Queueing Properties of Feedback Flow Control Systems

Dongyu Qiu and Ness Shroff

School of Electrical and Computer Engineering

Purdue University

West Lafayette, IN 47907, U.S.A.

E-mail:{dongyu, shroff}@ecn.purdue.edu

Abstract

In this paper, we consider a network with both controllable and uncontrollable flows. Uncontrollable flows are typically generated from applications with stringent QoS requirements and are given high priority. On the other hand, controllable flows are typically generated by elastic applications and can adapt to the available link capacities in the network. We provide a general model of such a system and analyze its queueing behavior. Specially, we obtain a lower bound and an asymptotic upper bound for the tail of the workload distribution at each link in the network. These queueing results provide us with guidelines on how to design a feedback flow control system. Simulation results show that the lower bound and asymptotic upper bound are quite accurate and that our feedback control method can effectively control the queue length in the presence of both controllable and uncontrollable traffic. Finally, we describe a distributed strategy that uses the notion of Active Queue Management (AQM) for implementing our flow control solution.

1 Introduction

In communication networks, we can classify flows as being either controllable or uncontrollable. The controllable flows can adjust their data rates in response to the feedback information received from the network. Typical examples of controllable flows are TCP flows in the Internet and ABR flows in an ATM network. On the other hand, the data rate of an uncontrollable flow is determined by the application and cannot usually be adapted to the network congestion status (this is typical of flows with stringent QoS requirements). Because of the potential for networks to carry applications with diverse features, we expect to see both controllable and uncontrollable flows in future networks. Even in a network with only TCP-flows, some flows are so short-lived that they leave the network before being able to adequately respond to any feedback. For example, measurements of the Internet traffic show that a significant fraction of TCP flows are short-lived flows (mainly because of the popularity of http protocol) or *mice*. Since these flows do not respond well to feedback information, they can also be viewed as being uncontrollable.

Uncontrollable flows are often generated from applications with QoS requirements, such as the loss probability, queueing delay, and jitter etc. Hence, when there are only uncontrollable flows in the network, analyzing their queueing behavior is especially important because these QoS metrics are directly related to the queue length distribution. On the other hand, when there are only controllable flows in the network, queueing is not such a critical issue and most research has focused on how to distribute the link capacities (via flow control) among the different flows such that some fairness criteria are satisfied, or the total network performance is maximized [1, 2, 3, 4]. All of these works assume that the available link capacity of each link is fixed (not time-varying). Under this assumption, in these works, distributed and iterative algorithms are given and it is proven that, under suitable conditions,

the rate allocation vector (rate of each flow) converges to an optimal point (or an allocation that satisfies the given fairness criterion). Since the data rate of each flow will eventually converge, the aggregate input rate to a given link will also converge to a constant. This constant is either less than the link capacity (non-bottleneck) or equal to the link capacity (bottleneck). So, the queue length associated with each link is either zero or a finite constant. Hence, queueing is not an important issue in this case. However, when both types of flows are present, uncontrollable flows, because of their QoS requirements, are generally given a higher priority over controllable flows. While this ensures that the QoS of uncontrollable flows is not affected by controllable flows, it also means that controllable flows can only utilize the residual link capacity (time-varying). In this case, the objective of flow control is to maintain high link utilization, low loss probability, and fairness [5, 6, 7, 8, 9, 10]. Most previous works in this area have focused on a single bottleneck link and it is not easy to extend the single bottleneck link results to a network with multiple bottleneck links. Further, those works mainly focus on the flow control algorithm and do not shed much insight on the queueing behavior of the controlled queue. In this paper, we will first provide a general model of a feedback flow control system with both types of flows. Under this framework, we show that the single link results can easily be extended to a network with multiple links. We then analyze the queueing behavior of such a system. We believe that the queueing analysis of such a system has significant importance because it can provide appropriate guidelines on how to design the feedback control system. We then give an example application and discuss how our scheme could be implemented in a distributed way. Finally, we provide simulation results to illustrate the efficiency of our techniques.

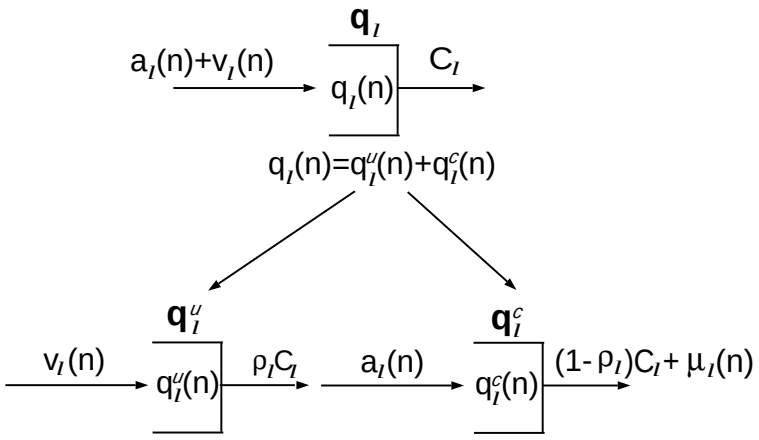

Figure 1: Model of a Single Link l

2 Queueing Analysis of Feedback Flow Control Systems

In this section, we analyze the queueing behavior of a feedback flow control system with both uncontrollable and controllable flows. We consider a network with L bottleneck links and S controllable flows. The set of bottleneck links is $\{1, \cdots, L\}$ and the set of controllable flows is $\{1, \cdots, S\}$. Any given bottleneck link l in the network has associated with it a queue denoted by $\mathbf{q}_l$ (Fig 1). For the purpose of analysis, we consider an infinite buffer discrete-time fluid queueing model. Let $a_l(n)$ be the aggregate input rate of the controllable flows and $v_l(n)$ be the aggregate input rate of the uncontrollable flows at time n. Further, let $q_l(n)$ be the workload at time n, C_l be the link capacity, and $\rho_l \leq 1$ be the target link utilization. We now describe our general flow control model. In Section 4.1, we will see how this model can be significantly simplified under certain conditions. In our model, we assume that uncontrollable flows always have priority over controllable flows, but at any time n, the amount of uncontrollable traffic that leaves the queue $\mathbf{q}_l$ cannot exceed $\rho_l C_l$.

This assumption, while not necessary for the later development of our results, ensures that at least a minimum amount of capacity is available for the controllable flows. We now define two queueing system $\mathbf{q}_l^u$ and $\mathbf{q}_l^c$ whose workloads $q_l^u(n)$ and $q_l^c(n)$ correspond to the workload caused by uncontrollable flows and controllable flows, respectively. The sum of $q_l^u(n)$ and $q_l^c(n)$ equals $q(n)$ at all time n. The system $\mathbf{q}_l^u$ is defined as the queueing system with only $v_l(n)$ as the input and $\rho_l C_l$ as the link capacity. We then have

$$q_l^u(n) = [q_l^u(n-1) + v_l(n) - \rho_l C_l]^+,$$

where $[x]^+ = x$ if $x \geq 0$ and 0 otherwise. Here, $q_l^u(n)$ is the workload caused by the uncontrollable flows and thus we cannot control it in any way. Let

$$\mu_l(n) = [\rho_l C_l - v_l(n) - q_l^u(n-1)]^+ \tag{1}$$

be the residual link capacity of $\mathbf{q}_l^u$. Then, $\mu_l(n) + (1-\rho_l)C_l$ is the available link capacity for controllable flows at time n. Note, however, that the controllable flows will only utilize a fraction of the available link capacity to ensure that the total link utilization is ρ_l. Ideally, we want $a_l(n) = \mu_l(n)$ for all time n. But this is impossible to achieve in practice because of network delays, estimation errors, etc. Hence, the best we can do is to control $a_l(n)$ such that it can track the changes in $\mu_l(n)$. We define $\mathbf{q}_l^c$ as the queueing system with $a_l(n)$ as input and $\mu_l(n) + (1-\rho_l)C_l$ as the link capacity. Then $q_l^c(n)$ is the workload caused by the controllable flows, and is what we will focus on. The total workload $q_l(n)$ will then be $q_l(n) = q_l^u(n) + q_l^c(n)$. *The idea behind separating $q_l(n)$ into these components is that since we cannot control $q_l^u(n)$, if we minimize $q_l^c(n)$, we will also minimize $q_l(n)$.*

We characterize $v_l(n)$, the aggregate input rate of the uncontrollable flows on link l, by a stationary stochastic process. Then $\mu_l(n)$ is also a stationary stochastic process and will be used to control the data rates of the controllable flows. Let $\mu(n) = [\mu_1(n), \cdots, \mu_L(n)]^T$ and $a(n) = [a_1(n), \cdots, a_L(n)]^T$. We assume that the feedback control system is linear (i.e.,

$a(n)$ is a linear transformation of $\mu(n)$) or can be approximately modeled as a linear system (an example will be discussed in Section 3). A linear feedback control has been found to give good results when we have video traffic as uncontrollable traffic [9, 11], and we find that it gives good results for other types of traffic as well [10]. Note that we only assume that the feedback control is linear. All queueing systems considered here are still non-linear. In practice, the linear feedback control may be an approximation of a real control system. This approximation is valid under certain conditions as noted in [12]. Note that if we set the target link utilization ρ_l to be strictly less than one for all links l, it will be equivalent to Assumption 1 of [12], which justifies the linearized system. We don't need further assumptions on the queueing system. On the other hand, the non-linearity of a queueing system comes from the fact that the queue size can never be less than zero. This is also the main difficulty in the analysis of queues. We will take into account this non-linearity in our analysis. In fact, if we set ρ_l to be strictly less than one, it is very likely that the queues will be frequently empty, as pointed out by [12]. This is exactly why we need to consider the non-linearity of the queueing system.

