AIMD Congestion Control: Stability, TCP-friendliness, Delay Performance

Lijun Wang, Lin Cai *Member, IEEE,* Xinzhi Liu and Xuemin(Sherman) Shen *Senior Member, IEEE*

***Abstract*— In this report, a class of generalized AIMD/RED (Additive Increase and Multiplicative Decrease/Random Early Detection) model for the Internet is studied. Sufficient conditions for asymptotic stability by using indirect Lyapunov method are obtained for both the homogeneous-flow system and the heterogeneous-flow system. The TCP (Transmission Control Protocol)-friendly condition and average queuing delay are also derived. The analytical results can provide important guidelines for system parameters setting, in order to efficiently and fairly support emerging multimedia applications with a wide variety of Quality of Service (QoS) requirements in future heterogeneous networks. Numeric results by Matlab and simulation results by NS-2 are given to verify the model and validate the analysis.**

***Index Terms*— Congestion control, AIMD, TCP-friendly, Stability, Fairness, Lyapunov method.**

I. Introduction

The proliferation and universal adoption of the Internet have escalated it as the key information transport platform. The explosive growth of the Internet depends on the design of the stateless core network. The intermediate nodes, e.g., routers, forward packets with their best efforts, without quality of service (QoS) guarantee: packets are forwarded with the first in first out (FIFO) queue management strategy, and are discarded when buffer overflows. The intermediate nodes do not maintain any state information about end-to-end sessions, which makes the core network simple, robust, and scalable.

In the Internet, it is the end systems, instead of the core network, that take the responsibility of maintaining stability and integrity of the whole network. Since the end systems have no complete knowledge of the network internal conditions, e.g., network topology, global traffic, and link capacity, etc., they have to estimate the available bandwidth and take appropriate actions without explicit feedback from the core network. When the network suffers congestion, the most important and robust indicator which end systems can capture is packet losses. The end systems should appropriately throttle their sending rates to avoid network congestion collapse (when the network power, defined as throughput over delay, dramatically decreases to zero). The first network congestion collapse was seen in the late 1980's. Since then, the dominant Internet transport layer protocol, Transmission Control Protocol (TCP) [1], had been re-engineered to incorporate the end-to-end congestion control mechanism [2], [3], which is acknowledged as one of the key factors to the success of the Internet.

TCP implements an Additive Increase and Multiplicative Decrease (AIMD) [4] congestion control mechanism. Specifically, a TCP sender additively increases the sending rate to probe for available bandwidth when no congestion occurs and exponentially (multiplicatively) decreases its sending rate in response to network congestion indicators (packet losses). With the AIMD congestion control mechanism, TCP can utilize network resources efficiently, guarantee network stability and maintain the fairness among co-existing TCP flows.

TCP controls the sending rate by a congestion window ($cwnd$), which bounds the maximum number of unacknowledged packets being sent. The $cwnd$ is used to estimate the product of available bandwidth in the bottleneck and the round trip time (rtt). The $cwnd$ is increased by one segment[1] per rtt when no congestion occurs, to probe for available bandwidth; and it is halved when packet losses occur, to respond to network congestion. To distribute the network congestion indicators to all on-going flows fairly, active queue management (AQM) schemes [5], [11], [19], e.g., the Random Early Detection (RED) queue management scheme, can be deployed in the intermediate nodes. With the RED scheme [20], [21], the intermediate nodes discard packets of all on-going flows randomly when the queue length exceeds a pre-defined threshold, so the packet loss rate of each flow is roughly proportional to the flow sending rate.

Although TCP congestion control has been adopted and succeeded over the past two decades, it meets great challenges, mainly from two aspects. First, a large number of multimedia applications with a wide variety of QoS requirements have been emerging. Many multimedia applications are time-sensitive: they can tolerate certain degree of packet losses, but not the excessive packet delay and jitter. TCP's increase-by-one and decrease-by-half strategy may lead to severe throughput variation and delay jitter. Second, with the rapid advances in optical and wireless communications, the Internet is becoming a more heterogeneous and disparate system: link capacity varies from several Kbps to several Gbps, with six orders of magnitude; transmission bit error rates vary from $< 10^{-9}$ to 10^{-3}, also with about six orders of magnitude; and end-to-end delay varies from several milliseconds to several seconds. TCP's increase-by-one and decrease-by-half strategy becomes less efficient when the bandwidth and rtt product is

L. Wang and X. Liu are with the Department of Applied Mathematics, University of Waterloo. L. Cai is with the Department of Electrical & Computer Engineering, University of Victoria. X. Shen is with the Department of Electrical & Computer Engineering, University of Waterloo.

[1]Modern transport protocols can negotiate maximum segment size on its connection establishment to avoid IP fragmentation. In the sequel, the term *packet* is used generically to represent the network layer *packet* and transport layer *segment*.

large (more than a hundred), or small (less than three).

For continuous growth of the Internet, it is critical to redesign and re-engineer TCP congestion control, in order to fairly and efficiently support heterogeneous applications over heterogeneous networks. Therefore, it is important to first fully understand the performance of the AIMD congestion control mechanism, in terms of stability, fairness, and delay. Then, the congestion control parameters in the end systems can be properly adjusted according to the application QoS requirements and the characteristics of the bottleneck to achieve the design goal. In addition, the queue parameters of the intermediate nodes can be tuned, considering the tradeoff between resource utilization efficiency and queuing delay. Since TCP is dominant in the Internet, non-TCP controlled flows should fairly share network resources with co-existing TCP flows, i.e., being TCP-friendly.

This report systematically studies the performance of AIMD controlled flow with mathematical modeling and analysis. In specific, we use Lyapunov stability theorem to prove the asymptotic stability of the generalized AIMD/RED system. The derived equilibria of the AIMD/RED system are used to obtain the TCP-friendly conditions for AIMD parameters and the average queuing delay. The analytical results can provide important guidelines for system parameters setting, in order to *efficiently* and *fairly* support emerging multimedia applications with diversified QoS requirements in future heterogeneous networks. Extensive simulations with the Network Simulator (NS-2) are conducted to verify the analysis and evaluate the performance of AIMD controlled flows.

