Dynamical Behavior of Rate-Based Flow Control Mechanisms[1]

Jean-Chrysostome Bolot
A. Udaya Shankar[2]

Department of Computer Science
University of Maryland
College Park, Maryland 20742

bolot@cs.umd.edu, shankar@cs.umd.edu

Abstract

Flow control mechanisms are essential for the efficient and stable operation of store-and-forward networks. New transport protocols, such as VMTP and NETBLT, intend to use rate-based flow control. We present a model in which a network with connections subject to rate-based flow control is considered as a dynamical system, i.e., a set of coupled differential equations. We consider two scenarios: (1) a single connection over a long delay path involving a bottleneck; and (2) two connections with different roundtrip delays that share a common bottleneck. For a recently proposed control scheme, we obtain closed-form solutions for the dynamical model in both transient and steady state regimes, and evaluate appropriate performance measures. We compare our results with those obtained by others using experimental and simulation approaches.

1 Introduction

A computer network typically uses store-and-forward routing to transfer data packets between users at geographically distributed nodes. Packets generated by a source node are delivered to their destination by routing them via a sequence of intermediate nodes. The traffic flowing through an intermediate node depends upon the number of source-destination pairs that are routed through that node and the rates at which these sources introduce packets into the network. If the source rates are increased without constraint, queues of packets waiting to be routed build up at bottleneck nodes. Eventually, the buffering capacity of these nodes is exceeded and packets are dropped, resulting in low throughput and high delay.

Flow control mechanisms attempt to avoid such breakdown by imposing constraints on the source. Two types of constraint are used. In rate-based flow control, a limit is placed on the rate at which the source can send packets [4, 2]. In window-based flow control, at any time there is a limit to the number of outstanding packets at the source, but there is no constraint on the rate at which packets can be sent [3].

We can formulate the objective of flow control as follows: To maximize the throughput for the source, while minimizing packet loss due to buffer overflows. Consider a source-destination pair

[1]This work was supported in part by National Science Foundation grant NCR 89-04590. The first author was also supported by a University of Maryland Graduate School Dissertation Fellowship.

[2]Also with the Institute for Advanced Computer Studies, University of Maryland.

whose packets are routed via intermediate nodes. Let μ_i be the service rate offered by intermediate node i to packets of this source- destination pair. For two nodes i and j, μ_i and μ_j can differ for several reasons: the total traffic through them may differ because they support different source-destination pairs, their hardware may differ, etc. Assuming that the network is in steady-state, the ideal flow control policy would be to limit the source rate to $\mu = \min_i(\mu_i)$ [6, 2, 4]. Then, no packet is lost and the bottleneck node is utilized 100%. A higher rate would result in packet loss, and a lower rate in underutilization.

In a real network, however, the μ_i's vary with time because connections (i.e., source-destination pairs) are constantly set up and terminated, and because sources do not maintain constant data rates. Consequently, the rate, as well as the identity of the bottleneck node can change with time. The source must be informed somehow of such a change, and a control mechanism should adapt the source rate to the change in the bottleneck rate in minimum time, losing a minimum number of packets in the process. We will look at different feedback and control mechanisms below.

Because the feedback from intermediate nodes reaches the source only after some delay τ, the two objectives of adapting in minimum time and losing a minimum number of packets are in conflict: If the bottleneck rate decreases at time t, then the source keeps sending at a rate higher than the bottleneck can handle until time $t + \tau$, resulting in a large queue at the bottleneck. In order to minimize packet loss, the bottleneck queue size at time t would have to be minimized. If the bottleneck rate increases at time t, then the bottleneck is underutilized until time $t + \tau$ unless an appropriately large queue size existed at time t.

Comparison of different flow control algorithms

One of the most important characteristics of a flow control mechanism is its *feedback*: What information does the source obtain about the state of the intermediate nodes. For example, the source could be informed of the queue sizes at each intermediate node along the path toward its destination. A less ambitious variation of this is the scheme proposed by Ramakrishnan and Jain [13], in which each data packet has a reserved bit initially set to 0. The bit is set to 1 by an intermediate node if the average queue size at that node is greater than 1. Thus, even if only one node among all intermediate nodes has an average queue size greater than 1, the destination receives data packets with bits set to 1. The destination then sends this information back to the source in the acknowledgement packets. Upon receipt of an acknowledgement, the source decreases its window size by a multiplicative factor if the bit is set to 1. Otherwise, it increases the window size linearly. Clearly, the objective here is to keep the bottleneck node 100% utilized and with no data packet waiting for service. Thus, timeouts and messages losses should be very infrequent occurrences here.

The situation is different in TCP, where the feedback consists solely of the arrival times of acknowledgement messages. Jacobson noted that, assuming the network is in steady-state, the interval between the reception of successive acknowledgement packets equals the bottleneck service time [7]. Thus, the ideal policy in steady-state would be to send a data packet whenever an acknowledgement is received. However, this is not appropriate in a dynamic environment; for example, the source would not be able to adapt to an increase in the bottleneck service rate. The only way to adapt to this situation is to send at a rate higher that the bottleneck rate. Jacobson describes a scheme in which the window size increases with time, either exponentially (if it is currently less than half the size it was before the last timeout) or linearly (otherwise). At every timeout, the window size is reset to 1, the idea being that a packet loss means that the bottleneck queue is overflowing. It is clear that in this scheme, the bottleneck queue size is maintained at

higher levels than in [13] and packet losses and timeouts are more regular occurrences. A good round trip delay estimate is an essential part of this scheme.

Observe that in both schemes described above, the send window size is controlled at the source. NETBLT [4, 10] and VMTP [2] are two protocols where the data transmission rate is intended to be controlled by adjusting interpacket gaps. However, it is not clear from the literature how these are adjusted or what the feedback would be in a wide-area network.

Three basic approaches, namely experimental, simulation, and analytic approaches, have been taken to evaluate the performance of flow control mechanisms. Jacobson [7] analyzed the performance of his window mechanism by implementing it on a network and observing traces of various parameters characterizing network congestion, delays, etc. We have used an instrumentation of TCP to examine the effect of clock resolution and the use of different roundtrip delay estimators [14]. Ramakrishnan and Jain [13] studied the performance of their window flow control mechanism with a deterministic simulation model of a connection in a wide-area network. Each link along the connection is characterized by a fixed service time. The simulation model is run to obtain the time-dependent behavior of the window size for various increase and decrease algorithms, feedback schemes, etc. In [1], we presented a Markov model of the performance of window protocols over channels whose delay and loss characteristics depend significantly upon the number of messages in transit. Such channels are typical of most store-and-forward networks, including the Internet. We solved the Markov model numerically to obtain performance measures such as throughput, response time, congestion in the channels, etc.

Most of the analytic models reported in the literature pertain to the steady-state analysis of stationary queuing systems. However, the analysis of control mechanisms that dynamically regulate data flows according to changing network conditions requires understanding of the dynamic, i.e., time-dependent network behavior. In this paper, we consider deterministic analytic models of connections with rate-based flow control in a wide-area network. We show how to solve the models to obtain closed-form expressions describing the time-dependent behavior of the sending rate and the queue size at the bottleneck in both steady state and transient regimes.