Now let $\boldsymbol{\mu}(z)$ and $\mathbf{a}(z)$ be the Z-transforms of $\mu(n)$ and $a(n)$ respectively. We have

$$\mathbf{a}(z) = \mathbf{H}(z)\boldsymbol{\mu}(z),$$

where $\mathbf{H}(z)$ is an $L \times L$ matrix and represents a causal, stable, linear, time-invariant system [13]. For example, if there is only one controllable flow and one link in the network and the round trip delay for the flow is 5, we may have $a(n) = 0.5\mu(n-5) + 0.5\mu(n-6)$ and $\mathbf{H}(z) = 0.5z^{-5} + 0.5z^{-6}$. Next, we will see how to design $\mathbf{H}(z)$ to achieve certain desirable properties for the controlled queueing system $\mathbf{q}_l^c$.

Proposition 1 *If* $\mathbf{H}(1) = I$ *and* $\bar{v}_l < \rho_l C_l$ *for all* l*, then in the steady state of the system, we have* $\bar{a}_l + \bar{v}_l = \rho_l C_l$ *for all* l*, where* $\bar{v}_l = \mathbb{E}\{v_l(n)\}$ *and* $\bar{a}_l = \mathbb{E}\{a_l(n)\}$.

Proof: Let $d_l(n)$ be the amount of uncontrollable traffic that leaves $\mathbf{q}_l^u$ at time n. Then, from Eq. (1), the definition of $\mu_l(n)$, we have $\mu_l(n) + d_l(n) = \rho_l C_l$ for all n.

Because $\bar{v}_l < \rho_l C_l$, $\mathbf{q}_l^u$ is a stable system. So, $\bar{v}_l = \bar{d}_l$. Now, since $\mathbf{H}(1) = I$, $\mathbb{E}\{\mu_l(n)\} = \bar{a}_l$. Hence, we have $\bar{a}_l + \bar{v}_l = \rho_l C_l$. ■

Proposition 1 tells us that under the condition $\mathbf{H}(1) = I$, the actual link utilization of link l is fixed at ρ_l, our target utilization. We next focus on the behavior of the workload for a given utilization.

Proposition 2 *If $\mathbf{H}(1) = I$, there exists a constant C_q such that $q_l^c(n) \leq C_q$ for all n and all l.*

Proof: Let $Y_l(n) = \sum_{j=0}^{n}(a_l(j) - \mu_l(j))$ and $Y(n) = [Y_1(n), \cdots, Y_L(n)]^T$. The Z-transform of $Y(n)$,

$$\mathbf{Y}(z) = \frac{1}{1 - z^{-1}}(\mathbf{a}(z) - \boldsymbol{\mu}(z)) = \frac{1}{1 - z^{-1}}(\mathbf{H}(z) - I)\boldsymbol{\mu}(z). \tag{2}$$

Since $\mathbf{H}(1) = I$, $z = 1$ will be a zero point of $\mathbf{H}(z) - I$. Hence, $\mathbf{H}(z) - I$ can be written as $\mathbf{H}_1(z)(1 - z^{-1})$ where $\mathbf{H}_1(z)$ is still a stable system. Therefore,

$$\mathbf{Y}(z) = \mathbf{H}_1(z)\boldsymbol{\mu}(z).$$

Note that this is a multiple input, multiple output system. Because each input $0 \leq \mu_l(n) \leq C_l$ (i.e., $\mu(n)$ is bounded) and $\mathbf{H}_1(z)$ is a stable system, $Y(n)$ will also be bounded. For any l, n, and n_0, there will exist a constant C_q such that

$$Y_l(n) - Y_l(n_0) \leq C_q.$$

Let $\mathbf{q}_l^c$ be empty at time $n = 0$, then $q_l^c(n)$, the workload caused by controllable traffic at time n, can be expressed as [14] [15]:

$$q_l^c(n) = \sup_{0 \leq n_0 \leq n} \left\{ \sum_{j=n_0+1}^{n} \Big(a_l(j) - \mu_l(j) - (1 - \rho_l)C_l \Big) \right\} \tag{3}$$

From Eq. (3), we know,

$$q_l^c(n) = \sup_{0 \le n_0 \le n} (Y_l(n) - Y_l(n_0) - (n - n_0)(1 - \rho_l)C_l) \le \sup_{0 \le n_0 \le n} (Y_l(n) - Y_l(n_0)) \le C_q.$$

■

Proposition 2 tells us that $q_l^c(n)$ can be bounded by a constant (independent of n) when $\mathbf{H}(z)$ is appropriately chosen. However this condition may not be sufficient to guarantee a good flow control mechanism because the value of this constant could be loose. We are more interested in the details of the distribution of the workload $\mathbb{P}\{q_l^c(n) > x\}$. Since $\mu(n)$ is stationary and $a(n)$ is a linear transformation of $\mu(n)$, $a(n)$ is also stationary. The steady state workload distribution of q_l^c will be given by [14] [15]

$$\mathbb{P}\{Q_l^c > x\} = \mathbb{P}\left\{\sup_{n \ge 0} X_{l,n} > x\right\}, \tag{4}$$

where $X_{l,n} = \sum_{j=-n+1}^{0}(a_l(j) - \mu_l(j) - (1 - \rho_l)C_l)$ (note that $X_{l,n}$ is the sum of the aggregate input rates minus the sum of the available link capacities from time $-n+1$ to 0 at link l). From now on, we will focus on the steady state properties of the queueing system.

From Eq. (4), it follows that the stochastic properties of $X_{l,n}$ will directly affect the workload distribution. If $\mathbf{H}(1) = I$, $\mathbb{E}\{a_l(j)\} = \mathbb{E}\{\mu_l(j)\}$. From the definition of $X_{l,n}$, we know that, $\mathbb{E}X_{l,n} = -(1 - \rho_l)C_l n = -k_l n$, where $k_l = (1 - \rho_l)C_l$. In [16], it has been shown that when k_l is fixed, $\mathrm{Var}X_{l,n}$ (the variance of $X_{l,n}$) plays an important role in the queue distribution. In general, when n goes to infinity, $\mathrm{Var}X_{l,n}$ will also go to infinity. For example, if the input process to link l is a long range dependent process with Hurst parameter $H \in [1/2, 1)$ and the link capacity is not time-varying, we have $\mathrm{Var}X_{l,n} \sim Sn^{2H}$, when $n \to \infty$, where S is a constant. But in a controlled queueing system, we show that $\mathrm{Var}X_{l,n}$ can be bounded, as is given by the next lemma.

Lemma 1 *If* $\mathbf{H}(1) = I$ *and* $Var\{\mu_l(n)\}$ *is finite for all* l*, then in the steady state of the system, for each link* l*, there exists a constant* D_l *such that* $VarX_{l,n} \le D_l$ *for all* n*.*

Proof: Since $\mu(n)$ is stationary, we can easily see that $\mathrm{Var}X_{l,n} = \mathrm{Var}\{Y_l(n-1)\}$. From the proof of Proposition 2, we know that $\mathbf{Y}(z) = \mathbf{H}_1(z)\boldsymbol{\mu}(z)$, where $\mathbf{H}_1(z)$ is a stable system. Because $\mathrm{Var}\{\mu_l(n)\}$ is finite for all l and $\mathbf{H}_1(z)$ is stable, $\mathrm{Var}\{Y_l(n)\}$ will also be finite and bounded for all n and l. Hence, there exists a constant D_l such that $\mathrm{Var}X_{l,n} \leq D_l$ for all n.

■

Note that in practice, since $0 \leq \mu_l(n) \leq C_l$, $\mathrm{Var}\{\mu_l(n)\}$ will always be finite. But in Lemma 1, we only require $\mathrm{Var}\{\mu_l(n)\}$ to be finite and do not require $\mu_l(n)$ to be bounded. This will be useful when we model $\mu(n)$ by a Gaussian process (as described next). We will also see how the fact that the $\mathrm{Var}X_{l,n}$ can be bounded will affect the workload distribution. For the purpose of analysis, we assume that $\mu(n)$ is a joint Gaussian process. A Gaussian process is a good model for the aggregate traffic in a high-speed network. Although the traffic from each individual application may not be accurately characterized by a Gaussian process, the aggregate traffic from many different applications is modeled quite effectively by a Gaussian process. Note that in Fig. 1, $\mu_l(n)$ is the residual link capacity in $\mathbf{q}_l^u$ and hence it is approximately $\rho_l C_l - v_l(n)$ when $\mathbf{q}_l^u$ is lightly loaded (it is exactly $\rho_l C_l - v_l(n)$, if $v_l(n) < \rho_l C_l$ for all n). We believe that the condition of lightly loaded $\mathbf{q}_l^u$ is not atypical because uncontrollable flows are generated by applications with stringent QoS requirements. Hence, we expect that queues will be lightly loaded for uncontrollable flows. Further, this also reflects the traffic pattern in the current Internet. Now since $v_l(n)$ is the aggregate input rate of uncontrollable flows, it is effectively characterized by a Gaussian process, and hence $\mu_l(n)$ can also be approximated by a Gaussian process (we will also justify this approximation numerically in Section 5). Now, if $\mu(n)$ is Gaussian, $X_{l,n}$ will also be Gaussian. When $\mathbf{H}(1) = I$, we know that $\mathbb{E}X_{l,n} = -(1-\rho_l)C_l n = -k_l n$ and $\mathrm{Var}X_{l,n}$ is bounded. Let

$V_{l,n} = \mathrm{Var}X_{l,n}$. We have,

$$\mathbb{P}\left\{Q_l^c > x\right\} = \mathbb{P}\left\{\sup_{n\geq 0} X_{l,n} > x\right\} \geq \sup_{n\geq 0} \mathbb{P}\left\{X_{l,n} > x\right\} = \sup_{n\geq 0} \Psi\left(\frac{x + k_l n}{\sqrt{V_{l,n}}}\right),$$

where $\Psi(x)$ is the tail of the standard Gaussian distribution, i.e., $\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{z^2}{2}} \mathrm{d}z$. It has been shown [17] that,

$$\frac{1 - z^{-2}}{\sqrt{2\pi}} z^{-1} e^{-\frac{z^2}{2}} \leq \Psi(z) \leq \frac{1}{\sqrt{2\pi}} z^{-1} e^{-\frac{z^2}{2}} \quad \text{for} \quad z > 0 \tag{5}$$

Let $n_{l,x} = \mathrm{argmin}_n \frac{x+k_l n}{\sqrt{V_{l,n}}}$. Then $n_{l,x}$ is the time at which $\mathbb{P}\left\{X_{l,n} > x\right\}$ attains its maximum value, i.e., $n_{l,x}$ is the *dominant time scale*. Further let $\sigma_{l,x}^2 = \frac{x V_{n_{l,x}}}{(x+k_l n_{l,x})^2}$. It has been shown in [16] that the tail of the workload distribution is asymptotically of the form $e^{-\frac{x}{2\sigma_{l,x}^2}}$ if $V_{l,n} \sim S n^{2H}$ when n is large, where $H \in [1/2, 1)$ and S are constants. In our case, however, $V_{l,n}$ is not of this form since it can be bounded as shown by Lemma 1. We next study the behavior of $\mathbb{P}\left\{Q_l^c > x\right\}$ when $V_{l,n}$ is bounded.

