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 those applications with stringent QoS requirements and are given a 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 guides on how to design a feedback flow control system. Simulation results show that the lower bound and asymptotic upper bound are in fact quite accurate and that our feedback control method can effectively control the queue length in the presence of both controllable and uncontrollable traffic.

1 Introduction

In communication networks, we can classify flows as controllable and uncontrollable flows. The controllable flows can adjust their data rates according to the feedback information 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 typically 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 major 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 typically generated from applications with QoS requirements. Hence, when there are only uncontrollable flows in the networks, queueing analysis is especially important because QoS requirements such as loss probability, queueing delay, and jitter are directly related to the queue length distribution. On the other hand, when

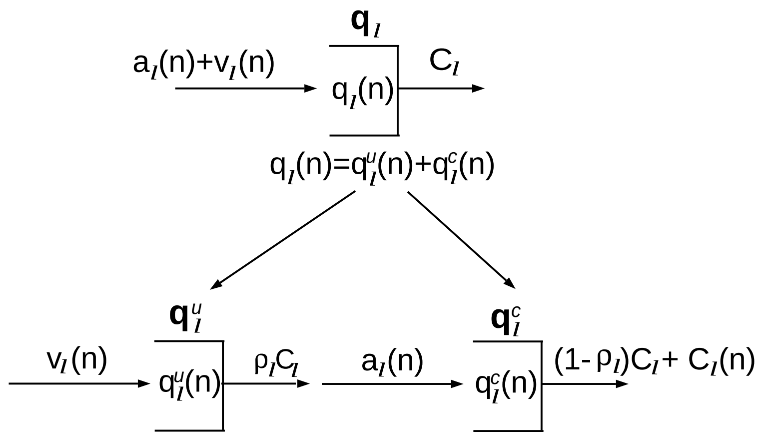

Figure 1: Model of a Single Link l

there are only controllable flows in the network, 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 proved 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. Queueing is not an important issue in this case because all the traffic is assumed to be controllable. 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 mainly high link utilization, low loss probability, and fairness [5, 6, 7, 8]. 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 are mainly on the flow control algorithm and do not give us 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. We then analyze the queueing behavior of such a system. Our main results can be applied to both single-link and multi-link cases. We believe that the queueing analysis of such a system is important because it can guide us on how to design the feedback control system. We will then briefly discuss how our control strategy can be used within the confines of AQM (Active Queue Management) and provide some numerical results.

2 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 analysis, we consider an infinite buffer discrete-time fluid queueing model. Let $a_l(n)$ be the aggregate input rate of controllable flows and $v_l(n)$ be the aggregate input rate of 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 introduce our general flow control model. (in Section 3.1, 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 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. $\mathbf{q}_l^u$ 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,

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

Then, $C_l(n)$ is the residual link capacity in $\mathbf{q}_l^u$. Therefore, $C_l(n)+(1-\rho_l)C_l$ 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 $a_l(n) = C_l(n)$ 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 $C_l(n)$. We define $\mathbf{q}_l^c$ as the queueing system with $a_l(n)$ as input and $C_l(n) + (1 - \rho_l)C_l$ as the link capacity. Then $q_l^c(n)$ is the workload caused by the controllable flows and is the one 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 uncontrollable flows to link l, by a stationary stochastic process. Then $C_l(n)$ is a time-varying (but stationary) process and will be used to control the data rates of the controllable flows. Let $C(n) = [C_1(n), \cdots, C_L(n)]^T$ 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 $C(n)$) or can be approximately modeled as a linear system (an example will be discussed in Section 3.3). A linear feedback control has been found to give good results when we have video traffic as uncontrollable traffic [7, 9], and we find that it gives good results for other types of traffic as well [8]. Note that we only assume that the feedback control is linear. All queueing systems considered here are non-linear. Let $\mathbf{C}(z)$ and $\mathbf{a}(z)$ be the Z-transforms of $C(n)$

and $a(n)$ respectively. We have

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

where $\mathbf{H}(z)$ is an $L \times L$ matrix and represents a causal, stable, linear, time-invariant system [10]. 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 $a(n) = C(n-5)$ and $\mathbf{H}(z) = z^{-5}$. Next, we will see how to design $\mathbf{H}(z)$ to achieve desirable properties for the controlled queueing system $\mathbf{q}_l^c$. Because of page limitations, we omit all proofs in this paper. Interested readers are referred to our technical report [11].