The remainder of the report is organized as follows. Section II introduces the AIMD and RED algorithms and the model of the system. Section III proves the stability of the generalized AIMD/RED system, and derives the TCP-friendly condition and average queuing delay. Simulation results are given in Section IV, followed by concluding remarks in Section V.

II. Network Models and Algorithms

A. A Class of AIMD/RED Networks

TCP's congestion control algorithm operates in two phases [2]:

i) Slow-Start Phase

The $cwnd$ size is initialized to one, and it is increased by one packet for every acknowledgements (ack) received. This continues till the window size reaches the slow-start threshold ($ssthresh$). Thereafter, the slow-start phase ends, and the next phase called congestion avoidance begins. If a packet loss is detected before the window size reaches $ssthresh$, then $ssthresh$ is set to half of the current window size, and the $cwnd$ size is reset to one followed by the slow-start phase again.

ii) Congestion Avoidance Phase

In this phase, the window size is increased by $1/cwnd$ packet for every ack received. This is roughly equivalent to increase the window size by one packet after every $cwnd$ of acks are received.

If a packet is lost and the following packets can arrive at the receiver successfully, the TCP sender will detect the packet loss by receiving three duplicated acks. Then, it will set the $ssthresh$ and $cwnd$ to half of the current window size, followed by the congestion avoidance phase.

If the network is in severe congestion, the TCP sender cannot receive any ack which will trigger the timeout event; thereafter, it will set the $ssthresh$ to half of the current window size, and re-initialize the $cwnd$, followed by the slow-start phase.

In this report, we consider the high bandwidth and delay product networks where the average window size of TCP is large ($>> 10$), and timeout events are rare. Thus, TCP works in the Congestion Avoidance phase most of time, and it is characterized as AIMD$(1, 0.5)$.

With dynamic window control, TCP has been successful in supporting data-transfer applications like FTP, email, HTTP. However, for multimedia applications with stringent delay requirements and less stringent reliability requirements, the TCP protocol with the *increase-by-one and decrease-by-half* strategy is not desirable. Although emerging Internet-based multimedia applications can use scalable and error-resilient source coding schemes to adapt to network dynamics, they cannot tolerate their sending rates being frequently halved for any packet losses. For multimedia applications, instead of (1, 0.5), a pair of parameters (α, β) can be used to control the increase rate and decrease ratio of the $cwnd$ [13]–[16]. Specifically, when there is no packet loss, the $cwnd$ is increased by α packet per rtt; when there is packet losses, the $cwnd$ is decreased to β times its current value. By choosing a small value of α and a large value of β, the flow throughput variation can be reduced. The AIMD parameters and other protocol parameters can be flexibly chosen according to application requirements and network conditions.

B. Random Early Detection Queue Management Scheme

Since TCP/AIMD senders use packet losses as network congestion indicators, intermediate nodes (routers) must assume a role in network management by sensing congestion and pre-emptively signaling TCP/AIMD senders. A RED-capable router estimates congestion by monitoring its average queue length (and the speed at which the queue length increases). If the length is below a lower threshold, no packet is dropped; if it is above an upper threshold, all packets are dropped. When the queue length is between the two thresholds, packets are dropped with a certain probability, which is a function of the average queue length.

RED can also choose to marking instead of dropping packets when the queue length is above the lower threshold, and routing the marked packets to the receiver. The receiver, in turn, completes the feedback by acknowledging the reception of marked packets to the sender. Upon the reception of such acknowledgments, the sender adjusts its $cwnd$ according to the AIMD(α, β) algorithm.

C. A Fluid-flow Model of AIMD/RED

A stochastic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis [7].

Simulation results demonstrated that this model accurately captured the dynamics of TCP. We extend the fluid-flow model for AIMD, which relates to the average values of key network variables and is described by the following coupled, nonlinear differential equations:

$$
\frac{dW(t)}{dt} = \frac{\alpha}{R} - \frac{2(1-\beta)}{1+\beta} W(t) \frac{W(t-R(t))}{R(t-R(t))} p(t-R(t))
$$

$$
\frac{dq(t)}{dt} = \begin{cases} \dfrac{N(t)\cdot W(t)}{R(t)} - C & q>0 \\ \{\dfrac{N(t)\cdot W(t)}{R(t)} - C\}^{+} & q=0 \end{cases} \tag{1}
$$

where $\{a\}^{+}=\max\{a,0\}$, $\alpha>0$, $\beta\in[0,1]$, $W\in[0,\ W_{max}]$ is the ensemble average of TCP window size (packets); $q \in [0,\ q_{max}]$ is the ensemble average of queue length (packets); R is the ensemble average of round-trip time with $R = \frac{q}{C}+T_p$ (secs), where C is the queue capacity (packets/sec) and T_p is the deterministic delay. Let N be the number of TCP sessions and p the probability of a packet being marked (or dropped). The ensemble average of queue length q and window size W are positive and bounded quantities.

The first differential equation describes the AIMD(α,β) window control dynamic. Roughly speaking, α/R represents the window's additive increase, while $\frac{2(1-\beta)}{1+\beta}W$ represents the window's multiplicative decrease in response to packet marking (or dropping) probability p. This is because the flow's window size always oscillates between βW_{max} to W_{max}, the average window size over a round[2] is $(1+\beta)W_{max}/2$. Each time, the window size is cut by $(1-\beta)W_{max} = 2(1-\beta)/(1+\beta)W$ [6]. The second equation models the bottleneck queue length as simply an accumulated difference between packet arrival rate NW/R and link capacity C. $\{\cdot\}^{+}$ in the model guarantees queue length is a non-negative number.

It is noticed that, (1) is a generalized TCP/AQM congestion control model, which includes the model studied in [7]–[12]. If we choose $\alpha = 1, \beta = 0.5$, (1) is equivalent to the traditional TCP/AQM model. We will also show in next section that the stability properties of the specific model in the literature is compatible with the corresponding properties of this generalized model as well.

III. Stability and Fairness Analysis

A. Stability of AIMD/RED Networks

With the fluid-flow model (1), we assume that the traffic load (N AIMD flows) is time-invariant, i.e., $N(t)=N$. Since we consider a large bandwidth and delay networks where C and R is large, the variation of queuing delay is negligible compared to R, and we assume that the round-trip time of each flow is a constant, $R(t)=R$.

