Fairness of Traffic Controls for Inelastic Flows In the Internet

Dah Ming Chiu Adrian Sai-Wah Tam

***Abstract*— In best-effort networks, fairness has been used as a criterion to guide the design of traffic controls. The notion of fairness has evolved over time, from simple equality to a form of equality modulated by the user's need (e.g. max-min and proportional fairness). However, fairness has always been defined on a per-user basis for a deterministic workload. In this paper, we argue that we must redefine the notion of fairness when we study traffic controls for the co-existence of elastic and inelastic traffics. Our results indicate that subjecting inelastic flows to fairness congestion control on a per-flow basis does not necessarily maximize the network's utility. Instead, inelastic flows may follow their own form of traffic control, such as admission control (without congestion control). At the aggregate level, our results indicate that it still makes sense to maintain a balance between elastic and inelastic traffic. In order to support our arguments, we develop a methodology for comparing different traffic controls for given utility functions and different workloads, both deterministic and stochastic.**

***Index Terms*— congestion control, admission control, fairness, utility maximization, non-convex utility function, stochastic traffic model**

I. Introduction

Internet is a connectionless network. It relies on congestion control implemented in the end-systems to prevent offered load exceeding network capacity, as well as evenly allocate network resources to different users and applications. In the past, the applications (such as email, file transfer) were predominantly *elastic*, or in other words *flexible* in their bandwidth requirements. The Internet architecture served these applications well.

Dah Ming Chiu and Adrian Sai-Wah Tam are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (email: {dmchiu,swtam3}@ie.cuhk.edu.hk).

This work was partially supported by University Grants Committee, Hong Kong SAR under RGC grant 4232/04E (project 2150420) and AoE grant AoE/E-01/99 (project 4801312).

As the Internet shifts to support increasing volume of multimedia applications, there has been continuous debate on the next generation Internet architecture. Various proposals have been made for building a multi-services network to support different types of applications, for example ATM [1], Integrated Service [2], Differentiated Service [3] and MPLS-based traffic engineering [4]. Despite such efforts, a prevalent belief is that the Internet's connectionless service model does not need to change, as long as there is adequate provisioning of network bandwidth (see [5] for a systematic discussion of this viewpoint). What is needed instead is a relaxed end-system congestion control that co-exists with the widely used TCP congestion control.

So what is a suitable alternative congestion control for multimedia applications to practice? The orthodox solution requires all applications to share network bandwidth fairly, as existing TCP flows do. If an application needs more bandwidth than the prevailing fair share, then it should adapt its bandwidth (down) in favor of fairness. It is recognized that multimedia flows need more gradual adaptation to fair bandwidth share, so the effort of designing the alternative control is focused on a smooth transient response in bandwidth adaption. Proposals of such congestion control schemes are generally referred to as *TCP-friendly congestion controls* in the literature [6]–[8].

The thesis of this paper is to argue for abandoning the traditional notion of *fairness* in designing end-system traffic controls for different types of applications. Unlike the case with elastic traffic, the best way to deal with congestion for inelastic traffic should be some form of admission control, which is by definition *unfair* in the traditional sense. Furthermore, it is important to consider the stochastic nature of bandwidth allocation, rather than the allocation of bandwidth to a fixed number of flows. The fact that flows arrive at different times and have different demands for the network should also be taken into account in fair bandwidth allocation.

While we argue against insisting on per-flow TCP-friendliness, our results show that in a self-regulated network, it is still sensible to apply TCP-friendliness principles at the aggregate level. In other words, it yields higher utility for both kinds of traffic to maintain a balanced allocation according to the respective demand.

The methodology used to substantiate and support these arguments is to define the bandwidth allocation problem as a network utility optimization problem. There is significant prior literature on this approach which we will review. Our contribution is to extend the standard network utility maximization by considering non-concave utility functions to model inelastic flows, as well as considering both the elastic and inelastic traffic as stochastic processes with finite sizes. Both of these extensions have been considered recently in separate contexts [9]–[11], but we apply these extensions together to develop a methodology to evaluate different traffic controls for the co-existence of TCP and non-TCP flows. In particular, it is challenging to account for the utility of flows of different sizes. We adopt a decomposition model for accounting the utility of flows of different sizes which makes the computation feasible.

The rest of the paper is organized as follows. In section II, we briefly review the classic results on bandwidth allocation and fairness of network traffic controls. In section III, we develop our methodology for studying network fairness and discuss the validity of the TCP-friendliness approach. In section IV, we apply our approach to a homogeneous class of flows, the utility function is parametrized to model both elastic as well as inelastic flows. In section V, we apply our methodology to a scenario where heterogeneous flows (both elastic and inelastic flows) co-exist. We introduce idealized models of several traffic controls for inelastic flows, and apply our methodology to compare them. Specifically, for inelastic flows, we compare how admission control fares against TCP-friendly (like) controls, for the deterministic workload case in section VI and stochastic workload in section VII. Finally, the significance of this work and future directions are discussed in the concluding section.

II. Review of previous work

Actually, some of the basic ideas we are espousing have been discussed at some length in a seminal paper by Shenker ten years ago [5]. Shenker pointed out the obvious benefits in merging different types of networks (i.e. data, voice and TV) into a single network. While *over-provisioning* can always satisfy the needs of such an integrated network, Shenker argued that it is more effective to introduce multiple services into the network to support the different applications. Most importantly, to measure how good a network is, he introduced the notion of utility maximization. Shenker elaborated on some different forms of utility function for different types of (e.g. elastic and inelastic) flows, and applied utility maximization to justify the role for admission control in the situation when the utility function is non-concave. Although [5] was written in the context of advocating Integrated Service [2], which pre-dated subsequent work on end-system admission control [12] and TCP-friendly congestion control [6], its analysis and discussion of network design goals, viz. the utility optimization framework, still applies to end-system based traffic controls.

Before the utility maximization formulation, various performance metrics had been adopted in studying network traffic control algorithms. In addition to the obvious goals of maximizing the throughput and minimizing the delay, *fairness* was also adopted as an important goal [13], [14]. In a simple setting of a fixed number of flows sharing a common bottleneck, the simple notion of fairness corresponding to dividing bandwidth equally among competing flows seems particularly appealing. Both distributed algorithms (e.g. AIMD [13]) and centralized algorithms (e.g. fair queueing [15]) were proposed to implement fair traffic control. The emphasis of equality was also extended to the case when not all flows share the same path. In a general network topology with arbitrary flows, fair bandwidth allocation is first applied to the most limiting bottleneck, and iteratively to all bottlenecks, leading to the definition of *max-min fairness*.

The theory of network utility maximization blossomed when Kelly et al applied it to create a fluid model of the Internet with elastic applications under congestion control [16]. Kelly's model reduces the complicated network under distributed congestion control to a standard convex optimization problem

$$\max_{\mathbf{x}} \sum_i U_i(x_i)$$
$$\text{subject to } \mathbf{Rx} \leq \mathbf{c}$$

where $\mathbf{x}$ denotes the transmission rate of a set of flows; the flows' routes (which links are use by a flow) are represented by the zero-one matrix $\mathbf{R}$; the capacity of the links are represented by $\mathbf{c}$, and the utility of the i-th flow is represented by $U_i(x_i)$.

One way to derive the solution for this problem is through decomposing the problem into a set of sub-problems for the flows and the network, and this has a straight correspondence to Internet's end-system congestion control with network feedback. Kelly showed that the simplistic AIMD-like [13] distributed congestion control adopted in the Internet is stable, and achieves a sensible operating point assuming logarithmic utility functions. The resultant bandwidth allocation is referred to as being *proportionally fair*. In [17], a more general family of utility functions (of which the logarithmic utility is only a special case) is introduced. Based on this family of utility functions, it is possible to relate the different types of network fairness, such as max-min fairness, proportional fairness, and maximum-throughput allocation, as the solutions of utility maximization corresponding to different utilities. While this theory is very elegant, because its assumption of concave utility functions (to make the problem tractable as convex optimization problems), it is limited to characterize a network with elastic flows only. Furthermore, we also believe the assumption that all flows stay indefinitely long for the system to reach a steady state is too limiting.

Recently, some authors begin to consider the utility maximization problem when not all users have concave utility functions [10], [11]. This considerably complicates the utility maximization problem. However, they showed that under some continuity conditions, distributed price-based (congestion feedback based) congestion algorithms can still converge to the optimal solution. These studies mostly consider a given fixed number of flows, derive the best steady state bandwidth allocation to these flows. These results give new insights to co-existence of elastic and inelastic traffic in a network under distributed traffic controls. For more realistic evaluation, it is still important to consider a stochastic workload of flows.

There is a body of work studying bandwidth allocation with stochastic workloads as well [9], [18]. See [19] for a survey of this topic. The thrust of these studies often focus on how to create a stochastic model for which there is a product form solution (e.g. Whittle networks), or such product-form solution can be used as a performance bounds for the original model [20].

III. A New Approach for Evaluating Network Traffic Controls

Most of the notations for the rest of the paper are introduced in this section. Table I provide a summary.

A. Type of Flows and Utility Functions

In our new problem formulation, we consider two types of flows that use the network in different ways:

- *Holding time flows*. Such a flow has a holding time. After using the network for the holding time, the flow leaves.
- *File transfer flows*. Such a flow has a file size. It stays in the network for as long as necessary until the complete file has been transferred from source to destination.

Holding time flows are normally inelastic flows whereas file transfer flows are normally elastic flows. The degree of elasticity of each kind of flow, however, is determined by the respective utility function which will be introduced below. In the following discussion, we assign an elastic utility function to file transfer flows and refer to them as elastic flows. The elasticity of the utility function assigned to holding time flows is controlled by a parameter. For the study of different traffic controls in this paper, we assume the holding time flows have a rather inelastic utility function, hence we also refer to holding time flows as inelastic flows unless noted otherwise.

Since we are considering flows of finite size, it is necessary to account for how the utility of a flow depends on its size. We consider the two types of flows separately.

a) Holding time flows: We find the utility of a holding time flow by decomposition.

