A Case for TCP-Friendly Admission Control

Adrian Sai-wah Tam*, Dah-Ming Chiu*, John C. S. Lui† and Y. C. Tay‡
*Department of Information Engineering
The Chinese University of Hong Kong, Shatin, Hong Kong
Email: {swtam3,dmchiu}@ie.cuhk.edu.hk
†Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, Hong Kong
Email: cslui@cse.cuhk.edu.hk
‡Department of Mathematics and Department of Computer Science
National University of Singapore, 117543, Singapore
Email: mattyc@leonis.nus.edu.sg

***Abstract*— Admission control has been shown to be a preferred alternative to TCP-friendly congestion control for inelastic flows in heterogeneous networks shared by elastic and inelastic traffic [1]. However, it is possible for an inelastic flow to adopt different level of aggressiveness in implementing the admission control. How these different levels of aggressiveness affect the system performance remains an open issue. In this paper, we evaluate a full spectrum of (abstract) admission control algorithms in terms of their aggressiveness towards elastic flows. A totally aggressive version would admit an inelastic flow even if this means elastic flows' fair bandwidth share is reduced to close to zero. In the other extreme, a *TCP-friendly* version would only admit an inelastic flow if its desired rate is no higher than what the elastic flows will receive after its arrival. We show that the performance of inelastic flows is *asymptotically* insensitive to their aggressiveness without strong assumptions about flow file size or holding time distributions. This makes a strong case for adopting a less aggressive, yet TCP-friendly admission control in a heterogeneous network. Extensive simulations are carried out to validate the performance, stability and asymptotic behavior the the proposed TCP-friendly admission control policy.**

I. Introduction

Internet applications were predominantly *elastic* in their bandwidth requirements. These applications can function with different available bandwidth, although more is better. However, Internet is increasingly being used to support multimedia applications as well. These multimedia applications are *inelastic* in their bandwidth demands, which means they may not run if certain minimum bandwidth is not available.

There has been a long debate about how to support both kinds of applications. Numerous studies proposed ways to support *quality of service* (QoS) in the Internet (e.g., [2]). However, after a period of high rate of network business expansion and bandwidth over-provisioning, very little QoS mechanisms have been adopted, or are even deemed necessary. Instead, the prevalent wisdom is that the Internet should be kept simple by continuing to rely on relatively simple end-to-end adaptation.

This work is supported in part by University Grants Committee, Hong Kong SAR under RGC grant 4232/04E (project 2150420) and AoE grant AoE/E-01/99 (project 4801312), and by National University of Singapore under ARF grant R-146-000-051-112.

The result of this laissez-faire situation is that the elastic applications continue to do congestion control via TCP, and most multimedia applications use UDP. This is normally not a problem in the over-provisioned Internet, but from an architectural point of view, this does not seem to be fair or stable. If an increasing number of applications are no longer participating in congestion control, it would certainly affect the operation of the shared network.

The well-intentioned solution is to develop a class of *TCP-friendly* congestion control mechanisms [3]–[5] for all multimedia applications. This will make all applications in the network *responsible citizens* so that they all share the burden of congestion avoidance and control. In the proposals, the end result is for a multimedia flow to smoothly converge to a transmission rate that is the same as that used by other TCP flows sharing the same bottleneck.

In [1], it is argued that this notion of TCP-friendliness is too narrow. Rate adaptation is inherently the wrong way for multimedia flows to deal with network congestion. Rather, it is more suitable for multimedia flows to use some form of *admission control* at times of congestion. To support this argument, [1] developed a stochastic model and methodology to compare how the network performs when different traffic controls are used by the inelastic flows when coexisting with elastic traffic in a shared network. It demonstrated that by adopting admission control, the inelastic flows can benefit itself under heavy load; and admission-controlled inelastic flows (without congestion control) can also benefit elastic flows, when compared to TCP-friendly congestion control.

A. Previous work on admission control

Admission control is hardly a new idea. It has been proposed as part of various QoS mechanisms to be added to the Internet, such as IntServ in early 1990s [6]. The IntServ type of admission control requires the participation of the routers. In an effort to avoid complicated roles by routers, distributed admission control schemes [7]–[11] were proposed. Some of these schemes work in the context of DiffServ architecture. In DiffServ, network administrators can configure network

1-4244-0476-2/06/$20.00

devices to partition the bandwidth for different classes of flows to ensure adequate performance for elastic flows.

Some other proposals for distributed admission control assume no network assistance, for example [9]–[11]. Most of these schemes try to probe for available bandwidth and admit an inelastic flow as long as bandwidth can be made available, which is equivalent to the aggressive admission control in this paper.

The most related works are [11], [12]. The authors of [11] proposed a form of admission control that is sensitive to the coexisting elastic flows. Ref. [12] proposed a flow level model of a network with file transfer and streaming traffic, in which the authors discussed admission control. The model in [12] is somewhat different than ours. They model admission control as a random decision depending on the relative magnitude of the prevailing TCP rate and the inelastic flow's desired rate. Despite the difference, our conclusions are consistent with the observations in [12]. However, we have the following improvement over [12]: Firstly, we do not involve utility functions to find a fair allocation, which in turn was defined as TCP-friendly. Rather, we define TCP-friendliness based on the bandwidth allocation directly. Secondly, [12] assuming a Markov chain and we relax the constraint to include the non-exponential distribution of file size, which more accurately reflects the traffic pattern in the Internet.

B. Contributions in this paper

If we consider distributed admission control as a form of congestion control to be exercised by inelastic flows, it turns out that we can make some interesting observation about the different ways admission control is to be designed. We looked at two example admission controls which are two cases in a spectrum of different admission control designs that give different amount of consideration to the elastic flows. We find that all these admission controls yield the same steady state performance for inelastic flows, asymptotically as we scale up the network. However, the different admission controls yield very different performance for elastic flows. This observation naturally leads us to the notion of *TCP-friendly admission control*. It is a pseudo-Nash-equilibrium strategy for inelastic flows to adopt; namely, if it is adopted, it does not hurt the adopter but helps others. The use of 'TCP-friendliness' here is subtly different than its use in 'TCP-friendly congestion control'. In our case, it refers to achieving a balance between the aggregate TCP traffic and inelastic traffic, therefore it is not on a per flow basis.

The contributions of this paper are as follows: Firstly, we derive the blocking probability and population size for inelastic flows in a stochastic fluid model of mixed traffic. Secondly, we give the stability conditions for such traffic models. These two lead to the conclusion that the performance of inelastic flows are insensitive to how aggressive the admission control is, asymptotically as we scale up the network. Thirdly, we also present numerical results to show that elastic traffic experience different performance, and argue the case for TCP-friendly admission control.