With RED, the packet marking probability is proportional to the average queue length, as shown in Fig. 1. It can be seen that $p = K_p(q_{act} - min_{th})$ with $K_p > 0$ and $p\in[0,1]$. When $q_{act}\leq min_{th}$, $\frac{dW(t)}{dt} = \frac{\alpha}{R}$, i.e., the marking probability

[2]A round is defined as the time interval between two instants at which the sender reduces its $cwnd$ consecutively.

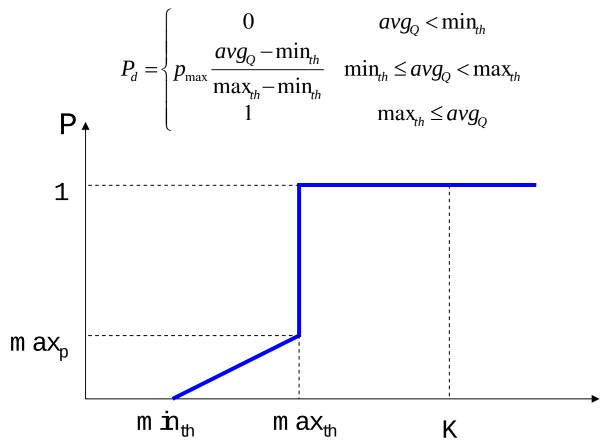


Fig. 1. RED Marking Scheme

is zero, so the window size will keep increasing and won't converge to any value. Thus, in the following, we will discuss the stability property of this model when $q_{act}>min_{th}$. Let $q(t) = q_{act}(t) - min_{th}$, then the original model (1) can be written in a closed-loop dynamics:

$$
\begin{aligned}
\frac{dW(t)}{dt} &= \frac{\alpha}{R} - \frac{2(1-\beta)}{1+\beta} W(t) \frac{W(t-R)}{R(t-R)} K_p q(t-R) \\
\frac{dq(t)}{dt} &= \begin{cases} \frac{N\cdot W(t)}{R} - C & q>0 \\ \{\frac{N\cdot W(t)}{R} - C\}^{+} & q=0 \end{cases}
\end{aligned} \tag{2}
$$

For a one-bottleneck system, the equilibrium point $(W_0^*,\ q_0^*)$ for (2) is given by

$$
W_0^* = \frac{RC}{N}; \qquad q_0^* = \frac{\alpha(1+\beta)N^2}{2(1-\beta)R^2C^2K_p} \tag{3}
$$

With the transformed variables $\tilde{W}{:=}W - W_0^*$, $\tilde{q}{:=}q - q_0^*$, (2) becomes

$$
\begin{aligned}
\dot{\tilde{W}}(t) = & - \frac{2(1-\beta)}{1+\beta} \frac{(\tilde{W}(t)+W_0^*)^2}{R} K_p \tilde{q}(t-R) \\
& - \frac{\tilde{W}^2(t) + 2\tilde{W}(t)W_0^*}{R} K_p q_0^*
\end{aligned} \tag{4}
$$

$$
\dot{\tilde{q}}(t) = \frac{N}{R}\cdot \tilde{W}(t) \tag{5}
$$

The equilibrium point for (4) and (5) is then $(\tilde{W}^*,\ \tilde{q}^*)$=$(0,0)$. Note that $\tilde{q} \geq -q_0^*$ since $q > 0$.

In the case of delay-free marking, i.e., $p(t) = K_p q(t)$, we will show that the AIMD/RED system's equilibrium point $(W_0^*,\ q_0^*)$ is asymptotically stable for all positive gains K_p. Then, (4) and (5) for delay-free marking can be written as

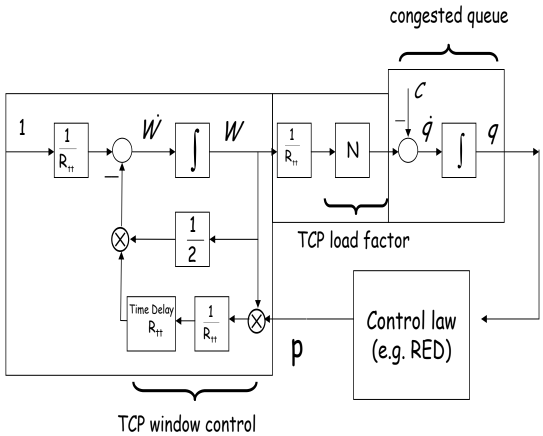


Fig. 2. Block Diagram of AIMD/RED Network

$$\begin{aligned}\dot{\tilde{W}}(t) = &-\frac{2(1-\beta)}{1+\beta}\frac{(\tilde{W}(t)+W_0^*)^2}{R}K_p\tilde{q}(t)\\ &-\frac{2(1-\beta)}{1+\beta}\frac{\tilde{W}^2(t)+2\tilde{W}(t)W_0^*}{R}K_p q_0^*\end{aligned} \tag{6}$$

$$\dot{\tilde{q}}(t) = \frac{N}{R}\cdot\tilde{W}(t)$$

Consider the positive-definite Lyapunov function

$$V(\tilde{W},\,\tilde{q}) = \frac{(1+\beta)N^3}{2(1-\beta)R^2C^2}\cdot\tilde{W}^2(t)+\frac{1}{2}K_p\tilde{q}^2(t)$$

We can get the following theorem for AIMD/RED networks, which is similar to the argument in [11] for TCP/AQM networks.

Theorem 1: The equilibrium point of (6) is asymptotically stable for all $K_p > 0$.

From the viewpoint of control theory, the block diagram is depicted in Fig. 2. By suitable control law, we relate the output q with the input p, which makes the original open loop systems into a closed loop control system to achieve asymptotic stability.

B. Fairness of different (α, β) flows

In the previous section, we discussed the stability property of the homogeneous-flow system when there is only one type of flows with the parameter pair (α, β). In this section, we will study the fairness issue for the heterogeneous-flows system when there are two or more types of flows with the parameter pairs (α_1, β_1), (α_2, β_2), $\cdots$, (α_n, β_n), respectively.