Consider a *holding time* flow f with holding time T, arriving at time t_0, and achieving rate $x_f(t)$ for $t_0 \leq t \leq t_0 + T$. The

TABLE I

SYMBOLS AND NOTATIONS

Symbol	Explanation
$f,\ f_j$	Flows or subflows
$x(t),\ x_E(t),\ x_I(t)$	Bandwidth allocated to a flow at time t, elastic, or inelastic
$a_E(n,m),\ a_I(n,m)$	Bandwidth allocation to an elastic/inelastic flow as a function of system state, (n,m)
$T,\ T_j$	Holding time of an inelastic flow, j
$S,\ S_j$	File size of an elastic flow, j
$\mathcal{U}_f$	Utility accrued by a flow f
$u_E(x)$	Utility accrued by a byte of elastic flow transferred at rate x. The unit is utility per byte.
$u_I(x)$	Instantaneous utility of an inelastic flow with allocated bandwidth x. The unit is utility per unit time.
$s_k()$	Sigmoidal utility function with parameter k, as in (1)
G	Utility throughput, also known as u-put
$G_E(t),\ G_I(t)$	Instantaneous utility throughput of elastic/inelastic flows
$\bar{G}_E,\ \bar{G}_I$	Average utility throughput of elastic/inelastic flows
$\bar{G}_E^C,\ \bar{G}_I^C$	Average utility throughput of elastic/inelastic flows under traffic control C
n,$n(t)$	Number of elastic flows, at time t
m,$m(t)$	Number of inelastic flows, at time t
$P[n,m]$	Probability that the network has n elastic flows and m inelastic flows
$P_C[n,m]$	Probability that the network has n elastic flows and m inelastic flows under control C
α	Desired rate of inelastic flows
ε	Minimal rate allowed for each elastic flow by inelastic flows when performing aggressive admission control. See section VII.
λ_E	Arrival rate of elastic flows
$1/\mu_E$	Mean file size of elastic flows, $E[S]=1/\mu_E$
λ_I	Arrival rate of inelastic flows
$1/\mu_I$	Mean holding time of inelastic flows, $E[T]=1/\mu_I$
ρ_E	Offered load of elastic flows, $\rho_E=\lambda_E/\mu_E$
ρ_I	Traffic intensity of inelastic flows, $\rho_I=\lambda_I/\mu_I$
$\alpha\rho_I$	Offered load of inelastic flows.
ρ	$\rho=\rho_E+\alpha\rho_I$, this is the total bandwidth demand by both elastic and inelastic flows

utility achieved by flow f is denoted $\mathcal{U}_f$. We can view f as the composition of subflows f_j ($j=1,\ldots,h$) of holding times T_j ($j=1,\ldots,h$, $\sum_j T_j=T$). We assume that the utility of f is comparable to the sum of the utility of the subflows. In the limit, as each subflow becomes sufficiently short, the utility of a subflow (at time t) becomes the utility of achieving an instantaneous rate $x(t)$ which we denote using $u_I(x(t))$. Note, the unit of $u_I()$ is utility per unit time (e.g. seconds). The assumption above becomes

Assumption 1 (Utility composition of holding time flows): The utility of a *holding time* flow is the sum of the utility of the instantaneous rates the flow achieves, namely

$$\mathcal{U}_f=\int_T u_I(x_f(t))dt.$$

This assumption obviously does not always hold. For example, consider a flow f that achieves rate $x_f(t)=1$ for $t\in[0,T/2]$ and $x_f(t)=0.5$ for $t\in(T/2,T]$. Alternatively, consider another flow f' with the same holding time T but achieves rate

$$x_{f'}(t)=\begin{cases}1 & \text{for } t\in[t_k,t_(k+1)],\ k=0,2,4,\ldots\\ 0.5 & \text{for } t\in(t_k,t_(k+1)],\ k=1,3,5,\ldots\end{cases}$$

where $t_k=t_0+\delta$ for some small $\delta\ll T$. For each scenario, the flow's utility comes out the same according to assumption 1. In reality, we expect the first flow, f to get a lower utility for most inelastic applications since it is easier to tolerate occasional loss than continuous loss. Nonetheless, this assumption makes the inelastic flows decomposable, and allows a reasonable way to account for the utility of flows of different sizes.

b) File transfer flows: Intuitively, the utility of a *file transfer* flow depends on two things:

- the time it takes the transfer to complete
- the file size

We expect it to be a non-increasing function of the former, and a non-decreasing function of the latter. The dependency on completion time is equivalent to (the reciprocal) dependency on average transfer rate. Consider a file transfer flow f of file size S_f, that achieved a certain transfer rate $x_f(t)$ between some start of finish time of the file transfer, and $\bar{x_f}$ denotes the average transfer rate of the flow f. The assumption for the utility of a file transfer flow can be stated as:

Assumption 2 (Utility composition of file transfer flows): The utility of a *file transfer* flow is the sum of utility for each byte which is a function of the average rate, namely

$$\mathcal{U}_f=S_f u_E(\bar{x_f}).$$

Again, we derive the utility of a flow by decomposition. The function $u_E(\bar{x})$ is a (usually concave) utility function on the *average rate*. Note, the unit of $u_E()$ is utility per unit of data (e.g. bytes). Therefore, the total utility of a flow can be thought of as being composed from the utility of each byte of the flow. But unlike in the holding time case, each component utility is based on the same average rate instead of the instantaneous rate.

Some alternative definitions of the utility composition rule of file transfer flows were also considered. One possibility is to let the utility of each data unit be dependent on the instantaneous rate of transferring that unit of data. This would give rise to the following paradox. Let f be a flow of size S_f and achieved rate $x_f(t)$, and f is decomposed as two flows g and h so that $S_g + S_h = S_f$, and $x_g(t)$ and $x_h(t)$ cover different intervals of the original horizon of $x_f(t)$. Since $u_E()$ is concave, by Jensen's inequality we know

$$\mathcal{U}(f) \leq \mathcal{U}(g) + \mathcal{U}(h)$$

So the utility of a flow increases as we decompose a flow (without any other change in the transfer of a flow), not a satisfying property. Another possibility is to make flow utility be a non-linear function (e.g. concave) of the file size. This would also cause the flow utility change as we split up a flow. In real life, a file transfer can be split up into multiple smaller file transfers, and there is a large class of traffic controls that may take advantage of such flow splitting (e.g. schedule them to be transferred sequentially or in parallel, possibly on different paths). The analysis of these cases is beyond the scope of the current paper.

B. Utility function

In the previous subsection, we established the utility of a flow in relation to functions $u_E()$ or $u_I()$. Actually, the elasticity of a flow are determined by the shape of $u_E()$ and $u_I()$.

If the utility function of instantaneous rate is non-decreasing and concave, then we consider the associated flow *elastic*. For an inelastic flow, its utility function (for instantaneous rate) is still non-decreasing, but the slope might not always be decreasing. A good example of an elastic utility function is a logarithmic function, and that for an inelastic utility function is a sigmoidal function:

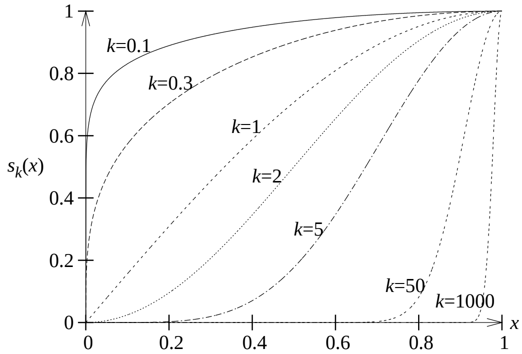


Fig. 1. A family of sigmoidal function $s_k(x)$ with different values of k

$$s_k(x) = \begin{cases} 0 & \text{if } x \leq \ell_1 \\ \sin^k\left(\frac{\pi}{2} \cdot \frac{x-\ell_1}{\ell_2-\ell_1}\right) & \text{if } \ell_1 < x \leq \ell_2 \\ 1 & \text{if } x > \ell_2. \end{cases} \quad (1)$$

A family of the sigmoidal functions $s_k()$ is plotted in Figure 1. Note, x is the given rate; ℓ_1 and ℓ_2 are thresholds so that when $x \leq \ell_1$, $s_k(x) = 0$ and when $x > \ell_2$, $s_k(x) = 1$ (in Figure 1, $\ell_1 = 0$ and $\ell_2 = 1$). The parameter k controls the shape of $s_k(x)$. The larger the value of k, the closer the function is to a step function. For k larger than 1, there is an inflexion point where the change in the gradient goes from positive to negative. When we use this sigmoidal function as the inelastic utility function, we assume $k > 1$.

In this paper, we set $u_E(x) = \log(1+x)$ and $u_I(x) = s_k(x)$ with $\ell_1 = 0$ and $\ell_2 = \alpha$, where α is the rate at which the inelastic flow is intended to transfer its data. We use this particular set of function to ensure the utility of a flow is bounded and always positive.

C. Workload

What we have introduced so far allows us to define a *workload* which is simply a set of elastic and inelastic flows that use the network. A workload can be stochastic or deterministic. A deterministic workload would be defined by a fixed set of flows with given arrival times, given file sizes (and holding times), and a utility function for each flow. A stochastic workload, on the other hand, would be defined by given arrival rates (e.g. assuming Poisson arrivals), and given distributions of file size and holding time (e.g.

assuming exponential distribution), and the corresponding utility functions for elastic and inelastic flows.

In this paper, we consider both deterministic workloads as well as stochastic workloads. When studying stochastic workloads, we assume an exponential distribution for file size and holding time to make the model reducible to a Markov chain and use standard techniques to compute the performance metrics.

Assumption 3 (Flow size distribution): Flow size (file size or holding time) is exponentially distributed.

More realistic flow size distributions (e.g. heavy-tailed) and packet level (rather than flow level) modeling is beyond the scope of the current paper, and is under further study.

D. Comparison of Traffic Controls

We start by restricting ourselves to the simplest network, with the following assumption, which applies to the rest of this paper:

Assumption 4 (Network): The network consists of a single link.

This is a reasonable assumption when the focus is to consider a set of flows sharing a single bottleneck. In a general network setting, network utility maximization tends to favor shorter flows (those traversing fewer links) than longer flows, as made clear by [16]. This is a different dimension in designing heterogeneous traffic controls that should be considered separately in the future.

Given a workload and network topology and static workload (fixed number of flows), it is possible to decompose and transform the network utility optimization problem to use distributed algorithms to obtain the optimal allocation. For example, this has been studied for elastic flows in [16], and for mixed elastic and inelastic flows in [10] and [11]. For stochastic workloads, however, the problem becomes more complicated. One established framework for studying optimal controls is through Markov decision processes. Using this approach, the derived optimal traffic control will likely be state-dependent, which is not easy to implement in a distributed fashion. In this paper, we focus on answering a simpler question: how do we compare two traffic controls? This is more tractable as well as quite useful for practical reasons. Once we establish the methodology, we can compare any two traffic controls which are considered reasonable to implement in practice. In particular, we will try to compare fair congestion control with admission control as two alternatives for managing inelastic traffic in a mixed traffic environment.