Lemma 2 *For any given link l, let $D_l = \sup_{n\geq 0} V_{l,n}$ be finite, then in the steady state of the system,*

$$\lim_{x\to\infty} V_{l,n_{l,x}} = D_l,$$

$$\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = 1,$$

$$\lim_{x\to\infty} \frac{n_{l,x}}{x} = 0.$$

Proof: For any $\varepsilon > 0$, because $D_l = \sup_{n\geq 0} V_{l,n}$, the set $\{n | V_{l,n} \geq D_l - \varepsilon\}$ will not be empty. Let $n_0 = \min\{n | V_{l,n} \geq D_l - \varepsilon\}$. Then for any $i < n_0$, we have $V_{l,i} < V_{l,n_0}$. Let $x_i = k_l \frac{n_0\sqrt{V_{l,i}} - i\sqrt{V_{l,n_0}}}{\sqrt{V_{l,n_0}} - \sqrt{V_{l,i}}}$. Then if $x > x_i$, we have,

$$\frac{(x + k_l n_0)^2}{V_{l,n_0}} < \frac{(x + k_l i)^2}{V_{l,i}}.$$

Let $x_0 = \max\{x_i | 1 \le i < n_0\}$. If $x > x_0$,

$$\frac{(x + k_l n_0)^2}{V_{l,n_0}} < \frac{(x + k_l i)^2}{V_{l,i}},$$

for all $i < n_0$. So, we have $n_{l,x} \ge n_0$. From

$$\frac{(x + k_l n_{l,x})^2}{V_{l,n_{l,x}}} \le \frac{(x + k_l n_0)^2}{V_{l,n_0}},$$

it is easy to show that

$$\frac{V_{l,n_{l,x}}}{V_{l,n_0}} \ge \frac{(x + k_l n_{l,x})^2}{(x + k_l n_0)^2} \ge 1.$$

So, for any $\varepsilon > 0$, there exists x_0, when $x > x_0$, $V_{l,n_{l,x}} \ge V_{l,n_0} \ge D_l - \varepsilon$, i.e., $\lim_{x\to\infty} V_{l,n_{l,x}} = D_l$.

Next, we will prove that $\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = 1$. For any $\varepsilon > 0$, because $D_l = \sup_{n\ge 0} V_{l,n}$, there must exist a n_0 such that $V_{l,n_0} > \frac{D_l}{1+\varepsilon}$. So,

$$\frac{D_l}{x\sigma_{l,x}^2} = \frac{D_l(x + k_l n_{l,x})^2}{x^2 V_{l,n_{l,x}}} \le \frac{D_l(x + k_l n_0)^2}{x^2 V_{l,n_0}}.$$

Since $\lim_{x\to\infty} \frac{D_l(x+k_l n_0)^2}{x^2 V_{l,n_0}} = \frac{D_l}{V_{l,n_0}} < 1 + \varepsilon$, there must exist x_0, when $x > x_0$,

$$\frac{D_l(x + k_l n_0)^2}{x^2 V_{l,n_0}} < 1 + \varepsilon.$$

This also means,

$$1 \le \frac{D_l(x + k_l n_{l,x})^2}{x^2 V_{l,n_{l,x}}} < 1 + \varepsilon.$$

So, $\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = 1$.

From $\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = \lim_{x\to\infty} \frac{D_l(x+k_l n_{l,x})^2}{x^2 V_{l,n_{l,x}}} = 1$ and $\lim_{x\to\infty} V_{l,n_{l,x}} = D_l$, it is easy to see that

$$\lim_{x\to\infty} \frac{n_{l,x}}{x} = 0.$$

■

Now, we are ready to prove our main result.

Theorem 1 *If $D_l = \sup_{n\geq 0} V_{l,n}$ is finite for link l and $\mu(n)$ is a Gaussian process, then in the steady state of the system,*

$$\begin{aligned} -1 &\leq \liminf_{x\to\infty} \frac{1}{\log x}\left(\log \mathbb{P}\left\{Q_l^c > x\right\} + \frac{x}{2\sigma_{l,x}^2}\right) \\ &\leq \limsup_{x\to\infty} \frac{1}{\log x}\left(\log \mathbb{P}\left\{Q_l^c > x\right\} + \frac{x}{2\sigma_{l,x}^2}\right) \leq 0 \end{aligned}$$

Proof: We first have,

$$\begin{aligned} \mathbb{P}\left\{Q_l^c > x\right\} &= \mathbb{P}\left\{\sup_{n\geq 0} X_{l,n} > x\right\} \geq \mathbb{P}\left\{X_{l,n_{l,x}} > x\right\} \\ &= \Psi(\sqrt{x/\sigma_{l,x}^2}) \geq \frac{1-\sigma_{l,x}^2/x}{\sqrt{2\pi x/\sigma_{l,x}^2}} e^{-\frac{x}{2\sigma_{l,x}^2}}. \end{aligned} \tag{6}$$

So,

$$\frac{1}{\log x}\left(\log \mathbb{P}\left\{Q_l^c > x\right\} + \frac{x}{2\sigma_{l,x}^2}\right) \geq \frac{1}{\log x}\left(\log(1-\sigma_{l,x}^2/x) - \frac{1}{2}\log(2\pi x/\sigma_{l,x}^2)\right).$$

From Lemma 2, we have $\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = 1$. Hence,

$$\lim_{x\to\infty} \frac{1}{\log x}\left(\log(1-\sigma_{l,x}^2/x) - \frac{1}{2}\log(2\pi x/\sigma_{l,x}^2)\right) = -1.$$

We now have

$$\liminf_{x\to\infty} \frac{1}{\log x}\left(\log \mathbb{P}\left\{Q_l^c > x\right\} + \frac{x}{2\sigma_{l,x}^2}\right) \geq -1.$$

Now, for the lim sup part. Let $\widetilde{n_x} = (\sqrt{\frac{D_l}{V_{l,n_{l,x}}}} - 1)\frac{x}{k_l} + \sqrt{\frac{D_l}{V_{l,n_{l,x}}}} n_{l,x}$. Then,

$$\frac{x}{\sigma_{l,x}^2} = \frac{(x+k_l n_{l,x})^2}{V_{l,n_{l,x}}} = \frac{(x+k_l\widetilde{n_x})^2}{D_l} \tag{7}$$

$$\begin{aligned} \mathbb{P}\left\{Q_l^c > x\right\} &= \mathbb{P}\left\{\sup_{n\geq 0} X_{l,n} > x\right\} \\ &\leq \sum_{n\geq 0} \mathbb{P}\left\{X_{l,n} > x\right\} = \sum_{n\geq 0} \Psi\left(\frac{x+k_l n}{\sqrt{V_{l,n}}}\right) \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{n\geq 0}\frac{1}{\sqrt{2\pi}}\frac{\sqrt{V_{l,n}}}{x+k_l n}e^{-\frac{(x+k_l n)^2}{2V_{l,n}}}\\
&\leq \sum_{n\geq 0}\frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{(x+k_l n)^2}{2V_{l,n}}}\\
&= \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\sum_{n\geq 0}e^{-\frac{1}{2}\left[\frac{(x+k_l n)^2}{V_{l,n}}-\frac{x}{\sigma_{l,x}^2}\right]}\\
&\leq \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{n\geq\widetilde{n_x}}e^{-\frac{1}{2}\left[\frac{(x+k_l n)^2}{V_{l,n}}-\frac{x}{\sigma_{l,x}^2}\right]}\right)\\
&= \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{n\geq\widetilde{n_x}}e^{-\frac{1}{2}\left[\frac{(x+k_l n)^2}{V_{l,n}}-\frac{(x+k_l\widetilde{n_x})^2}{D_l}\right]}\right) \quad \text{FromEq. (7)}\\
&\leq \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{n\geq\widetilde{n_x}}e^{-\frac{1}{2}\left[\frac{(x+k_l n)^2}{D_l}-\frac{(x+k_l\widetilde{n_x})^2}{D_l}\right]}\right)\\
&= \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{n\geq\widetilde{n_x}}e^{-\frac{(2x+k_l n+k_l\widetilde{n_x})(k_l n-k_l\widetilde{n_x})}{2D_l}}\right)\\
&\leq \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{n\geq\widetilde{n_x}}e^{-\frac{(k_l n-k_l\widetilde{n_x})^2}{2D_l}}\right)\\
&= \frac{\sqrt{D_l}}{\sqrt{2\pi}x}e^{-\frac{x}{2\sigma_{l,x}^2}}\left(\widetilde{n_x}+\sum_{m\geq 0}e^{-\frac{(k_l m)^2}{2D_l}}\right) \quad \text{let } m=n-\widetilde{n_x}\\
&\leq \frac{\sqrt{D_l}}{\sqrt{2\pi}x}(\widetilde{n_x}+A)e^{-\frac{x}{2\sigma_{l,x}^2}},
\end{aligned}
$$

where A is a constant. Since $\lim_{x\to\infty}\frac{\sqrt{D_l}}{\sqrt{2\pi}x}(\widetilde{n_x}+A)=0$, when x is large enough, we have

$$\mathbb{P}\left\{Q_l^c>x\right\}\leq e^{-\frac{x}{2\sigma_{l,x}^2}}.$$

Hence,

$$\limsup_{x\to\infty}\frac{1}{\log x}\left(\log\mathbb{P}\left\{Q_l^c>x\right\}+\frac{x}{2\sigma_{l,x}^2}\right)\leq 0.$$