Outline of the paper

We model a network with connections subject to rate-based flow control as a dynamical system, i.e., a set of coupled differential equations involving the source rates and the queue sizes. In Section 2, we formulate a dynamical model of a single connection. The source-to-destination path is assumed to traverse a sequence of intermediate nodes, one of them being a bottleneck. The source rate is controlled with a linear increase/exponential decrease algorithm. We also present performance measures regarding the source rate and the bottleneck queue size for both transient and steady-state regimes.

In Section 3, we obtain analytical closed-form solutions for the transient and steady-state behavior of this dynamical model. Following a finite duration transient behavior, the bottleneck queue size and the sending rate at the source stabilize into a steady-state limit cycle. We give closed-form expressions for the performance measures. Our results show that good steady-state performance can be achieved at the expense of long transient duration; i.e., there is a tradeoff between rapid relaxation to steady state and the efficiency of that steady state. We also show that it is important to take the current roundtrip delay into account when adjusting the sending rate at the source.

In Section 4, we formulate a dynamical model of two connections with different roundtrip delays that share a common bottleneck. Both connections use the same rate-based flow-control scheme as in Section 2. We present numerical solutions of this dynamical model. Our results show that if

sources have identical rate adjustment algorithms, their sending rates converge to fair values; i.e., both users evenly share the bottleneck's bandwidth irrespective of differences in their roundtrip delays. We also show the importance of choosing an appropriate initial send rate.

In Section 5, we point out directions for extending this work.

2 Dynamical Model of a Single Connection

We consider a connection between a source and a destination, where the source and the destination are on different high-speed local area networks that are connected by a slower, wide-area network. In this very common scenario, unless care is taken the source can very easily congest the low-speed network [7]. We model this by representing the slower network by a single bottleneck node and the high-speed networks by fixed delays as shown in Figure 1.

The parameters describing the connection are:

$q(t)$: the queue size at time t at the bottleneck
μ : the bottleneck service rate
τ_F : the propagation delay from the source to the bottleneck
τ_R : the propagation delay from the bottleneck to the source via the destination (including processing time at the destination)
$\lambda(t)$: the rate at which data is sent by the source

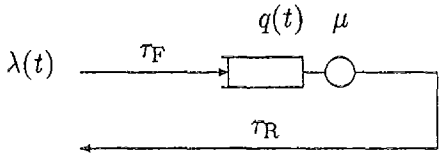

Figure 1: Model of a single connection

In this model, we consider $\lambda(t)$ and $q(t)$ to be real-valued continuous variables. There are several reasons for this choice. First, it allows the adjustments to $\lambda(t)$ to be simply described by specifying $\dot{\lambda}(t)$ (we use $\dot{\lambda}(t)$ to denote $d\lambda(t)/dt$). Second, the dynamical behavior of the model can be conveniently described in terms of coupled differential equations involving $\lambda(t)$, $\dot{\lambda}(t)$, $q(t)$ and $\dot{q}(t)$. Third, this is a standard formulation of control theory; we intend to make use of the results and insights available in that field. Continuous real-valued variables can also be thought of as first order fluid approximations of stochastic processes [9]. For example, $q(t)$ can be thought of as the expectation of a stochastic process $Q(t)$ which represents the number of packets present at time t at the bottleneck [9, 11].

We now derive the equations describing $\dot{q}(t)$ and $\dot{\lambda}(t)$ in terms of $q(t)$ and $\lambda(t)$. Observe that data sent by the source at time t arrives at the bottleneck at time $t + \tau_F$. Thus, the data arrival rate at the bottleneck at time t is $\lambda(t - \tau_F)$. If $q(t) > 0$, data departs from the bottleneck at rate μ. If $q(t) = 0$, data departs at the same rate as it arrived, i.e., $\lambda(t - \tau_F)$. This leads us to the following equation for $\dot{q}(t)$:

$$\dot{q}(t) = \begin{cases} 0 & \text{if } q(t) = 0 \text{ and } \lambda(t - \tau_F) - \mu < 0 \\ \lambda(t - \tau_F) - \mu & \text{otherwise} \end{cases} \quad (1)$$

We now derive the equation for $\dot{\lambda}(t)$. We assume that, at every time t, a bit value indicating whether $q(t) > 0$ or $q(t) = 0$ is fed back to the source from the bottleneck node using a mechanism similar to that described in [13]. This bit indicates how to adjust $\lambda(t)$ as follows: $\lambda(t)$ is decreased exponentially with time constant β if $q(t - \tau_R) > 0$ at time $t - \tau_R$, i.e., the bottleneck was 100% utilized. $\lambda(t)$ is increased linearly with rate α if $q(t - \tau_R) = 0$. Thus, we get the following:

$$\dot{\lambda}(t) = \begin{cases} \alpha & \text{if } q(t - \tau_R) = 0 \\ -\frac{\lambda(t)}{\beta} & \text{if } q(t - \tau_R) > 0 \end{cases} \qquad (2)$$

Clearly, the situation of 100% bottleneck utilization and no waiting delay corresponds to $\lambda(t) = \mu$ and $q(t) = 0$. Observe that $\lambda(t)$ is changed not in response to acknowledgement packets sent by the destination node, with increase and decrease done on a per-packet basis. Rather, the feedback information merely indicates how to adjust $\lambda(t)$, i.e., the rate is increased linearly in time until information is received that indicates to switch to exponential decrease. In a system where data would come in finite-size discrete packets, adjustments to $\lambda(t)$ would not be made continuously.

In our model, adjustments to $\lambda(t)$ are made based on the instantaneous value of $q(t - \tau_R)$. In recently proposed schemes, the bottleneck node computes an average $\hat{q}$ of recent values of the queue size to eliminate short-lived transient variations of $q(t)$. The feedback information then indicates whether $\hat{q} > 0$ or $\hat{q} = 0$. We will not consider such a mechanism here.

Performance measures

We now define performance measures for the dynamical behavior of the system modeled above. As was mentioned in Section 1, it is important to consider measures that characterize both steady-state and transient behavior. The system described by our model turns out to have an initial transient of finite duration, which we shall denote by t_0, followed by a steady-state limit cycle with a period that we shall denote by T.

We consider the following measures in steady-state

q_{max}	: maximum value of $q(t)$
$\overline{q}$	: average of $q(t)$
$\sigma(q)$	: standard deviation of $q(t)$
$\overline{\lambda} - \mu$	: average of $\lambda(t) - \mu$
$\sigma(\lambda - \mu)$	: standard deviation of $\lambda(t) - \mu$

where average and standard deviation of a function $f(t)$ are defined as follows:

$$\overline{f} = \frac{1}{T} \int_{t_0}^{T+t_0} f(t)dt$$

$$\sigma^2(f) = \frac{1}{T} \int_{t_0}^{T+t_0} [f(t) - \overline{f}]^2 dt$$

We characterize the transient behavior by its duration time t_0. We could consider measures similar to those defined above for steady-state, such as the time average $\frac{1}{t_0} \int_0^{t_0} q(t)dt$ of $q(t)$, but they do not prove very interesting as we shall see below.