Note that, we are not proposing a specific implementation for admission control in this paper. Rather, we study the implication of admission control of inelastic flows to the elastic flows in coexistence. We show that, adopting admission control for inelastic flows is a viable option to achieve fair bandwidth allocation with TCP traffic and it is inherently more justified than TCP-friendly congestion control.

II. NETWORK AND ADMISSION CONTROL MODELS

This section first describes our network model and a generic admission control scheme. We then introduce AC-A and AC-F, which are two extreme examples of this scheme, and their Markov chain models.

A. Network model: Elastic and inelastic flows

We focus on a network with a single bottleneck shared by two classes of fluid traffic.

The first traffic class consists of *elastic flows*. These flows model TCP traffic—their objective is to transfer some files of finite size and, when necessary, perform congestion control. They try to fully utilise and share the available bandwidth. Elastic flows, by themselves, will converge to a fair bandwidth allocation [13]. For simplicity, we assume this convergence is immediate in our fluid model.

The second traffic class consists of *inelastic flows*. These flows model multimedia traffic—their objective is to transfer some files with a fixed *playback rate* α and for some finite *holding times*. The fixed rate α implies that inelastic flows do not perform congestion control.

Unlike elastic flows that adjust their transfer rates, the fixed rate for an inelastic flow raises the need for *admission control*: Before it begins, an inelastic flow checks if the bandwidth is sufficient to support α; if so, it is admitted, the flow begins, and continues without congestion control until the end of its holding time. Otherwise, it is not admitted and leaves. For simplicity, we assume the assessment for available bandwidth (e.g. through probes [9], [11], [14]) has negligible bandwidth cost in our fluid model.

We assume the bottleneck bandwidth is 1 without loss of generality. Further, the elastic and inelastic flows have Poisson arrivals, with arrival rates λ_{e} and λ_{i}, respectively. The elastic flows are assumed to have exponentially distributed file sizes with mean $1/\mu_{\mathrm{e}}$. Hence, if all the bandwidth is used to serve elastic flows, then they have departure rate μ_{e}. The inelastic flows have exponentially distributed holding times $1/\mu_{\mathrm{i}}$ and playback rate $\alpha \leq 1$; once an inelastic flow starts, it consumes bandwidth α until its departure.

Thus, when there are n elastic flows and m inelastic flows over the link, the inelastic flows consume bandwidth $m\alpha$, and elastic flows share equally the remaining $1 - m\alpha$.

B. A generic admission control scheme

Even if the bottleneck link is fully utilised, it is possible for an inelastic flow to be admitted by reducing the rate of elastic flows. However, the admission control must judge if the reduced rate provides acceptable performance for elastic flows.

We therefore model the admission control scheme as follows: If n and m are the current number of elastic and inelastic flows, an inelastic flow is admitted only if

$$n\epsilon + (m+1)\alpha \leq 1, \tag{1}$$

where $\epsilon > 0$ is a parameter for the scheme. In effect, an inelastic flow is admitted only if the elastic flows will each get at least bandwidth ϵ after the admission. Here, ϵ models the fact that each elastic flow takes some minimal bandwidth to maintain its connection.

Current concern in network engineering is over inelastic flows hogging too much of the bandwidth, rather than favoring elastic flows (i.e. $\epsilon > \alpha$) in the bandwidth allocation. We therefore assume $\epsilon \leq \alpha$.

C. Examples of admission control: AC-A and AC-F

We study the above generic admission control scheme by considering its two extremes. The first scheme, denoted AC-A, has $\epsilon \ll \alpha$, so the criterion (1) says that an inelastic flow is admitted if and only if

$$\frac{1-(m+1)\alpha}{n} \geq \epsilon. \tag{AC-A}$$

Since the bottleneck bandwidth is normalized to 1, the available bandwidth after admitting the inelastic arrival is $1-(m+1)\alpha$; hence AC-A requires that each existing elastic flow get at least bandwidth ϵ after admission. AC-A stands for aggressive version of admission control.

The second scheme, denoted AC-F, has $\epsilon = \alpha$, so criterion (1) says that an inelastic flow is admitted if and only if

$$\frac{1-(m+1)\alpha}{n} \geq \alpha, \tag{AC-F}$$

i.e. each elastic flow must get at least the same bandwidth as an elastic flow after admission. AC-F thus models current interest on imposing TCP-friendliness upon inelastic flows [3], [4]. AC-F stands for friendly version of admission control.

D. A Markov chain model of admission control

The network we have described is equivalent to a processor-sharing queue, and can be modeled by a Markov chain. Let (n, m) represent a state with n elastic flows and m inelastic flows. Then AC-A and AC-F each define a Markov chain with two-dimensional state space $\{(n, m) : n, m \in \mathbb{Z}_0^+\}$ where $\mathbb{Z}_0^+$ is the set of all integers greater than or equal to zero. Figure 1 illustrates this for AC-F.

The arrival rate of elastic flows is always λ_e because there is no admission control. For inelastic flows, the arrival rate is λ_i unless the arrival is rejected by the admission control. For elastic flows, the departure rate is $(1-m\alpha)\mu_e$ if $m\alpha < 1$ and zero otherwise, since each of the n elastic flows have bandwidth $(1-m\alpha)/n$. The departure rate of inelastic flows is μ_i, i.e. the reciprocal of their average holding time $1/\mu_i$. Table I lists the transition rates for AC-A and AC-F. Table II summarizes the parameters for this Markov chain.

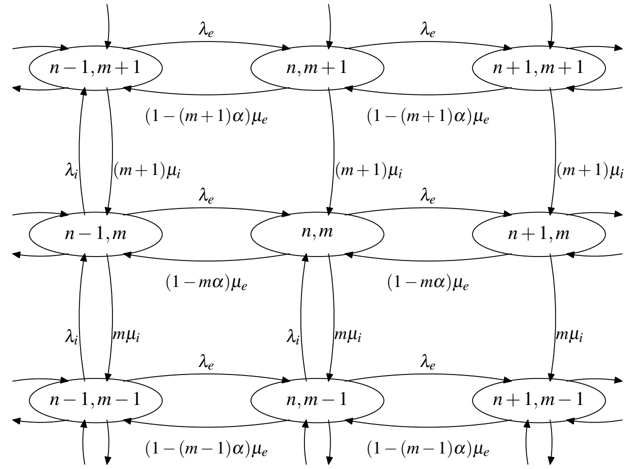


Fig. 1. Markov model of two types of flows with different controls in AC-F scheme, assuming $n\epsilon + m\alpha = 1$ and $\epsilon = \alpha$ in the transitions shown.