■

Corollary 1 *If $D_l = \sup_{n\geq 0} V_{l,n}$ is finite for link l, then in the steady state of the system, when x is large,*

$$\mathbb{P}\left\{Q_l^c > x\right\} \leq e^{-\frac{x^2}{2D_l}}.$$

Proof: From the proof of Theorem 1, we know that when x is large enough, we have

$$\mathbb{P}\left\{Q_l^c > x\right\} \leq e^{-\frac{x}{2\sigma_{l,x}^2}}.$$

From Lemma 2, we also have,

$$\lim_{x\to\infty} \frac{D_l}{x\sigma_{l,x}^2} = 1.$$

So, it is easy to show that when x is large enough,

$$\mathbb{P}\left\{Q_l^c > x\right\} \leq e^{-\frac{x^2}{2D_l}}.$$

■

Theorem 1 tell us that when $V_{l,n}$ is bounded, $e^{-\frac{x}{2\sigma_{l,x}^2}}$ is a good approximation to the tail probability $\mathbb{P}\{Q_l^c > x\}$ (note that although the theorem requires x to be large, our simulations show that the bounds on $\mathbb{P}\left\{Q_l^c > x\right\}$ are accurate even when x is small). From Corollary 1, we know that when x is large, the tail probability of $\mathbf{q}_l^c$ will decrease on the order of $e^{-\frac{x^2}{2D_l}}$. Note that this is quite different from the uncontrolled case in [16] where $V_{l,n} \sim Sn^{2H}$ for $H \in [1/2, 1)$ and when x is large, the tail probability will decrease only on the order of $e^{-bx^{2-2H}}$, where b is a constant. This tell us that when $\mathrm{Var}X_{l,n}$ is bounded, the tail probability of $\mathbf{q}_l^c$ will asymptotically decrease much faster than when $\mathrm{Var}X_{l,n}$ is not bounded. Hence, it is important to choose the design parameters correctly (e.g., set $\mathbf{H}(1) = I$). From the theorem, it also follows that an effective way to control the workload is to bound $\mathrm{Var}X_{l,n}$ and minimize the upper bound D_l.

3 AQM Implementation Strategy

3.1 An Example of a Linearized Feedback Flow Control System

An example of a linearized feedback flow control system is studied in [12]. In [12], the actual feedback control system is non-linear (mainly because of the non-linearity of the utility functions). In addition, there are no uncontrollable flows and the available link capacity for controllable flows is fixed. The feedback flow control algorithm used in [12] is called the "optimization flow control algorithm" [2]. It has been shown in [2] that, under the condition that the available link capacity of each link is fixed, the data rate allocation vector will eventually converge to an optimal point or an allocation that satisfies the given fairness criterion. A linear model is used in [12] to study the stability at the optimal equilibrium point. Similarly, in our system with both type of flows, if the available link capacity for controllable flows does not change significantly, the linear model should be a good approximation to the actual system (we will verify this via simulations). Note that in [12], only bottleneck links are considered and all data rates and link capacities are the actual value minus the equilibrium value. For example, $a_l(n)$ in the linear model is in fact the $a_l(n) - \bar{a}_l$ in the actual system, where $\bar{a}_l$ is the equilibrium value of the aggregate input rate of controllable flows to link l. We will use the same notation here. *But in our system, since the available link capacity for controllable flows is time-varying, the data rate allocation vector may never converge and there may be no equilibrium value.* Hence, we will use the mean value instead of the equilibrium value. For example, $\bar{a}_l$ will now be the mean value of $a_l(n)$. Remember that when we return back to the actual system, we need to add the mean value to get the actual value. One exception is the variance (e.g., $\mathrm{Var}X_{l,n}$) because the variance of a random variable dose not change with the addition of a constant. It is also straightforward to check that our main result, Theorem 1, still holds after adding the mean value because

the workload distribution is determined by $\mathrm{Var}X_{l,n}$ for a given utilization.

Our linear model of the optimization flow control system [2] is given as follows. At each link l, feedback information or *price* $p_l(n)$ is calculated.

$$p_l(n) = p_l(n-1) + m_l(a_l(n) - \mu_l(n)), \tag{8}$$

where $m_l > 0$ is a parameter (step size) used in the link algorithm at link l. Note that in [2, 12], since there are no uncontrollable flows in their system, the available link capacity C_l is fixed. But in our system, it is time-varying. Hence, we replace C_l by $\mu_l(n)$ in Eq. (8). Let $p(n) = [p_1(n), \cdots, p_L(n)]^T$ and $\mathbf{p}(z)$ be its Z-transform. We have

$$\mathbf{p}(z) = \frac{M}{1 - z^{-1}}(\mathbf{a}(z) - \boldsymbol{\mu}(z)), \tag{9}$$

where M is an $L \times L$ diagonal matrix, $M = \mathrm{diag}(m_l)$. The price information is fed back to the sources of the controllable flows . Let $r_s(n)$ be the aggregate price of all links used by flow s, $r(n) = [r_1(n), \cdots, r_S(n)]^T$, and $\mathbf{r}(z)$ be the Z-transform of $r(n)$. Then,

$$\mathbf{r}(z) = \mathbf{R_b^T}(z)\mathbf{p}(z), \tag{10}$$

where $\mathbf{R_b}$ is the delayed backward routing matrix. If flow s uses link l, $[\mathbf{R_b}(z)]_{l,s} = z^{-n^b_{s,l}}$ ($n^b_{s,l}$ is the delay from link l to the source of flow s) and 0 otherwise. At each source s, the price $r_s(n)$ is used to calculate the data rate $x_s(n) = -k_s r_s(n)$, where $k_s > 0$ is a constant that depends on the utility function of source s. The minus before k_s means that when the price increases, the data rate will be decreased and vice versa. Let $x(n) = [x_1(n), \cdots, x_S(n)]^T$ and $\mathbf{x}(z)$ be its Z-transform. We have,

$$\mathbf{x}(z) = -K\mathbf{r}(z), \tag{11}$$

where K is a $S \times S$ diagonal matrix, $K = \mathrm{diag}(k_s)$. Finally, the aggregate input rate $\mathbf{a}(z)$ will be

$$\mathbf{a}(z) = \mathbf{R_f}(z)\mathbf{x}(z), \tag{12}$$

where $\mathbf{R_f}$ is the delayed forward matrix, $[\mathbf{R_f}(z)]_{l,s} = z^{-n^f_{s,l}}$ if flow s uses link l ($n^f_{s,l}$ is the delay from source s to link l) and 0 otherwise.

From Eqs. (9)-(12), it is easy to get,

$$\mathbf{a}(z) = -\frac{1}{1-z^{-1}}\mathbf{R_f}(z)K\mathbf{R_b^T}(z)M(\mathbf{a}(z) - \boldsymbol{\mu}(z)).$$

Let

$$\mathbf{G}(z) = \mathbf{R_f}(z)K\mathbf{R_b^T}(z)M, \tag{13}$$

we then have

$$\mathbf{a}(z) = [(1-z^{-1})I + \mathbf{G}(z)]^{-1}\mathbf{G}(z)\boldsymbol{\mu}(z). \tag{14}$$

So, in this example, we have $\mathbf{H}(z) = [(1-z^{-1})I + \mathbf{G}(z)]^{-1}\mathbf{G}(z)$. Note that for the system to be stable [12], $[(1-z^{-1})I + \mathbf{G}(z)]^{-1}$ should exist and be stable. It is also easy to see that when $z = 1$, $\mathbf{H}(1) = I$. Next, we will use this example to show how we can apply our result to effectively control the workload caused by controllable flows.

3.2 Application

The main application of our result is to effectively control the workload caused by controllable flows when there are both controllable and uncontrollable flows. Here, we still consider the linearized feedback flow control system. $\mathbf{H}(z)$ is the feedback control system that we need to design. Of course, $\mathbf{H}(z)$ cannot take an arbitrary form (because of delays, etc). Our goal is to design a feasible $\mathbf{H}(z)$ that satisfies $\mathbf{H}(1) = I$ and also ensure that the resultant queue length is small. Note that the algorithm used to choose $\mathbf{H}(z)$ (or choose the parameters of $\mathbf{H}(z)$) is not the flow control algorithm. The time-scale that it runs over is much larger than the time-scale of the flow control algorithm. There are two approaches to choose $\mathbf{H}(z)$. The first one is to minimize D_l, the upper bound of $\mathrm{Var}X_{n,l}$. The second one is to minimize the tail probability $\mathbb{P}\{Q_l > b_l\}$, which can be viewed as an approximation of the loss probability

$\mathbb{P}_L\{b_l\}$, where b_l is the buffer size of link l. Under the Gaussian assumption, we already have a good upper bound of $\mathbb{P}\{Q_l > b_l\}$ from Theorem 1. Let $\mathbb{P}_u\{b_l\}$ be the upper bound of $\mathbb{P}\{Q_l > b_l\}$, then we could minimize $\mathbb{P}_u\{b_l\}$. Note that the first approach is more general (see Section 4.2), while the second approach may have better performance when the Gaussian assumption holds. When there is only one bottleneck link in the network, examples using the first approach are shown in [10, 18]. In a network with multiple bottleneck links, the problem is more complicated. Each bottleneck link l may have different D_l or different $\mathbb{P}_u\{b_l\}$. In most cases, we cannot minimize all the D_l's or all the $\mathbb{P}_u\{b_l\}$'s at the same time. Hence, we have to set an objective, for example, $\min\max D_l$, or $\min\sum_l D_l$, or $\min\max_l \mathbb{P}_u\{b_l\}$, or $\min\sum_l \mathbb{P}_u\{b_l\}$. In the rest of this paper, we will use $\min\sum_l \mathbb{P}_u\{b_l\}$ as the objective when there are multiple bottleneck links. Next, we will use the example that is discussed in Section 3.1 to show how to design $\mathbf{H}(z)$ in a multiple bottleneck link network.