3 Analytic Solution of the Single Connection Model

We now solve equations (1) and (2) of the single connection model, which are repeated below:

$$\dot{q}(t) = \begin{cases} 0 & \text{if } q(t) = 0 \text{ and } \lambda(t - \tau_F) - \mu < 0 \\ \lambda(t - \tau_F) - \mu & \text{otherwise} \end{cases}$$

$$\dot{\lambda}(t) = \begin{cases} \alpha & \text{if } q(t - \tau_R) = 0 \\ -\frac{\lambda(t)}{\beta} & \text{if } q(t - \tau_R) > 0 \end{cases}$$

The evolution of $\lambda(t)$ and $q(t)$ is depicted pictorially in Figure 2. We now explain it. Assume that the source starts sending data at time $t = 0$ at a rate $\lambda(0) < \mu$. $\lambda(t)$ increases at rate α and reaches the value μ at time $t_0 = \mu/\alpha$. Only now is $\lambda(t)$ large enough to create a queue buildup at the bottleneck. Because of the delay τ_F, the queue at the bottleneck starts building up at time $t_0 + \tau_F$. Therefore, the source detects the existence of the queue at the bottleneck only at time $t_0 + \tau$, where the round trip delay $\tau = \tau_F + \tau_R$. Thus, $\lambda(t)$ increases linearly from $t = 0$ to $t = t_1 = t_0 + \tau$, at which point $\lambda(t_1) = \mu + \tau\alpha$.

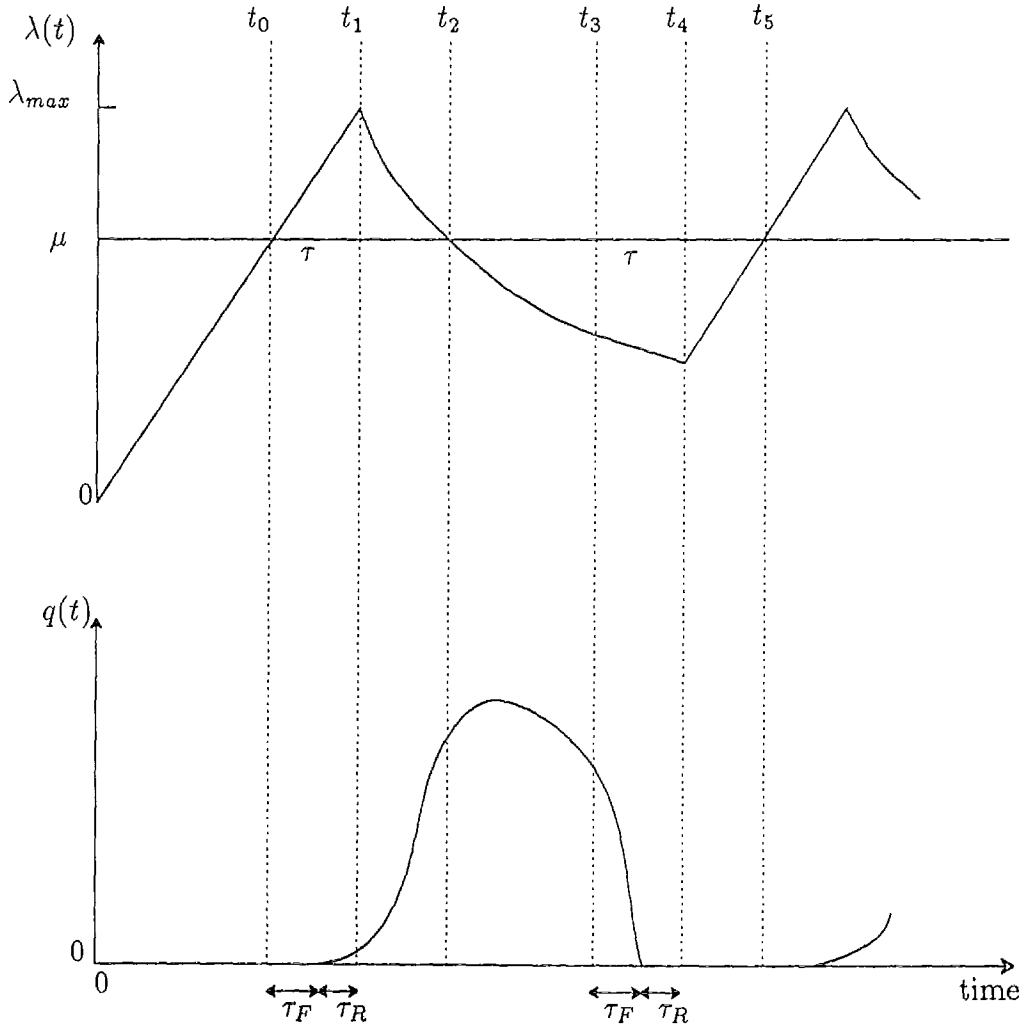

Figure 2: Analytic behavior of $\lambda(t)$ and $q(t)$

At time t_1, $\lambda(t)$ starts decreasing. However, the queue keeps increasing since $\lambda(t)$ remains higher than μ for some time. Let t_2 be the time when $\lambda(t)$ again equals μ. Then $q(t)$ reaches its maximum at time $t_2 + \tau_F$, after which it starts decreasing. We define t_3 as the time at which $\lambda(t)$ has been

lower than μ long enough to drain out the extra packets introduced between t_0 and t_2 (this time is evaluated below). $q(t)$ becomes zero at time $t_3 + \tau_F$. Meanwhile, $\lambda(t)$ keeps decreasing until the source detects, at time $t_4 = t_3 + \tau$, that the queue size at the bottleneck is down to 0, at which point it starts increasing again.

Let t_5 be the time at which $\lambda(t_5) = \mu$. Observe that $\lambda(t_5) = \lambda(t_0)$ and $q(t_5) = q(t_0)$, i.e., the state of the system at time t_5 is identical to that at time t_0. Therefore, the steady-state behavior of both $\lambda(t)$ and $q(t)$ is cyclic with period $T = t_5 - t_0$. The transient duration t_0 is then $t_0 = \mu/\alpha$.

To summarize, the transient behavior of the system is defined by :

$$\left.\begin{array}{l} \lambda(t) = \alpha t \\ q(t) = 0 \end{array}\right\} 0 \leq t \leq t_0$$

and the steady-state behavior is defined by :

$$\lambda(t) = \begin{cases} (\mu + \alpha\tau) exp(-\frac{1}{\beta}(t - t_1 - nT)) & t_1 + nT \leq t \leq t_4 + nT \\ \alpha(t - t_4 - nT) + \lambda_{min} & t_4 + nT \leq t \leq t_1 + (n+1)T \end{cases}$$

$$q(t) = \int_{t_0}^{t} (\lambda(t - \tau_F) - \mu) dt$$

Example evolutions of $q(t)$ and $\lambda(t)$ are shown in Figure 3; the periodic behavior of $q(t)$ and $\lambda(t)$ is obvious.

Performance measures

From the steady-state solution, we now derive closed-form expressions for the values of q_{max}, λ_{max}, and the period $T = t_5 - t_0$.