III. Stability of the admission control schemes

We now study the stability of AC-A and AC-F. A stability condition is a constraint on the load for a system, so we must first define offered load.

A. Offered load

From Table I, we see that if the number of inelastic flows is fixed at m, then the elastic flows behave identically as an M/M/1 processor-sharing queue [15] with capacity $1-m\alpha$. On the other hand, the inelastic flows—failed admission aside—behave like an M/M/∞ queue with each server having capacity α. Following standard queueing notation, we define the *offered load* for elastic flows as $\rho_e = \lambda_e/\mu_e$, and $\rho_i = \lambda_i/\mu_i$ for inelastic flows.

From queueing theory, ρ_e is the utilization of the M/M/1 queue (i.e. the fraction of bottleneck bandwidth that is used), whereas ρ_i is the number of busy servers in the M/M/∞ queue; therefore, the bottleneck bandwidth utilization by inelastic flows is $\alpha\rho_i$. Thus, the total offered load to the network is $\rho = \rho_e + \alpha\rho_i$, which represents the aggregated bandwidth demand by both flow classes.

A basic concern for any queue is whether it is *stable*, i.e. there is a steady state with a bounded number of jobs, rather than a steadily increasing queue. In our system, it may not be obvious that this is an issue; after all, inelastic arrivals will be rejected if congestion is too high, and elastic flows will always suitably adjust their flow rate.

In fact, stability is an issue for the elastic flows: When congestion forces them to reduce their flow rate, their transfer duration will increase; if the duration increases beyond the inter-arrival time $1/\lambda_e$, elastic flows will arrive faster than they depart, n increases without bound, and the queue is thus unstable.

One may reasonably guess that the stability regime for our network is $\rho < 1$, i.e. the bandwidth demand is less than

TABLE I

STATE TRANSITION RATES CORRESPONDING TO THE TWO MARKOV MODELS

		$(n,m)\to(n,m+1)$	$(n,m)\to(n+1,m)$	$(n,m)\to(n,m-1)$	$(n,m)\to(n-1,m)$
AC-A	$n\epsilon+(m+1)\alpha\le 1$	λ_{i}	λ_{e}	$m\mu_{\mathrm{i}}$	$(1-m\alpha)\mu_{\mathrm{e}}$
	$n\epsilon+(m+1)\alpha>1$	0	λ_{e}	$m\mu_{\mathrm{i}}$	$\max(0,(1-m\alpha)\mu_{\mathrm{e}})$
AC-F	$(n+m+1)\alpha\le 1$	λ_{i}	λ_{e}	$m\mu_{\mathrm{i}}$	$(1-m\alpha)\mu_{\mathrm{e}}$
	$(n+m+1)\alpha>1$	0	λ_{e}	$m\mu_{\mathrm{i}}$	$\max(0,(1-m\alpha)\mu_{\mathrm{e}})$

TABLE II

MODEL PARAMETERS AND PERFORMANCE METRICS

Symbol	Explanation
λ_e	arrival rate of elastic flows
λ_i	arrival rate of inelastic flows
μ_e	reciprocal of the average file size sent by elastic flows
μ_i	reciprocal of the average holding time for inelastic flows
α	the desired throughput for inelastic flows (also known as playback rate)
ϵ	the minimum bandwidth that an elastic flow must receive after admitting an inelastic flow
n	number of elastic flows
m	number of inelastic flows
ρ_e	offered load due to elastic flows, $\rho_e=\lambda_e/\mu_e$
ρ_i	offered load due to inelastic flows, $\rho_i=\lambda_i/\mu_i$
ρ	total offered load, $\rho=\rho_e+\alpha\rho_i$
ρ'	actual load, i.e. utilization
R	admission probability for inelastic flows

the bottleneck bandwidth. We shall see that, because inelastic flows can be rejected, stability is in fact possible for $\rho\ge 1$.

To facilitate our stability analysis, we define an admission function $I(n,m)$ to be the indicator for whether criterion (1) is satisfied, so an inelastic flow is admitted if and only if $I(n,m)=1$. Thus,

$$\text{For AC-A:}\qquad I(n,m)=\begin{cases}1 & \text{if } n\epsilon+(m+1)\alpha\le 1\\ 0 & \text{otherwise;}\end{cases}$$

$$\text{For AC-F:}\qquad I(n,m)=\begin{cases}1 & \text{if } (n+m+1)\alpha\le 1\\ 0 & \text{otherwise.}\end{cases}$$

B. Stability analysis

By our definition of the admission function, both n and m may increase with flow arrival if $I(n,m)=1$. If $I(n,m)=0$, then an inelastic flow that arrives at the moment will not be admitted. The number of inelastic flows is thus bounded; specifically,

$$0\le m\le\left\lfloor\frac{1}{\alpha}\right\rfloor.$$

The M/M/∞ queue for the inelastic flows is therefore (unconditionally) stable.

For the elastic flows, their M/M/1 processor-sharing queue has a reduced bandwidth of $1-m\alpha$, so its stability condition is $\rho_{\mathrm{e}}<1-m\alpha$, if m is fixed. However, if $\rho_{\mathrm{e}}\ge 1-m\alpha$, the M/M/1 queue becomes unstable, and n increases without bound, then eventually $n\epsilon>1$ for AC-A or $n\alpha>1$ for AC-F; by then, $I(n,m)=0$, new inelastic arrivals are rejected, m eventually decreases almost surely, and the capacity $1-m\alpha$ for the M/M/1 queue increases, possibly restoring stability.

We thus see that rejection of inelastic flows helps to restore stability, so stability may not be constrained by $\rho_{\mathrm{e}}+\alpha\rho_i<1$. What then is the stability condition for our network?

According to [15], we know that in a M/M/1 processor-sharing queue, the queue is stable, that is the number of customers in queue is bounded, if and only if the offered load $\rho<c$, where c is the capacity of the queue. Then we can state the following:

Theorem 1: In the admission controlled network as described in section II, the network is stable if and only if the offered load due to elastic flows ρ_{e} satisfies $\rho_{\mathrm{e}}<1$.