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

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 $q_l^c(n) \le C_q$ for all n and all l.*

Proposition 2 tells us that $q_l^c(n)$ can be bounded by a constant (independent of n) when $\mathbf{H}(z)$ is chosen appropriately. However this may not be sufficient 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 $C(n)$ is stationary and $a(n)$ is a linear transformation of $C(n)$, $a(n)$ is also stationary. The steady state workload distribution of q_l^c will be given by [12] [13]

$$\mathbb{P}\{Q_l^c > x\} = \mathbb{P}\left\{\sup_{n\ge 0} X_{l,n} > x\right\}, \tag{2}$$

where $X_{l,n} = \sum_{j=-n+1}^{0}(a_l(j) - C_l(j) - (1-\rho_l)C_l)$.

From Eq. (2), it follows that the stochastic properties of $X_{l,n}$ will directly affect the workload distribution. If $\mathbf{H}(1) = I$, $\mathbb{E}\{a_l(j)\} = \mathbb{E}\{C_l(j)\}$. 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 $k_l = (1-\rho_l)C_l$ (note that $X_{l,n}$ is the sum of the aggregate input rates minus the sum of the available link capacities from time $-n$ to 0 at link l). In [14], it has been shown that when k_l is fixed, $\mathrm{Var}X_{l,n}$ plays an important role in the queue distribution. In general, when n go to infinity, $\mathrm{Var}X_{l,n}$ will also goes 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 can 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$\{C_l(n)\}$ is finite for all l, then for each link l, there exists a constant D_l such that Var$X_{l,n} \le D_l$ for all n.*

Note that in practice, since $0 \le C_l(n) \le C_l$, $\mathrm{Var}\{C_l(n)\}$ will always be finite. But in Lemma 1, we only require $\mathrm{Var}\{C_l(n)\}$ to be finite and do not require $C_l(n)$ to be bounded. This will be useful when we model $C(n)$ 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 $C(n)$ 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, $C_l(n)$ is the residual link capacity in $\mathbf{q}_l^u$ and hence it is approximately $\rho_l C_l - v_l(n)$ (it is exactly $\rho_l C_l - v_l(n)$, if $v_l(n) < \rho_l C_l$ for all n). Since $v_l(n)$ is the aggregate input rate of uncontrollable flows, it can be well modeled by a Gaussian process and hence $C_l(n)$ can also be modeled by a Gaussian process (we will also justify this approximation numerically in Section 4). Now, if $C(n)$ 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., $\Psi(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{z^2}{2}} \mathrm{d}z$.

Let $n_{l,x} = \mathrm{argmin}_n \frac{x+k_l n}{\sqrt{V_{l,n}}}$. Then $n_{l,x}$ is the time where $\mathbb{P}\left\{X_{l,n} > x\right\}$ attains its maximum value, i.e., $n_{l,x}$ is the *dominant time scale*. Let $\sigma_{l,x}^2 = \frac{x V_{n_{l,x}}}{(x+k_l n_{l,x})^2}$. It has been shown in [14] that the tail of the workload distribution can be well approximated by $e^{-\frac{x}{2\sigma_{l,x}^2}}$ if $V_{l,n} \sim S n^\beta$ when n is large, where $\beta \in [1,2)$ and S is a constant. In our case, however, $V_{l,n}$ can be bounded as shown in Lemma 1. Hence, we would like to determine the behavior of $\mathbb{P}\left\{Q_l^c > x\right\}$ when $V_{l,n}$ is bounded. This behavior is described by Theorem 1 stated below.