We first consider λ_{max}. Since $\lambda(t)$ increases linearly at rate α for $t \leq t_1$, it follows that

$$\boxed{\lambda_{max} = \mu + \alpha\tau}$$

The maximum value q_{max} of the bottleneck queue is given by $q_{max} = \int_{t_0}^{t_2} (\lambda(t) - \mu) dt$. Since $\lambda(t) = (\mu + \alpha\tau) \exp(-\frac{1}{\beta}(t - t_1))$ for $t_1 \leq t \leq t_2$, it follows that

$$q_{max} = \alpha\frac{\tau^2}{2} + \int_0^{t_2 - t_1} ((\mu + \alpha\tau)e^{-\frac{1}{\beta}t} - \mu) dt$$

where $t_2 - t_1$ is the solution of $\exp(-\frac{1}{\beta}(t_2 - t_1)) = \mu/(\mu + \alpha\tau)$. We obtain

$$\boxed{q_{max} = \alpha\frac{\tau^2}{2} + \alpha\beta\tau + \mu\beta \ln(\frac{\mu}{\mu + \alpha\tau})}$$

We now compute the period $T = t_5 - t_0$. Observe that $T = (t_1 - t_0) + (t_3 - t_1) + (t_4 - t_3) + (t_5 - t_4) = \tau + (t_3 - t_1) + \tau + (\mu - \lambda_{min})/\alpha = 2\tau + (t_3 - t_1) + (\mu - \lambda_{min})/\alpha$. Since $\lambda_{min} = (\mu + \alpha\tau) \exp(-\frac{1}{\beta}(\tau + t_3 - t_1))$, the problem reduces to that of finding $t_3 - t_1$. However, t_3 is defined such that

$$\int_{t_0}^{t_3} (\lambda(t) - \mu) dt = \alpha\frac{\tau^2}{2} + \int_0^{t_3 - t_1} ((\mu + \alpha\tau)e^{-\frac{1}{\beta}t} - \mu) dt = 0$$

After some algebraic manipulations, we obtain

$$\boxed{t_3 - t_1 = \beta Root(\frac{\mu}{\mu + \alpha\tau}, \frac{\alpha\tau^2}{2\beta(\mu + \alpha\tau)})}$$

where $Root(a,b)$ denotes the unique real solution of $1-e^{-x}=ax-b$, with $0<a<\infty$ and $0<b$. We have from above

$$\boxed{T = 2\tau + (t_3 - t_1) + \frac{\mu - \lambda_{min}}{\alpha}}$$

This concludes the computation of T.

First order approximation

We now consider a first-order approximation to our original model. Specifically, we consider a system model in which $\lambda(t)$ is adjusted as follows

$$\dot{\lambda}(t) = \begin{cases} \alpha & \text{if } q(t-\tau_R) = 0 \\ -\frac{\mu}{\beta} & \text{if } q(t-\tau_R) > 0 \end{cases}$$

This amounts to approximating the exponentially decreasing function $(\mu+\alpha\tau)\exp(-\frac{1}{\beta}(t-t_1))$ by the linearly decreasing function $(\mu+\alpha\tau)-\frac{\mu}{\beta}(t-t_1)$. Then, assuming that $\lambda_{min}>0$, we obtain:

$$\boxed{\begin{array}{lcl} \lambda_{max} & = & \mu+\alpha\tau \\ q_{max} & = & \alpha\frac{\tau^2}{2}(1+\frac{\alpha\beta}{\mu}) \\ T & = & \tau(1+\frac{\alpha\beta}{\mu})(1+\frac{\mu}{\alpha\beta}+\sqrt{1+\frac{\mu}{\alpha\beta}}) \\ \overline{\lambda} & = & \mu-\frac{\alpha\tau}{2}(\frac{\mu}{\alpha\beta}+\sqrt{1+\frac{\mu}{\alpha\beta}}-1) \end{array}}$$

Setting $\alpha\beta = \mu$

For convenience and clarity of the presentation, we eliminate the parameter β by assuming, throughout the rest of the paper, that coefficients α and β satisfy the relation $\alpha\beta=\mu$. This means that, near the fixed point $\lambda(t)=\mu$, $q(t)=0$, the rate at which $\lambda(t)$ increases is equal to the rate at which it decreases.

We can then simplify the results derived above. For convenience of reference, we now present the values of the performance measures obtained with the first order approximation. Assuming that $\lambda_{min}>0$, i.e., $\mu>\alpha\tau(1+\sqrt{2})$, we have:

$$\boxed{\begin{array}{lcl} \lambda_{max} & = & \mu+\alpha\tau \\ q_{max} & = & \alpha\tau^2 \\ T & = & 2\tau(2+\sqrt{2}) \\ \overline{\lambda} & = & \mu-\frac{\alpha\tau}{\sqrt{2}} \end{array}} \qquad (3)$$

We will see later that these approximations for q_{max}, T, and $\overline{\lambda}$ are quite accurate. We observe that the duration of the transient behavior given by $t_0=\mu/\alpha$ is inversely proportional to α. Therefore, steady-state is reached quickly if α is large. However, a large value of α implies a large value of q_{max} and thus a higher probability of packet loss at the bottleneck due to buffer overflow. In addition, equation (3) indicates that the average steady-state sending rate $\overline{\lambda}$ moves farther away from the ideal value μ as α increases. Therefore, a tradeoff has to be found between a rapid relaxation to steady-state and the efficiency of the steady-state behavior. This is consistent with the results reported in [13].

In a real network, τ varies with time. Therefore, keeping q_{max} at a given value requires α to vary with time such that α is proportional to $1/\tau^2$. Similarly, keeping $\overline{\lambda}$ at a given value requires α

to vary with time as $1/\tau$. Which objective is more important will dictate how α should vary with time. But in any case, these results indicate the need for a time-varying value of α.

Example behaviors of a single connection

We now present example behaviors of the single-connection model described above. We fix the bottleneck rate at $\mu = 1s^{-1}$, and we assume that $\tau_F = \tau_R$ so that the round trip propagation delay $\tau = 2\tau_F$. Thus, the free parameters are α and τ.

We fix $\tau = 20s$ and consider different values of α. Figure 3(a) shows the evolutions of $\lambda(t)$ and $q(t)$ for $\alpha = 1/40$. Figure 3(b) shows the evolutions of $\lambda(t)$ and $q(t)$ for $\alpha = 1/160$. In each case, we solve the differential equations for t in the range $0 \leq t \leq t_{max} = 400s$. In Table 1, we show the values of the performance measures (defined in Section 2) for different α's. We show the measures obtained by solving the differential equations and the measures obtained with the first order approximation.