Sketch of Proof: ($\Rightarrow$) If the network is stable, then we let the average number of inelastic flows be $\bar{m}\ge 0$. Therefore, the average reduced bandwidth, which is used to serve the elastic flows, is $1-\bar{m}\alpha$. From the property of M/M/1 processor-sharing queue, we have

$$\rho_{\mathrm{e}}<1-\bar{m}\alpha<1.$$

($\Leftarrow$) Let us assume that $\rho_e<1$ but the network is unstable. By the definition of instability, we have

$$\lim_{t\to\infty}\Pr[n(t)=\infty]=1.$$

Therefore we have some t' such that

$$n(t)\epsilon>1 \qquad\qquad \forall t\ge t'.$$

In other words,

$$I(n(t),m(t))=0 \qquad\qquad \forall t\ge t' \tag{2}$$

where $m(t)\ge 0$ denotes the number of inelastic flows in the network at time t. Let $m(t')=m'$, due to (2), there would be no inelastic admission at $t\ge t'$. If $m'\ne 0$, then almost surely,

$$m(t'+\Delta t)=0$$

for some random time Δt, which satisfies the density function

$$f_{\Delta t}(x) = \mu_{\mathrm{i}} e^{-\mu_{\mathrm{i}} x} \frac{(\mu_{\mathrm{i}} x)^{m'-1}}{(m'-1)!}.$$

In other words, we have

$$m(t) = 0 \qquad\qquad \forall t \geq t''$$

almost surely for some $t'' \geq t'$. Thus at $t \geq t''$, the network is serving only the elastic flows and behaves like a M/M/1 processor-sharing queue. Therefore, $\rho_{\mathrm{e}} < 1$ implies the network is stable and $n(t)$ should be bounded, which contradicts our assumption. ■

This result, based on a similar model of heterogeneous networks, has already been established in previous works [1], [12]. We state it and prove it here because it also leads to our main observations of this paper, namely,

1) the stability condition for AC-A and AC-F are the same, and
2) the stability condition is as if there are no inelastic flows.

In particular, the network with $\rho_{\mathrm{e}} < 1$ will be stable even if $\rho > 1$.

We should emphasize that Theorem 1 does not mean that n is bounded for any positive m. Rather, an increasing n will eventually set $I(n, m) = 0$ (as previously described), shut out inelastic arrivals, and thus return the queue to stability.

One can thus see that the stability of the generic admission control scheme (1) is independent of the choice of ϵ and the parameters $(\alpha, \lambda_{\mathrm{i}}, \mu_{\mathrm{i}})$ for the inelastic flows. We will see similar insensitivity in next section.

IV. Admission probability for inelastic flows

Given that inelastic flows, once admitted, are guaranteed the playback rate α. The service they get is 'all or nothing', so the only performance measure of an admission control that is of interest to inelastic flows is their probability of admission.

In this section, we will treat the flows as fluids and use stochastic differential equations to derive a closed-form expression for this admission probability R; it leads us to a surprising corollary that R is insensitive to ϵ. We first introduce some notation and briefly review stochastic differential equations.

A. Notations

The stochastic behavior of our system can be described by the following metrics:

We refer to $\tau_k(t)$ as the remaining amount of data to be sent by flow k at time t. By our fluid approximation, if flow k is an elastic flow,

$$\frac{d}{dt}\tau_k(t) = -\frac{1 - m\alpha}{n}$$

when the system is in state (n, m). If flow k is inelastic, $\frac{d}{dt}\tau_k(t) = -\alpha$. These derivatives are simply the respective flow rates. Then, the total remaining amount of data to be sent by all the flows at t is $\tau(t) = \sum_k \tau_k(t)$.

$N_{\mathrm{e}}(t)$ and $N_{\mathrm{i}}(t)$ are the Poisson counters measuring the number of elastic and inelastic arrivals, respectively, in the time interval $[0, t)$.

Now we can relate $\tau(t)$, $N_{\mathrm{e}}(t)$ and $N_{\mathrm{i}}(t)$ using *Poisson counter driven stochastic differential equations* [16], [17], which we will use to derive the admission probability for inelastic flows.

Let $N(t)$ be a Poisson counting process with parameter λ. Suppose $N(t')$ is unchanged for all $t' \in [t, t+\delta t)$ and $N(t+\delta t) = N(t) + 1$, then δt has density $\lambda e^{-\lambda x}$. Therefore, for an infinitesimal dt, the probability

$$\Pr[N(t+dt) = N(t) + 1] = \lambda dt. \tag{3}$$

The differential of $N(t)$ is defined with dt by

$$dN(t) = \lim_{\delta t \to 0} N(t+\delta t) - N(t) = N(t+dt) - N(t),$$

so $dN(t)$ is 0 if $N(t+dt) = N(t)$ and 1 if $N(t+dt) = N(t) + 1$. It follows from (3) that

$$E[dN(t)] = \lambda dt. \tag{4}$$

If we have a stochastic process $X(t)$ on real numbers and it depends on Poisson counters $N_1, N_2, \ldots, N_r$, the stochastic differential equation is of the form

$$dX(t) = f(X(t), t)dt + \sum_{j=1}^{r} g_j(X(t), t) dN_j(t) \tag{5}$$

for some real-valued functions f and g_j. We can solve for $X(t)$ by applying stochastic integration on (5), but it may suffice, and is usually easier, to just solve for the expectation $E[X(t)]$, through

$$dE[X(t)] = E[f(X(t), t)]dt + \sum_{j=1}^{r} E[g_j(X(t), t) dN_j(t)].$$

B. Admission probability R

The admission probability for inelastic flows is $R = \Pr[I(n, m) = 1] = E[I(n, m)]$. The following result gives a closed-form expression for R.

Theorem 2: In steady state, the probability of admission R for an arriving inelastic flow in our generic admission control scheme (section II-B) is

$$R = \frac{\Pr[\tau > 0] - \rho_{\mathrm{e}}}{\alpha \rho_{\mathrm{i}}}, \qquad\qquad \text{for any } \epsilon > 0.$$

Proof: Consider the remaining work $\tau(t)$. When there are no arrivals, $\tau(t)$ would decrease with rate $\mathbf{1}(\tau > 0)$, where the indicator function $\mathbf{1}(\tau(t) > 0) = 1$ if $\tau(t) > 0$, and 0 otherwise.