Theorem 1 *If $D_l = \sup_{n\geq 0} V_{l,n}$ is finite for link l, then,*

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

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

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

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 of $\mathbb{P}\left\{Q_l^c > x\right\}$ are quite 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 in [14], $V_{l,n} \sim S n^\beta$ for $\beta \in [1,2)$, when x is large, the tail probability will decrease only on the order of $e^{-bx^{2-\beta}}$, 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 Discussion

3.1 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 $C_l(n) = \rho_l C_l - v_l(n)$. The available link capacity for controllable flows will be $C_l - v_l(n)$. This simplified model has been widely used [5, 6, 7]. 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 high priority or not. If the uncontrollable flows are given a high 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

$$q_l(n) = [q_l(n-1) + a_l(n) + v_l(n) - C_l]^+. \tag{3}$$

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. (3). So, the workload will not change whether the uncontrollable flows are given high priority or not. 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 be true because, as we have shown, that the workload will not be 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.

3.2 Non-Gaussian Process

Theorem 1 is based on a Gaussian assumption on the available capacity. However, our method 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 $C(n)$ is not Gaussian. The not-so-rigorous explanation is as follows. From Eq. (2), we know that

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

Let $n_{l,x}$ be the time at which $\mathbb{P}\{X_n > x\}$ attain its maximum value. Then it is well known that a good lower bound approximation to $\mathbb{P}\{Q_l^c > x\}$ is $\mathbb{P}\{X_{n_{l,x}} > x\}$. Since $\mathbb{E}X_{n_{l,x}} = -k_l n_{l,x}$ where k_l is fixed once the link utilization is fixed, we expect that if we can make the variance of $X_{n_{l,x}}$ smaller, $\mathbb{P}\{X_{n_{l,x}} > x\}$ 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.

3.3 AQM Implementation Strategy

An example of a linearized feedback flow control system is studied in [15]. In [15], there are no uncontrollable flows and the available link capacity for controllable flows is fixed.

The feedback flow control algorithm used in [15] is the so-called 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. A linear model is used in [15] to study the stability at the optimal equilibrium point. Similarly, in our system, if the available link capacity for controllable flows does not change significantly, the linear model should be a good approximation to the real system (we will verify this in the simulations). Note that in [15], 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 real 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 notations 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 map back to the real system, we need to add the mean value back. Note that our main result (Theorem 1) will still hold because the workload distribution is determined by $\mathrm{Var}X_{l,n}$ and the variance of a random variable does not change when a constant is added to the random variable.

In our simulations (Section 4), we will use the similar feedback flow control system as in [15] (note that, however, our method to control workload can be applied to any linear or linearized feedback flow control system). The only difference is that in [15], there are no uncontrollable flows and the available link capacity C_l is fixed. But in our system, the available link capacity for controllable flows is time-varying. Hence, we replace C_l with $C_l(n)$ when we calculate the feedback information (price) $p_l(n)$ at link l.

$$p_l(n) = p_l(n-1) + m_l(a_l(n) - C_l(n)), \tag{4}$$

where $m_l > 0$ is a parameter (step size) used in the link algorithm at link l and hence, also a parameter in the linear system $\mathbf{H}(z)$. Our goal is to effectively control the workload caused by the controllable flows. $\mathbf{H}(z)$ is the feedback control system that we need to design. Of course, $\mathbf{H}(z)$ cannot take an arbitrary form (because of delays, etc). In this example, we will tune m_l to modify $\mathbf{H}(z)$. In [15], m_l corresponds to important AQM parameters. If m_l is not correctly chosen, the feedback control system may not be stable. Some guidelines are given in [15] on how to choose m_l to make the system stable. However when there are uncontrollable flows, even if the feedback control system is stable, a poor choice of m_l may cause a large workload. Hence, the value of m_l should be carefully chosen such that not only is the system stable, but also the workload is effectively controlled. The value of m_l can be chosen in both a centralized and a distributed way. Because of page limitations, we will not discuss how to choose m_l here. The simulation results will be shown in Section 4. Note that the algorithm to choose m_l (or choose $\mathbf{H}(z)$) 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. Hence, this algorithm can run over a much larger (slower) time-scale than the flow control mechanism.

