, depending upon the rate at which probe packets are generated, a large number of probe packets may also require a source to wait for an unacceptably long time before it can make an admission decision. The parameter of interest in the probing process is the number of probe packets generated, which will be denoted by w . Ideally, we would like a distributed admission control scheme which performs well for small values of w . This would mean that a call does not have to wait very long before the admission decision and that probing does not significantly add to congestion.

2.2 Marking strategy

The marking strategy has to be simple enough to minimize the amount of intelligence required in the routers. Yet, marking must be done in a meaningful manner and must be tightly related to the current ability of the network to satisfy the QoS requirement. If marking is not aggressive enough, too many calls will be admitted into the network and the QoS requirement will not be satisfied. On the contrary, if marking is too aggressive, calls might be denied access to the network while enough capacity is available to accommodate them without violating the QoS requirement. The simplest strategy uses tail marking to signal impending congestion. In the following sections, we will show the disadvantage of such a technique and we will consider more robust marking strategies.

2.3 Decision strategy

The source has to make the best use of the information represented by marked probe packets. A simple decision strategy is the following: if the number of marked probe packets is greater than some threshold, the call is blocked; otherwise, it is admitted into the network [9]. We will assume such a decision strategy in this paper.

3 Traffic and System Model

Consider a single link of capacity C (in packets per second) accessed by users who expect the delay experienced by a packet to be less than some value D_{max} . This is equivalent to stating that the QoS guarantee is violated if the number of packets in the queue exceed B , where $B = D_{max}C$. Therefore, we assume that the queue has a finite buffer of size B and packets entering when the buffer is full are dropped. The goal of the network is to keep the packet drop probability (equivalently, the per-packet probability of violating the delay constraint) below a certain threshold ϵ .

Call requests arrive according to a Poisson process of rate λ . A call seeking admission uses the following decision strategy to decide whether or not to join the network: it sends w probe packets and joins the network if and only if the number of marked probe packets is less than or equal to a threshold r . The choice of the parameters w and r is discussed later. Each admitted call remains in the system for a duration which is exponentially distributed with mean $1/\mu$. The number of packets generated by an admitted call in an interval of length τ is a random variable with mean $\lambda_p\tau$ and variance $\sigma_p^2\tau$. Similarly, the number of probe packets generated by probing calls in an interval of length τ are assumed to have mean $\lambda_m\tau$ and variance $\sigma_m^2\tau$. The packet generation processes of distinct calls are mutually independent.

In the remainder of this section, we describe the system dynamics at two time scales: packet-level time scale, where packets belonging to different admitted calls compete to access the resource and the call-level time scale, where calls enter and leave the system at random instants. In Section 3.1, we calculate packet marking and drop probabilities assuming a separation of time scales between the call-level and packet level dynamics, so that the queue length process comes to equilibrium within the lifetime of a single call occupancy state. With this assumption, the call blocking probability is simply a function of the current call occupancy state, the function being determined by the probing, marking and decision strategies. (In the centralized setting, this function only takes values 0 or 1. Loosely speaking, the goal is to come up with a decentralized scheme that approximates this $\{0, 1\}$ -valued function.) We use this fact to describe the call-

level dynamics in Section 3.3. We also introduce here the objectives of the admission control scheme and relate it to the packet and call-level dynamics. Subsequently, we use these objectives to evaluate different marking schemes.

First, some remarks on the model are in order. We have chosen a particularly simple model where there is a single type of call. It is easy to generalize the analysis to consider J different call types with corresponding packet generation parameters λ_m^j and $(\sigma_m^j)^2$, $j = 1, \dots, J$. We could further extend the model by assuming that each call generates packets according to a Markov-modulated process, with the parameters λ_m and σ_m depending on the state of the modulating Markov chain. We have not pursued these generalizations so as not to burden the exposition or the notation. Intuitively, in a “large” system, we do not expect these generalizations to alter our main conclusions, for the following reason. (It may be easier to follow this argument after going through the derivation of the diffusion approximation in Section 3.3.) By the separation of time scales assumption, the call blocking probability is a function of the current call occupancy state; in the extended model, this would include the number of admitted and probing calls of each type, and the state of each of the modulating Markov chains. If we consider a sequence of systems with C and λ going to infinity but with their ratio fixed, then it can be shown that the dynamics of the call occupancy process (suitably rescaled) converge to a fixed point on the fluid scale and to an Ornstein-Uhlenbeck process on the diffusion scale. Suppose the fixed point corresponds to a proportion p_j of calls being of type j , $j = 1, \dots, J$. (These proportions can be calculated easily from the type-specific call arrival rate and holding time if the call admission probability does not depend on its type.) Then, in the vicinity of the fixed point, the average packet arrival rate per call has mean $\bar{\lambda}_p = \sum_{j=1}^J p_j \lambda_p^j$ and variance $\bar{\sigma}_p^2 = \sum_{j=1}^J p_j (\sigma_p^j)^2$. The state influences the call blocking probability only through the mean and variance of the aggregate packet arrival process generated in that state. Hence, it suffices (for the purposes of the limiting analysis to be described in Section 3.3) to consider a single call type with parameters $\bar{\lambda}_p$ and $\bar{\sigma}_p^2$. Similar arguments apply if these parameters depend on the state of some underlying modulating Markov chain. With this justification, we now proceed to describe the packet and call level dynamics for the single class model.

3.1 Buffer overflow and packet drop probability

We assume a separation between packet time-scale and call time-scale, i.e., we assume that the queue reaches steady-state between successive call-level events, namely call arrivals and departures. Thus, for a given traffic model, we can compute the probability that a buffer level B is exceeded, given that k calls are currently admitted into the system and l calls are currently probing the system to decide whether or not to join the

network. This quantity is denoted by $p(k, l, B)$. As suggested in [13], we calculate $p(k, l, B)$ using a diffusion approximation [17, 10]. Recall that the number of packet arrivals generated by admitted calls in an interval of length τ have a mean $\lambda_p \tau$ and variance $\sigma_p^2 \tau$. Similarly, the number of probe packets generated by probing calls in an interval of length τ have a mean $\lambda_m \tau$ and variance $\sigma_m^2 \tau$. In particular, the number of packets arriving in disjoint intervals are uncorrelated. Then, from the diffusion approximation, the state of the buffer is approximated by a reflected Brownian motion with drift $d := C - k\lambda_p - l\lambda_m$ and infinitesimal variance $v^2 := k\sigma_p^2 + l\sigma_m^2$. The steady-state probability that the number of packets in the infinite-buffer queue exceeds B is given by [17, 10]:

$$p(k, l, B) = \exp\left(-\frac{2Bd}{v^2}\right). \quad (1)$$

This is clearly an upper bound on the probability that the number of packets in the finite-buffer queue is equal to the buffer size, B . (This can be seen by a coupling construction, whereby both models are constructed on the same probability space of packet arrival times, starting from the same initial conditions. It is clear that any given packet leaves the finite buffer queue no later than its infinite buffer counterpart - it may leave earlier because it is dropped. Hence, the queue size in the finite buffer queue is no bigger than that in the infinite buffer queue.) By ergodicity, the steady state probability that the buffer is full in the finite-buffer queue is the fraction of time that the queue spends in this state. It is also equal to the fraction of arriving packets that see the queue full, because the packet arrival process has been assumed to have independent increments. Thus, the packet drop probability is bounded above by $p(k, l, B)$.

3.2 Marking probability

We consider three different marking/drop mechanisms in this paper: tail drop, REM and virtual-queue marking. In the rest of this subsection, we define each of these mechanisms and develop expressions for computing the probability that a probing packet is marked under each of these mechanisms.

3.2.1 Tail drop

This is the simplest strategy, and can be implemented by end-systems with no support from the router. An incoming packet is dropped if there are already B packets in the buffer when it arrives. Thus, the packet drop probability is bounded above by $p(k, l, B)$ computed above. Suppressing the dependence on B , we write this as

$$p_m(k, l) = \exp\left(-\frac{2Bd}{v^2}\right). \quad (2)$$

Here, the call relies on acknowledgements from the receiver to detect dropped packets and make its call admission decision accordingly.

3.2.2 Random-early marking (REM)

This technique is closely related to the RED mechanism proposed for the Internet [8]. An incoming packet that sees b packets in the buffer upon arrival is marked with probability $1 - e^{-\gamma b}$ [2]. In RED, a different function of the queue length is used to compute the probability of marking.

Given that there are k admitted and l probing calls, the marking probability in an infinite buffer queue is given by:

$$p_m(k, l) = \int_0^{+\infty} -\frac{\partial p(k, l, b)}{\partial b} (1 - e^{-\gamma b}) db, \quad (3)$$

where $p(k, l, b)$ is given in (1). Thus,

$$p_m(k, l) = \frac{\gamma}{\gamma + \frac{2d}{v^2}}. \quad (4)$$

As with the tail drop scheme above, this is an upper bound on the marking probability in the finite buffer queue; it will be a good approximation in the regime where buffer overflow is rare, which is the targeted operating regime.

3.2.3 Virtual-queue marking

This marking strategy is based on the idea of virtual queue, a technique first explored by Gibbens and Kelly [9] and later studied in [12, 15]. Routers maintain the state of a virtual queue, which corresponds to a buffer of size B drained by a link of capacity θC , where $\theta < 1$. Arriving packets are enqueued in the real queue, but a counter which tracks of the contents of the virtual queue is incremented by one for each arriving packet. The virtual queue is drained at rate θC . Incoming packets are marked if the counter denoting the number of packets in the virtual queue is larger than B . The purpose of such a technique is to provide early congestion indication: since the virtual queue is drained at a rate slower than the real queue, it will overflow faster than the real queue.

The marking probability with a virtual-queue mechanism is the probability of exceeding a level B in the virtual queue; hence, it is given by (1) with the expressions for v^2 and d modified by replacing C with θC . Thus,

$$p_m(k, l) = \exp \left(-\frac{2B\theta}{k\sigma_p^2 + l\sigma_m^2} (\theta C - k\lambda_p - l\lambda_m) \right). \quad (5)$$

3.3 Call-level dynamics

There are two types of calls in the network at any time: admitted calls that are sending data packets and probing calls that are sending probe packets to decide whether or not to join the network. Let $k(t)$ denote the number of admitted calls and $l(t)$ the number of probing calls in the system at time t . Recall that a user seeking admission sends w probe packets and joins the network if and only if the number of marked probe packets is less than or equal to a threshold r . Assuming that each probe packet is marked with probability $p_m(k, l)$, independently of the others, the probability that a call is admitted when there are k admitted calls and $l - 1$ other probing calls in the network is given by

$$a(k, l) = \sum_{j=0}^r \binom{w}{j} p_m(k, l)^j (1 - p_m(k, l))^{w-j}. \quad (6)$$

Note that $p_m(k, l)$ depends on the marking strategy employed at the router. The aim is to choose the marking strategy and the parameters w and r , and thereby the admission function $a(k, l)$, so as to achieve performance comparable to the centralized scheme which admits calls only when it can do so without violating the QoS constraint. Observe that this admission control law makes the process $(k(t), l(t))$ stationary. It is also ergodic because it regenerates whenever the system is empty, and the mean busy cycle length¹ is finite. Indeed, it is bounded above by the busy cycle length in the system in which every probing call is admitted at the end of its probing period; this latter system corresponds to an $M/G/\infty$ queue. Thus, the process $(k(t), l(t))$ possesses a unique equilibrium distribution, denoted $\pi(k, l)$.

Now, given $\varepsilon > 0$, let $A_\varepsilon = \{(k, l) : p(k, l, B) < \varepsilon\}$, be the set of call occupancy states in which the QoS requirement is satisfied. Here $p(k, l, B)$, given by (1), is the packet drop probability. For notational convenience, we assume that $\lambda_p = \lambda_m$ and $\sigma_p^2 = \sigma_m^2$, i.e., traffic generated by admitted and probing calls is identical. Then $p(k, l, B)$ depends only on $k + l$ and B , and the set A_ε has the form

$$A_\varepsilon = \{(k, l) : k + l \leq \eta_\varepsilon\}, \quad (7)$$

where $\eta_\varepsilon = \max\{k + l : p(k, l, B) < \varepsilon\}$. Thus, the probability that the call-level system is in a state where the QoS will be violated is given by

$$\Delta_\varepsilon = 1 - \sum_{(k, l) \in A_\varepsilon} \pi(k, l), \quad (8)$$

while the link utilization is given by

$$u = \frac{E(k)}{C} = \frac{1}{C} \sum_{k, l} k \pi(k, l). \quad (9)$$

¹A busy cycle begins when the first call enters an empty system and ends when the system again becomes empty.