When there is an elastic arrival at t, $\tau(t)$ increases by a random value S_{e} according to the file size distribution for elastic flows. Similarly, $\tau(t)$ increases by some S_{i} when an inelastic flow is admitted at time t. Therefore, omitting t's for convenience, we have

$$d\tau = -\mathbf{1}(\tau > 0)dt + S_{\mathrm{e}} dN_{\mathrm{e}} + I(n, m) S_{\mathrm{i}} dN_{\mathrm{i}}$$

where dN_e and dN_i are 1 if there is an elastic or inelastic arrival (respectively) at time t, and 0 otherwise. Since S_e, S_i and the arrival processes are independent, taking expectation and using (4),

$$\begin{aligned} dE[\tau] &= E[-\mathbf{1}(\tau > 0)]dt + E[S_\mathrm{e}dN_\mathrm{e}] + E[I(n,m)S_\mathrm{i}dN_\mathrm{i}] \\ &= -E[\mathbf{1}(\tau > 0)]dt + E[S_\mathrm{e}]E[dN_\mathrm{e}] + \\ &\quad E[I(n,m)]E[S_\mathrm{i}]E[dN_\mathrm{i}] \\ &= -\Pr[\tau > 0]dt + \frac{1}{\mu_\mathrm{e}}\lambda_\mathrm{e}dt + \Pr[I(n,m) = 1]\alpha\frac{1}{\mu_\mathrm{i}}\lambda_\mathrm{i}dt \end{aligned}$$

since elastic flows have mean file size $1/\mu_\mathrm{e}$ and inelastic flows have flow rate α and mean duration $1/\mu_\mathrm{i}$. As $R = Pr[I(n,m) = 1]$, we get

$$dE[\tau] = -\Pr[\tau > 0]dt + \rho_\mathrm{e}dt + R\alpha\rho_\mathrm{i}dt,$$

so

$$\frac{dE[\tau]}{dt} = -\Pr[\tau > 0] + \rho_\mathrm{e} + R\alpha\rho_\mathrm{i}.$$

If the network is in steady state, $dE[\tau]/dt = 0$ and the theorem follows. ■

Recall from Theorem 1 that our system is stable if and only if $\rho_\mathrm{e} < 1$, even if offered load $\rho > 1$. Since inelastic arrivals may be rejected, the *actual load* or *utilization* is ρ', for some $\rho' \leq \rho$. Now, utilization is $\Pr[\tau > 0]$, so it follows from Theorem 2 that

$$R = \frac{\rho' - \rho_\mathrm{e}}{\alpha\rho_i}. \tag{6}$$

C. An approximation for R

When ρ is small, the system is lightly loaded and rejections are rare, so $\rho' \approx \rho$. On the other hand, when $\rho > 1$, the offered load exceeds capacity, so one expects the system to be fully utilised, i.e. $\rho' \approx 1$. We can therefore approximate ρ' by

$$\rho' \approx \begin{cases} \rho & \text{if } \rho < 1 \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

Equation (6) now gives the following corollary to Theorem 2:

Corollary 3: For any $\epsilon > 0$,

(a) the average number of inelastic flows is

$$\bar{m} = (\rho' - \rho_\mathrm{e})/\alpha;$$

(b) the admission probability is

$$R \approx \frac{\min(\rho, 1) - \rho_\mathrm{e}}{\alpha\rho_\mathrm{i}}.$$

Proof: (a) follows from $\bar{m} = R\rho_\mathrm{i}$ and (6); (b) follows from applying approximation (7) to (6). ■

Recall from Theorem 1 the surprising result that stability of our admission control scheme is independent of ϵ. The above corollary is another surprise: it suggests that the bandwidth share by inelastic flows and the admission probability are also independent of ϵ. Moreover, this result *does not* depend on the assumption that the file size of elastic flows or the holding time of inelastic flows is exponentially distributed. We will verify this in the next Section.

V. Verification by Simulation and Discussions

This section presents results from discrete-event simulations which simulated the fluid model Markov chain in section II-D. In particular, we validate the stability condition in Theorem 1, verify the approximation in Corollary 3, and study the scalability of the admission scheme and our approximations. We also examine the case for TCP-friendliness, in light of our theoretical and simulation results.

In our simulation, we increase ρ from 0.1 until $\rho_\mathrm{e} = 0.99$. We considered three ratios for $\rho_\mathrm{e} : \alpha\rho_\mathrm{i}$, namely 3:7, 1:1 and 7:3. In other words, we simulated scenarios where the elastic offered load ρ_e is 30%, 50% and 70% of total offered load ρ.

A. R and $\bar{m}$

We started with $\mu_\mathrm{e} = \mu_\mathrm{i} = 10$, $\alpha = 0.05$ and $\epsilon = 0.001$. Figure 2(a) compares the approximated R in Corollary 3(b) against the measured value. It shows that the approximation is excellent for AC-A. The agreement is also good for AC-F, except around $\rho = 1$.

The 'all or nothing' admission criterion (1) implies that, the bigger ϵ is, the harder it is for an inelastic flow to be admitted, and the poorer our approximation for R. This is why Corollary 3 gives a better approximation for AC-A ($\epsilon = 0.001$) than for AC-F ($\epsilon = \alpha = 0.05$). In any case, Figure 2(a) confirms our observation from Corollary 3 that AC-A and AC-F have approximately equal admission probabilities.

For the average number of inelastic flows, we compared the measured $\bar{m}$ against the value obtained by substituting the approximation (7) into Corollary 3(a). Figure 2(b) shows that the agreement is similarly good.

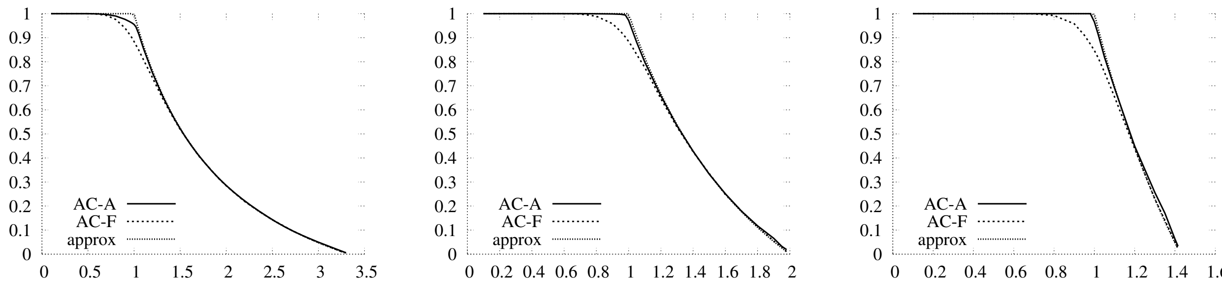
These figures confirm the theoretical result in Theorem 1: elastic flows continue to be admitted, i.e. $R > 0$ (Figure 2(a)), and $\bar{m}$ remains nonzero (Figure 2(b)) even for $\rho > 1$; when R reaches 0, all the bandwidth goes to the elastic flows and the stability of the queue is then governed by $\rho_\mathrm{e} < 1$. When $\rho_\mathrm{e} > 1$, the M/M/1 processor-sharing queue is unstable and thus no steady state result can be obtained from simulation.