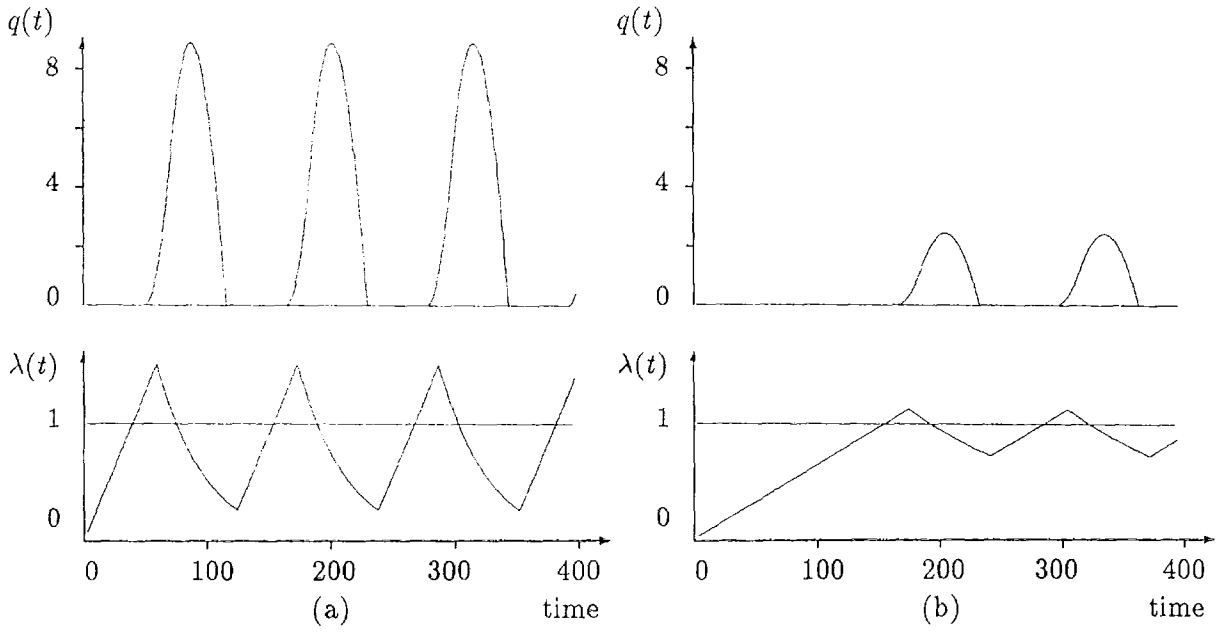

Figure 3: Evolutions of $\lambda(t)$ and $q(t)$ for $\tau = 20s$: $\alpha = 1/40$ in part (a); $\alpha = 1/160$ in part (b)

	exact measures				approximate measures				transient
α	q_{max}	$\overline{q}$	$\overline{\lambda}$	T	q_{max}	$\overline{q}$	$\overline{\lambda}$	T	t_0
1/10	29	10.8	0.75	99.7	40	11.7	−0.4	136.6	10
1/20	16.1	5.6	0.75	104.4	20	5.8	0.29	136.6	20
1/40	8.8	2.8	0.80	114.4	10	2.9	0.65	136.6	40
1/80	4.65	1.4	0.87	123.4	5	1.45	0.82	136.6	80
1/160	2.4	0.7	0.92	129.3	2.5	0.73	0.91	136.6	160
1/320	1.2	0.36	0.96	132.7	1.25	0.37	0.96	136.6	320
1/640	0.62	0.18	0.98	134.5	0.625	0.183	0.98	136.6	640

Table 1: Performance measures versus α for $\tau = 20s$

As expected, we observe that small values of α provide good steady-state performance, but long transient duration. We verify that the analytical results obtained with the first order approximation are quite accurate for $\mu > \alpha\tau(1+\sqrt{2})$, i.e., $\alpha < 1/50$. In particular, the values in Table 1 are consistent with the fact that q_{max} is proportional to α and $\overline{\lambda} - \mu$ decreases as α decreases.

We now fix $\alpha = 1/40$ and consider different values of τ. Figure 4(a) shows the evolutions of $\lambda(t)$ and $q(t)$ when $\tau = 20s$ for $0 \leq t \leq t_{max}/2 = 400s$ and $\tau = 30s$ for $t_{max}/2 \leq t \leq t_{max} = 800s$. We observe that q_{max} and $\overline{\lambda} - \mu$ increase as τ increases, and it can be shown that the first order approximation is again accurate when $\mu > \alpha\tau(1+\sqrt{2})$, i.e., $\tau < 17$.

Recall that, in the approximation, q_{max} is proportional to τ^2. Therefore, q_{max} should be constant as τ varies if α varies proportionally to $1/\tau^2$. This is clear is Figure 4(b), which shows the evolutions of $\lambda(t)$ and $q(t)$ for $\alpha = 10/\tau^2$, i.e., $\alpha = 1/40$ and $\tau = 20s$ for $0 \leq t \leq t_{max}/2$, and $\alpha = 1/90$ and $\tau = 30s$ for $t_{max}/2 \leq t \leq t_{max}$.

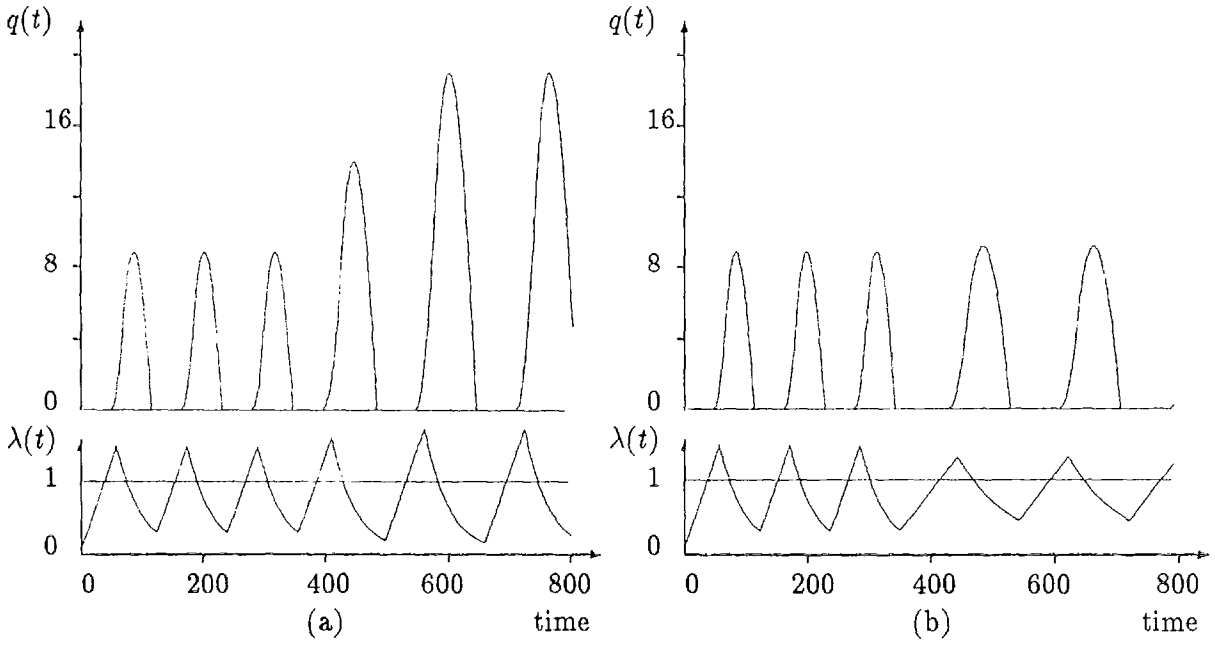

Figure 4: Evolutions of $\lambda(t)$ and $q(t)$ with $\tau = 20s$ for $0 \leq t \leq 400s$ and $\tau = 30s$ for $400 \leq t \leq 800s$: $\alpha = 1/40$ in part (a); $\alpha = 1/40$ for $0 \leq t \leq 400s$ and $\alpha = 1/90$ for $400 \leq t \leq 800s$ in part (b)