From Section 3.1, we know that $\mathbf{H}(z) = [(1 - z^{-1})I + \mathbf{G}(z)]^{-1}\mathbf{G}(z)$, where $\mathbf{G}(z) = \mathbf{R_f}(z)K\mathbf{R_b^T}(z)M$. It is easy to show that $\mathbf{H}(1) = I$. Hence, the first condition is satisfied. We also see that $\mathbf{R_f}(z)$ and $\mathbf{R_b}(z)$ are determined by routes and delays and hence cannot be changed. K is dependent on the utility functions which are chosen by the end users and cannot be changed either. So, the only parameters that can be tuned are the elements of matrix M. Remember that M is an $L \times L$ diagonal matrix whose components are given by the step-size parameters m_l in Eq. (8). In [12], m_l corresponds to important AQM (Active Queue Management) parameters. If M is not correctly chosen, the feedback control system may not be stable. Some guidelines are given in [12] on how to choose M to make the system stable. However when there are uncontrollable flows, even if the system is stable, a poor choice of M may result in a large workload. Hence, M needs to be carefully chosen such that not only is the system stable, but also the workload is effectively controlled. We now briefly describe how to choose M if we knew the global information such as $\mathbf{R_f}(z)$, $\mathbf{R_b}(z)$, K, and

the stochastic properties of $\mu(n) = [\mu_1(n), \cdots, \mu_L(n)]^T$. In Section 3.3, we will discuss how to choose M in a distributed way. For any given M, since we know $\mathbf{R_f}(z)$, $\mathbf{R_b}(z)$, and K, we can easily get $\mathbf{G}(z)$ and hence $\mathbf{H}(z)$. From $\mathbf{a}(z) = \mathbf{H}(z)\boldsymbol{\mu}(z)$ and the stochastic properties of $\mu(n)$, we can calculate the stochastic properties of $a(n)$. Since $\mathrm{Var}X_{l,n}$ only depends on the stochastic properties of $a(n)$ and $\mu(n)$, we can calculate $\mathrm{Var}X_{l,n}$ for any l, n, and hence $\mathbb{P}_u\{b_l\}$ for any l and $\sum_l \mathbb{P}_u\{b_l\}$. Next, we can change the value of the matrix M and do the same thing. We can calculate $\sum_l \mathbb{P}_u\{b_l\}$ for all the different M's that we are interested in and choose the M that minimizes $\sum_l \mathbb{P}_u\{b_l\}$. Remember that M is an $L \times L$ matrix. When the number of links is large, this method requires not only global information but also a lot of computation time. Hence, even though the algorithm runs over a much larger time-scale than that of flow control, it may still be not practically viable.

3.3 Distributed Algorithm

From Section 3.2, we know that the feedback flow control parameters (e.g., M) need to be carefully chosen according to the stochastic properties of the uncontrollable flows. The main difficulty is how to choose a good set of parameters in a simple and distributed way. Since different flow control algorithms use different set of parameters, there is no general distributed method to choose parameters. In this section, we will still use the example discussed in Section 3.1 and 3.2. In addition, we assume that $\mu_{l1}(n)$ and $\mu_{l2}(n)$ are independent if $l1 \neq l2$. This assumption is reasonable if most flows that use $l1$ are different from flows that use $l2$ (if this is not true, i.e., most flows that use $l1$ are the same flows that uses $l2$, typically, only one of the links, $l1$ or $l2$ will be a bottleneck link and we can ignore the non-bottleneck link). We also assume that $m_l \ll 1$ for all l. Let $\mathbf{G}_{ij}(z)$ be the element at row i and column j of matrix $\mathbf{G}(z) = \mathbf{R_f}(z)K\mathbf{R_b^T}(z)M$. Since $m_l \ll 1$ for all l, we will ignore all terms with order of m_l^2 or with higher order. Remember that $\mathrm{Var}X_{l,n} = \mathrm{Var}Y_l(n-1)$ and from Eq. (2),

we have,

$$\mathbf{Y}(z) = \frac{1}{1-z^{-1}}(\mathbf{H}(z) - I)\boldsymbol{\mu}(z) = -[(1-z^{-1})I + \mathbf{G}(z)]^{-1}\boldsymbol{\mu}(z).$$

Now, if we ignore all terms with order m_l^2 or higher, we have

$$\mathbf{Y}_l(z) = -\frac{\boldsymbol{\mu}_l(z)}{1 - z^{-1} + \mathbf{G}_{ll}(z)}, \tag{15}$$

where $\mathbf{Y}_l(z)$ and $\boldsymbol{\mu}_l(z)$ are the lth item of $\mathbf{Y}(z)$ and $\boldsymbol{\mu}(z)$ respectively. Since $\mathrm{Var}X_{l,n} = \mathrm{Var}Y_l(n-1)$ and $\mathrm{Var}X_{l,n}$ is the one that determines the workload distribution, from Eq. (15), we know that the queue buildup at link l is caused mainly by the change in the available link capacity at link l itself. Although the available link capacity change at other links also causes queue buildup at link l, it can be ignored compared to the queue buildup caused by link l itself. From Eq. (15), we can also see that now $\mathbb{P}_u\{b_l\}$ will only depend on $\mathbf{G}_{ll}(z)$ and the stochastic properties of $\mu_l(n)$. Hence, each link can consider itself to be the only bottleneck link in the network and choose m_l locally to minimize $\mathbb{P}_u\{b_l\}$. In return, $\sum_l \mathbb{P}_u\{b_l\}$ will also be minimized automatically. Of course, to make the algorithm distributed, we still need to find $\mathbf{G}_{ll}(z)$ and the stochastic properties of $\mu_l(n)$ locally. The stochastic properties of $\mu_l(n)$ (mean, covariance) can be measured locally. The difficulty now is how to obtain $\mathbf{G}_{ll}(z)$. From Eq. (13), the definition of $\mathbf{G}(z)$, we can see that $\mathbf{G}_{ll}(z)$ should have the following form,

$$\mathbf{G}_{ll}(z) = \left(\sum_{i=1}^{N} f_{li} z^{-i}\right) m_l,$$

where N is the maximum delay. Now, our task is to obtain f_{li} for $1 \le i \le N$. From Eqs. (9), (13), and (14), making the same approximation as before, we get

$$\mathbf{a}_l(z) = -\frac{\mathbf{G}_{ll}(z)}{m_l}\mathbf{p}_l(z) + \mathbf{O}_l(z) = -\left(\sum_{i=1}^{N} f_{li} z^{-i}\right)\mathbf{p}_l(z) + \mathbf{O}_l(z), \tag{16}$$

where p_l is the price and O_l is a linear combination of $\mu_{l1}(n)$ for $l1 \neq l$. Writing Eq. (16) in the time domain, we have,

$$a_l(n) = -\sum_{i=1}^{N} f_{li} p_l(n-i) + O_l(n). \tag{17}$$

Now, we multiply $\mu_l(n-1)$ to both sides of Eq. (17) and take expectations. Since $\mu_l(n)$ and $\mu_{l1}(n)$ are independent if $l1 \neq l$ and $O_l(n)$ is a linear combination of $\mu_{l1}(n)$ for $l1 \neq l$, we have

$$\mathrm{Cov}_{a_l,\mu_l}(1) = -\sum_{i=1}^{N} f_{li}\mathrm{Cov}_{p_l,\mu_l}(1-i).$$

Repeating the procedure for $\mu_l(n-2), \cdots, \mu_l(n-N)$, we will have N equations and now we can calculate f_{li}, $1 \leq i \leq N$. Note that $a_l(n)$, $\mu_l(n)$, and $p_l(n)$ are all parameters that can be locally obtained at link l. Hence, we can estimate f_{li} locally at link l.

The following summarizes our distributed algorithm for finding a good set of values of matrix M. We initially set M to be a value such that the feedback control system is stable (e.g., we can follow the guidelines of [12] to find such a M). The initial set of M may not be good in terms of performance (i.e., effectively controlling workload). We then run the flow control algorithm (e.g., optimization flow control algorithm [2] with $\mu_l(n)$ in place of C_l). Each link l can then measure the stochastic properties of $\mu_l(n)$, $a_l(n)$, and $p_l(n)$ and estimate f_{li}, $1 \leq i \leq N$ (or $\mathbf{G}_{ll}(z)$). Once $\mathbf{G}_{ll}(z)$ and the stochastic properties of $\mu_l(n)$ are known, from Eq. (15) and Theorem 1, we can calculate $\mathbb{P}_u\{b_l\}$ for any given m_l. We can then find the value of m_l that minimizes $\mathbb{P}_u\{b_l\}$ and set that value to m_l. Note that our distributed algorithm is not the flow control algorithm (in our example, the flow control algorithm is the optimization flow control algorithm [2]). It is the algorithm to find a good set of parameters for the flow control algorithm. Once the network configuration (K, $\mathbf{R_f}(z)$, $\mathbf{R_b}(z)$, and the stochastic properties of $\mu_l(n)$ here) do not change significantly, the set of parameters that are chosen by our algorithm will keep working well. Hence, this algorithm does not need to run on the time-scale of the flow control algorithm. It only needs to run on the time-scale of changes in the network configuration.