B. Implication for the admission control schemes

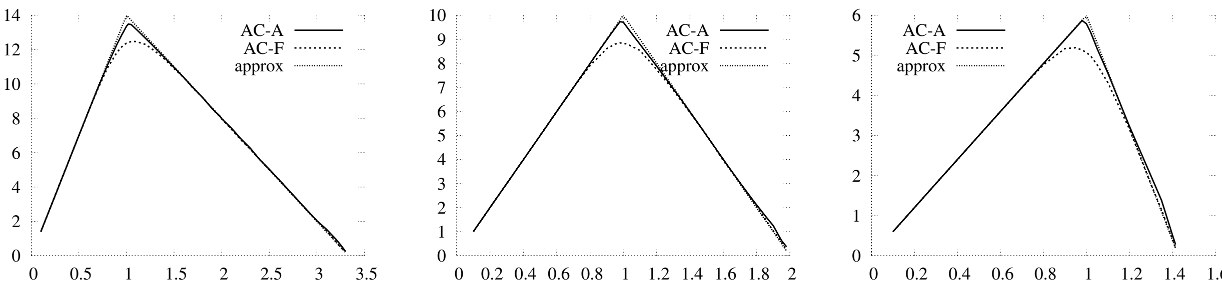
The close agreement between AC-A and AC-F in Figure 2(a) and 2(b) has an important implication for admission control: namely, using a 'friendly' scheme like AC-F would not significantly reduce the admission probability and bandwidth share for inelastic flows.

For elastic flows, however, an 'aggressive' scheme like AC-A is significantly worse than AC-F. Figure 2(c) plots the average number $\bar{n}$ of elastic flows for the two schemes. Note the logarithmic scale for the vertical axis, which shows that AC-A has $\bar{n}$ values that are orders of magnitude greater than AC-F when $\rho > 1$.

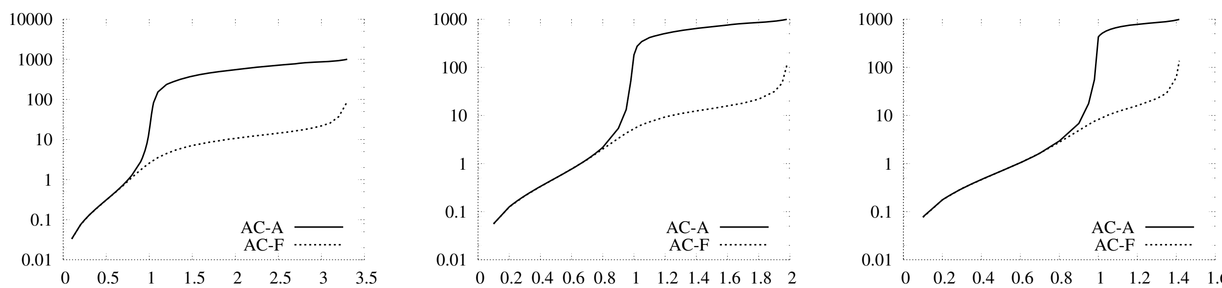
Since AC-A and AC-F have similar values for $\bar{m}$, the elastic flows must share a similar amount of bandwidth $1 - \bar{m}\alpha$ in both schemes. Since AC-A has a much higher $\bar{n}$ than AC-F, it follows that AC-A has much lower bandwidth per elastic flow.



(a) R vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(b) $\bar{m}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(c) $\bar{n}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ

Fig. 2. Simulation results with $\alpha = 0.05$

In contrast, notice from Figure 2(c) that $\bar{n}$ for AC-F increases gently, so elastic flows do not suffer the sudden, drastic drop in flow rate when ρ exceeds 1.

To summarise, an admission control scheme like AC-F that is TCP-friendly ($\epsilon = \alpha$) provides performance for inelastic flows that is as good as an unfriendly scheme ($\epsilon \ll \alpha$), but at the same time much better performance for the elastic flows.

C. Asymptotic behavior

In this section, we study the sensitivity of our analysis to the network scale. We find our approximations become asymptotically exact as we scale up the network.

We study this issue by scaling up the model parameters by a factor of 2. To help us see the trend, we also scale them down by 0.8. This scaling can be considered as replacing the network with a bottleneck of 2 or 0.8 times the original capacity but serving 2 or 0.8 times of the original population of users. Table III lists the parameter values, and the simulation results are shown in Figures 4(a), 4(b) and 4(c) for scaling up and Figures 3(a), 3(b) and 3(c) for scaling down.

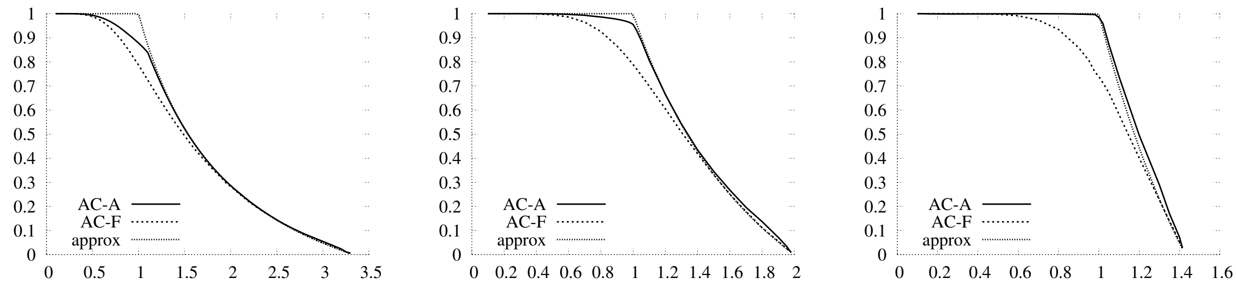
TABLE III

PARAMETER SETS USED, x AND y ARE VARYING QUANTITIES.

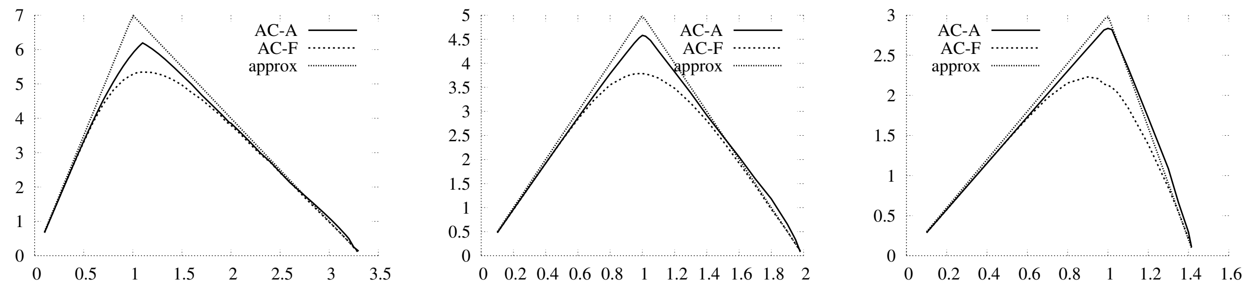
Figure	α	ϵ	μ_e	μ_i	λ_e	λ_i	Scaled factor
3	0.0625	0.00125	8	10	$0.8x$	$0.8y$	0.8
2	0.05	0.001	10	10	x	y	1
4	0.025	0.0005	20	10	$2x$	$2y$	2

As the scale increases from Figure 3(a) to Figure 2(a) to Figure 4(a), we see that the difference in admission probability between AC-A and AC-F disappears.