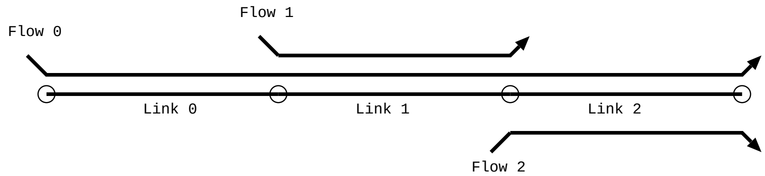

Figure 2: A Multi-Link Network

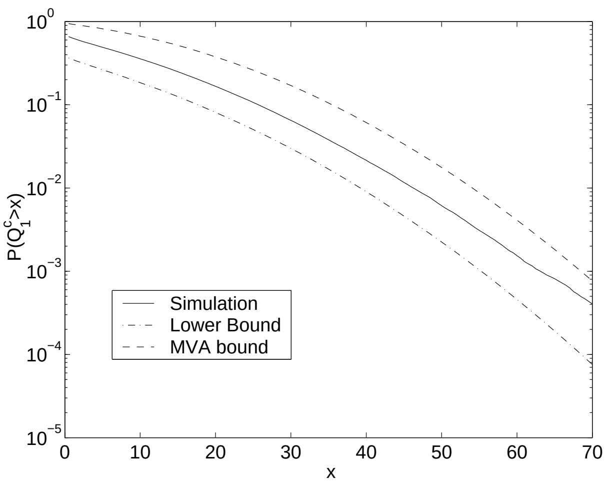

Figure 3: Tail probability at link 1

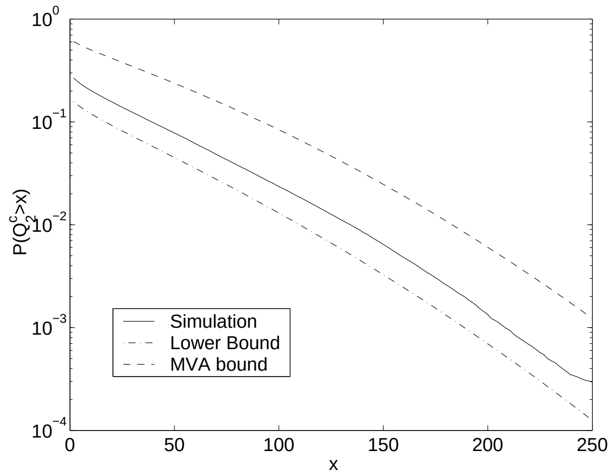

Figure 4: Tail probability at link 2

4 Numerical Results

From Theorem 1, we can see that when x is large, we have $\Psi(\sqrt{x/\sigma_{l,x}^2}) \leq \mathbb{P}\{Q_l^c > x\} \leq e^{-\frac{x}{2\sigma_{l,x}^2}}$. We call $\Psi(\sqrt{x/\sigma_{l,x}^2})$ the lower bound of $\mathbb{P}\{Q_l^c > x\}$ and $e^{-\frac{x}{2\sigma_{l,x}^2}}$ the MVA upper bound (following the abbreviation used in [14]). In our simulations, we consider a network with three links (as shown in Fig. 2). The link capacities of links 0, 1, 2 are 500, 200, and 400, respectively. The propagation delays of the three links are 1, 2, and 3, respectively. There is no uncontrollable traffic on link 0. The aggregate input traffic of uncontrollable flows to link 1 is a Gaussian process with mean 100. The uncontrollable flows to link 2 correspond to 3000 voice flows and the mean rate of aggregate traffic is 341. 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, the target utilizations are set to 99.5% and 98% respectively. We use the modified optimization flow control algorithm [2] described in Section 3.3 to control the controllable flows. The utility functions used are the same as the one suggested in [15].