4 Discussion

4.1 Simplified Flow Control Model

The flow control model (Fig. 1) that we have described in Section 2 can be significantly simplified if $v_l(n) \leq \rho_l C_l$ for all n. Under this condition, $\mathbf{q}_l^u$ will always be empty and $\mu_l(n) = \rho_l C_l - v_l(n)$. The available link capacity for controllable flows will be $C_l - v_l(n)$. This simplified model has been widely used [6, 7, 9]. However we should keep in mind the requirement $v_l(n) \leq \rho_l C_l$ for all n, otherwise, the available link capacity calculated by $C_l - v_l(n)$ may be negative. An interesting property of this simplified model is that the workload $q_l(n)$ will be the same regardless of whether the uncontrollable flows are given a higher priority than controllable flows or not. If the uncontrollable flows are given a higher priority, $\mathbf{q}_l^u$ will always be empty and $q_l(n) = q_l^c(n)$. Since the input rate to $\mathbf{q}_l^c$ is $a_l(n)$ and the available link capacity is $C_l - v_l(n)$, we will have

$$q_l(n) = [q_l(n-1) + a_l(n) + v_l(n) - C_l]^+. \tag{18}$$

If the uncontrollable flows are not given high priority, the input rate to $\mathbf{q}_l$ will be $a_l(n)+v_l(n)$ and the link capacity will be C_l. The workload $q_l(n)$ will take the exact same form as in Eq. (18). So, the workload will be the same in either case. This property makes the simplified model suitable for a TCP network where the only uncontrollable flows are short-lived TCP flows. Internet traffic measurement shows that although a major fraction of TCP flows are short-lived, the total bandwidth utilized by those short-lived TCP flows is in fact quite small compared to the total link capacity. Hence, it is reasonable to assume that $v_l(n) < \rho_l C_l$ and to use the simplified model. In a real TCP network, short-lived TCP flows will have the same priority as other TCP flows. But our analytical results will still hold because, as we have shown, the workload is not affected by whether the short-lived TCP flows are given

high priority or not. Although our analysis does not require this simplification, this model appears reasonable and useful in the context of TCP traffic control. In our simulation of TCP networks, we use this simplified model.

4.2 Non-Gaussian Process

Theorem 1 is based on a Gaussian assumption on the available capacity. However, our first approach to control the workload (i.e., bound $\mathrm{Var}X_{l,n}$ and minimize the upper bound D_l) is general and we expect it to perform well even when $\mu(n)$ is not Gaussian. The not-so-rigorous explanation is as follows. From Eq. (4), we know that

$$\mathbb{P}\left\{Q_l^c > x\right\} = \mathbb{P}\left\{\sup_{n\geq 0} X_{l,n} > x\right\}.$$

Let $n_{l,x}$ be the time at which $\mathbb{P}\{X_{l,n} > x\}$ attain its maximum value. Then it is well known that a good lower bound approximation to $\mathbb{P}\{Q_l^c > x\}$ is $\mathbb{P}\{X_{l,n_{l,x}} > x\}$ [15]. Since $\mathbb{E}X_{l,n_{l,x}} = -k_l n_{l,x}$ where k_l is fixed once the link utilization is fixed, we expect that if we can make the variance of $X_{l,n_{l,x}}$ smaller, $\mathbb{P}\{X_{l,n_{l,x}} > x\}$ and hence $\mathbb{P}\{Q_l^c > x\}$ will also be smaller. Since we know that $\mathrm{Var}X_{l,n}$ can be bounded, if we can minimize the upper bound D_l, we should be able to effectively control the workload.

5 Numerical Results

From Theorem 1, we can see that when x is large, we have $\Psi(\sqrt{x/\sigma_{l,x}^2}) \leq \mathbb{P}\{Q_l^c > x\} \leq e^{-\frac{x}{2\sigma_{l,x}^2}}$. Hence, $\Psi(\sqrt{x/\sigma_{l,x}^2})$ is a lower bound on $\mathbb{P}\{Q_l^c > x\}$, while $e^{-\frac{x}{2\sigma_{l,x}^2}}$ is an asymptotic upper bound, which we will call the MVA bound following the terminology used in [16]. Note that all simulations here are all Monte-Carlo except for the last one, which uses an *ns2* simulator. The duration of each simulation (except for the *ns2* simulation) is around 1 hour

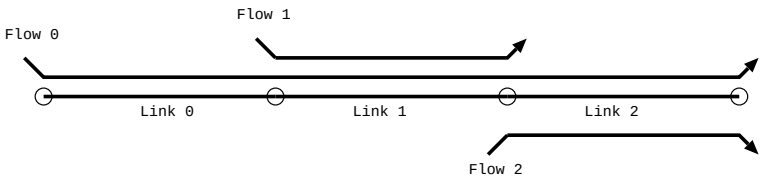

Figure 2: A Network with 3 Links

and each step in the simulation is one time unit, which is $1msec$. The duration of the *ns2* simulation is around 10 minutes.

We first simulate a network with three links (as shown in Fig. 2). The link capacities of link 0, 1, 2 are $500Mbps$, $200Mbps$, and $400Mbps$, respectively. The propagation delay of the three links are $1msec$, $2msec$, and $3msec$, respectively. There is no uncontrollable traffic on link 0. The aggregate input of uncontrollable flows to link 1 is a Gaussian process generated by using the following ARMA model

$$\begin{aligned} v(n) &= 2.889v(n-1) - 2.77911v(n-2) + 0.890109v(n-3) \\ &\quad +0.01x(n) - 0.0197325x(n-1) + 0.0097335x(n-2), \end{aligned} \tag{19}$$

where $x(n)$ are $i.i.d$ Gaussian random variables with mean $100Mbps$ and variance $10^4 Mbps^2$. It is easy to see that the mean rate of aggregate uncontrollable flows is $100Mbps$. The uncontrollable flows to link 2 are 2000 voice flows. Each individual voice flow is modeled by a Markov modulated On-Off fluid process. The state transition matrix and rate vector are given as follows.

$$\text{State transition matrix : } \begin{bmatrix} 0.9833 & 0.01677 \\ 0.025 & 0.975 \end{bmatrix}$$

$$\text{Input rate vector : } \begin{bmatrix} 0 \\ 0.425 \end{bmatrix},$$

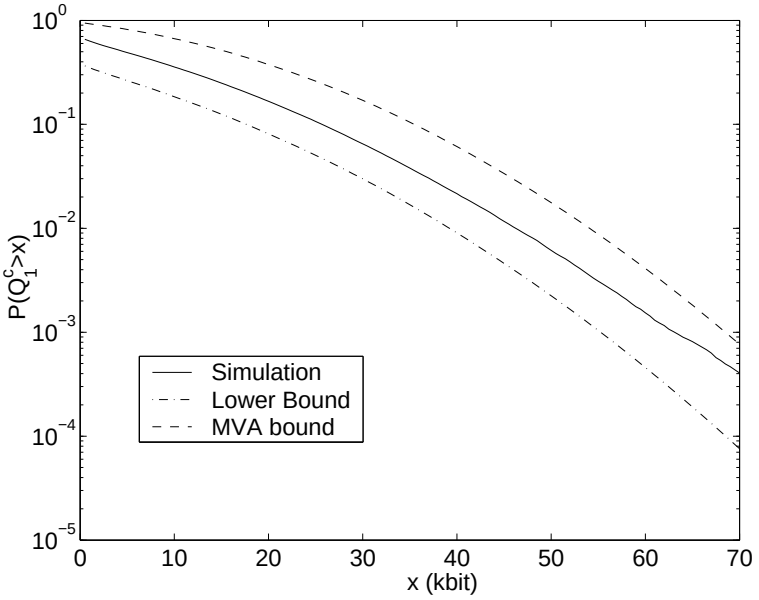

Figure 3: Tail probability at link 1

where the unit of the input rate is $Mbps$. We can show that the mean rate of aggregate voice flows is $341Mbps$. There are three controllable flows. Flow 0 uses all links. Flow 1 uses only link 1. Flow 2 uses only link 2. It is easy to see that link 0 is not a bottleneck link. For the two bottleneck links, link 1 and link 2, the target utilizations are set to 99.5% and 98% respectively. We use the modified version of the optimization flow control algorithm [2] described in Section 3 to control the controllable flows. The utility functions used are the same as the one suggested in [12]. The utility function for flow i is

$$U_i(x) = \frac{M_i \tau_i}{\alpha_i} x \left[1 - \log \frac{x}{x_{max,i}}\right], \quad \text{for } x \leq x_{max,i}, \tag{20}$$

where x is the data rate of flow i, M_i is the upper bound on the number of bottleneck links in the path of flow i, τ_i is the round trip delay of flow i, α_i is a constant between 0 and 1, and $x_{max,i}$ is the maximum rate of flow i. In this simulation, we set $M_i = 2$ and $\alpha_i = 0.9$ for all controllable flows. We also set $x_{max,0} = x_{max,2} = 100Mbps$ and $x_{max,1} = 200Mbps$.