In addition to workload, we need to define (a) network state, (b) traffic control, and (c) utility throughput, as follows.

The *state* of such a simple network can be characterized by the flow population in the network and the per-flow states. At time t, the flow population of the network is given by the ordered pair $(n(t), m(t))$, where $n(t)$ is the number of elastic (file transfer) flows and $m(t)$ the number of inelastic (holding time) flows. The state of flow f_j is given by the current rate $x_{f_j}(t)$.

Given the network state, it is possible to define *traffic controls* — the object of our study. A traffic control can be thought of as a function that maps the network state (current state plus history) and other workload information to bandwidth allocation. We are mainly interested in *distributed* traffic controls, which means a flow's allocation is based only on its own flow state and some summary information about the network's state. For example, if the traffic control is TCP, the summary of network state would be a binary variable indicating whether the total load exceeds the nominal system capacity or not. For each traffic control we study, we will list out what network summary information must be gathered later when we describe those traffic controls.

In a real-life system, it takes time for the traffic control implemented in each flow to gather needed information for making adjustments, and it also takes time for the controls to take effect. The ensuing dynamics is extremely complicated. For our purposes, we assume idealized controls; in other words

Assumption 5 (Fluid model with zero feedback delay): Traffic controls can implement any rate within the link capacity and sense the network condition instantaneously, and the controls take effect immediately.

This assumption makes it possible to account for the different traffic controls as different transition rates in a Markov process model, and compute the state space probability distribution, $P[n,m]$.

E. Utility Throughput

Finally, we define *utility throughput* (u-put for short) as the rate the network is generating utility based on the different flows getting served by the network. This is a function of the traffic control adopted, and consequently the basis for comparing different traffic controls. Theoretically, each flow finishes accruing certain amount of utility from the network at its departure time. The u-put would then be the rate the network accrues utility based on this process.

Let $(f_1, f_2, \ldots f_K)$ be the set of flows that complete in time interval $[0,T]$. Then the u-put (denoted G) can be generally defined as

$$G = \frac{\sum_{j=1}^{K} \mathcal{U}_{f_j}}{T} \tag{2}$$

This definition can be used for both stochastic workloads as well as deterministic workloads.

In the stochastic workload case, flows are assumed to be of finite size (given by some distribution) and are assumed to arrive at a certain rate λ with some inter-arrival time distribution. The systems of interest are those that are stable, or in other words converge to a steady state. In the steady state, it is possible to derive an average flow utility for holding time and file transfer flows, $\mathcal{U}_I$ and $\mathcal{U}_E$ respectively.

$$G = G_I + G_E = \lambda_I \mathcal{U}_I + \lambda_E \mathcal{U}_E \tag{3}$$

Equation 3 can in turn be computed in terms of the components of flow utility according to Assumption 1 and 2.

At time t, let there be a set $M(t)$ ($|M(t)| = m(t)$) inelastic flows, and let $x_{f_j}(t)$ be the transmission rate of inelastic flow $f_j \in M(t)$. The network is thus accumulating inelastic utility at the rate:

$$G_I(t) = \sum_{j \in M(t)} u_I(x_{f_j}(t)).$$

Here, $u_I()$ is the inelastic utility function.

If the traffic control under consideration is such that all inelastic flows (in the network) transmit at the same rate $x_I(t)$, then we can write down the traffic controls as simple function $a_I()$ of the system state[1]:

$$x_I(t) = a_I(n(t), m(t)).$$

Then the instantaneous u-put equation becomes

$$G_I(t) = m(t) u_I(x_I(t)).$$

If the system is in steady state, then there is a steady state distribution for $m(t)$ (and $n(t)$ elastic flows), $P[n,m]$. So we can calculate $\bar{G_I} := E[G_I]$ as:

$$\bar{G_I} = \sum_n \sum_{m \neq 0} m u_I(a_I(n,m)) P[n,m] \tag{4}$$

The u-put of elastic flows, according to Assumption 2 can be computed based on the utility of each data unit (byte) transferred. During a time interval t to $t+\delta t$, let a total of $R(t)$ bytes (from $n(t)$ elastic flows) be transferred. The u-put of elastic flows at time t can be written as:

$$G_E(t) = \sum_{j=1}^{R(t)} u_E(\bar{x_j})$$

for δt approaching zero. The average rate $\bar{x_j}$ here denotes the average rate of the flow the j-th transferred byte belongs to, and $u_E()$ is the elastic utility function defined earlier. Unlike in the inelastic case, however, $\bar{x_j}$ is some property of a flow and we do not have a steady state distribution for that. To facilitate the computation, we make another assumption:

Assumption 6 (Same average rate): For a stochastic workload, the average rate seen by each *file transfer* flow is the same as the steady state file transfer flow rate averaged over all elastic flows, and it is independent of file size, namely

$$\bar{x_j} = \sum_m \sum_{n \neq 0} a_E(n,m) P[n,m] = \bar{x}.$$

Here, n is the number of elastic flows in the network and $a_E(n,m)$ denotes the bandwidth allocated to each elastic flows under certain traffic control in state (n,m). Intuitively, this assumption is more reasonable if flows are not too small, relative to the fluctuation of network utilization.

[1]Note, $a_I()$ in this form is not general enough to represent all traffic controls. For example, not all flows necessarily have the same rate. Even in our case, those flows rejected by admission control will get zero rates. But for our purpose of computing u-put, we are only interested in admitted flows and they are assumed to have the same rate as other flows in the same class.

Now we can readily write down the u-put for elastic flows

$$\begin{aligned}\bar{G}_E &= \lambda_E E[\mathcal{U}_f] \\ &= \lambda_E E[S_f] u_E(\bar{x_f}) \\ &= \rho_E u_E\big(\sum_m \sum_{n\neq 0} a_E(n,m) P[n,m]\big). \end{aligned} \tag{5}$$

The definition of u-put is quite general, and can be readily applied to the case of deterministic workloads, as long as the bandwidth allocation functions $a_I(n(t),m(t))$ and $a_E(n(t),m(t))$ are well-defined. We will see an example of this in section VI.

The reason we separately account for the elastic and inelastic utility is that it is very difficult to calibrate the utility functions to make the two kinds of utilities addable. This slightly complicates the comparison of different traffic controls, but is a more reasonable way to account for the situations.

Thus we have explained for each traffic control, C, how to compute the u-put as a pair of values, $(\bar{G}_E^C, \bar{G}_I^C)$, corresponding to the elastic and inelastic utility throughput. This affords us a way to compare traffic controls. If $\bar{G}_E^H > \bar{G}_E^K$ and $\bar{G}_I^H > \bar{G}_I^K$, then we can say traffic control H is better than K. Note, this would be just for the one workload that is analyzed. In order to do a complete comparison for traffic control H and K, it is necessary to evaluate a spectrum of representative workloads that are of interest.

Let $\bar{G}_E^C(w)$ and $\bar{G}_I^C(w)$ be the elastic and inelastic u-put for traffic control C under different workloads index by w. In order to conclude that a traffic control H is better than K, we require

$$\begin{aligned} \bar{G}_E^H(w) &\geq \bar{G}_E^K(w) \qquad && \forall w \in W \\ \bar{G}_I^H(w) &\geq \bar{G}_I^K(w) \qquad && \forall w \in W \end{aligned}$$

where W is some set of *benchmark* workloads.

This completes the description of our methodology for comparing two traffic controls. In essence, this methodology (when considering stochastic workloads) is rather similar to the *Markov reward processes* [21]. In the terminology of [21], comparing u-put is analogous to comparing the system *gain* under two different *policies* to find out which is the better *policy*. In our case, we do need to deal with an infinite (or potentially large) population model, and multiple kinds of *rewards*[2].

In the following sections, we apply this methodology to a comparative study of some specific traffic controls.

IV. Case Study: Traffic Controls for Homogeneous Flows

Before analyzing the case of two classes of traffic, we first compare different traffic controls for holding time flows when it is the only kind of traffic (when elastic traffic is absent). We vary the utility function to see how it affects the selection of the best traffic control.

Let the holding time flows arrive at a rate of λ_I and let each flow have an average holding time of $1/\mu_I$. Let the utility function be

$$s_k(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \sin^k\left(\frac{\pi}{2}\cdot\frac{x}{\alpha}\right) & \text{if } 0 < x \leq \alpha \\ 1 & \text{if } x > \alpha \end{cases} \tag{6}$$

Which is (1) with $\ell_1 = 0$ and $\ell_2 = \alpha$. The utility varies between zero and one. As illustrated in Figure 2, the utility is zero for rate $x \leq 0$, and increases with x until it reaches 1 when $x = \alpha$ and stays the same for larger values of x. The parameter k controls the shape of the curve. In one extreme, when $k \leq 1$, the utility function is concave. In this case, a holding-time flow behaves a little like an elastic flow, in the sense that it can still benefit from a rate significantly smaller than its desired rate α. In the other extreme, for larger values of k the utility function approaches a step function. This corresponds to the classic utility function of an inelastic flow with α as its desired rate. These type of flows cannot benefit from a rate sufficiently smaller than its desired rate.

We consider two types of traffic controls: congestion control (with fair allocation) and admission control. Under Assumptions 3, 4 and 5 both can be analyzed by standard queueing models.

The fair congestion control can be modeled by a M/M/∞ queue. In a state with m jobs, since all flows are of the

[2]This is why it is not straightforward to consider using policy iteration method to optimize our traffic controls. But this is something we are working on.

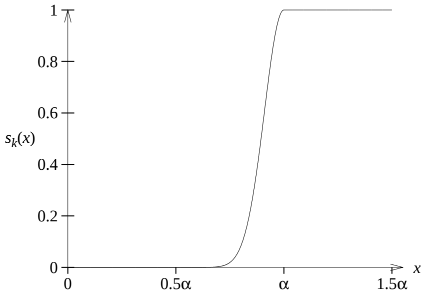


Fig. 2. A particular utility function for inelastic flows, with $k = 50$.