First, we consider the case when there are two different types of flows W_I and W_{II}, with the parameters (α_1, β_1), (α_2, β_2), respectively. The number of W_I flow is N_1, and the number of W_{II} flow is N_2. Then, the corresponding mathematical model has the following form,

$$\frac{dW_I(t)}{dt} = \frac{\alpha_1}{R}-\frac{2(1-\beta_1)}{1+\beta_1}\cdot\frac{W_I(t)^2}{R}\cdot K_p q(t)$$

$$\frac{dW_{II}(t)}{dt} = \frac{\alpha_2}{R}-\frac{2(1-\beta_2)}{1+\beta_2}\cdot\frac{W_{II}(t)^2}{R}\cdot K_p q(t)$$

$$\frac{dq(t)}{dt} = \begin{cases}\dfrac{N_1W_I(t)+N_2W_{II}(t)}{R}-C & q>0\\ \{\dfrac{N_1W_I(t)+N_2W_{II}(t)}{R}-C\}^+ & q=0\end{cases} \tag{7}$$

The equilibrium point (W_I^*, W_{II}^*, q_0^*) of (7) can be obtained as

$$W_I^* = \frac{RC}{N_1+(\frac{\alpha_2(1-\beta_1)(1+\beta_2)}{\alpha_1(1+\beta_1)(1-\beta_2)})^{1/2}\cdot N_2};$$

$$W_{II}^* = \frac{RC}{(\frac{\alpha_1(1+\beta_1)(1-\beta_2)}{\alpha_2(1-\beta_1)(1+\beta_2)})^{1/2}\cdot N_1+N_2};$$

$$q_0^* = \frac{\alpha_1(1+\beta_1)[N_1+(\frac{\alpha_2(1-\beta_1)(1+\beta_2)}{\alpha_1(1+\beta_1)(1-\beta_2)})^{1/2}\cdot N_2]^2}{2R^2C^2K_p(1-\beta_1)} \tag{8}$$

With the transformed variables $\tilde{W}_I(t) := W_I(t) - W_I^*$, $\tilde{W}_{II}(t) := W_{II}(t) - W_{II}^*$ and $\tilde{q}(t) := q(t) - q_0^*$, (7) becomes

$$\begin{aligned}\dot{\tilde{W}}_I(t) = &-\frac{2(1-\beta_1)}{1+\beta_1}\frac{(\tilde{W}_I(t)+W_I^*)^2}{R}K_p\tilde{q}(t)\\ &-\frac{2(1-\beta_1)}{1+\beta_1}\frac{\tilde{W}_I^2(t)+2W_I^*\tilde{W}_I(t)}{R}K_p q_0^*\end{aligned}$$

$$\begin{aligned}\dot{\tilde{W}}_{II}(t) = &-\frac{2(1-\beta_2)}{1+\beta_2}\frac{(\tilde{W}_{II}(t)+W_{II}^*)^2}{R}K_p\tilde{q}(t)\\ &-\frac{2(1-\beta_2)}{1+\beta_2}\frac{\tilde{W}_{II}^2(t)+2W_{II}^*\tilde{W}_{II}(t)}{R}K_p q_0^*\end{aligned}$$

$$\dot{\tilde{q}}(t) = \frac{N_1\cdot\tilde{W}_I(t)+N_2\cdot\tilde{W}_{II}(t)}{R} \tag{9}$$

The equilibrium point for (9) is then ($\tilde{W}_I^*$, $\tilde{W}_{II}^*$, $\tilde{q}_0^*$) $=(0,0,0)$. Note that $\tilde{q}(t) \geq -q_0^*$ since $q(t) > 0$.

With (9), choose the following positive-definite Lyapunov function,

$$\begin{aligned}&V(\tilde{W}_I(t),\,\tilde{W}_{II}(t),\,\tilde{q}(t))\\ &=\frac{(1+\beta_1)N_1}{2(1-\beta_1)W_I^{*2}}\cdot\tilde{W}_I^2(t)+\frac{(1+\beta_2)N_2}{2(1-\beta_2)W_{II}^{*2}}\cdot\tilde{W}_{II}^2(t)\\ &+K_p\tilde{q}^2(t)\end{aligned}$$

Then,

$$
\begin{aligned}
\dot{V} &= \frac{(1+\beta_1)N_1}{(1-\beta_1)W_I^{*2}}\tilde{W}_I(t)\dot{\tilde{W}}_I(t) \\
&\quad + \frac{(1+\beta_2)N_2}{(1-\beta_2)W_{II}^{*2}}\tilde{W}_{II}(t)\dot{\tilde{W}}_{II}(t) + 2K_p\tilde{q}(t)\dot{\tilde{q}}(t) \\
&= -\frac{2N_1K_p}{W_I^{*2}R}\tilde{W}_I^2(t)(\tilde{W}_I(t)+2W_I^*)(\tilde{q}(t)+q_0^*) \\
&\quad -\frac{2N_2K_p}{W_{II}^{*2}R}\tilde{W}_{II}^2(t)(\tilde{W}_{II}(t)+2W_{II}^*)(\tilde{q}(t)+q_0^*) \\
&\leq 0
\end{aligned}
$$

From the physics constraint point of view, the positive-definite Lyapunov function is like the total energy function of a mechanical system, i.e., the sum of kinetic and potential energy. Here $\dot{V} \leq 0$, since $\tilde{W}_I(t) + 2W_I^* > 0$, $\tilde{W}_{II}(t) + 2W_{II}^* > 0$ and $\tilde{q}(t) + q_0^* \geq 0$, which means the energy of the system is non-increasing. Thus, we prove that the equilibrium point is stable. To conclude asymptotic stability, we first consider the set of states where $\dot{V} = 0$,

$$
\begin{aligned}
\mathcal{M}: &= \{(\tilde{W}_I, \tilde{W}_{II}, \tilde{q}) : \dot{V} = 0\} \\
&= \{(\tilde{W}_I, \tilde{W}_{II}, \tilde{q}) : \tilde{W}_I{=}\tilde{W}_{II}{=}0 \text{ or } \tilde{q}{=}-q_0^*\}.
\end{aligned}
$$

By LaSalle's Invariance Principle [17], [22], trajectories of (9) converge to the largest invariant set contained in $\mathcal{M}$. We will then prove that the only invariant set contained in $\mathcal{M}$ is the equilibrium point (0, 0, 0). If $(\tilde{W}_I, \tilde{W}_{II}, \tilde{q})$ is equal to $(0, 0, \tilde{q})$ or $(\tilde{W}_I, \tilde{W}_{II}, -q_0^*)$, by using (9), we can conclude that $(\tilde{W}_I(t^+), \tilde{W}_{II}(t^+), \tilde{q}(t^+))$ is not in $\mathcal{M}$, which implies that no trajectory can stay in $\mathcal{M}$, other than the point (0, 0, 0). Therefore, asymptotic stability is obtained, which we now summarize:

Theorem 2: For any $K_p > 0$, the equilibrium point of (9) is asymptotically stable for any positive pairs (α_1, β_1) and (α_2, β_2).