4 Dynamics of Two Connections Sharing a Bottleneck

In this section, we consider the situation where two rate-based flow controlled connections share a common bottleneck as shown in Figure 5. The parameters describing the system are as follows

- $q(t)$: the queue size at time t at the bottleneck
- μ : the bottleneck service rate
- τ_{1F}, τ_{1R} : the forward and reverse propagation delay for connection 1
- τ_{2F}, τ_{2R} : the forward and reverse propagation delay for connection 2
- $\lambda_1(t)$: the rate at which data is sent by source 1
- $\lambda_2(t)$: the rate at which data is sent by source 2

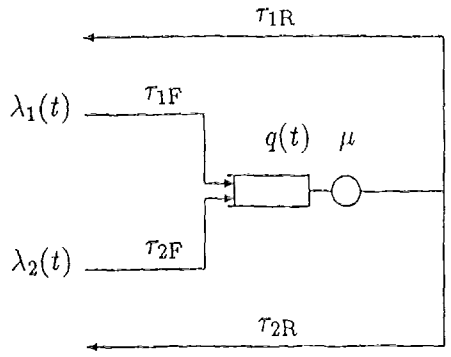

Figure 5: A model of two connections sharing a bottleneck node

We now derive the equations describing the dynamical behavior of this system. We proceed as in Section 2. The arrival rate at the bottleneck at time t is now $\lambda_1(t-\tau_{1F}) + \lambda_2(t-\tau_{2F})$. Therefore, $\dot{q}(t)$ is given by the following equations

$$\dot{q}(t) = \begin{cases} 0 & \text{if } q(t) = 0 \text{ and } \lambda_1(t-\tau_{1F}) + \lambda_2(t-\tau_{2F}) < \mu \\ \lambda_1(t-\tau_{1F}) + \lambda_2(t-\tau_{2F}) - \mu & \text{otherwise} \end{cases}$$

Assuming that the sending rate of each source is controlled in the way described in Section 2, we have the following equations for $\dot{\lambda}_1(t)$ and $\dot{\lambda}_2(t)$

$$\dot{\lambda}_1(t) = \begin{cases} \alpha_1 & \text{if } q(t-\tau_{1R}) = 0 \\ -\alpha_1\lambda_1(t)/\mu & \text{if } q(t-\tau_{1R}) > 0 \end{cases}$$

$$\dot{\lambda}_2(t) = \begin{cases} \alpha_2 & \text{if } q(t-\tau_{2R}) = 0 \\ -\alpha_2\lambda_2(t)/\mu & \text{if } q(t-\tau_{2R}) > 0 \end{cases}$$

We observe that the two sources interact via their feedback $q(t)$: the feedback used by source i depends dynamically on the behavior of $\lambda_j(t)$. Therefore, we expect the dynamics of $\lambda_i(t)$ to depend on $\lambda_j(t)$.

In the remainder of this section, we consider a scenario in which source 1 starts sending data at time $t = 0$ and source 2 starts sending data at a time $t_c > 0$ when source 1 has already reached steady-state . We expect that the transient behavior initiated by source 2 will last for a finite duration t_0. Thus, at time $t_0 + t_c$, both connections will have settled into a steady-state limit cycle. Henceforth, the term 'steady-state' applies to the interval $t \geq t_0 + t_c$ and the term 'transient' to $t_c \leq t \leq t_0 + t_c$.

The ideal steady-state behavior would be that each source sends at an average rate $\mu/2$. Therefore, the performance measures we consider in steady-state are the averages $\overline{\lambda_1}$ and $\overline{\lambda_2}$, the standard deviations $\sigma(\lambda_1 - \mu/2)$ and $\sigma(\lambda_2 - \mu/2)$, and the maximum queue size q^{ss}_{max}. We characterize the transient behavior by its duration t_0 and the maximum queue size q^{t}_{max} achieved during the transient phase.

Example behaviors

We now present example behaviors of the above dynamical system. They were obtained by numerically solving the differential equations describing the system over the range $0 \leq t \leq t_{max} = 800s$. We fix the bottleneck rate $\mu = 1s^{-1}$ and the time t_c at which source 2 starts sending data at $t_c = t_{max}/4 = 200s$. Throughout the rest of this paper, we fix we fix $\tau_{1F} = \tau_{1R} = 10s$ and $\alpha_1 = 1/40$ for connection 1. For connection 2, we assume $\tau_{2F} = \tau_{2R}$. Thus, α_2 and $\tau_2 = \tau_{2F} + \tau_{2R} = 2\tau_{2F}$ are the free parameters of the system. In the next three subsections, we examine the behaviors for the following variations of α_2 and τ_2: (a) different α_2's with $\tau_2 = \tau_1$ fixed, (b) different τ_2's with α_2 fixed and (c) different initial sending rate for source 2 with α_2 and τ_2 fixed.

Different α_2's with $\tau_2 = \tau_1$

We fix $\tau_2 = \tau_1 = 20s$. Figure 6(a) shows the evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\alpha_2 = 1/20$. Figure 6(b) shows the evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\alpha_2 = 1/160$.