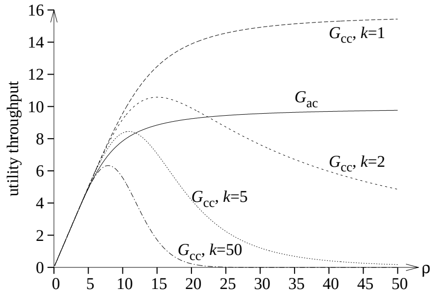


Fig. 3. Utility throughput in homogeneous case, with $\alpha = 0.1$

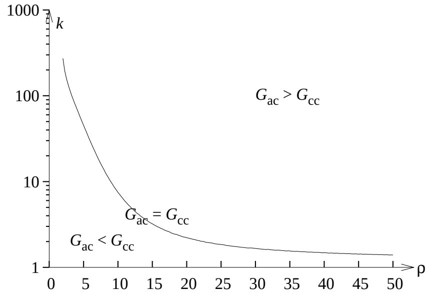


Fig. 4. Crossover value of k at different offered load ρ, with $\alpha = 0.1$

holding time type, the departure rate is $m\mu_I$. The steady state probability distribution of number of flows is given by

$$P_{\mathrm{cc}}[m] = \frac{\rho_I^n e^{-\rho_I}}{m!}.$$

The u-put in state m, however, is given by $ms_k(\frac{1}{m})$. According to equation (4), the steady state average u-put is

$$G_{\mathrm{cc}} = \sum_{m=0}^{\infty} ms_k(\tfrac{1}{m})P_{\mathrm{cc}}[m].$$

The admission control case can be modeled by a M/M/h/h loss system, where $h = \lfloor 1/\alpha \rfloor$ denotes the maximum number of flows the network can accommodate giving them their desired rate α. The steady state probability distribution of the number of flows is given by

$$P_{\mathrm{ac}}[m] = \frac{\rho_I^m}{m!}\Big(\sum_{j=0}^{h}\frac{\rho_I^j}{j!}\Big)^{-1}.$$

Since the admission control scheme does not over-admit, all admitted flows get utility of 1, and all blocked flows get utility of zero. This means the steady state u-put is simply

$$\begin{aligned} G_{\mathrm{ac}} &= \sum_{m=0}^{h} ms_k(\tfrac{1}{m})P_{\mathrm{ac}}[m] \\ &= \sum_{m=0}^{h} mP_{\mathrm{ac}}[m] \\ &= \rho(1 - P_{\mathrm{ac}}[h]) \end{aligned}$$

where $P_{\mathrm{ac}}[h]$ is the *blocking probability*.

Since we have closed-form expressions for the u-puts of both systems we want to compare, we can plot them against the offered load, ρ, as in Figure 3.

The figure shows that the u-put for the admission controlled system, G_{ac}, increases monotonically as ρ increases and reaches a saturated level. This is expected behavior for the Erlang-B model. When fair congestion control is applied, the situation becomes interesting. The system u-put depends on the utility function, parametrized by k. When $k = 1$, the utility is still elastic (concave). The fair congestion control consistently gives higher u-put for all values of ρ. When $k = 2$, i.e. when the utility becomes mildly inelastic, there is a *crossover point* in the u-put curves. In other words, when the offered load is sufficiently high, there is a point when it is better to start applying admission control (for $k = 2$, this point is roughly $\rho = 20$). As the utility function becomes more inelastic, the crossover point appears earlier and earlier. It is clear that the admission control approach becomes asymptotically better than fair congestion control as k becomes large.

Figure 4 shows the crossover value of k versus each offered load ρ. The curve serves as a dividing line: for all the values of (ρ, k) above the curve, it is better to use admission control, and vice versa. Actually, for many points below the curve (e.g. $\rho < 5$), admission control is also reasonable since the u-put curve for admission control is approximately the same as the fair congestion control curve until the crossover point, as shown in Figure 3.

V. Case Study: Traffic Controls for Heterogeneous Flows

The case of considerable practical interest is when both elastic and inelastic flows sharing the network. We let the traffic control for elastic flows be some form of fair congestion control, as in the Internet. For inelastic flows, there are three options we want to compare: (1) no control (NC), (2) fair congestion control (CC), (3) admission control (AC). We discuss how bandwidth is allocated in each case.

- *No control (NC)*. Perform neither congestion control nor admission control. This is meant to model the behavior of UDP flows. In this case, when $1/\alpha$ or more inelastic flows are in the network, the network capacity is totally consumed (equally shared) by the inelastic flows, and the elastic flows get no service. Otherwise, if there are $m < 1/\alpha$ inelastic flows, each inelastic flow gets α and the elastic flows share the remaining $1 - m\alpha$.
- *Fair congestion control (CC)*. Perform TCP-friendly congestion control. Here, we model it as the same fair congestion control as adopted for elastic flows, with a slight difference. When the fair share is smaller than α, then the fair share is used, but when the fair share is greater than α, the inelastic flows would still consume α. This is a different treatment than the model in [18] where rate allocated to inelastic flow is always the same as elastic flows.
- *Admission control (AC)*. Perform admission control but no congestion control once admitted. In general, the admission control function can be quite sophisticated. For example, it may depend on the playback rate α and the holding time μ_i. In our analysis, we assume the playback rate for all inelastic flows are the same, and consider a rather simple form of admission control that is oblivious to the holding time. Assume the network already has n elastic flows and m inelastic flows, a newly arriving (inelastic) flow is admitted if

$$n\varepsilon + (m+1)\alpha \leq 1$$

  The parameter ε represents some minimal rate the admission control scheme tries to leave alone for each elastic traffic. It is a knob to tune how aggressive to make the admission control try. We consider two extreme cases:

  – *Aggressive admission control (AC1)*. In this case, $\varepsilon \ll \alpha$. In other words, the arriving flow is admitted as long as it is possible to allocate to it the desired rate of α, even if this means all elastic flows have to run at their minimum rate of ε.

  – *Fair admission control (AC2)*. In this case, $\varepsilon = \alpha$. Precisely, the arriving inelastic flow is admitted if

$$\frac{1 - n\alpha}{m+1} \geq \alpha.$$

  In other words, the newly arriving flow is admitted only if its desired rate (α) is no greater than the prevailing fair share for each elastic flow. The above condition actually reduces to

$$(n+m+1)\alpha \leq 1.$$

  Because this form of admission control tries to be friendly to elastic (hence TCP) flows, we refer to it as *TCP-friendly admission control*.

To summarize, we can write down the traffic control functions for NC, CC, AC1 and AC2 in terms of what they allocate to elastic flows ($a_E(n,m)$) and what they allocate to inelastic flows ($a_I(n,m)$) when there are n and m flows of each kind respectively:

$$\begin{aligned}
\text{NC:} \quad a_E(n,m) &= \begin{cases} \frac{1-m\alpha}{n} & \text{if } m\alpha \leq 1 \\ 0 & \text{if } m\alpha > 1 \end{cases} \\
\text{CC:} \quad a_E(n,m) &= \begin{cases} \frac{1-m\alpha}{n} & \text{if } (n+m)\alpha \leq 1 \\ \frac{1}{n+m} & \text{if } (n+m)\alpha > 1 \end{cases} \\
\text{AC1, AC2:} \quad a_E(n,m) &= \begin{cases} \frac{1-m\alpha}{n} & \text{if } m\alpha \leq 1 \end{cases}
\end{aligned} \tag{7}$$

$$\begin{aligned}
\text{NC:} \quad a_I(n,m) &= \begin{cases} \alpha & \text{if } m\alpha \leq 1 \\ \frac{1}{m} & \text{if } m\alpha > 1 \end{cases} \\
\text{CC:} \quad a_I(n,m) &= \begin{cases} \alpha & \text{if } (n+m)\alpha \leq 1 \\ \frac{1}{n+m} & \text{if } (n+m)\alpha > 1 \end{cases} \\
\text{AC1, AC2:} \quad a_I(n,m) &= \begin{cases} \alpha & \text{if } m\alpha \leq 1 \end{cases}
\end{aligned} \tag{8}$$

For AC1 and AC2, $a_E(n,m)$ and $a_I(n,m)$ are defined assuming the m inelastic flows have been admitted. Therefore, $a_E(n,m)$ and $a_I(n,m)$ for the $m > \lfloor \frac{1}{\alpha} \rfloor$ case are not specified.

For inelastic flows, the utility function $u_I()$ is the same sigmoidal function introduced before in Equation 6. For elastic flows, the utility function $u_E()$ is the frequently adopted log function, with normalization:

$$u_E(x) = \log(1+x). \tag{9}$$

Now we are ready to compare the four cases: NC, CC, AC1 and AC2 as alternative traffic controls for inelastic flows in co-existence with elastic flows. We first consider a deterministic workload case, and then the more general stochastic workload case in the next two sections.

VI. CASE STUDY: STATIC WORKLOAD

We first study a special workload when all n elastic and m inelastic flows arrive at the same time[3] and stay on indefinitely. If the flow were an elastic flow, it has an infinite file size; if it were an inelastic flow, it has an infinite holding time. We referred to these as *infinite flows*. This is a special case of deterministic workload, and it is the workload commonly used in the study of congestion control and bandwidth allocation for homogeneous networks with elastic traffic flows only. We also refer to this as the *static* workload. An important reason for taking a special look at this workload is that it is possible to derive closed-form solutions, so that the results will lead to more insights and certain degree of validation of the general result. Furthermore, since all the flows (of each type) are the same, these results are independent of the assumptions on how the utility of flows of different lengths are computed.

Given the above definition of workload with infinite flows, the network state (n,m) remains unchanged and in a steady state by definition. The utility throughput for elastic and inelastic flows are also constant, respectively:

$$\begin{aligned}\bar{G}_E &= \rho_E u_E \big(\sum_m \sum_{n\neq 0} a_E(n,m)P[n,m]\big)\\ &= \rho_E u_E(a_E(n,m))\\ &= n a_E(n,m) u_E(a_E(n,m)); \qquad (10)\\ \bar{G}_I &= \sum_n \sum_{m\neq 0} m u_I(a_I(n,m))P[n,m]\\ &= m u_I(a_I(n,m)).\end{aligned}$$

according to (4) and (5). The four traffic controls can be compared by substituting $a_I(n,m)$ and $a_E(n,m)$ for these controls into the above u-put equations.

Given the static workload, it is necessary to assume an order of arrival for the n and m flows so that $a_E(n,m)$ and $a_I(n,m)$ can be defined for AC1 and AC2. For the following analysis, we assume all the n elastic flows arrived before the m inelastic flows. This assumption creates a certain bias to the results, which will be discussed at the end of Section VI.B.

[3]The order of arrival does matter for the evaluation of admission controls.