The distributed admission control scheme faces a multi-objective problem: keep Δ_ε small, while making u large. In other words, there is a tradeoff involved between high levels of QoS and high network utilization. To study this tradeoff, we consider the following quantity, which we call the *service objective*:

$$s_\alpha := u - \alpha \Delta_\varepsilon,$$

where α is a parameter that determines the relative weight of QoS and link utilization.

The service objective s_α can be computed if $\pi(k, l)$ is known. But, in general, it is difficult to compute this equilibrium distribution exactly. Therefore, we approximate the call-level dynamics by an Ornstein-Uhlenbeck process and use this approximation to calculate s_α . To this end, consider a family of systems, indexed by their capacity C , and let $C \rightarrow \infty$. For notational convenience, we scale time so that $\lambda_p = 1$.

Without providing a formal proof, we will argue that the scaled behavior tends to a fluid limit, and the deviations from this limit follow an Ornstein-Uhlenbeck process. The asymptotic limits can then be used as an approximation for a finite system with a large, but fixed, value of C . Let C be the capacity of the system, and λC the arrival rate of the C^{th} system, with the departure rate μ of the admitted calls left unscaled. Unlike the earlier sections, we make the further assumption that the probing calls last for an (unscaled) exponential time, with mean T where $T = w/\lambda_p$. Let $\mathbf{N}^C(t) = (k^C(t), l^C(t))$ be the number of active and probing calls in the C^{th} system. We now consider the scaled process, $\mathbf{n}^C(t) = (n_k^C(t), n_l^C(t)) = (k^C(t)/C, l^C(t)/C)$. Let $a^C(k, l)$ be the acceptance probability given k active and l probing calls, and assume that $a^C(\mathbf{N}^C(t)) = a(\mathbf{n}^C(t))$, which holds in our examples, where the function a is a function of the normalized load.

Assume that $\mathbf{n}^C(\mathbf{0}) \rightarrow \mathbf{n}(\mathbf{0})$ almost surely for some fixed $\mathbf{n}(\mathbf{0})$, the scaled process $\mathbf{n}^C(t)$ converges uniformly on compact sets in t to the fluid limit process $\mathbf{n}(t)$, for $t \geq 0$, which is the solution to the following differential equations:

$$\begin{aligned} \frac{d}{dt} n_k(t) &= n_l(t) a(n_k(t), n_l(t)) / T - \mu n_k(t) \\ \frac{d}{dt} n_l(t) &= \lambda - (1/T) n_l(t) [a(n_k(t), n_l(t)) \\ &\quad + 1 - a(n_k(t), n_l(t))] \\ &= \lambda - n_l(t) / T \end{aligned}$$

The system converges to the unique equilibrium which solves

$$\bar{n}_l = \lambda T \tag{10}$$

$$\bar{n}_k = \frac{\lambda}{\mu} a(\bar{n}_k, \lambda T) \tag{11}$$

Now consider the deviations from the fluid limit. Let $\hat{\mathbf{n}}(t)$ be the vector of differences from the equilibrium, so $\hat{\mathbf{z}}_t = (n_k(t) - \bar{n}_k, n_l(t) - \bar{n}_l)$. Let $\stackrel{\mathcal{D}}{=}$ denote convergence in distribution. Then, as $C \rightarrow \infty$, using the results in [7], it can be verified that $\sqrt{C}\hat{\mathbf{z}}_t \stackrel{\mathcal{D}}{=} \hat{\mathbf{n}}_t$, where $\hat{\mathbf{n}}_t$ is an Ornstein-Uhlenbeck process, the solution to the stochastic differential equation

$$d\hat{\mathbf{n}}_t = H\hat{\mathbf{n}}_t dt + F d\mathbf{B}_t, \quad (12)$$

where \mathbf{B}_t is a four-dimensional Brownian motion. The matrix H is the linearization of the fluid equations about the equilibrium, given by

$$\begin{aligned} H &= \begin{pmatrix} \frac{1}{T}\bar{n}_l a_k(\bar{n}_k, \bar{n}_l) - \mu & \frac{1}{T}\bar{n}_l a_l(\bar{n}_k, \bar{n}_l) + \frac{1}{T}a(\bar{n}_l, \bar{n}_k) \\ 0 & -\frac{1}{T} \end{pmatrix} \\ &= \begin{pmatrix} \lambda\bar{a}_k - \mu & \lambda\bar{a}_l + \bar{a}/T \\ 0 & -1/T \end{pmatrix} \end{aligned}$$

where $a_k(n_k, n_l)$ denotes the partial derivative of a with respect to n_k and similarly for a_l , and \bar{a}_k is shorthand for a_k evaluated at the equilibrium point (\bar{n}_k, \bar{n}_l) . Note that these derivatives have negative sign. The matrix F is given by

$$\begin{aligned} F &= \begin{pmatrix} \sqrt{\bar{n}_l \bar{a}/T} & -\sqrt{\mu \bar{n}_k} & 0 & 0 \\ -\sqrt{\bar{n}_l \bar{a}/T} & 0 & \sqrt{\lambda} & -\sqrt{\bar{n}_l(1-\bar{a})/T} \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{\lambda \bar{a}} & -\sqrt{\lambda \bar{a}} & 0 & 0 \\ -\sqrt{\lambda \bar{a}} & 0 & \sqrt{\lambda} & -\sqrt{\lambda(1-\bar{a})} \end{pmatrix} \end{aligned}$$

In steady state, the scaled difference vector $\hat{\mathbf{n}}_t$ has a multivariate normal distribution with mean zero and covariance matrix Σ given by [1]

$$\Sigma = \int_{-\infty}^0 e^{-uH} F F^T (e^{-uH})^T du,$$

which is also the solution to the following Lyapunov equation:

$$\Sigma H^T + H \Sigma + F F^T = 0.$$

Performing the calculations

$$\Sigma = \begin{pmatrix} -\left(\frac{\lambda(\lambda^2 T^2 a_l^2 + \bar{a}(1+T\mu - \lambda T a_k + \lambda T a_l))}{(1+T\mu - \lambda T a_k)(-\mu + \lambda a_k)} \right) & \frac{\lambda^2 T^2 a_l}{(1+T\mu - \lambda T a_k)} \\ \frac{\lambda^2 T^2 a_l}{(1+T\mu - \lambda T a_k)} & \lambda T \end{pmatrix}$$

and the variance of the total load is given by

$$\begin{aligned} &\lambda T + \frac{2\lambda^2 T^2 a_l}{(1+T\mu - \lambda T a_k)} \\ &- \frac{\lambda(\lambda^2 T^2 a_l^2 + \bar{a}(1+T\mu - \lambda T a_k + \lambda T a_l))}{(1+T\mu - \lambda T a_k)(\lambda a_k - \mu)}. \end{aligned}$$