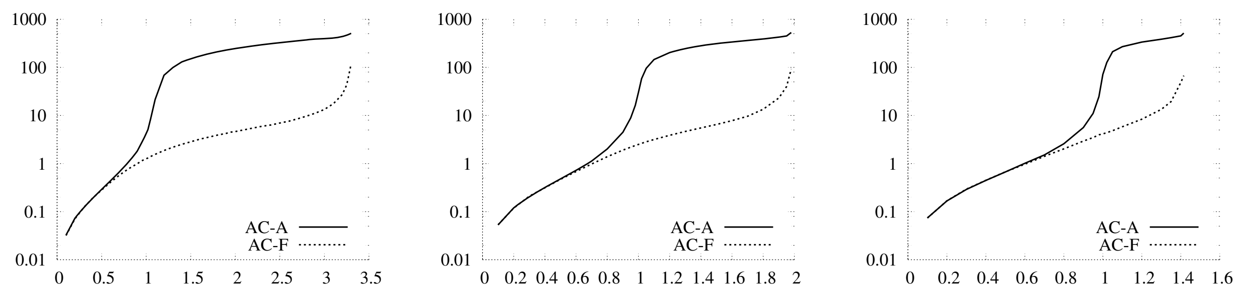
The same is true for the difference in $\bar{m}$ as we go from Figure 3(b) to Figure 2(b) to Figure 4(b); this implies that the difference between AC-A and AC-F in bandwidth share for inelastic flows disappears when the parameters are scaled up. Figure 3(b) shows there is a significant discrepancy in $\bar{m}$ from



(a) R vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(b) $\bar{m}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(c) $\bar{n}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ

Fig. 3. Simulation results with $\alpha = 0.0625$, i.e. after scaling by a factor of 0.8.

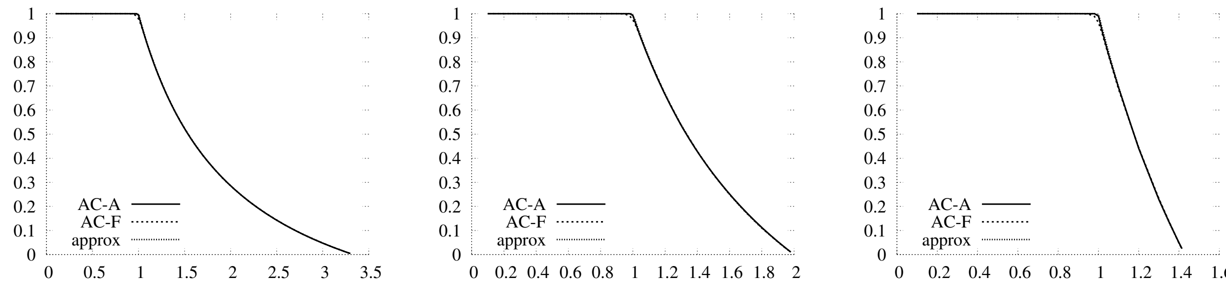
simulation and the approximation by (7) with Corollary 3(a). This is also caused by the coarse approximation of R when ϵ is large and increasing the granularity of ϵ can significantly improve the approximation.

For the elastic flows, we see that, going from Figure 3(c) to Figure 2(c) to Figure 4(c), the increase in $\bar{n}$ near $\rho = 1$ becomes more abrupt. This means that, even a TCP-friendly admission control like AC-F cannot prevent a sudden performance degradation for elastic flows when ρ exceeds 1. Even so, Figure 4(c) shows that this degradation is nowhere near as bad as for an unfriendly scheme like AC-A.

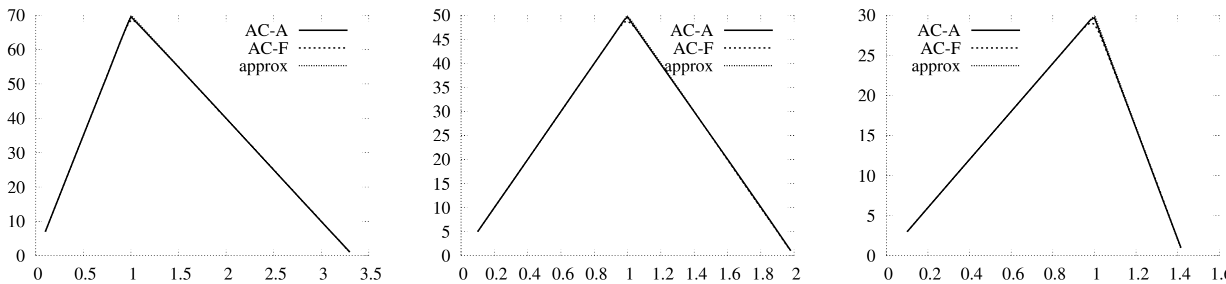
VI. Implementation issue

The above discussion is based on a fluid model and a set of assumptions. It is reasonable to ask whether the suggested form of admission control can be implemented, and whether the theoretical results still hold in a network without the above idealized assumptions.