TABLE II

COMPARING $\bar{G}_E$ OF DIFFERENT TRAFFIC CONTROLS

Range of m	Comparison	Region
$0 \le m \le \frac{1-n\alpha}{\alpha}$	$\bar{G}_E^{AC2} = \bar{G}_E^{CC} = \bar{G}_E^{AC1} = \bar{G}_E^{NC}$	(i)
$\frac{1-n\alpha}{\alpha} < m < \frac{1-n\varepsilon}{\alpha}$	$\bar{G}_E^{AC2} > \bar{G}_E^{CC} > \bar{G}_E^{AC1} = \bar{G}_E^{NC}$	(ii)
$\frac{1-n\varepsilon}{\alpha} < m < \frac{1-n\varepsilon}{\varepsilon}$	$\bar{G}_E^{AC2} > \bar{G}_E^{CC} > \bar{G}_E^{AC1} > \bar{G}_E^{NC}$	(iii)
$\frac{1-n\varepsilon}{\varepsilon} \le m$	$\bar{G}_E^{AC2} > \bar{G}_E^{AC1} > \bar{G}_E^{CC} > \bar{G}_E^{NC}$	(iv)

A. Comparing Elastic Utility Throughput

Let $\bar{G}_E^{NC}, \bar{G}_E^{CC}, \bar{G}_E^{AC1}$ and $\bar{G}_E^{AC2}$ be the elastic u-put for NC, CC, AC1 and AC2 respectively. After some straightforward algebraic manipulations, we have

Proposition 1: The relationship between the elastic u-put for NC, CC, AC1 and AC2 is as given in Table II.

The proof is given in the appendix. NC favors inelastic flows, hence always produces the least elastic u-put, as expected. Interestingly, AC2 always produces the most (or equal to the most) elastic u-put. Finally, when the network is *overprovisioned* (i.e. few flows are competing for the network), all four traffic controls generate the same elastic u-put. Figure 5 shows the four regions of the (n,m) space corresponding to the four different rankings in Proposition 1.

For any fixed n, the value $\bar{G}_e$ is monotonically decreasing with m, as increasing inelastic flows grab bandwidth away from inelastic flows. Figure 8 shows $\bar{G}_e$ against m for a fixed n. The merit of this class of admission control algorithms is that after certain point, the inelastic flows will not continue to take away bandwidth because it would be detrimental to themselves.

B. Comparing Inelastic Utility Throughput

Similarly, let $\bar{G}_I^{NC}, \bar{G}_I^{CC}, \bar{G}_I^{AC1}$ and $\bar{G}_I^{AC2}$ denote the inelastic u-put for NC, CC, AC1 and AC2 respectively.

The comparative situation for inelastic u-put is more complicated, as illustrated in Table III for a specific set of parameters. The ten different *regions* in Table III refer to those depicted in Figure 9.

As in the previous case, region (i) corresponds to the underloaded case where all traffic controls are the same. Region (ii) is a small region where CC is better than AC2. This is the case when n and m are such that the offered load

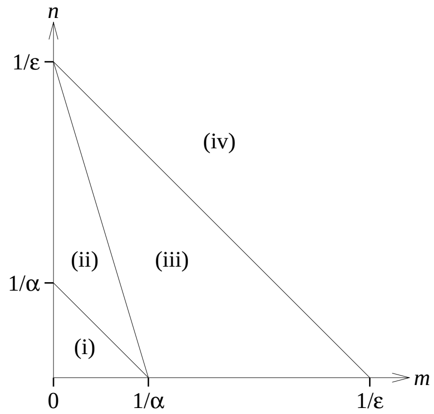


Fig. 5. Regions of different comparative outcome for elastic u-put

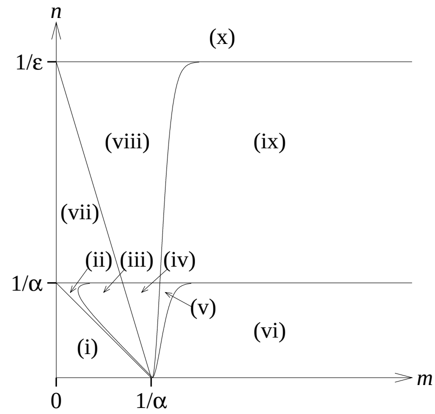


Fig. 6. Regions of different comparative outcome for inelastic u-put

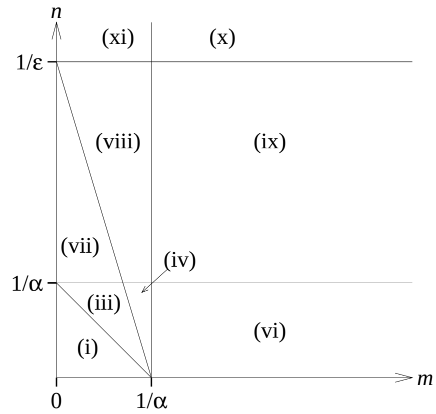


Fig. 7. Regions of different comparative outcome for inelastic u-put, when $k = \infty$

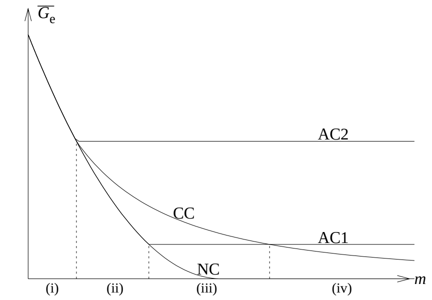


Fig. 8. Plot of $\bar{G}_e$

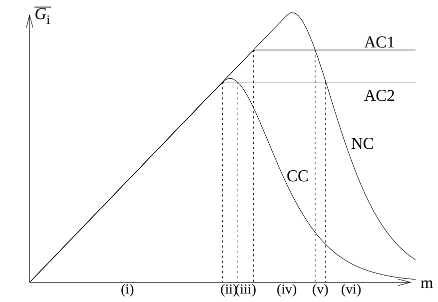


Fig. 9. Regions of different comparative outcome for inelastic u-put

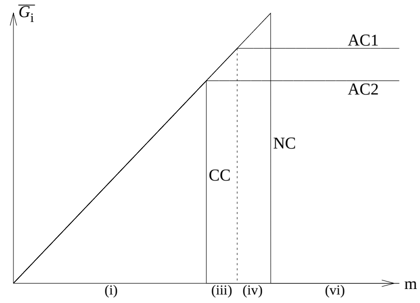


Fig. 10. Regions of different comparative outcome for inelastic u-put, when $k = \infty$

TABLE III

COMPARING $\bar{G}_I$ FOR DIFFERENT TRAFFIC CONTROLS

Comparison	Region
$G_i^{NC} = G_i^{AC1} = G_i^{CC} = G_i^{AC2}$	(i)
$G_i^{NC} = G_i^{AC1} > G_i^{CC} > G_i^{AC2}$	(ii)
$G_i^{NC} = G_i^{AC1} > G_i^{AC2} > G_i^{CC}$	(iii)
$G_i^{NC} > G_i^{AC1} > G_i^{AC2} > G_i^{CC}$	(iv)
$G_i^{AC1} > G_i^{NC} > G_i^{AC2} > G_i^{CC}$	(v)
$G_i^{AC1} > G_i^{AC2} > G_i^{NC} > G_i^{CC}$	(vi)
$G_i^{NC} = G_i^{AC1} > G_i^{CC} > G_i^{AC2} = 0$	(vii)
$G_i^{NC} > G_i^{AC1} > G_i^{CC} > G_i^{AC2} = 0$	(viii)
$G_i^{AC1} > G_i^{NC} > G_i^{CC} > G_i^{AC2} = 0$	(ix)
$G_i^{NC} > G_i^{CC} > G_i^{AC1} = G_i^{AC2} = 0$	(x)

barely exceeds the capacity to offer all flows what is desired by inelastic flows (α). By admitting one or two additional inelastic flows than AC2, if the inelastic utility function is not steep enough (k is not too large, hence there is still some elasticity), then CC would be slightly better than AC2. For regions (iii) to (iv), AC1 and AC2 are more effective (than CC) by blocking some inelastic flows in order to gain higher system utility for the rest of the flows. Finally, regions (vii) to (ix) correspond to the cases when n is large enough so that AC2 would not admit any inelastic flows, and region (x) corresponds to the case when n is so large that neither AC controls would admit any inelastic flows. In these severely overloaded cases, even though AC is out-performed by CC and NC, all flows are enjoying very low utilities (the allocated bandwidth is less than $1/\alpha$ or $1/\varepsilon$).

Figure 9 plots the inelastic u-put against m for a given value of n. The different regions of comparative outcomes are indicated.

The above example clearly shows the rankings of the traffic controls, although the exact characterization of these regions cannot be expressed in closed-form. We do have the following asymptotic result when k tends to infinity (i.e. the inelastic utility function approaches a step function.

Proposition 2: The relationship between the inelastic u-put for NC, CC, AC1 and AC2 asymptotically converges to that

TABLE IV

COMPARING $\bar{G}_I$ OF DIFFERENT TRAFFIC CONTROLS AT $k=\infty$

Range of m and n		Comparison	Reg.
$0 \le m \le \frac{1-n\alpha}{\alpha}$	$0 \le n < \frac{1}{\alpha}$	$G_i^{NC} = G_i^{AC1} = G_i^{CC} = G_i^{AC2}$	(i)
$\frac{1-n\alpha}{\alpha} < m \le \frac{1-n\varepsilon}{\alpha}$	$0 \le n < \frac{1}{\alpha}$	$G_i^{NC} = G_i^{AC1} > G_i^{AC2} > G_i^{CC} = 0$	(iii)
$\frac{1-n\varepsilon}{\alpha} < m \le \frac{1}{\alpha}$	$0 \le n < \frac{1}{\alpha}$	$G_i^{NC} > G_i^{AC1} > G_i^{AC2} > G_i^{CC} = 0$	(iv)
$\frac{1}{\alpha} < m$	$0 \le n < \frac{1}{\alpha}$	$G_i^{AC1} > G_i^{AC2} > G_i^{NC} = G_i^{CC} = 0$	(vi)
$\frac{1-n\alpha}{\alpha} < m \le \frac{1-n\varepsilon}{\alpha}$	$\frac{1}{\alpha} \le n < \frac{1}{\varepsilon}$	$G_i^{NC} = G_i^{AC1} > G_i^{CC} = G_i^{AC2} = 0$	(vii)
$\frac{1-n\varepsilon}{\alpha} < m \le \frac{1}{\alpha}$	$\frac{1}{\alpha} \le n < \frac{1}{\varepsilon}$	$G_i^{NC} > G_i^{AC1} > G_i^{CC} = G_i^{AC2} = 0$	(viii)
$\frac{1}{\alpha} < m$	$\frac{1}{\alpha} \le n < \frac{1}{\varepsilon}$	$G_i^{AC1} > G_i^{NC} = G_i^{CC} = G_i^{AC2} = 0$	(ix)
$\frac{1}{\alpha} < m$	$\frac{1}{\varepsilon} \le n$	$G_i^{NC} = G_i^{CC} = G_i^{AC1} = G_i^{AC2} = 0$	(x)
$\frac{1-n\varepsilon}{\alpha} < m \le \frac{1}{\alpha}$	$\frac{1}{\varepsilon} \le n$	$G_i^{NC} > G_i^{CC} = G_i^{AC1} = G_i^{AC2} = 0$	(xi)

given in Table IV, as the inelastic utility function converges to a step function ($k \to \infty$ if $u_I()$ is a sigmoidal function).