To calculate u , we solve the fixed-point equations (10) and (11) to find \bar{n}_k (which is approximately equal to $\mathbf{E}(k)$ when C is large). To compute Δ_ϵ , approximate $(k, l) = \mathbf{N}^C(t)$ by $C\bar{\mathbf{n}} + \sqrt{C}\hat{\mathbf{n}}(t)$ which has stationary covariance matrix $C\Sigma$. Informally

$$\begin{aligned}\Delta_\epsilon &= \mathbf{P}\{k + l > \eta_\epsilon\} \\ &= \mathbf{P}\{n_k^C(t) + n_l^C(t) > \eta_\epsilon/C\} \\ &\rightarrow \mathbf{P}\left\{\hat{n}_k(t) + \hat{n}_l(t) > \frac{\eta_\epsilon}{\sqrt{C}} - \sqrt{C}(\bar{n}_k + \bar{n}_l)\right\} \\ &= Q\left(\frac{\eta_\epsilon}{\sqrt{C}} - \sqrt{C}(\bar{n}_k + \bar{n}_l)\right)\end{aligned}$$

as C becomes large, where $Q(\cdot)$ is the complementary cumulation distribution of a Gaussian random variable with mean 0 and variance 1.

Further details on the limit process and analysis can be found by adapting the arguments in [3, 4], which looked at a different problem (rate adaptation for in-call probing) and derived a limiting functional law of large numbers and functional central limit theorem.

4 Numerical Results

In this section, we present numerical results obtained using the analytical formulas presented above in Section 3.3. In what follows, unless otherwise stated, we use the default parameters, $\epsilon = 0.01$, $\alpha = 100$, $c = 200$ packets/sec., $B = 20$ packets, $\lambda_p = 0.2$ packets/sec., $\sigma_p^2 = 0.8\lambda_p$, $\lambda_m = \lambda_p$, $\sigma_m^2 = \sigma_p^2$, and $1/\mu = 10000$ secs.

4.1 Tail Marking

Figure 1 shows the service objective for $\lambda = 0.11$, which corresponds to 10% overload (i.e., if there is no probing and all calls are accepted, the mean packet arrival rate will exceed capacity by 10%). To achieve the maximum value of the service objective, the required number of probe packets is close to 200, and the achieved service objective is around 0.8. The maximum possible value for the service objective, achieved by a centralized scheme, is approximately η_ϵ/C ; thus, it is always smaller than 1 and is close to 1 in a large system. Figure 2 depicts the service objective in more heavily loaded regimes, with $\lambda = 0.15$ and 0.2 (corresponding to 50% and 100% overload). In these instances, tail marking fails to achieve a positive service objective even with up to 300 probe packets. (Note that the trivial policy of rejecting all calls without

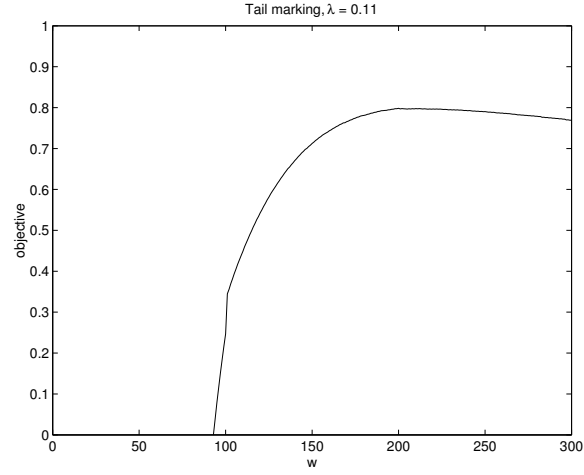


Figure 1: Service objective for tail marking, for $\lambda = 0.11$ with default parameters, optimal r .

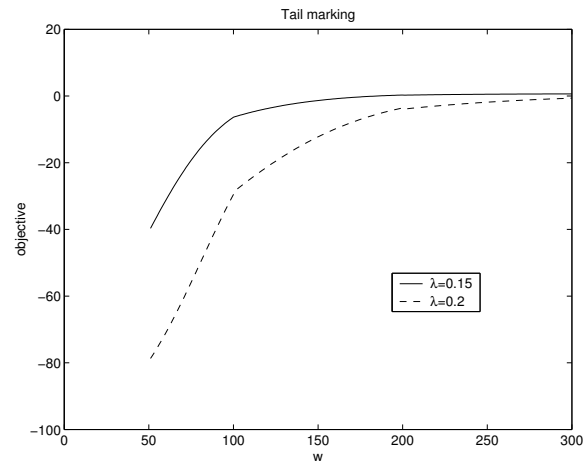


Figure 2: Service objective for tail marking, for $\lambda = 0.15$ and 0.2 , with default parameters, optimal r .

probing achieves a service objective of zero.) In all of the cases, $r = 0$ turned out to be optimal, i.e., each user will not join the network even if one probe packet is marked. As w increases, not only does the probing traffic increase, but also the probability of rejecting a call increases.

4.2 Random Early Marking (REM)

To compute the service objective with REM-based marking, we have to first decide on a value for γ , and then on a value for the threshold r corresponding to each value of w , the number of probe packets. Our experiments indicated that the performance of REM-based marking is not too sensitive to the choice of γ . In what follows, we use $\gamma = 0.2$. We report results for $\lambda = 0.11, 0.15$ and 0.2 , corresponding to 10%, 50% and 100% overloading. For each value of w , the threshold r (of number of marked packets at which a call is admitted) is chosen optimally for the worst-case λ , namely 0.2 , and the same r is used to evaluate performance for all values of λ .