From (8), we can also get the relationship between W_I^* and W_{II}^* as follow:

$$
\frac{W_I^*}{W_{II}^*} = [\frac{\alpha_1(1+\beta_1)(1-\beta_2)}{\alpha_2(1-\beta_1)(1+\beta_2)}]^{1/2} \tag{10}
$$

This means that the ratio of W_I^* and W_{II}^* depends only on the choices of (α_1, β_1) and (α_2, β_2), and regardless of the number of flows in the network and their initial states. Therefore, by choosing suitable (α_1, β_1) and (α_2, β_2), we can guarantee the fair share of bottleneck bandwidth for each flow. For AIMD(α, β) flows to be TCP-friendly (co-existing TCP and AIMD flows obtain the same share of bottleneck bandwidth), the necessary and sufficient condition is

$$
\alpha = \frac{3(1-\beta)}{1+\beta}. \tag{11}
$$

In the Internet, different types of services are provided with different resource requirements. Sometimes, they may require to share the bandwidth of the Internet with different weights. Eq. (10) indicates that we can easily adjust the AIMD parameters of the end systems to provide differentiated services according to the application QoS requirements. For instance, to let the throughput of an AIMD(α_1, β_1) flow be k times that of an AIMD(α_2, β_2) flow, the AIMD parameter pairs should satisfy

$$
\frac{\alpha_1}{\alpha_2} = \frac{k^2(1-\beta_1)(1+\beta_2)}{(1+\beta_1)(1-\beta_2)}. \tag{12}
$$

We can also extend our results to the case when more than two types of flows exist in the same network. Suppose there are M different types of flows (α_1, β_1), (α_2, β_2), $\cdots$, (α_m, β_m) sharing the resources, with the number $N_1, N_2, \cdots, N_m$, respectively, then those flows can be mathematically modeled as,

$$
\begin{aligned}
\frac{dW_I(t)}{dt} &= \frac{\alpha_1}{R} - \frac{2(1-\beta_1)}{1+\beta_1}\cdot\frac{W_I(t)^2}{R}\cdot K_p q(t) \\
\frac{dW_{II}(t)}{dt} &= \frac{\alpha_2}{R} - \frac{2(1-\beta_2)}{1+\beta_2}\cdot\frac{W_{II}(t)^2}{R}\cdot K_p q(t) \\
&\cdots\cdots\cdots\cdots \\
\frac{dW_M(t)}{dt} &= \frac{\alpha_m}{R} - \frac{2(1-\beta_m)}{1+\beta_m}\cdot\frac{W_M(t)^2}{R}\cdot K_p q(t) \\
\frac{dq(t)}{dt} &= \begin{cases} \frac{N_1W_I(t)+\cdots+N_mW_M(t)}{R} - C & q > 0 \\ \{\frac{N_1W_I(t)+\cdots+N_mW_M(t)}{R} - C\}^+ & q = 0 \end{cases}
\end{aligned} \tag{13}
$$

With (13), choose a positive-definite Lyapunov function as

$$
\begin{aligned}
&V(\tilde{W}_I(t), \tilde{W}_{II}(t), \cdots, \tilde{W}_M(t), \tilde{q}(t)) \\
&= \frac{(1+\beta_1)N_1}{2(1-\beta_1)W_I^{*2}}\cdot\tilde{W}_I^2(t) + \frac{(1+\beta_2)N_1}{2(1-\beta_2)W_{II}^{*2}}\cdot\tilde{W}_{II}^2(t) \\
&\quad + \cdots + \frac{(1+\beta_m)N_m}{2(1-\beta_m)W_M^{*2}}\cdot\tilde{W}_M^2(t) + K_p\tilde{q}^2(t)
\end{aligned}
$$

where $\tilde{W}_i(t)$, i=1, 2, $\cdots$, m, and $\tilde{q}(t)$ have the same meaning as before. Then,

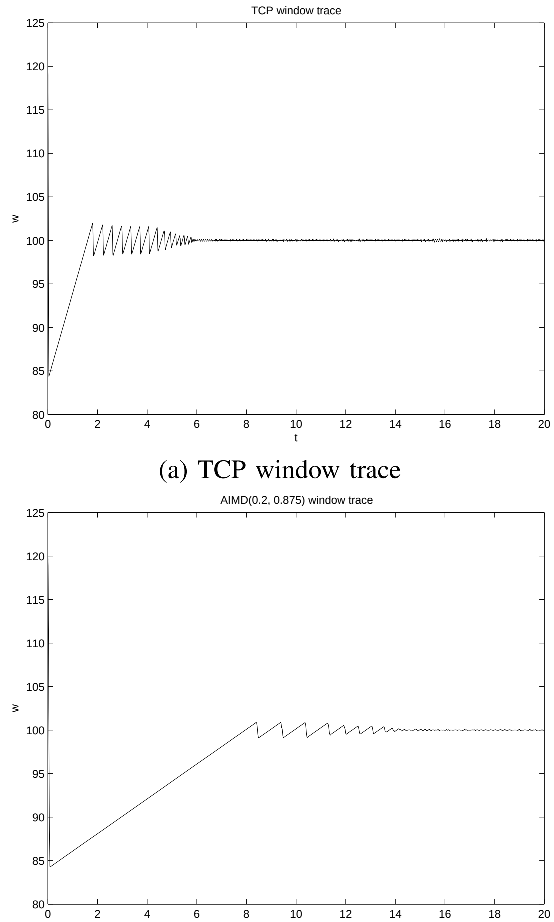
$$
\begin{aligned}
\dot{V} &= \frac{(1+\beta_1)N_1}{(1-\beta_1)W_I^{*2}}\tilde{W}_I\dot{\tilde{W}}_I + \frac{(1+\beta_2)N_2}{(1-\beta_2)W_{II}^{*2}}\tilde{W}_{II}\dot{\tilde{W}}_{II} \\
&\quad + \cdots + \frac{(1+\beta_m)N_M}{(1-\beta_m)W_M^{*2}}\tilde{W}_M\dot{\tilde{W}}_M + 2K_p\tilde{q}\dot{\tilde{q}} \\
&= -\frac{2N_1K_p}{W_I^{*2}R}\tilde{W}_I^2(\tilde{W}_I+2W_I^*)(\tilde{q}+q_0^*) - \cdots \\
&\quad -\frac{2N_mK_p}{W_M^{*2}R}\tilde{W}_M^2(\tilde{W}_M+2W_M^*)(\tilde{q}+q_0^*) \\
&\leq 0
\end{aligned}
$$