The proof is given in the appendix. Figure 7 depicts the regions in the (n,m) space that produce the rankings in Proposition 2. In this case, the boundaries between the regions are clear-cut. Region (ii) and (v) have disappeared because the CC and NC curves in Figure 10 drops sharply to zero as soon as one extra inelastic flow is admitted above the thresholds given by the AC1 and AC2 lines. For this asymptotic case, it is possible to conclude that AC2 is always better than CC, in the sense AC2 generates more u-put for both elastic and inelastic traffic.

In this comparison (for static workload), we have assumed that all n elastic flows arrived before any of the m inelastic flows. This assumption helps the admission control schemes (AC1 and AC2) to generate more elastic u-put and less inelastic u-put, when compared to the case where flows arrive randomly. This bias does not materially affect the overall results, since it does not exclusively favor one kind of control over another kind.

VII. CASE STUDY: STOCHASTIC WORKLOAD

Now we are going to study the case when the flows are finite. We have both elastic (file transfer) flows as well as inelastic (holding time) flows in the network. The inelastic flows are with arrival rate λ_I, average holding time $1/\mu_I$, a desired transmission rate of α, and a sigmoidal utility function with a sufficiently large k. The elastic flows have arrival rate λ_E, average file size $1/\mu_E$, and a concave utility function as (9).

TABLE V

THE WORKLOAD SETS FOR WHICH WE COMPARED THE TRAFFIC CONTROLS

	Workload parameter			Figures
1	$\rho_E/\alpha\rho_I = 1$	$0 \le \rho \le 2$	$\alpha = 0.05$	Figure 11(a)
2	$\rho_E/\alpha\rho_I = \frac{1}{9}$	$0 \le \rho \le 2$	$\alpha = 0.05$	Figure 11(c)
3	$\rho_E/\alpha\rho_I = 9$	$0 \le \rho \le 2$	$\alpha = 0.05$	Figure 11(b)
4	$\rho_E/\alpha\rho_I = 1$	$\rho = 0.50$	$0.01 \le \alpha \le 0.5$	Figure 13(a)
5	$\rho_E/\alpha\rho_I = 1$	$\rho = 0.95$	$0.01 \le \alpha \le 0.5$	Figure 13(b)
6	$\rho_E/\alpha\rho_I = 1$	$\rho = 1.40$	$0.01 \le \alpha \le 0.5$	Figure 13(c)
7	$\rho_E/\alpha\rho_I = \frac{1}{9}$	$\rho = 0.95$	$0.01 \le \alpha \le 0.5$	Figure 14(a)
8	$\rho_E/\alpha\rho_I = 9$	$\rho = 0.95$	$0.01 \le \alpha \le 0.5$	Figure 14(c)

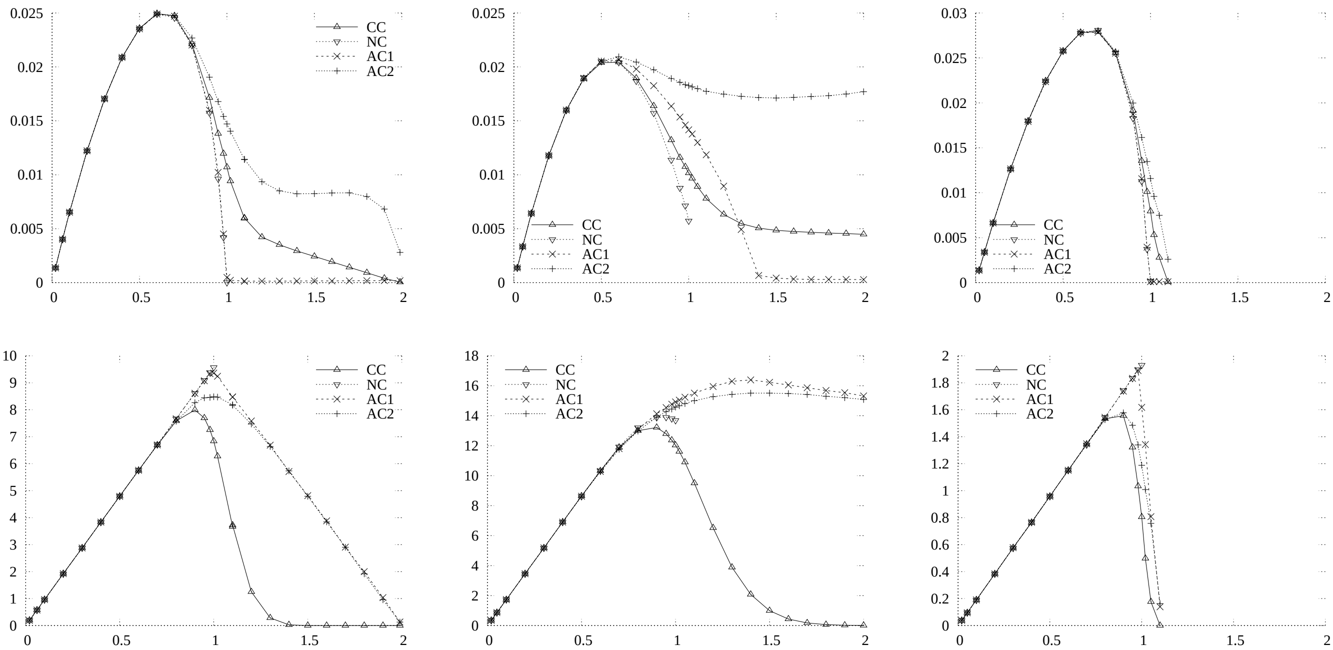
To compute the u-put for each traffic control, we plug $P_C[n,m]$ into (4) and (5), where $P_C[n,m]$ is defined in Table I. The u-put for the four traffic controls are compared for the workload sets shown in Table V. For all these cases, we set $k = 50$ in the sigmoidal function (a reasonably stiff inelastic utility function); and $\varepsilon = 0.001$ for AC1. The following notations for the workloads are adopted

$$\rho_E = \lambda_E/\mu_E$$
$$\rho_I = \lambda_I/\mu_I$$
$$\rho = \rho_E + \alpha\rho_I.$$

The ratio $\rho_E/\alpha\rho_I$ indicates the relative intensity of the elastic offered load to that of the inelastic offered load. The quantity ρ is the total offered load. Finally, recall α is the desired bandwidth for each inelastic flow (as a fraction of the bottleneck link bandwidth). In workload sets 1-3, we fixed the ratio of traffic intensities and desired rate, but varied the total offered load. In workload sets 4-8, we fixed the traffic ratios and total offered load, but varied the desired rate. These sets of workloads covered a wide spectrum of network loading scenarios.

A. Comparison for Different Offered Load

Figure 11(a) shows the u-put of elastic and inelastic flows, with equal traffic intensities ρ_E and $\alpha\rho_I$, as we vary total offered load ρ. When ρ is small, we can hardly differentiate the u-put for different controls. For each control, the u-put increases as ρ increases. This means it does not matter which control algorithm we use for inelastic flows when we are not in the congested regime.



(a) $\rho_E : \alpha\rho_I = 1:1$ (b) $\rho_E : \alpha\rho_I = 1:9$ (c) $\rho_E : \alpha\rho_I = 9:1$

Fig. 11. Elastic (top) and inelastic (bottom) utilities vs. ρ

When the total offered load approaches the network capacity, the service starts to degrade. The u-put is in a downward trend for all controls when $\rho > 1$. However, the result is more graceful for some controls than others. Under aggressive admission control (AC1) or no control (NC), the elastic u-put drops quickly to zero when ρ increases to 1. When $\rho > 1$, the NC case becomes unstable (elastic population blows up) whereas the u-put in the AC1 case will stay at almost zero. The fair congestion control (CC) and fair admission control (AC2) perform more gracefully as total offered load ρ increases to and beyond network capacity. AC2 always yields higher elastic u-put compared to the other three controls.

The overall trend for inelastic u-put is similar. When the total offered load is less than network capacity, the u-put gradually increases at an almost equal rate for all controls. As ρ approaches network capacity, the inelastic u-put reaches its peak for all controls. After that, the NC case first becomes unstable (due to total population blowing up). For the two admission control cases, as AC1 more aggressively admits inelastic flows it performs the best; but AC2 delivers almost the same performance. Both admission controls perform significantly better than fair congestion control (CC) when $\rho > 1$. The u-put of CC is almost zero in this congested regime. Intuitively, this is exactly caused by the *all-or-nothing* nature of the utility function for inelastic flows.

In summary, we make two observations from the analysis of this workload:

- Both elastic and inelastic u-put depend on offered load irrespective of the traffic control. When offered load is low, the u-put is low; as offered load approach the network capacity, the u-put peaks; as offered load exceed network capacity, the u-put declines, gracefully for some controls but much more precipitously for other controls.
- The admission control AC2 consistently out-performs fair congestion control (CC) in terms of both elastic and inelastic u-put. AC1 performs better than CC for inelastic u-put but worse for elastic u-put. The No Control (NC) case becomes unstable prematurely (when ρ approaches capacity). So no definitive statements can be made for AC1 or NC relative to CC.

B. Comparison for Different Traffic Mixes

Figures 11(b) and 11(c) are similar to Figure 11(a) but are for different traffic mixes. They show the utility throughput for elastic and inelastic flows with traffic ratio $\rho_E : \alpha\rho_I$ equal to 1:9 and 9:1 respectively, instead of 1:1 in Figure 11(a).

From these figures, we can see that the observations made at the end of the last section still hold. In particular, AC2 still consistently out-performs CC for these workloads. There are some quantitative differences as discussed below.