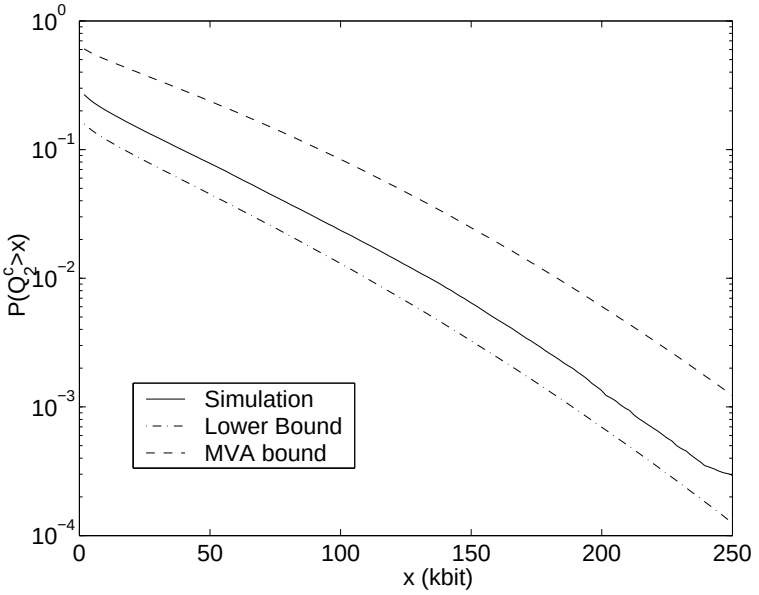

Figure 4: Tail probability at link 2

We first set the AQM parameters $m_1 = m_2 = 0.048$. Our simulation results are shown in Figs. 3 and 4. We can see that the lower bound and the MVA upper bound accurately characterize the tail probability for both bottleneck links (even when x is small). In Fig. 4, we can also see that although each voice flow traffic is not Gaussian, the aggregate traffic can be modeled quite accurately by a Gaussian process.

Next, we will show how the AQM parameters (m_1 and m_2 here) can affect the performance of the feedback flow control, and how to choose these AQM parameters. We compare three sets of AQM parameters. In the first set, we follow the guidelines in [12] and set $m_1 = m_2 = 0.005$. In the second set, we assume that we have global information. With the method discussed in Section 3.2, we get $m_1 = 0.055$ and $m_2 = 0.05$. In the third set, each link only has local information. We use the distributed method discussed in Section 3.3 and obtain $m_1 = 0.04$ and $m_2 = 0.05$. Our simulation results are shown in Figs. 5 and 6.

With all three sets of parameters, the measured bottleneck link utilization is the same as

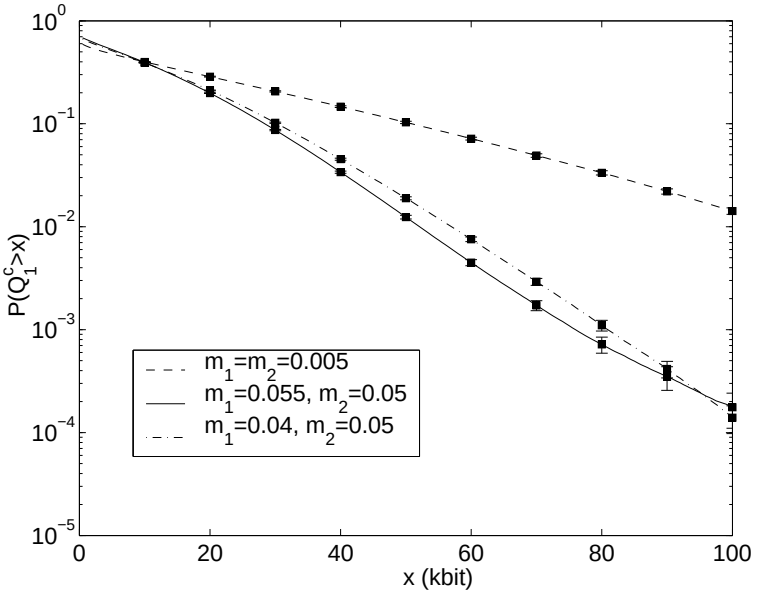

Figure 5: Tail probability at link 1

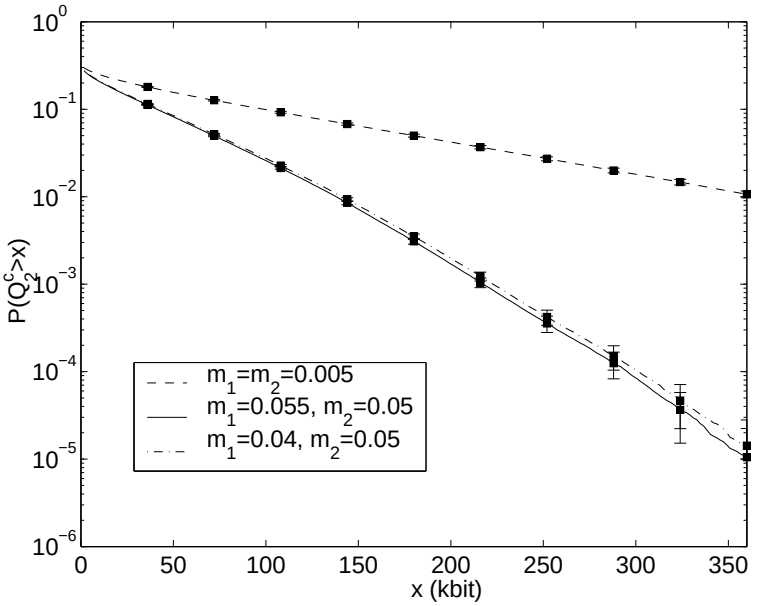

Figure 6: Tail probability at link 2

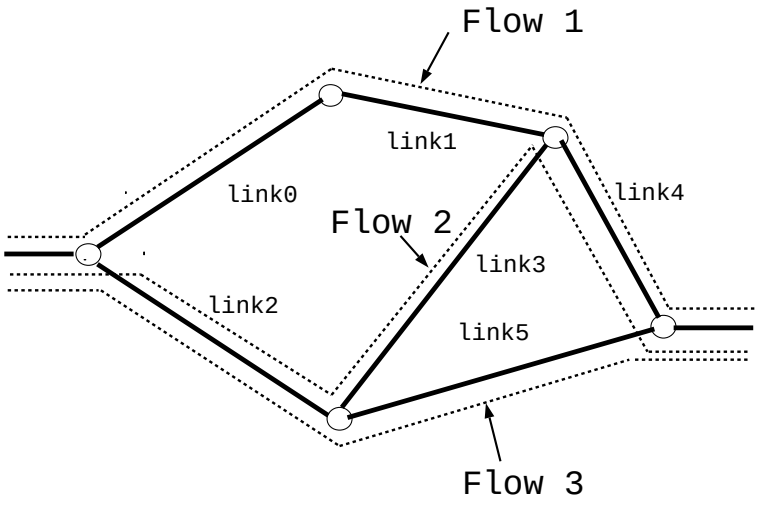

Figure 7: A Network with 6 Links

the target link utilization (99.5% and 98% for link 1 and 2 respectively). From Figs. 5 and 6, we can see that choosing M correctly is important to the performance of the feedback flow control. When the elements of M are properly chosen, the workload can be significantly reduced. We also see that when the parameter M is designed with only local information, the system performance is close to the case when the parameter is designed using global information.

We next simulate a more complicated network with six bottleneck links (Fig 7). For each link, the link capacity is $200Mbps$, the propagation delay is $2msec$, and the target link utilization is set to be 98%. Besides the three long flows shown in Fig 7, there is a short flow (not shown in Fig 7) for each link. So, totally we have nine controllable flows. The utility function for each flow is the same as Eq. (20) and we set $M_i = 3$, $\alpha_i = 0.9$ for all flows. For the six short flows, $x_{max,i} = 200Mbps$ and for the three long flows, $x_{max,i} = 100Mbps$. There are also uncontrollable flows at each link. The aggregate input rate of uncontrollable flows at each link has the same distribution and they are independent of each other. For a given