We can obtain its asymptotic stability by LaSalle's Invariance Principle, and thus have the following theorem.

Theorem 3: For any $K_p>0$, the equilibrium point of system(13) is asymptotically stable for any positive pairs (α_1, β_1), (α_2, β_2), $\cdots$, (α_m, β_m).

C. Numeric results

Matlab is used to obtain the numeric results of our analysis. The behaviors of window trace and queue length of 100 TCP flows and 100 AIMD(0.2, 0.875) flows with respect to time are given in Figs. 3 and 4, respectively. The parameters used are $C = 100000(packet/sec)$, $R = 0.1(sec)$, $K_p = 0.0001$, and $min_{th} = 200(packets)$. For the TCP-friendliness issue, we study the case when 100 TCP flows and 24 AIMD(0.2, 0.875) flows share the bottleneck, and the numeric results are shown in Fig. 5. These results show that when the flows in the network possess the same (α, β) parameter pair, either TCP or AIMD(0.2, 0.875), their window sizes and the bottleneck queue lengths converge to some certain values, i.e., the equilibrium points we derived in the previous analysis. When TCP and AIMD(0.2, 0.875) flows co-exist, they will fairly share the link capacity in the steady state, since (0.2, 0.875) satisfies the TCP-friendly condition (11). Thus, the numeric results do validate the theorems obtained in this report.



(a) TCP window trace

(b) AIMD(0.2, 0.875) window trace

Fig. 3. Window Trace

IV. Performance Evaluation

To further verify the analysis and evaluate the performance of TCP and TCP-friendly AIMD flows, extensive simulations have been conducted using the Network Simulator (*NS-2*) [18]. The logical simulation topology is the widely used shared bottleneck topology. The following parameters are used for simulations unless otherwise explicitly stated. The routers adjacent to the bottleneck link are RED-capable: all packets will be queued when the average queue length is less than 200 packets, and the packets will be discarded with probability K_p times the current average queue length minus 200. The packet size of all flows is $1,250$ bytes. The bottleneck link capacity is 1 Gbps, equivalent to $100,000$ packet/sec. To avoid the *phase effect* among competing flows, the round-trip time of each flow is made slightly different, ranging from $R-1$ ms to $R+1$ ms.

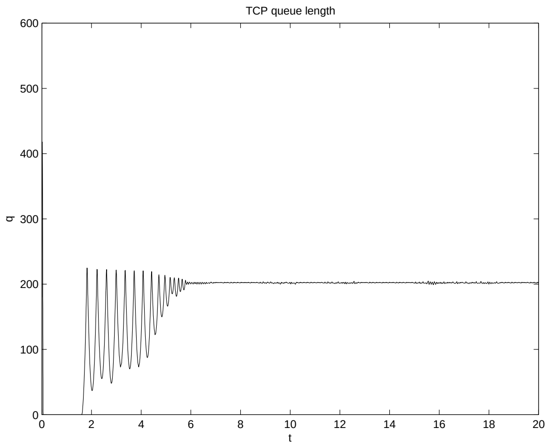
A. TCP

To examine the stability of TCP/RED, let 100 TCP connections be initialized within 10 ms to emulate a burst of traffic. Figs. 6 (a) and (b) show the bottleneck queue length and the average[3] window size of the TCP flows, respectively, with $K_p = 0.0001$ and $R = 100$ ms. It can be seen that within 5 seconds, the bottleneck average queue length is stabilized, and the instantaneous queue length oscillates between 0 and 300 packets, so the queuing delay varies between 0 and 3 ms. In the worst scenario, when most TCP flows are in the slow-start stage, the traffic load suddenly exceeds the bottleneck capacity with a large margin, the RED queue is built up, with the maximum value equal to $1,850$ packets and the maximum queuing delay 18.5 ms. Within 5 seconds, the instantaneous window size oscillates periodically between 85 and 105.

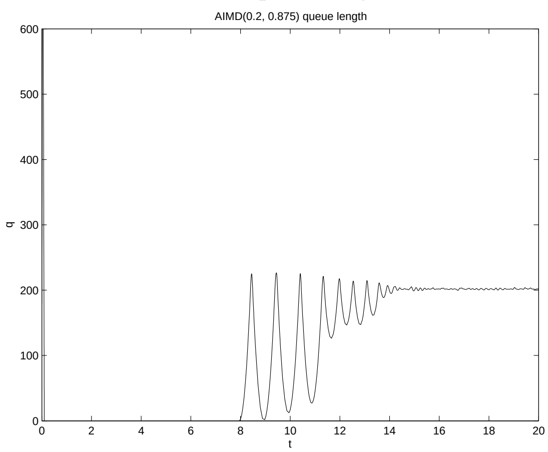
The analytical model and numeric results where the ensemble averages of window size and queue length are asymptotically stable; here, the simulation results show that the instantaneous window size and queue length oscillate periodically in the steady state. This is because, the analytical model in Section III captures the average behavior of AIMD/RED, and it cannot capture the microscope TCP behavior within a round; while the simulation results show the instantaneous window size and queue length. In the simulation results, window size and queue length oscillate *periodically* with almost unchanged amplitude in the steady state. If we average the values of window size and queue length over a round, the average values remain constant in steady-state, which confirms the analytical results that the average values are asymptotically stable.