In Figure 11(b), when the majority of the flows are inelastic, the choice of traffic control has a more significant impact, to both inelastic as well as elastic u-puts. When admission control (AC1 or AC2) is applied, inelastic u-put degrades gracefully from the peak in comparison to the fair congestion control case (CC). Intuitively, the reason is that the inelastic utility function prefers fewer flows operating at desired rate rather than more flows operating at reduced rate. For elastic u-put, fair admission control (AC2) is able to significantly out-perform not only CC, but also AC1[4].

In Figure 11(c), when the majority of the flows are elastic, the impact of the control used for inelastic flows is small, as expected. It is worth noting that as ρ_E approaches the network capacity (in this case, it means ρ approaches $1/0.9 \approx 1.11$) the elastic u-put quickly drops to zero no matter which control is used. This raises an interesting question: What is the set of offered load for which the system is stable (the population does not increase to infinity in steady state) for each traffic control we are studying? In the next subsection, we take a detour to discuss this question, before continuing with comparing traffic controls under other workloads.

C. Stability

A M/M/∞ queue is stable as long as the offered load is less than the service capacity; in other words, ρ is less than 1. In Section IV, however, we plotted the u-put of a network with homogeneous traffic for ρ greater than 1. Similarly, in this section we consider workloads with ρ greater than 1 (see Table V). Why is the system still stable when ρ is greater than 1? What is the stability criterion for our models?

It turns out this depends on how we model the flows. By modeling (inelastic) flows as *holding time* flows, it means under certain traffic controls the offered load is *compressible* [18]. For example, if the inelastic flows are subject to fair congestion control, then they may be allocated less instantaneous bandwidth than they desire. However, this does not affect the departure times of these flows. Therefore, under fair congestion control it is possible to receive arbitrarily heavy offered load of holding time flows without affecting the population size, as indicated by the analysis in Section IV.

Similarly if holding time flows are subject to admission control without any congestion control, the original offered load may be *compressed* since whatever the network cannot handle would have been blocked. But in this case, the compression affects those blocked flows only, rather than all flows. So in this situation, the network is also stable for arbitrary offered load as shown in Section IV.

The offered load presented by the file transfer flows, however, are not compressible. If less bandwidth is allocated to these flows, the file transfers will progress more slowly, and more load tend to be accumulated for later. In this case, it is well-established that if $\rho \geq 1$ the system is not stable.

We state without proof that in a network of mixed flows (both file transfer as well as holding time type of flows), if the traffic control adopts either CC or admission control (in the form of AC2 or AC1 with $\varepsilon \geq 0$, then it is sufficient to require $\rho_E < 1$ to ensure the network with mixed traffic is stable. Intuitively, this can be rationalized in the following sense. If $\rho_E < 1$, then consider the network's capacity be reduced by ρ_E, and there is still finite capacity left. Since the traffic controls for holding time flows can maintain stability for arbitrary offered load, as the file transfer workload accumulates eventually the holding time flows will be *compressed* sufficiently to accommodate the file transfer load [18]. Note, however, the NC case is an exception. Since there is no control whatsoever on the inelastic flows, it is necessary to have $\rho_E + \alpha\rho_I < 1$ to ensure stability.

In reality, our results demonstrate this is true. In Figure 11(a), we see the u-put is positive when the inelastic flows are under either fair congestion control or admission control. In Figure 11(b), although we only plot u-put for $0 \leq \rho \leq 2$ (hence $0 \leq \rho_E \leq 0.2$) for easy comparison, we did verify that the network is still stable for other values of $\rho_E < 1$. Finally,

[4]This is a very interesting property of the model, namely AC1 and AC2 give similar aggregate performance for inelastic flows; but AC2 is able to give significantly better aggregate performance for elastic flows. A detailed study of the reason for this behavior is given in a separate paper [22].

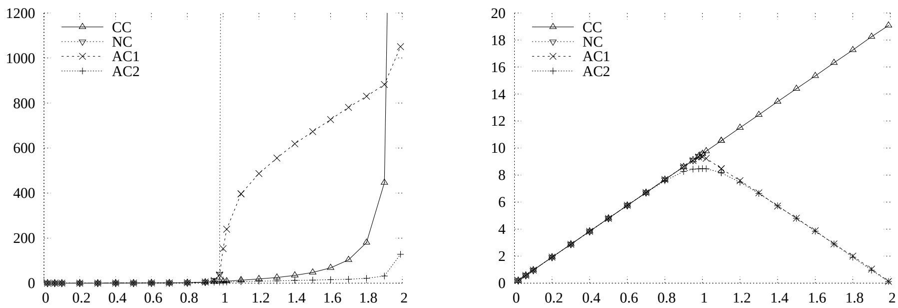


Fig. 12. Mean population of elastic (left) and inelastic (right) flows vs. ρ, with $\rho_E : \alpha\rho_I = 1 : 1$

for Figure 11(c), it is evident the network is only stable up to roughly $\rho < 1.1$ (hence $\rho_E < 1$).

Furthermore, we can plot the mean population size against total offered load ρ, as shown in Figure 12. Under light load, the mean population size is roughly the same for different traffic controls. At $\rho = 1$, the NC case first blows up, namely the elastic population goes to infinity. As ρ approaches 2 (i.e. ρ_E approaches 1), the elastic population under the other three controls also increase abruptly. For $1 \leq \rho \leq 2$, it is interesting to note that the elastic population under AC1 is significantly more than CC and AC2; and AC2 is the only control that can keep the elastic population low in this stable but congested regime.

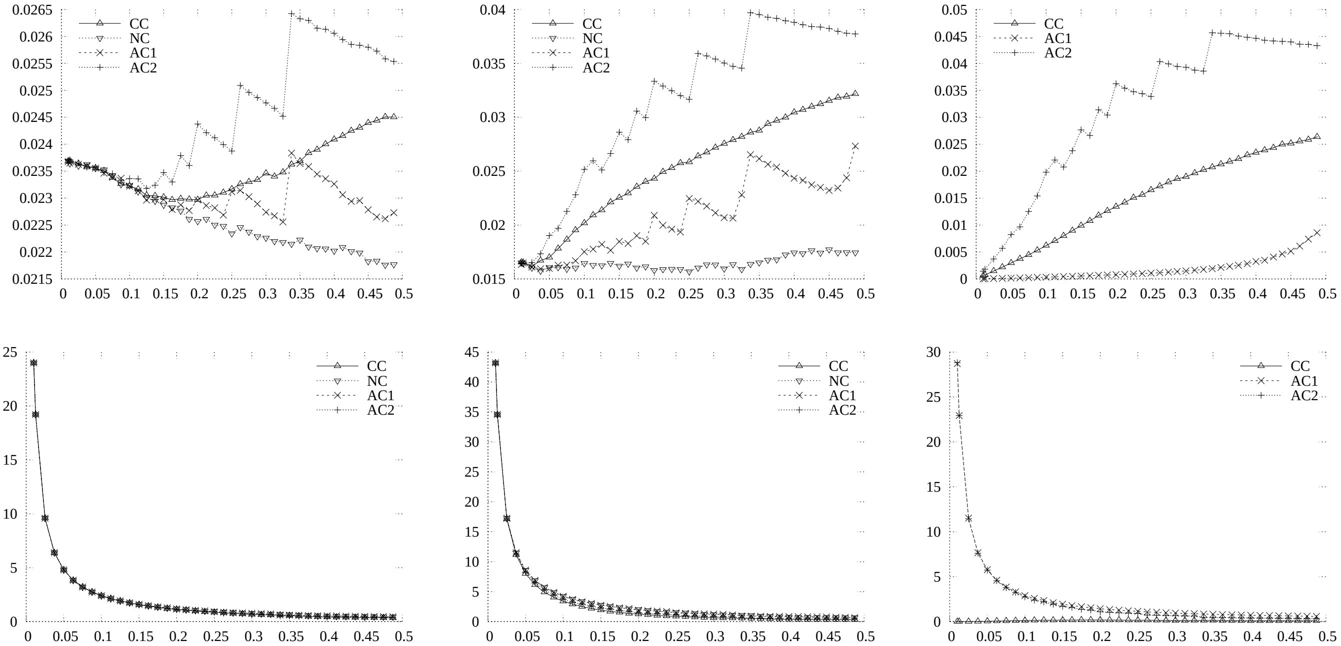
D. Sensitivity to Different Playback Rates

Another important parameter of the workload is the desired rate (or *playback rate*) α. In the above workloads, we picked a playback rate of 0.05 (equivalent to 5% of the network capacity). In this section, we study how the conclusion might change for different playback rates.

Figures 13(a), 13(b) and 13(c) plot the u-put against different α under moderate load ($\rho = 0.5$), heavy load ($\rho = 0.95$), and highly congested load ($\rho = 1.4$) respectively, all with balanced traffic mix. We can make the following observations:

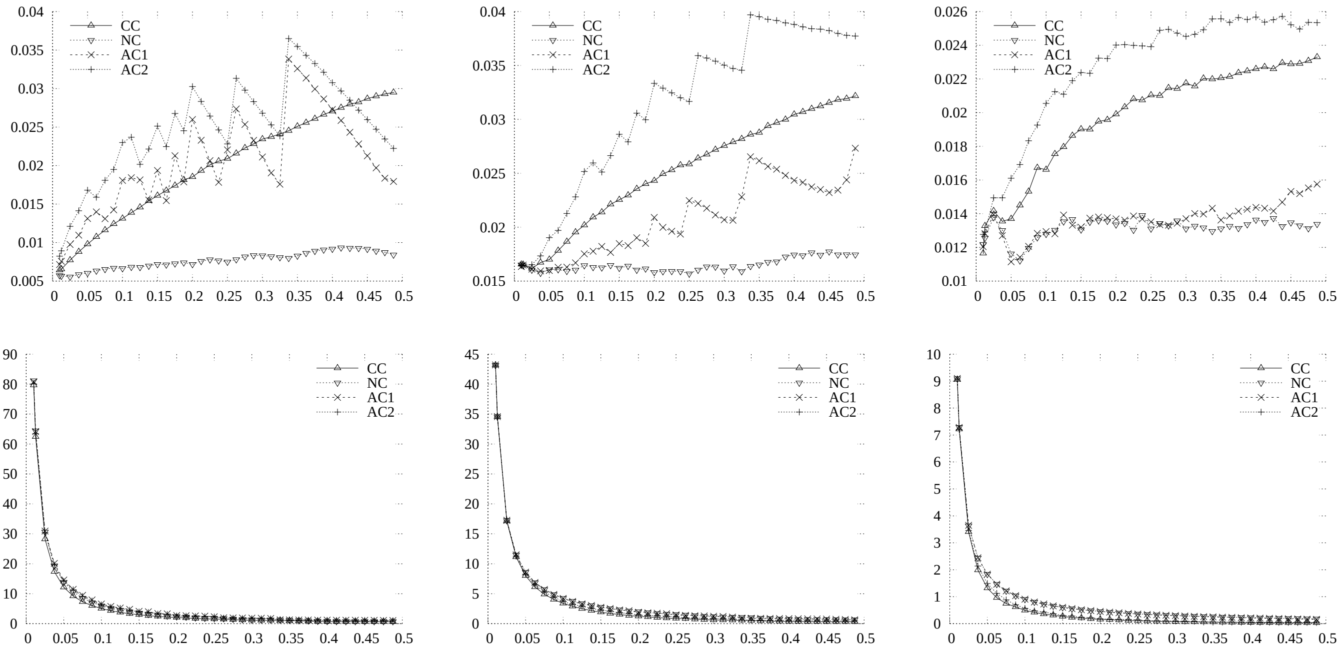
- Generally speaking, as α increases, the elastic u-put increases whereas the inelastic u-put decreases. This can be readily explained. If the traffic control is admission control, then a higher playback rate means the probability of accepting an inelastic flow would be lower; therefore more resources are left to elastic flows. If the traffic control is fair congestion control, a higher playback rate likely means more compression on the inelastic flows; hence more gain by the elastic flows.
- For small α, the performance of different traffic controls tends to be similar. For larger values of α, the performance difference between different controls widens. We attribute this to the *effect of discretization*. For larger α values, the difference between the effect of admitting a flow versus not admitting a flow is amplified. Just like packing large objects into a box, it becomes more difficult to achieve good efficiency as well.
- AC2 is always best for elastic u-put, while AC1 and NC are the worst for elastic flows. For inelastic u-put, AC1 and NC perform better. But AC2 performs at least as good as CC if not better. So our previous conclusion about superiority of AC2 over CC remains true.
- For the case when ρ exceeds 1, the NC case becomes unstable as we found out before. Fair congestion control (CC), while remains stable, performance quite poor in inelastic u-put. This is due to the steep inelastic utility function which values under-allocated inelastic flows very little.

Next we varied the traffic mix while keeping $\rho = 0.95$, as shown in Figure 14. The above observations about the general trends and the superiority of AC2 over CC remains true. In addition, Figure 14(a) and 14(c) show the sensitivity of α when the majority of flows is inelastic and the majority of flows is elastic respectively. As expected, the value of α has more effect when the majority of flows is inelastic (note, Figure 14(a) and 14(c) use different scales for the y-axes).



(a) $\rho = 0.5$ (b) $\rho = 0.95$ (c) $\rho = 1.4$

Fig. 13. Elastic (top) and inelastic (bottom) utilities vs. α with $\rho_E : \alpha\rho_I = 1:1$



(a) $\rho_E : \alpha\rho_I = 1:9$ (b) $\rho_E : \alpha\rho_I = 1:1$ (c) $\rho_E : \alpha\rho_I = 9:1$

Fig. 14. Elastic (top) and inelastic (bottom) utility vs. α with $\rho = 0.95$

VIII. Conclusion

As the Internet is increasingly being used to support both inelastic multimedia applications as well as traditional elastic applications, a central issue is how these applications should share network bandwidth. In this paper, we challenge the conventional wisdom of requiring all flows in the network to be TCP friendly on a per-flow basis. Instead, we propose letting inelastic flows adopt other traffic controls such as admission control.

To investigate the merit of this viewpoint, we propose a methodology to compare the different traffic controls based on utility maximization. We model the traffic as deterministic or stochastic workloads composed of file transfer (elastic) flows and holding time (inelastic) flows. Based on some general assumptions on the form of utility functions, we derive a systematic procedure for computing the rate the network generates aggregate utility of elastic and inelastic flows. We call these the elastic and inelastic *utility throughputs* (u-puts). Different traffic controls can then be compared based on comparing these u-puts.

From analysis based on this methodology, we find a form of TCP-friendly admission control, when applied to inelastic flows, out-performs TCP-friendly congestion control in terms of both elastic as well as inelastic u-puts. This implies, from maximizing the social welfare point of view, it is better to be *TCP-friendly* at the traffic class aggregate level rather than at the per-flow level.

We believe both the methodology and the conclusion opens up new directions for further research. One may continue to explore for more optimal form of traffic control than stay content with TCP-friendly admission control. The methodology itself can be further improved as it involves a number of strong assumptions. For example, it would be interesting to model the retry behavior of the rejected flows. In our analysis, while the conclusions under light load and heavy load are clear-cut, the moderate load case deserves further scrutiny. When applying our methodology to the stochastic workloads, we rely mostly on numeric methods to solve for the large Markov chains involved. We believe there are additional asymptotic results to be obtained. It would also be interesting to remove the exponential file size assumption and see how it affects the conclusions. Finally, how to implement TCP-friendly admission control in a distributed fashion is also a very challenging problem not addressed in this paper.

APPENDIX

Proof for Proposition 1

From Equation 10, the u-put equation for elastic flows is of the form $na_E(n,m)u_E(a_E(n,m))$ for any traffic control. Since $xu_E(x)$ is an increasing function in x, we can compare the ranking of the four traffic controls based on the respective allocation functions $a_E(n,m)$, if all m flows are admitted.

For region (i) in Figure 5, all m inelastic flows are admitted. The $a_E(n,m)$ is the same for all four controls, and their elastic u-puts are the same.

TABLE VI

NO. OF INELASTIC FLOWS ADMITTED FOR EACH REGION IN FIGURE 5

	Region (i)	Region (ii)	Region (iii)	Region (iv)
AC1	m	m	$\lfloor(1-n\varepsilon)/\alpha\rfloor$	$\lfloor(1-n\varepsilon)/\alpha\rfloor$
AC2	m	$\lfloor(1-n\alpha)/\alpha\rfloor$	$\lfloor(1-n\alpha)/\alpha\rfloor$	$\lfloor(1-n\alpha)/\alpha\rfloor$

Outside of region (i), namely when $(n+m)\alpha > 1$, NC allocates no bandwidth to elastic flows, hence it is ranked lower than the other three controls. The comparison of AC1, AC2 and CC depends on how many inelastic flows are admitted. Given that the elastic flows all arrive before the inelastic flows, Table VI lists the number of inelastic flows admitted by AC1 and AC2 in different regions respectively. Given this information, and the traffic control definition in Equation 7, it is straightforward to verify that the rankings in Table II are true.

Proof of Proposition 2

For AC1 and AC2, the allocated bandwidth for each admitted flow is always α. The comparison (hence the ranking) of AC1 and AC2 boils down to counting the number of inelastic flows admitted. Therefore, the ranking of AC1 and AC2 are the same in region (i), and AC1 generates more inelastic u-puts in the rest of the regions, based on Table VI.

Many of the regions correspond to special cases when allocations are zero for specific traffic controls. Since inelastic flows all arrive after the elastic flows, above the line $n\alpha = 1$ (region (vii) or higher), $a_I(n,m)$ is zero for AC2; and above the line $n\varepsilon = 1$ (region (x) and higher), $a_I(n,m)$ is zero for both AC1 and AC2. Given that $u_I()$ is a step function, to the right of the line $m\alpha = 1$ (region (vi), (ix) and (x)), $a_I(n,m)$ is zero for both CC and NC.

The regions when most of the allocations are greater than zero (regions (i), (iii), (iv) (vii) and (viii)) are of more interest in comparison. In each case, it is straightforward to derive the rankings in Table IV based on Equation 8 and Table VI that gives the number of inelastic flows admitted.

REFERENCES

[1] D. E. McDysan and D. L. Spohn, *ATM — Theory and Application*. New York, NY, USA: McGraw-Hill, 1995.

[2] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," *IETF RFC1633*, June 1994.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” *IETF RFC2475*, Dec. 1998.

[4] E. Osborne and A. Simha, *Traffic Engineering with MPLS*. Cisco Press, Mar. 2002.

[5] S. Shenker, “Fundamental design issues for the future internet,” *IEEE Journal on Selected Areas in Communications*, vol. 13, Sept. 1995.

[6] J. Mahdavi and S. Floyd, “TCP-friendly unicast rate-based flow control,” Technical note sent to the end2end-interest mailing list, Jan. 1997.

[7] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP friendly rate control (TFRC): Protocol specification,” IETF RFC3448, Jan. 2003, proposed Standard. [Online]. Available: ftp://ftp.internic.net/rfc/rfc3448.txt

[8] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, “A model based TCP-friendly rate control protocol,” in *Proceedings NOSSDAV*. Basking Ridge, NJ: International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), June 1999.

[9] L. Massoulié and J. Roberts, “Bandwidth sharing: Objectives and algorithms,” *IEEE/ACM Transactions on Networking*, vol. 10, 2002.

[10] M. Chiang, S. Zhang, and P. Hande, “Distributed rate allocation for inelastic flows: Optimization frameworks, optimality conditions, and optimal algorithms,” in *Proceedings of INFOCOM 2005*, Mar. 2005.

[11] J.-W. Lee, R. R. Masumdar, and N. B. Shroff, “Non-convex optimization and rate control for multi-class services in the internet,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, pp. 827–840, 2005.

[12] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, “Endpoint admission control: Architectural issues and performance,” in *Proceedings of SIGCOMM 2000*, Aug. 2000.

[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] R. Jain, D. M. Chiu, and W. Hawe, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” DEC, DEC Technical Report DEC-TR-301, Sept. 1984.

[15] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” in *Proceedings of the ACM SIGCOMM’89*, Sept. 1989.

[16] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control in communication networks: Shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[17] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 1998.

[18] P. Key, L. Massoulié, A. Bain, and F. Kelly, “A network flow model for mixtures of file transfers and streaming traffic,” in *Proceedings of ITC 18*, Berkin, Germany, Sept. 2003.

[19] J. Roberts, “A survey on statistical bandwidth sharing,” *Computer Networks*, July 2004.

[20] T. Bonald and A. Proutiere, “On performance bounds for the integration of elastic and adaptive streaming flows,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, pp. 235–245, 2004.

[21] R. Howard, *Dynamic Programming and Markov Processes*. Cambridge MA, USA: MIT Press, 1960.

[22] A. S.-W. Tam, D. M. Chiu, J. C. S. Lui, and Y. C. Tay, “A case for TCP-friendly admission control,” 2006, proceedings of IEEE IWQoS.