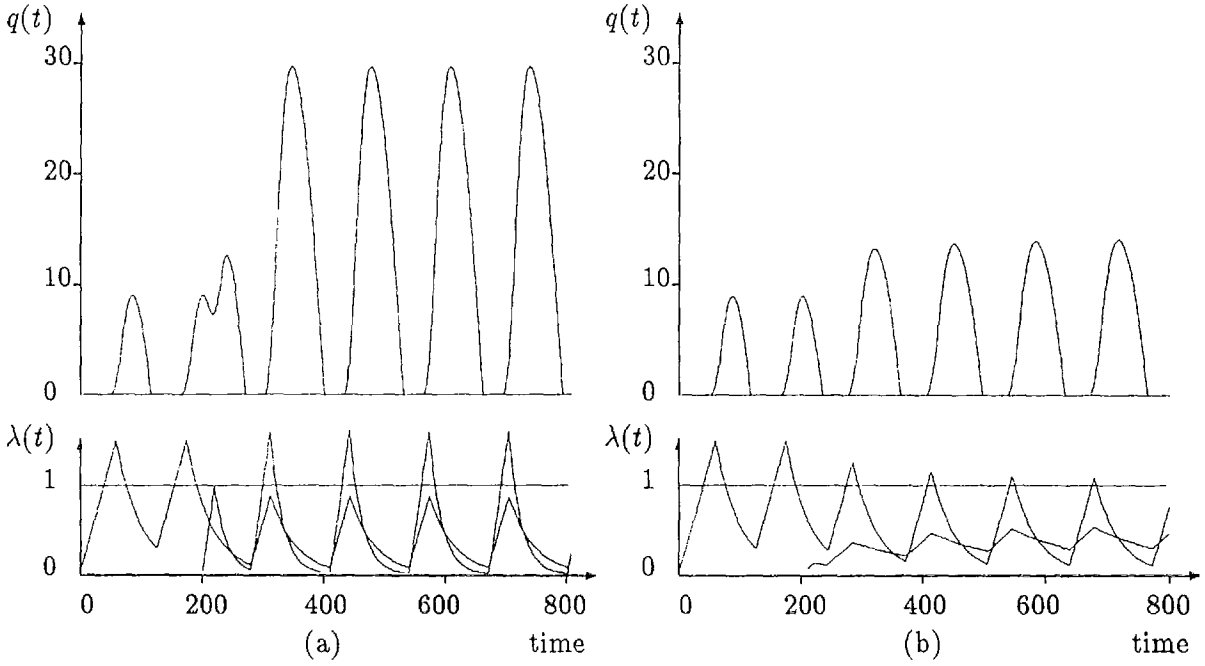

Figure 6: Evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\tau_2 = 20s$: $\alpha_2 = 1/20$ in part (a); $\alpha_2 = 1/160$ in part (b)

Consider the variations of $\lambda_2(t)$. We observe a phenomenon similar to that described in Section 3. Namely, a high value of α_2 results in a short transient duration and a large value of q_{max}. However, the dynamics of the system is complicated by the interaction between the two connections. In Table 2, we show the values of the performance measures defined above for different α_2's. We observe that, if $\alpha_2 \neq \alpha_1$, then the connections settle in a steady-state limit cycle in which they do not evenly share the bottleneck bandwidth. This unfair behavior is particularly clear when α_2 is larger than α_1.

Recall that the analytical results in Section 3 suggested that, if τ varies with time, then α should vary with time as $1/\tau$ or $1/\tau^2$. We now realize that this may lead to unfair sharing of the bottleneck bandwidth. Fair sharing would require a more elaborate mechanism such as the selective feedback mechanism described in [12].

α_2	queue size q_{max}^{ss}	connection 1 $\overline{\lambda_1}$	$\sigma(\mu/2-\lambda_1)$	connection 2 $\overline{\lambda_2}$	$\sigma(\mu/2-\lambda_2)$
1/2	123.0	0.148	0.15	0.74	2.14
1/10	40.2	0.308	0.197	0.52	0.76
1/40	22.3	0.417	0.265	0.417	0.265
1/160	13.8	0.46	0.28	0.413	0.07
1/640	11.3	0.48	0.28	0.417	0.02

Table 2: Performance measures versus α_2 for $\tau_2 = 20s$

Different τ_2's with α_2 fixed

We fix $\alpha_2 = 1/40$ and consider different values of τ_2. Figure 7(a) shows the evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\tau_2 = 10s$. Figure 7(b) shows the evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\tau_2 = 1s$.

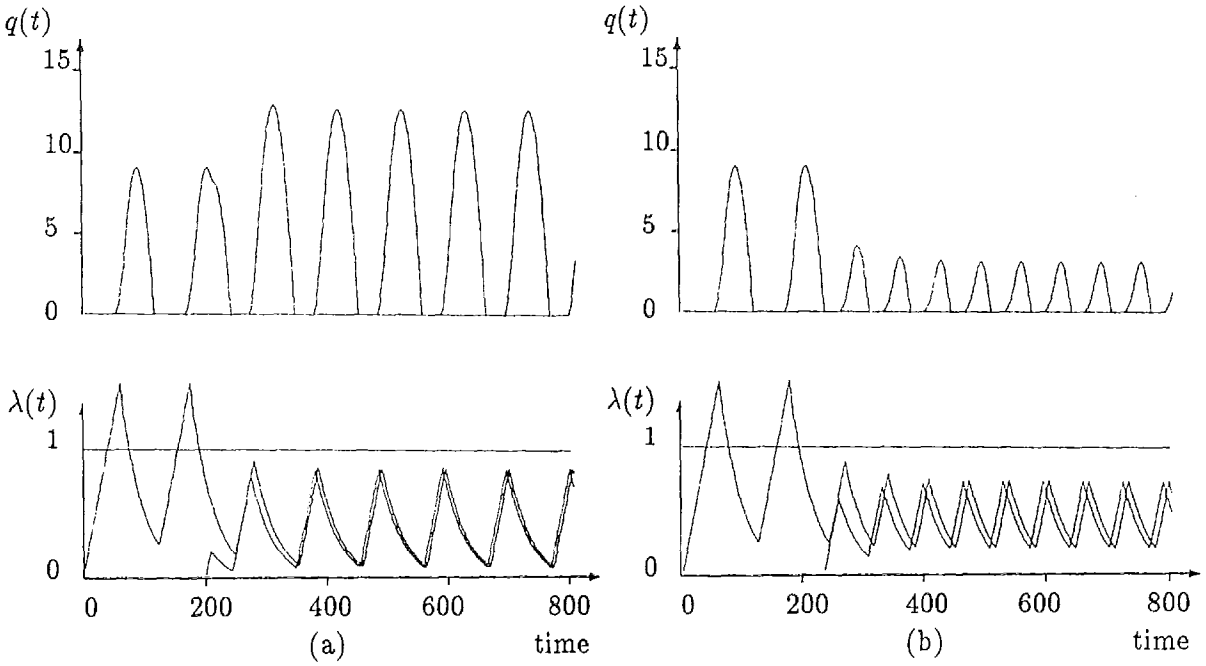

Figure 7: Evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$: $\tau_2 = 10s$ in part (a); $\tau_2 = 1s$ in part (b)

The figures clearly show that the rate adjustments made by source 1 and source 2 are not independent. In particular, we observe that, in steady-state, $\lambda_1(t)$ and $\lambda_2(t)$ oscillate with equal frequency. As may be expected, the phase difference, i.e., the time difference between a peak of $\lambda_1(t)$ and the corresponding peak of $\lambda_2(t)$, is equal to $(\tau_1 - \tau_2)/2$.

We observed above that the connections settle in a steady-state limit cycle in which they evenly share the bottleneck bandwidth if $\alpha_1 = \alpha_2$ and $\tau_1 = \tau_2$. Numerical solutions of the differential equations of the system appear to indicate that this result holds even if $\tau_1 \neq \tau_2$. This suggests that connections with identical rate adjustment algorithms evenly share the bottleneck bandwidth, irrespective of the values of the propagation delays τ_1 and τ_2.

Different initial sending rate for source 2 with $\tau_1 = \tau_2$

In this subsection, we consider a variation on the behavior of source 2. Specifically, when source 2 starts sending data at time t_c, it chooses an initial rate $\lambda_2(t_c)$ greater than zero. We shall now

see that the value chosen for t_c and the initial rate strongly affect the dynamics of the system. Figure 8(a) presents evolutions for $t_c = 200s$ and $\lambda_2(t_c) = 1s^{-1}$. Figure 8(b) presents evolutions for $t_c = 220s$ and $\lambda_2(t_c) = 1s^{-1}$.