Ideally, to fully utilize the link capacity, the TCP window size should converge to $W_0^* = 100$ without oscillation. The smaller the oscillation amplitude, the better the link utilization. Since TCP flows keep on increasing their window size at a constant pace in the congestion avoidance stage, with a smaller q_0^*, the traffic load will overshoot the link capacity more severely, and more packets will be discarded.

[3]For the simulation results, the *average* window size means the average of a number of flows' instantaneous window size. In the analytical model in Section III, W is the *ensemble average* of a flow's window size.

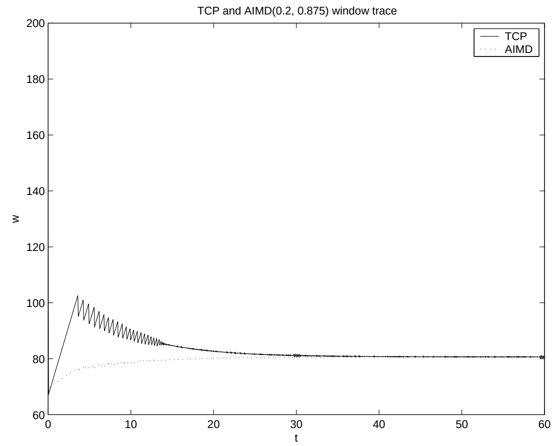


(a) TCP queue length

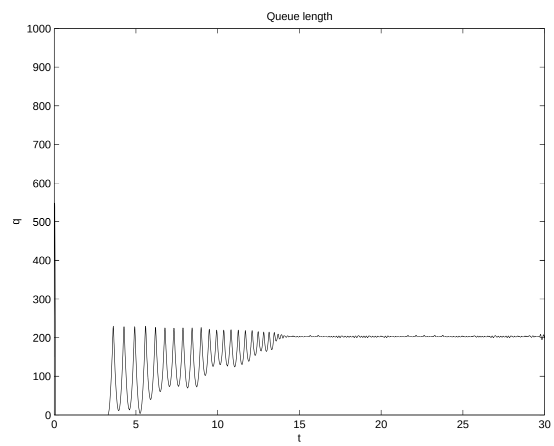


(b) AIMD(0.2, 0.875) queue length

Fig. 4. Queue Length



(a) TCP and AIMD(0.2, 0.875) window trace



(b) Queue length

Fig. 5. TCP-friendliness

Consequently, more flows will reduce their window size by half simultaneously, which results in larger oscillation of the average window size. According to the analytical model, q_0^* is inversely proportional to K_p. To reduce the oscillation and increase the link utilization, we set K_p to a smaller value (0.00002). Simulation results in Figs. 7 show that the window size oscillates with a much smaller amplitude, around 95 to 105, and the bottleneck queue length is slightly higher than that with $K_p = 0.0001$. In other words, with a smaller value of K_p, the network resources are more efficiently utilized, at the cost that the average queue length and queuing delay are slightly increased.

We further study the performance of TCP flows with a larger value of R. Figs. 8 and 9 show the simulation results with R equal to 200 ms and 400 ms, respectively. With a larger value of R, it takes longer time for the system to converge to the steady state, and the link utilization during the transient state is low. On the other hand, as shown in Figs. 6, 8 and 9, the average window size over R remains the same, and link utilization in steady state are independent of R.

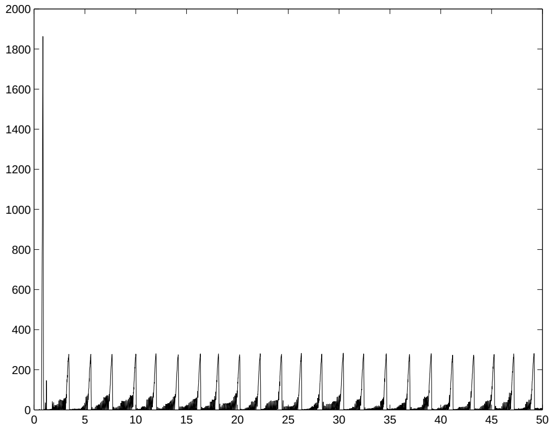
On the other hand, all simulation results in Figs. 6, 8 and 9 show that the instantaneous window size and queue length oscillate *periodically* in the steady state, so the average window size and queue length over a round are asymptotically stable.

B. AIMD

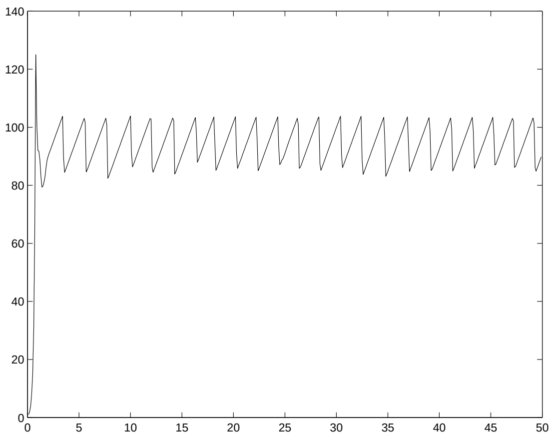
We then examine the stability and performance of AIMD-controlled flows. Figs. 10-12 show the bottleneck queue length and the average window size of 100 AIMD flows, with (α, β) pair equal to $(0.09677, 0.9375)$, $(0.2, 0.875)$, and $(0.4286, 0.75)$, respectively. It can be seen that with smaller value of α and larger value of β, it takes longer time for the system to converge to the steady state, and the link utilization during the transient stage is low; however, in the steady-state, the oscillation amplitudes of the instantaneous window size and queue length are smaller. In other words, with smaller value of α and larger value of β, the queuing delay jitter is smaller, and the link utilization in steady state is higher.