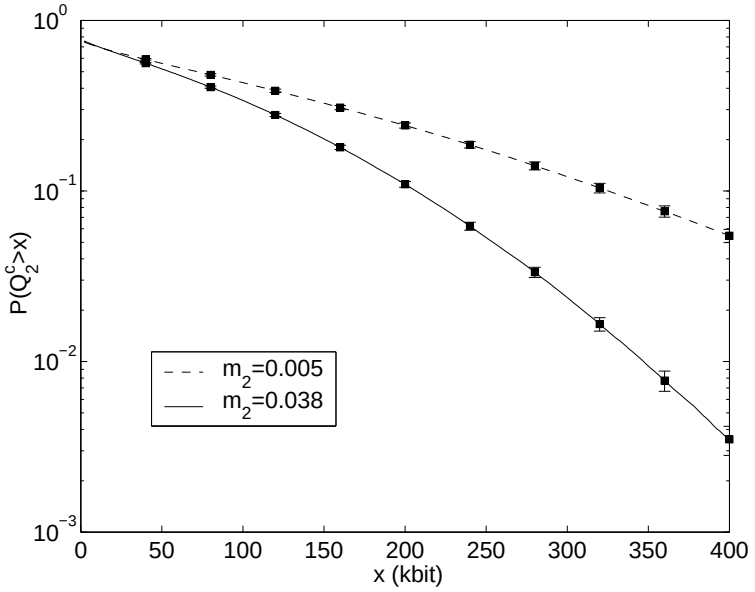

Figure 8: Tail probability at link 2

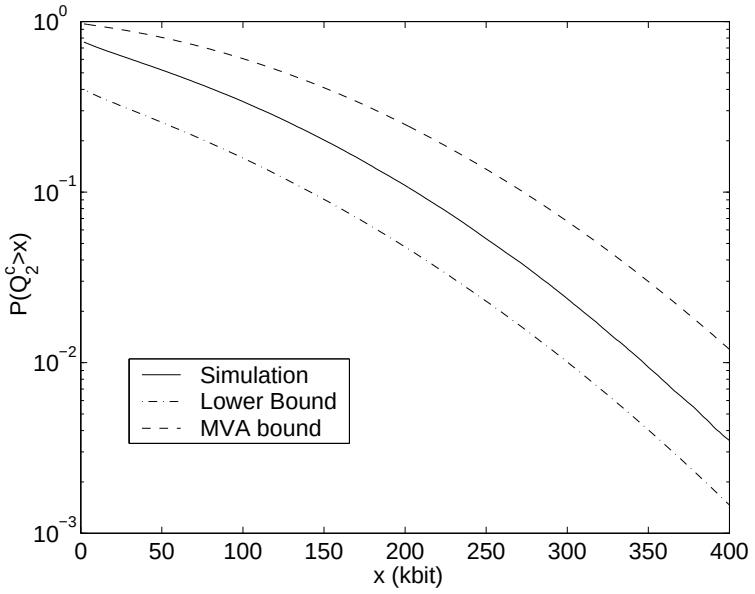

Figure 9: Tail probability at link 2

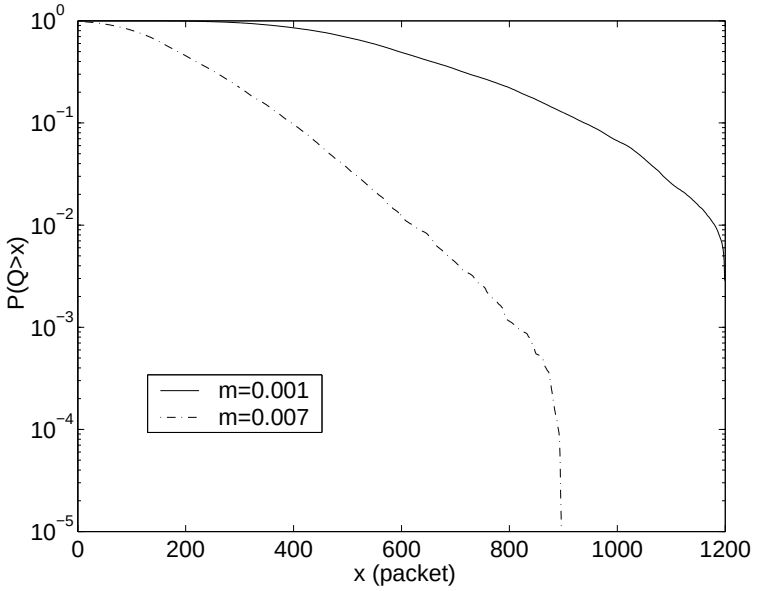

Figure 10: NS simulation: tail probability

link l, the aggregate input rate of uncontrollable flow is generated by

$$v_l(n) = 0.8v_l(n-1) + 0.2x_l(n),$$

where $x_l(n)$ are $i.i.d$ Gaussian random variables with mean $100Mbps$ and variance $2500Mbps^2$. We first follow the guidelines in [12] and set $m_l = 0.005$ for all links. Without loss of generality, we then pick link 2 and choose $m_2 = 0.038$ by using the distributed method discussed in Section 3.3 (note that m_l is still 0.005 for other links). The simulation result is shown in Fig 8. From Fig 8, we see that in a network with multiple links, even if the other links do not use our algorithm, an individual link can still improve its performance by using our distributed method to correctly choose the AQM parameter. Hence, our algorithm can be deployed incrementally in a large network. In Fig 9, we show the lower bound and MVA upper bound of the tail probability when $m_2 = 0.038$. Again, we see that the bounds accurately characterize the tail probability.

In the next simulation, we use the *ns2* simulator. We simulate one bottleneck link in

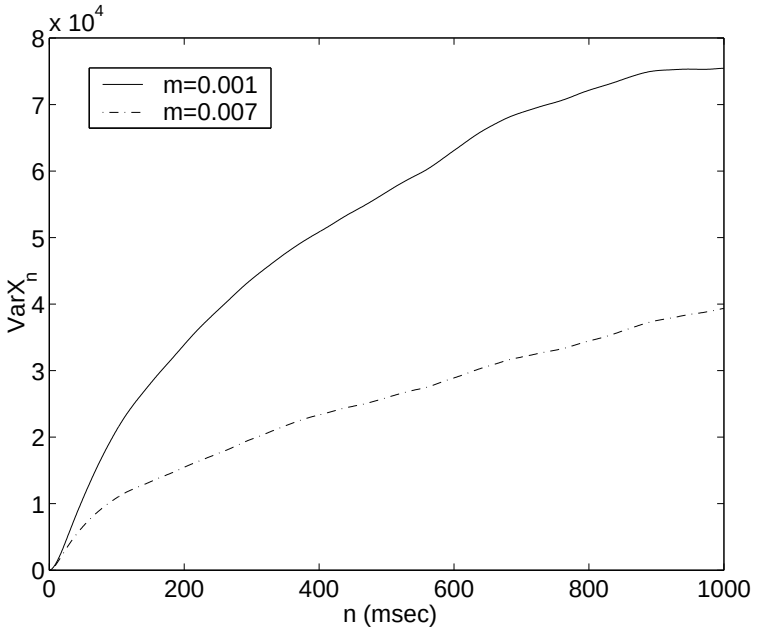

Figure 11: NS simulation: VarX_n

the network. The link capacity of the bottleneck link is $200Mbps$. The mean rate of the aggregate uncontrollable flows is $100Mbps$. The uncontrollable traffic is generated by using Eq. (19) and is carried by UDP packets. There are 100 TCP flows (controllable flows). The round trip time of each TCP flow is $10msec$. The router uses a modified REM algorithm in which the price is calculated by $p(n+1) = [p(n) + m(a(n) + v(n) - \rho C)]^+$. The target link utilization is $\rho = 96\%$ and the buffer size is 1200 packets. Each packet is 1000bytes. Our simulation results are shown in Figs. 10 and 11. From Fig. 10, we can see that with different REM parameters, the queue distribution can be quite different. Because TCP uses an AIMD type of window-based flow control, the MVA bound (derived from the fluid model) does not capture the tail probability as well any more. However VarX_n is still important for the queue distribution. To calculate VarX_n, we use $1msec$ as the time unit and count the number of arrival TCP packets and UDP packets in each time unit. They are used as the

aggregate input rate of controllable flows and uncontrollable flows respectively (the unit is $packets/msec$). The variance of X_n can then be calculated numerically by definition Eq. (4). In Fig. 11, we show the corresponding $\mathrm{Var}X_n$. We can see that the smaller the $\mathrm{Var}X_n$ (in the case $m = 0.007$), the smaller the queue length. Hence, in a TCP network, our first approach described in Section 3.2 (minimizing $\mathrm{Var}X_n$) is still an effective way to control the loss rate.

6 Conclusion

In this paper, we consider feedback flow control systems with both uncontrollable and controllable flows. We give uncontrollable flows high priority and focus on the workload that is caused by the controllable flows. We assume that the feedback control system is linear and find that, under certain conditions, the variance of the net input over a given time period can be bounded by a constant (not dependent on the length of the time period). We then analyze the queueing properties under a Gaussian assumption and derive a lower bound and an asymptotic upper bound for the tail probability of the workload. Our simulations show that these bounds are quite accurate when the aggregate traffic can be approximated by a Gaussian process. We also discuss how to apply our result to a network with multiple bottleneck links and how to find appropriate flow control parameters in a distributed way to effectively control the workload.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, NJ, 1992.

[2] S. H. Low and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–875,

Dec. 1999.

[3] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, pp. 237–252, March 1998.

[4] H. Yaiche, R. R. Mazumdar, and C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, Oct. 2000.

[5] E. Altman, F. Baccelli, and J. Bolot, "Discrete-Time Analysis of Adaptive Rate Control Mechanisms," *High Speed Networks and Their Performance (C-21)*, pp. 121–140, 1994.

[6] E. Altman, T. Basar, and R. Srikant, "Congestion Control as a Stochastic Control Problem with Action Delays," *Automatica (Special issue on Control Methods for Communication Networks)*, Dec 1999.

[7] L. Benmohamed and S. Meerkov, "Feedback Control of Congestion in Store-and-Forward Networks: The case of Single Congestion Node," *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, pp. 693–798, Dec. 1993.

[8] R. Jain, S. Kalyanaraman, and R. Viswandathan, "The OSU Scheme for Congestion Avoidance using Explicit Rate Indication," Tech. Rep., OSU, Sept. 1994.

[9] Y. D. Zhao, S. Q. Li, and S. Sigarto, "A linear dynamic model design of stable explicit-rate ABR control scheme," in *Proceedings of IEEE INFOCOM*, 1997, pp. 283–292.

[10] D. Qiu and N. B. Shroff, "A New Predictive Flow Control Scheme for Efficient Network Utilization and QoS," in *Proceedings of ACM SIGMETRICS*, 2001.

[11] S. Q. Li, S. Chong, and C. Hwang, "Link Capacity Allocation and Network Control by Filtered Input Rate in High Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 1, pp. 10–15, Feb. 1995.

[12] F. Paganini, J. Doyle, and S. Low, "Scalable Laws for Stable Network Congestion Control," in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, vol. 1.

[13] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, Prentice Hall, NJ, 1997.

[14] R. M. Loynes, "The Stability of a Queue with Non-independent Inter-arrival and Service Times," *Proc. Cambridge Philos. Soc.*, vol. 58, pp. 497–520, 1962.

[15] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," *Studies in Applied Probability*, pp. 131–155, 1994.

[16] J. Choe and N. B. Shroff, "Use of the supremum distribution of Gaussian processes in queueing analysis with long-range dependence and self-similarity," *Stochastic Models*, vol. 16, no. 2, Feb 2000.

[17] W. Feller, *An Introduction to Probability Theory and its Applications I*, John Wiley & Son, New York, 1968.

[18] D. Qiu and N. B. Shroff, "Study of Predictive Flow Control," Tech. Rep., Purdue University, May 2001, http://www.ifp.uiuc.edu/∼dqiu/paper/techrep.ps.