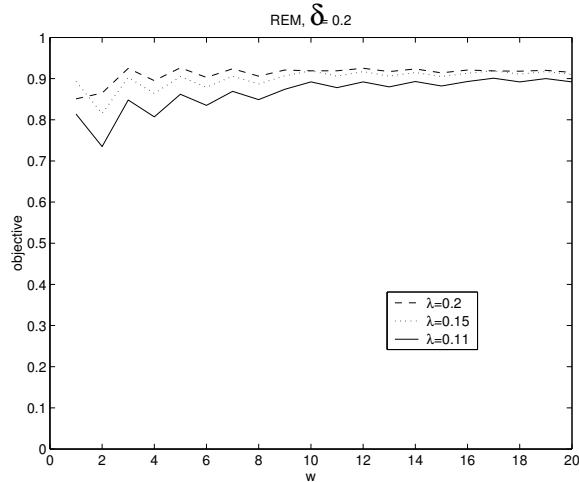


Figure 3: Service objective for REM as w is varied with $\gamma = 0.2$

Figure 3 shows how the service objective, s , varies with the number of probe packets, w , for $\gamma = 0.2$. In contrast to tail marking, a small number of probe packets suffice to achieve good performance. The plots show that a service objective of around 0.9 is achievable with 20 probe packets, while just 3 probe packets suffice to achieve a service objective of 0.85. Moreover, the performance achieved is not sensitive to the call arrival rate, λ , which will typically be unknown in applications. Thus, call admission based on REM marking performs well even in a heavily overloaded system. This contrasts with the tail marking case, where performance degrades substantially as the load increases. Note that the plots in this figure are not smooth

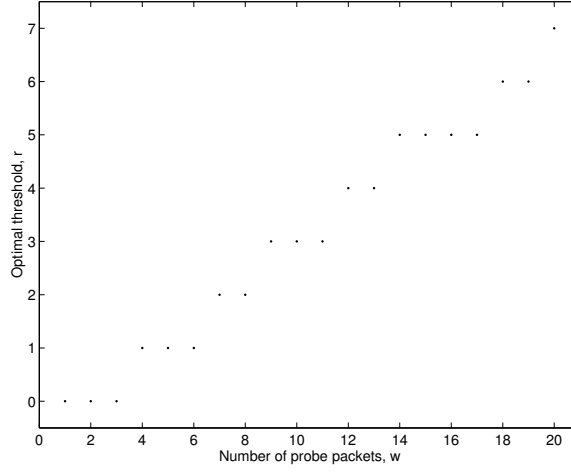


Figure 4: Optimal choice of r for REM as w is varied with $\gamma = 0.2$ and $\lambda = 0.2$.

due to the fact that w and r take only discrete values. Figure 4 shows how the optimal choice of r (for $\lambda = 0.2$) varies with w .

4.3 Virtual queue marking

For virtual-queue marking, we have to first decide on the value of θ . Roughly speaking, the maximum utilization achievable by such a scheme is around θ ; when the packet arrival rate begins to exceed θC , the virtual queue fills up and blocks the admission of new calls. This motivates choosing θ close to 1. On the other hand, a smaller value of θ reduces the probability that the number of calls in the system will exceed the threshold, η_ϵ , at which QoS is violated, i.e., calls suffer a packet drop rate in excess of ϵ . For our default parameters, we calculated $\eta_\epsilon = 966$, which corresponds to a capacity utilization of 0.966. We chose θ to be slightly smaller, at 0.95. This choice was adequate to ensure that the QoS violation probability, $\sum_{k+l > \eta_\epsilon} \pi(k, l)$, was very small and did not degrade the service objective. A smaller value of θ would share this property, but achieve correspondingly lower utilization. A larger value of θ runs the risk of violating the QoS requirement more frequently, which is heavily penalized as the parameter α is set to 100. The discussion below pertains to $\theta = 0.95$.

As in the previous subsection, we consider call arrival rates $\lambda = 0.11, 0.15$ and 0.2 . For fixed values of θ and w , we find the value of r that maximizes the service objective in the worst-case scenario, namely, $\lambda = 0.2$. The results are shown in Figure 6. The same value of r is then used to evaluate all 3 scenarios. Figure 5 presents the service objective as a function of the number of probe packets, w , for different values

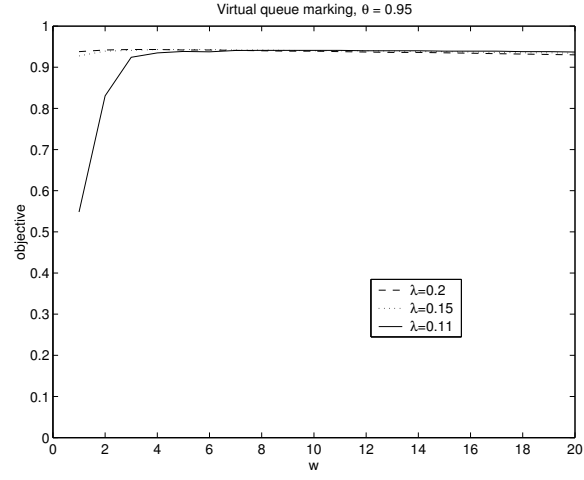


Figure 5: Service objective for VQ for $\theta = 0.95$ and different values of λ .

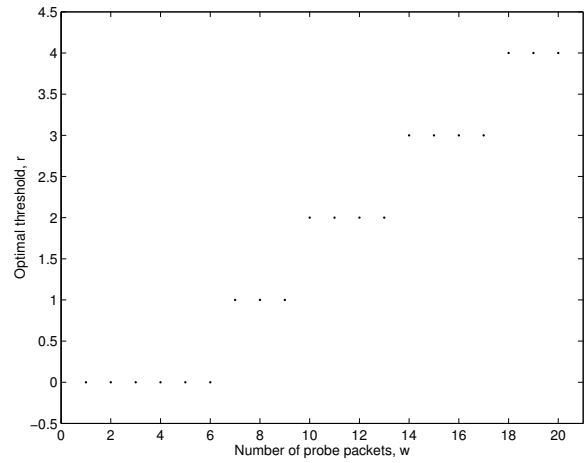


Figure 6: Optimal choice of r for VQ as w is varied, with $\theta = 0.95$ and $\lambda = 0.2$.

of the call arrival rate, λ . The plots show that a small number of probe packets suffice to achieve a very good service objective, close to what a centralized scheme could achieve. Moreover, the performance of the scheme is robust to wide variations in the actual call arrival rate, λ . This is similar to REM and contrasts vividly with the tail marking case.

5 Simulation Results

In this section, we complement the numerical results in the previous section with results from packet simulations. The conclusions regarding the relative performance of the three marking schemes are similar, so we only present a representative sample of the simulation results here. The simulations are slotted-time simulations with each slot equal to 1 sec. and the rest of the parameters are as follows: the capacity of the link is 200 packets/slot, with $B = 20$ packets. Each source sends 1 packet in a slot with probability 0.2 and none with probability 0.8. The holding time of a call is 10,000 slots. The parameter α is chosen to be 100. These parameters are identical to those chosen in the previous section.