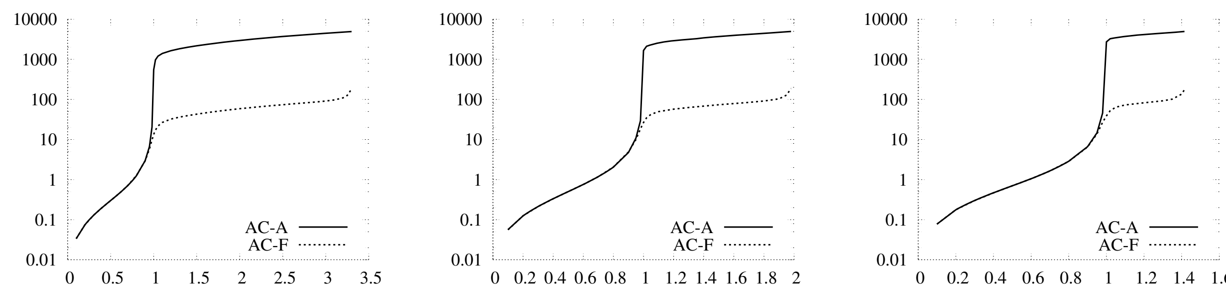
Although the implementation of admission control and experimentation are not the focus of this paper, we want to comment briefly on how one might approach the problem. Numerous papers have discussed various schemes to implement admission control, some being centralized schemes involving routers, and others decentralized schemes with varying degree of router support. We are particularly interested in a completely distributed implementation since it would be the easiest to deploy. Of course, such distributed schemes will not have perfect information and accuracy of the admission decision will need to be traded off with the overhead to probe the network state as well as admission delay. A plausible implementation may roughly go like this. An inelastic flow starts itself like a TCP flow, hence naturally probes for available bandwidth in a TCP-compatible way. After probing for some suitable period of time, the inelastic flow checks whether it has attained a transmission rate it desires, and makes



(a) R vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(b) $\bar{m}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ



(c) $\bar{n}$ vs. ρ, with ρ_{e} constituting 30% (left), 50% (middle), and 70% (right) of ρ

Fig. 4. Simulation results with $\alpha = 0.025$, i.e. after scaling by a factor of 2.

a decision to admit itself or quit accordingly. A procrastinating flow may indefinitely postpone the decision until it finds the network is hopeless and quits. These different implementations are being evaluated and the tradeoffs are under further study. The rough distributed admission control algorithm above is put in pseudo-code in Algorithm 1.

Note, there is a loop in Algorithm 1 (line 3–7), lasting a short duration t, used to probe the prevailing rate of TCP flows. Compare this algorithm to a typical TCP-friendly *congestion control* algorithm (such as TFRC [5]) as depicted in Algorithm 2. There is also a loop used to continuously adjust the sending rate to the prevailing TCP rate based on network feedback such as packet loss and round-trip time. But the loop in Algorithm 2 continues until the file transfer is over.

According to the result of this paper, using a smaller ϵ in line 9 of Algorithm 1 will not noticeably increase your chance of admission but setting $\epsilon = \alpha$ would make the scheme TCP-friendly. As we mentioned in the introduction, such *TCP-friendliness* is at the aggregated level (comparing total TCP flow rates with total inelastic flow rates), while that of TCP-friendly congestion control is a per-flow level fairness. The effect of the probing overhead, packetization and delay is being further investigated.

VII. CONCLUSION

In this paper, we consider admission control as a distributed mechanism for inelastic flows to negotiate its bandwidth usage in the presence of elastic flows regulated by congestion control. From [1], it is shown that such admission control schemes can potentially be even more fair to elastic (TCP) traffic than TCP-friendly congestion control. The main contribution of this paper is to show two important (asymptotic) properties of admission control in such systems: (a) the performance for inelastic flows is insensitive to how aggressive (relative to elastic traffic) these flows try to admit themselves; (b) the aggressiveness of inelastic flows, however, makes a big

Algorithm 1 Skeleton algorithm of admission control for inelastic flows

Set probe duration t base on flow parameters
  /* $t \ll$ holding time */
$r :=$ initial sending rate (≈ 0)
repeat
  Send data with rate r
  $\mathbf{x} :=$ Network feedback
  $r := f(\mathbf{x})$
   /* Update rate r according to TCP's algorithm */
until probe duration t expire
$\bar{r} :=$ average of r in probe duration
if $\bar{r} \leq \epsilon$ **then**
  Terminate transfer
  Disconnect
else
  Transfer with constant rate $r = \alpha$ until finish
end if

Algorithm 2 Skeleton algorithm of TCP-friendly congestion control like TFRC

$r :=$ initial sending rate (≈ 0)
repeat
  Send data with rate r
  $\mathbf{x} :=$ Network feedback
  $r := f(\mathbf{x})$
   /* Update rate r according to TCP's algorithm */
until Transfer complete

difference to the performance of elastic flows. This leads to the notion of a *TCP-friendly* admission control, where an inelastic flow admits itself only if its desired rate is no higher than the prevailing fair share for TCP flows. Not only is this TCP-friendly admission control fair to TCP flows, we argue that it is also incentive-neutral to the inelastic flows in the sense that it does not hurt themselves to be *nice*. Another contribution and difference between this paper and the previous results [1] is that there is no need to make any assumptions about utility functions.

Although our problem formulation in section II is based on a Markov Chain model, our main conclusion (the case for TCP-friendly admission control) is built upon the Poisson counter driven stochastic differential equation model, which does not depend on exponentially distributed file size or holding time. This makes our result more generally applicable.

ACKNOWLEDGMENT

We thank the reviewers for their valuable comments on the draft version of this paper.

REFERENCES

[1] D. M. Chiu and A. S.-W. Tam, "Network fairness for heterogeneous applications," in *Proc. of the 1st ACM SIGCOMM Asia Workshop*, 2005.
[2] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service." IETF RFC2212, Sep 1997.
[3] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, 1999.
[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. of SIGCOMM*, Aug 2000.
[5] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification." IETF RFC3448, Jan 2003. Proposed Standard.
[6] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," in *Proc. of ACM SIGCOMM*, 1995.
[7] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed admission control," *IEEE Journal on Selected Areas in Communications*, vol. 18, 2000.
[8] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: Architectural issues and performance," in *Proc. of SIGCOMM*, 2000.
[9] V. Elek, G. Karlsson, and R. Rönngren, "Admission control based on end-to-end measurements," in *Proc. of INFOCOM*, 2000.
[10] G. Bianchi, A. Capone, and C. Petrioli, "Throughput analysis of end-to-end measurement-based admission control in IP," in *Proceeding of INFOCOM*, 2000.
[11] G. Karlsson, H. Lundqvist, and I. Más Ivars, "QOS P.D.Q.," Tech. Rep. TRITA-IMIT-LCN R 03:06, KTH, Sweden, Dec. 2003.
[12] P. Key, L. Massoulié, A. Bain, and F. Kelly, "Fair internet traffic integration: Network flow models and analysis," *Annales des Télécommunications*, vol. 59, 2004.
[13] D. M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, 1989.
[14] P. Key and L. Massoulié, "Probing strategies for distributed admission control in large and small scale systems," in *Proc. of INFOCOM*, 2003.
[15] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice Hall, 1987.
[16] R. W. Brockett, "Stochastic control." Lecture Notes, Harvard University., 1983. Provided by the author.
[17] R. W. Brockett, W.-B. Gong, and Y. Guo, "New analytical methods for queueing systems," in *Proc. of the 38th Conference on Decision and Control*, IEEE, Dec 1999.