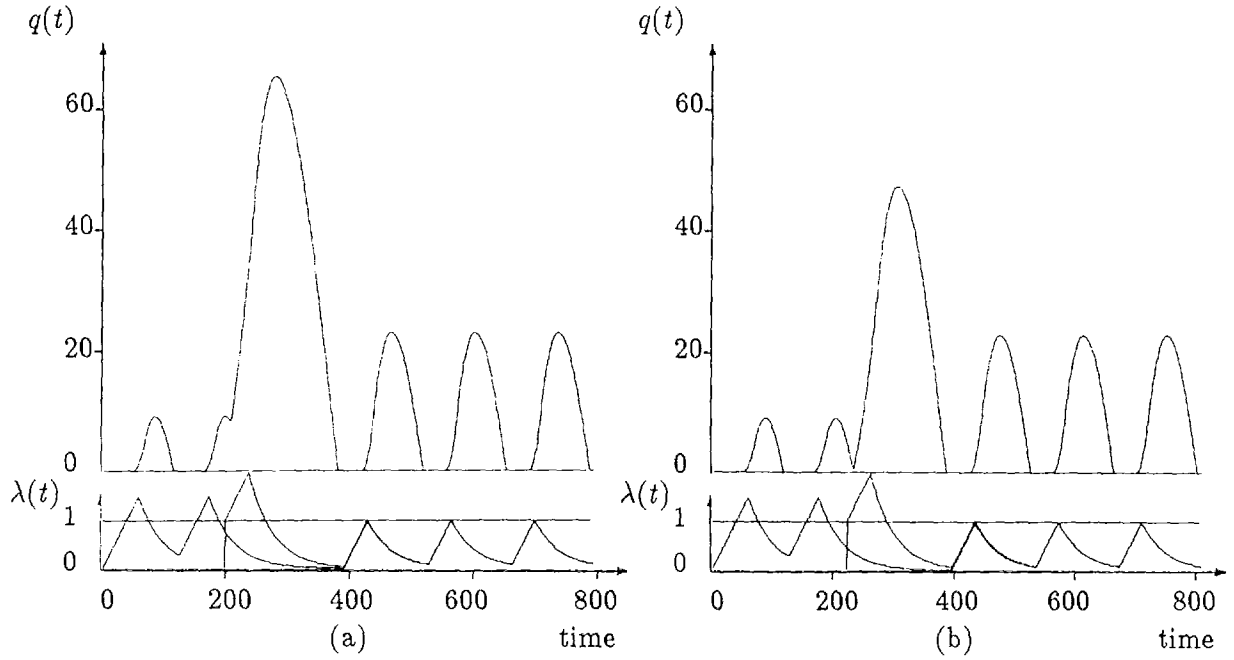

Figure 8: Evolutions of $\lambda_1(t)$, $\lambda_2(t)$ and $q(t)$ for $\lambda_2(t_c) = 1$: $t_c = 200$s in part (a); $t_c = 220$s in part (b)

We observe that a large queue can build up at the bottleneck during the transient. The value of q^t_{max} depends on the values of t_c and $\lambda_2(t_c)$. But in any case, the presence of the queue would result, in a real network, in increased packet loss and retransmissions. Therefore, care should be taken that the initial sending rate of user 2 be chosen at an appropriately low value. This is especially important if the bottleneck were already shared by n users when user 2 becomes active. This suggests that initial sending rates be chosen quite low, since the value of n is not known to a new connection as it starts sending data. This is consistent with the idea of slow-start [7].

5 Conclusion

It appears that dynamical modeling holds great promise in analyzing network congestion and flow control problems. It is very encouraging that the behavior (and thus design choices) indicated by the relatively simple dynamical model in this paper seems to tie in very well with the results obtained by others using experimental and simulation approaches. For example, our single-connection model brings out the tradeoff between good steady-state behavior and rapid adaptability to changing network conditions such as bottleneck rate and round trip delay changes. Our two-connections model indicated that when a connection starts, it should use a very low sending rate. Both these are consistent with the schemes proposed by others [7, 13]. However, further validation is required, and we intend to do so with discrete-event simulation and experiments [14].

We are extending the models in several ways. One is to consider a very large number of connections sharing a common bottleneck. Another is to consider the effects of averaging techniques

in the feedback mechanism. We are examining the control systems literature for more powerful solution techniques. Our efforts so far have not been particularly rewarding; it appears that not much is available in the area of coupled differential equations with delays.

References

[1] J.-C. Bolot, A. U. Shankar, B. D. Plateau, "Performance analysis of transport protocols over congestive channels", CS-TR-2004.2, Department of Computer Science, University of Maryland, College Park, April 1988. Rerevised July 1989. To appear in *Performance Evaluation.*

[2] D. R. Cheriton, "VMTP: a transport protocol for the next generation of communication systems", *Proc. ACM SIGCOMM '86*, Stowe, Vermont, pp. 406-415, Aug. 1986.

[3] D. D. Clark, "Window and Acknowledgement Strategy in TCP", RFC 813, Network Information Center, SRI international, July 1982.

[4] D. D. Clark, M. L. Lambert, L. Zhang, "NETBLT: a high throughput transport protocol", *Proc. ACM SIGCOMM '87*, Stowe, Vermont, pp. 353-359, 1987.

[5] Defense Data Networks, *Transmission Control Protocol*, MIL-STD-1778, August 1983.

[6] M. Gerla, L. Kleinrock, "Flow control: A comparative survey", *IEEE Trans. Comm.*, vol. COM-28, April 1980.

[7] V. Jacobson, "Congestion avoidance and control", *Proc. ACM SIGCOMM '88*, Stanford, California, pp. 314-329, August 1988.

[8] P. Karn, C. Partridge, "Improving round-trip time estimates in reliable transport protocols", *Proc. ACM SIGCOMM '87*, Stowe, Vermont, pp. 2-7, 1987.

[9] L. Kleinrock, *Queueing Systems. Volume 2: Computer Applications*, Wiley-Interscience, New York 1976.

[10] M. Lambert, "On testing the NETBLT Protocol over divers networks", RFC 1030, Network Information Center, SRI International, November 1987.

[11] G. F. Newell, *Applications of Queueing Theory*, Chapman and Hall, London, 1971.

[12] K. K. Ramakrishnan, D.-M. Chiu, R. Jain, "A selective binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer", DEC Technical Report *TR-510*, November 1987.

[13] K. K. Ramakrishnan, R. Jain, "A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer", *Proc. ACM SIGCOMM '88*, Stanford, California, pp. 303-313, August 1988.

[14] D. Sanghi, M. Subramanian, A. U. Shankar, O. Gudmundsson, P. Jalote, "Instrumenting a TCP implementation", CS-TR-2061, Department of Computer Science, University of Maryland, College Park, July 1988.

[15] L. Zhang, "Why TCP timers don't work well," *Proc. ACM SIGCOMM '86*, Stowe, Vermont, pp. 397-405, 1986.