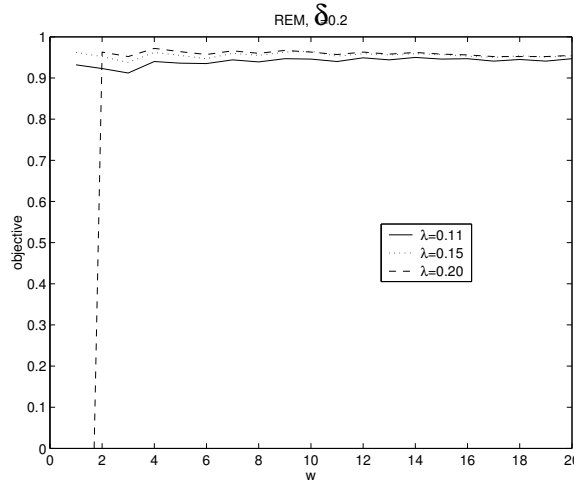


Figure 7: Service objective for REM as a function of the number of probe packets.

In Figures 7 and 8, the performance of REM and VQ-based marking schemes are shown for three different values of λ : 0.11, 0.15 and 0.2 calls per slot. The figures plot the service objective as a function of the number of probe packets. It is clear from the figures that near-optimal performance can be achieved with as few as two to four probe packets. The corresponding results for tail marking are shown in Figures 9 and 10. The first figure shows the performance for $\lambda = 0.11$ and the second figures shows the performance for

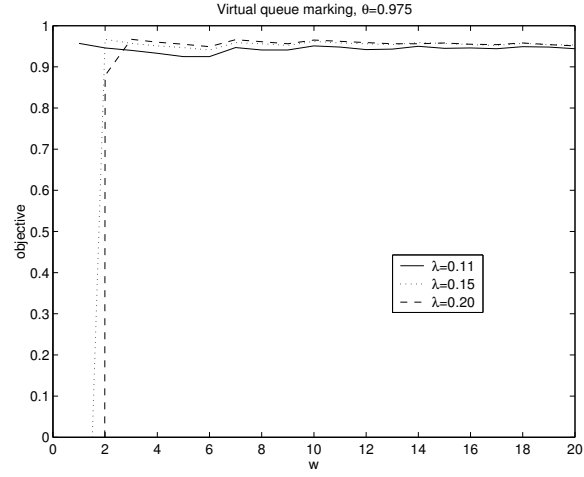


Figure 8: Service objective for VQ as a function of the number of probe packets.

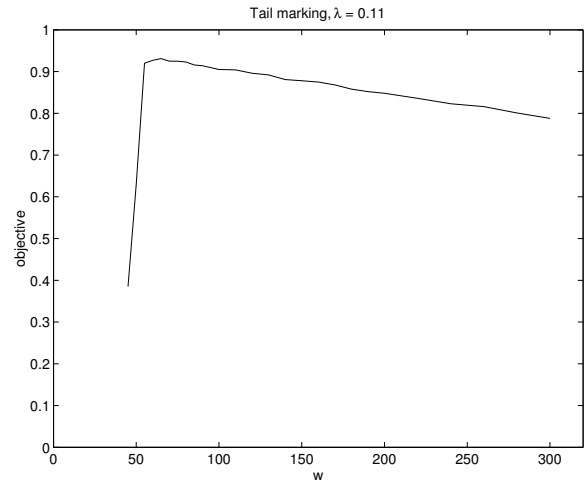


Figure 9: Service objective for tail marking as a function of the number of probe packets, $\lambda = 0.11$

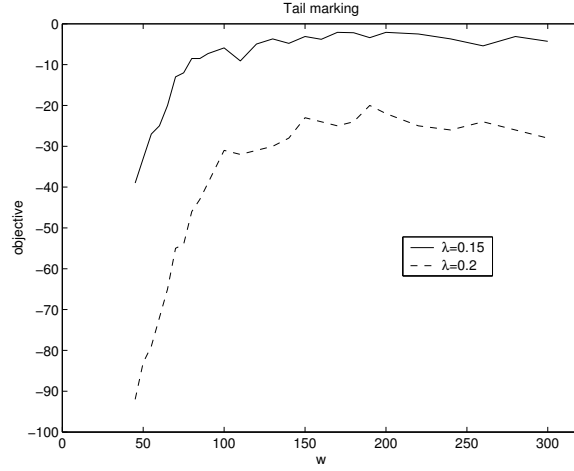


Figure 10: Service objective for tail marking as a function of the number of probe packets, λ is 0.15 and 0.2.

$\lambda = 0.15$ and $\lambda = 0.2$. The results for tail marking are shown in two separate figures since the performance is very different for the different values of λ , thus making it difficult to show the results on the same scale. From the figures, it is clear that when $\lambda = 0.11$, it requires about 300 to 400 probe packets to achieve a service objective of 0.9. When $\lambda = 0.15$ or $\lambda = 0.2$, even when 1500 probe packets are used, the service objective is negative! Thus, the simulation results strongly suggest that some form of active queue management-based marking is necessary to offer real-time services over the Internet.

5.1 Markovian Sources

All of our simulation results so far deal with sources whose traffic statistics are not correlated over time. Now, we consider Markovian On-Off sources which exhibit correlation over time and compare the results for such a model with the heavy-traffic models used earlier. Specifically, each source is now modelled as a Markov chain ξ_n on the state space $\{0, 1\}$, with state 0 denoting Off and 1 denoting On. We denote by

$$P = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix},$$

the transition matrix of this Markov chain, and by π its invariant distribution. We assume that the Markov chain is irreducible and aperiodic, i.e., $p, q \in (0, 1)$. Now, $\pi = (q/(p+q), p/(p+q))$. If the source is On in some time slot, then it transmits 1 packet in that slot with probability r and no packet with probability $1-r$. If it is Off, it transmits no packets. We assume below that the Markov chain is in stationarity.