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 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 well 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. We compare three sets of AQM parameters. In the first set, we follow the guidelines in [15] and set $m_1 = m_2 = 0.005$. In the second set, we choose m_l in a centralized way and

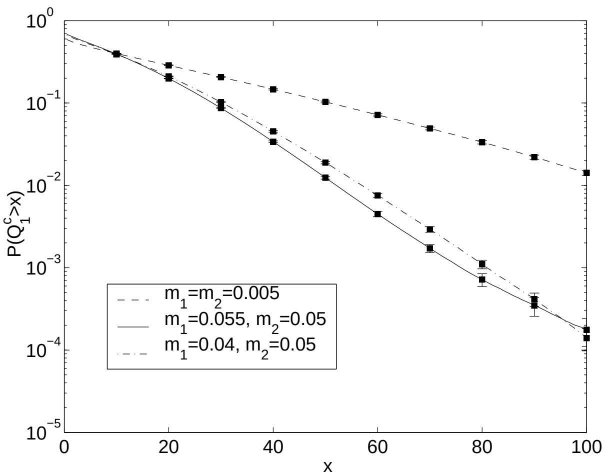

Figure 5: Tail probability at link 1

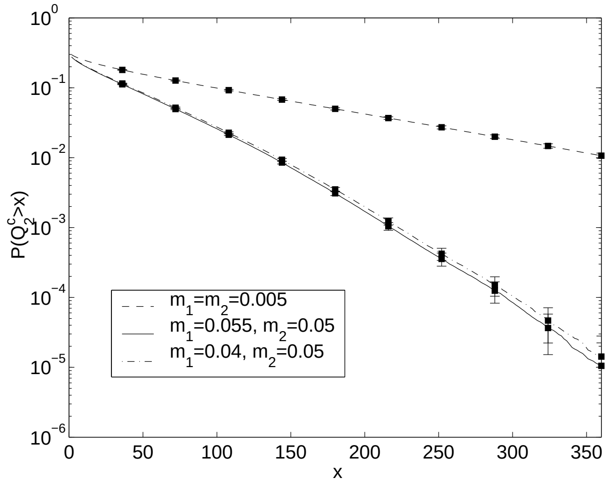

Figure 6: Tail probability at link 2

obtain $m_1 = 0.055$ and $m_2 = 0.05$. In the third set, we use a distributed method 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 the target link utilization (99.5% and 98% for link 1 and 2 respectively). From Figs. 5 and 6, we can see that m_l is important to the performance of the feedback flow control. When m_l is properly chosen, the workload can be significantly reduced. We also see that the parameter m_l designed with only local information (distributed way) gives very similar performance as the parameter designed with global information (centralized way).

5 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 the Gaussian assumption and give a lower bound and asymptotic upper bound for the tail probability of the workload. The upper bound also tells us that when a flow control algorithm is appropriately designed for this system, the tail of the workload distribution caused by the controllable traffic can be made to decay very rapidly in terms of the buffer size (squared exponential decay). Our simulations also show that these bounds are quite accurate. We also briefly discuss how to apply our result to a network with multiple bottleneck links 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, 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, 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.

[6] L. Benmohamed and S. Meekov, "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.

[7] 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.

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

[9] 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.

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

[11] D. Qiu and N. B. Shroff, "Queueing Properties of Feedback Flow Control Systems," Tech. Rep., Purdue University, West Lafayette, IN, Jul. 2002, http://min.ecn.purdue.edu/∼dongyu/paper/techrep02a.ps.

[12] 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.

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

[14] 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.

[15] 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.