Let X_n denote the number of packets transmitted in time slot n , and let $S_n = X_1 + X_2 + \dots + X_n$. We want to compute $E[S_n]/n$ and $\text{Var}(S_n)/n$ to relate it to the Gaussian approximation used earlier. These computations are well-known, but we provide it below for the reader's convenience. It is obvious that $EX_1 = pr/(p+q)$, and so

$$\frac{1}{n}E[S_n] = \frac{pr}{p+q} \quad \forall n = 1, 2, \dots \quad (13)$$

In order to compute $\text{Var}(S_n)$, we need to compute the covariances of X_i and X_j . By stationarity, this only depends on $|i - j|$. Hence, we compute $\text{Cov}(X_0, X_k)$ for $k \geq 0$. Now

$$\begin{aligned} E[X_0 X_k] &= P(X_0 = 1, X_k = 1) = r^2 P(\xi_0 = 1) P(\xi_k = 1 | \xi_0 = 1) \\ &= \frac{pr^2}{p+q} P(\xi_k = 1 | \xi_0 = 1). \end{aligned} \quad (14)$$

But $P(\xi_k = 1 | \xi_0 = 1)$ is just the $(1, 1)$ entry of the k -step transition probability matrix, P^k . Tedious but straightforward calculations yield that

$$P^k = \frac{1}{p+q} \begin{pmatrix} q + p(1-p-q)^k & p - p(1-p-q)^k \\ q - q(1-p-q)^k & p + q(1-p-q)^k \end{pmatrix}.$$

Hence, by (14),

$$E[X_0 X_k] = \frac{pr^2}{(p+q)^2} [p + q(1-p-q)^k], \quad \text{Cov}(X_0, X_k) = \frac{pqr^2(1-p-q)^k}{(p+q)^2}. \quad (15)$$

We can now calculate $\text{Var}(S_n)$. We have

$$\begin{aligned} \text{Var}(S_n) &= \sum_{i,j=1}^n \text{Cov}(X_i, X_j) = \sum_{k=0}^n (n-k) \text{Cov}(X_0, X_k) \\ &= \frac{pqr^2}{(p+q)^2} \sum_{k=0}^n (n-k)(1-p-q)^k. \end{aligned} \quad (16)$$

We are interested in $\text{Var}(S_n)/n$, for large n . Since p and q are in $(0, 1)$ by assumption, $1-p-q$ lies in $(-1, 1)$. Hence, $\sum_{k=0}^n (1-p-q)^k$ and $\sum_{k=0}^n k(1-p-q)^k$ both remain bounded by finite constants for all n . Thus, by (16),

$$\lim_{n \rightarrow \infty} \frac{1}{n} \text{Var}(S_n) = \frac{pqr^2}{(p+q)^2}. \quad (17)$$

For our earlier analysis, we modelled the packet arrival process as Gaussian with mean $\lambda_p t$ and variance $\sigma_p^2 t$ over a time interval of length t . The Markov modulated process described above can be approximated by such a Gaussian process if we take

$$\lambda_p = \frac{pr}{p+q}, \quad \sigma_p^2 = \frac{pqr^2}{(p+q)^2}. \quad (18)$$

For our numerical results based on the analytical model, we had chosen $\lambda_p = 0.2$ and $\sigma_p^2 = 0.16$. For the Markov model, we choose $p = 0.15$, $q = 0.225$ and $r = 0.5$ so that it has the same long-term mean $\lambda_p t$ and variance $\sigma_p^2 t$ as in the Gaussian model. The simulation results using these parameter choices are shown in Figures 11, 12, 13 and 14. These figures show that, for tail marking, when the arrival rate is high, even with

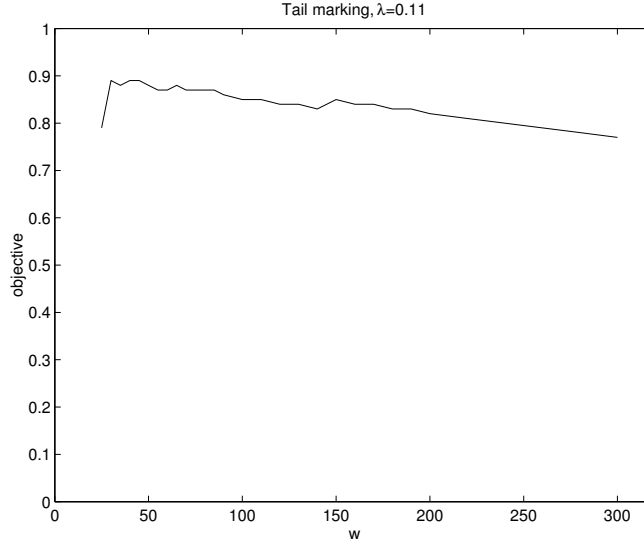


Figure 11: Service objective for tail marking as a function of the number of probe packets with Markovian traffic sources.

a very large number of probe packets, the service objective remains negative (Figure 12). For low arrival rates, the service objective can be made as high as 0.9, but the number of probe packets required is fairly large. On the other hand, for REM and VQ-based marking, with a very small number of probe packets, we can achieve service objective values that are close to 1 even for high arrival rates (Figures 13 and 14). These figures were obtained after tuning them for best performance for the case of $\lambda = 0.2$ in the Bernoulli source case and using the same parameter values for the Markov model. Thus, the figures also show the robustness of the REM and VQ mechanisms, while the tail marking results show a lack of robustness to arrival rate. These results are quite similar to the results for the Bernoulli source model indicating the source's correlation over time does not play a significant role in the performance of the admission control schemes under the different marking mechanisms.

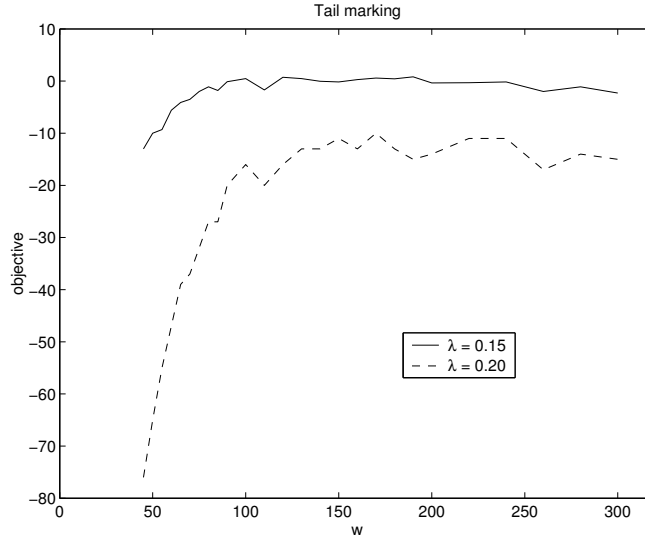


Figure 12: Service objective for tail marking as a function of the number of probe packets with Markovian traffic sources.

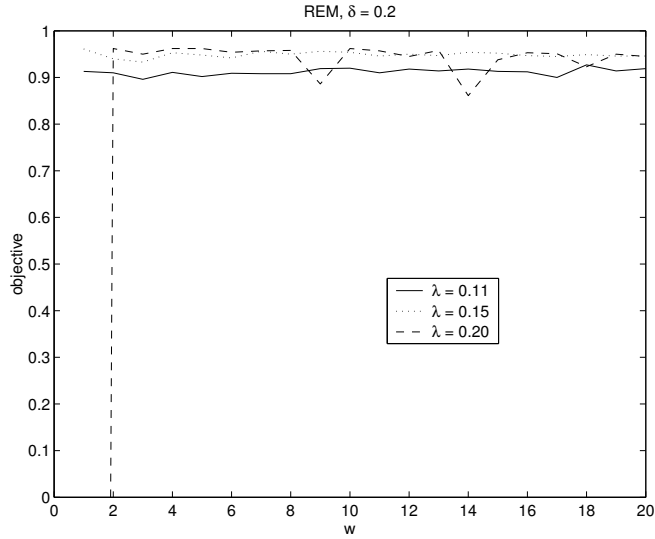


Figure 13: Service objective for REM as a function of the number of probe packets with Markovian traffic sources.

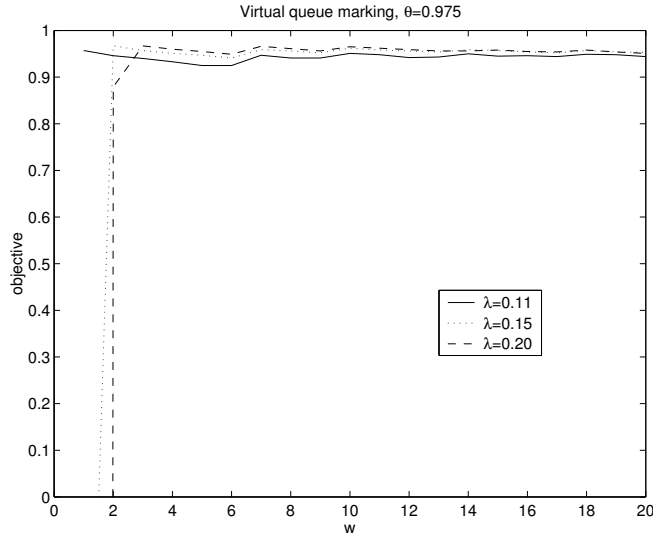


Figure 14: Service objective for VQ as a function of the number of probe packets with Markovian traffic sources.

6 Conclusions

In this paper, we have presented a framework for studying marking mechanisms to support distributed admission control for real-time traffic sources that do not adapt their transmission rate in response to congestion. We used a performance measure that trades off between utilization and QoS to characterize the performance of a congestion indication mechanisms. We presented a simple analytical framework to compute the performance measure.

Using this framework as well as simulations, we analyzed three particular marking strategies: tail drop, random-early marking and virtual-queue marking. Our conclusions are as follows:

- Tail drop requires a large number of probe packets to approach the performance of a centralized scheme.
- Random-early marking and virtual-queue marking approach the performance of a centralized scheme with only very few probe packets.
- Random-early marking and virtual-queue marking are highly robust to the actual load on the system, whereas the performance of tail drop degrades rapidly in a heavily overloaded system, where call admission is most needed.

The results in this paper suggest that both real-queue-based marking (REM) and virtual-queue-based marking perform equally well for admission control of non-adaptive sources. This is in contrast to the results in [16] which shows that virtual-queue marking is superior for achieving low queueing delays in networks with congestion-controlled sources. Evidently, the lack of transmission rate adaptation through congestion control leads to this difference in the performance of real-queue-based marking schemes. Further, in simulations not shown, other early marking schemes (such as tail marking at a lower buffer level) also seem to perform well in the context of distribution admission control. Thus, it appears as though early marking is more important than the actual mechanism used to implement such a congestion indication mechanism.

References

- [1] L. Arnold. *Stochastic Differential Equations: Theory and Applications*. John Wiley, New York, NY, 1974.
- [2] S. Athuraliya, D. E. Lapsley, and S. H. Low. Random early marking for Internet congestion control. In *Proceedings of IEEE GLOBECOM*, 1999.
- [3] A. Bain and P. B. Key. Modelling the performance of distributed admission control for adaptive applications. *ACM SIGMETRICS Performance Evaluation Review*, 29(3):21–22, December 2001.
- [4] A. Bain and P. B. Key. Modelling the performance of in-call probing for multi-level adaptive applications. Tech Report MSR-TR-2002-06, Microsoft Research, Jan 2002. http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-2002-06.
- [5] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 1987.
- [6] L. Breslau, E. Knightly, S. Shenker, I. Stoica and H. Zhang. Endpoint admission control: architectural issues and performance. In *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000.
- [7] S. N. Ethier and T. G. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley, 1994.
- [8] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, August 1993.

- [9] R.J. Gibbens and F.P. Kelly. Distributed connection acceptance control for a connectionless network. In *Proc. of the 16th Intl. Teletraffic Congress*, Edinburgh, Scotland, June 1999.
- [10] J. M. Harrison. *Brownian Motion and Stochastic Flow Systems*. Wiley, 1985.
- [11] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE J. Sel. Areas Comm.*, 21: 879894, August 2003.
- [12] F. P. Kelly, P. B. Key, and S. Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 18:2617–2628, 2000.
- [13] F.P. Kelly. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society*, A358:2335–2348, 2000.
- [14] P. B. Key and L. Massoulié. Probing strategies for distributed admission control in large and small scale systems. In *Proceedings of INFOCOM*, San Francisco, April 2003. IEEE.
- [15] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue algorithm for active queue management. In *Proceedings of ACM Sigcomm*, pages 123–134, San Diego, CA, August 2001.
- [16] A. Lakshmikantha, C. Beck, and R. Srikant. Robustness of real and virtual queue based active queue management schemes. In *Proceedings of the American Control Conference*, June 2003. To appear.
- [17] G. F. Newell. *Applications of queueing theory*. Chapman and Hall, London, UK, 1982.