C. TCP-friendliness

To study the fairness among TCP and AIMD flows, let 100 TCP flows and 24 AIMD flows share a link with capacity 1.1 Gbps ($11,000$ packet/sec). We compare the average window size of TCP and AIMD flows in Figs. 13-15. The AIMD parameter pairs, (0.09677, 0.9375), (0.2, 0.875),

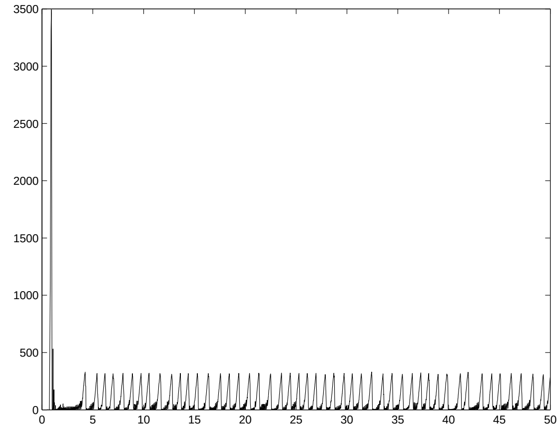


(a) Queue length

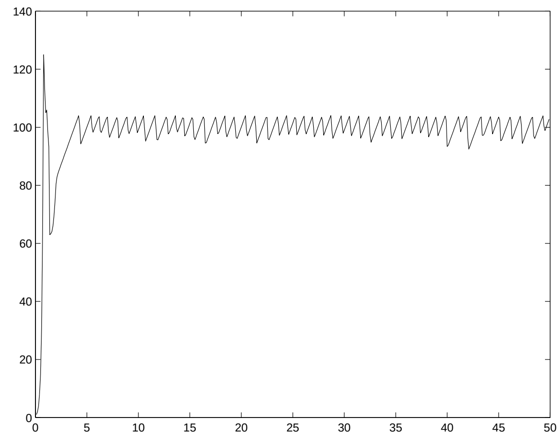


(b) Window trace

Fig. 6. TCP, $K_p = 0.0001$, $R = 100$ms



(a) Queue length



(b) Window trace

Fig. 7. TCP, $K_p = 0.00002$, $R = 100$ms

$(0.4286,\ 0.75)$, satisfy the TCP-friendly condition. The simulation results show that the average window size of co-existing TCP flows and that of AIMD flows are close to each other; therefore, the throughputs of TCP flows and that of AIMD flows are similar, and they can fairly share the network resources in a distributed manner. The simulation results and the numerical results show the same tendency, which verify the TCP-friendly condition derived.

D. Summary

The extensive simulation results demonstrate that

- TCP/RED and AIMD/RED systems are stable;
- to reduce the oscillation amplitude in the steady state in order to reduce delay jitter and increase link utilization, K_p should be small. On the other hand, a small value of K_p will lead to a larger value of q_0^* or average queuing delay. Therefore, K_p should be appropriately chosen to make the tradeoff between delay jitter, link utilization, and average delay;
- the TCP-friendly condition derived in the report is accurate and can be used as a guideline to choose AIMD parameter pairs for heterogeneous multimedia applications;
- for TCP-friendly AIMD flows, the larger the β, the smaller the oscillation amplitude in the steady state, which is desired. However, it takes longer time for the system to converge to the steady state;
- for links with larger propagation delay and link capacity product, it takes longer time for the system to converge to the steady state.

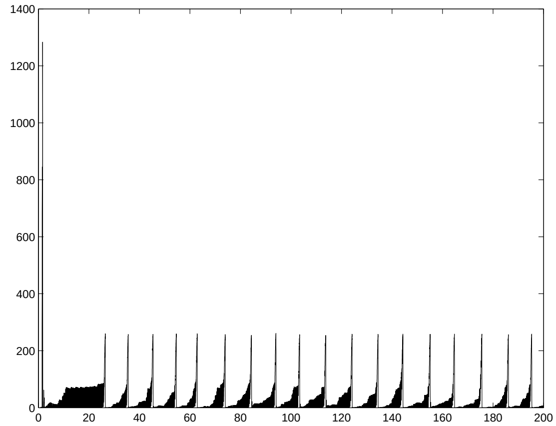
Since the link utilization during the transient state is low, further investigation on how to make the system quickly converge to the steady state is desired. Our future work will study how to appropriately use impulse-based control functions to solve the problem.

V. Conclusions

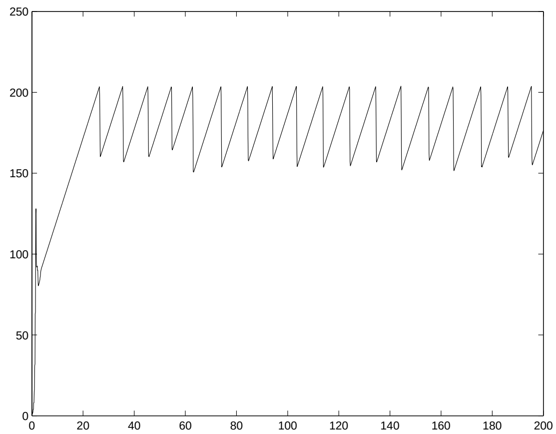
In this report, we have first generalized a class of TCP/RED model and then studied the stability property of this dynamical system. Sufficient conditions have been given for the asymptotic stability of the system mathematically. TCP-friendliness issue has also been discussed for multiple flows with different AIMD parameters. Numerical and simulation results have been given to verify the model and validate the analysis.

ACKNOWLEDGEMENT

This work has been supported in part by a research grant from the Natural Science and Engineering Council of Canada.

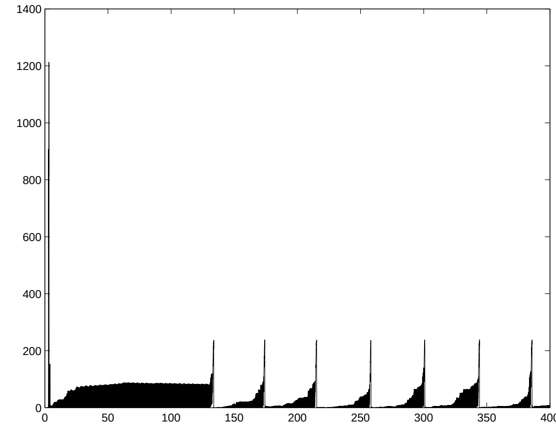


(a) Queue length

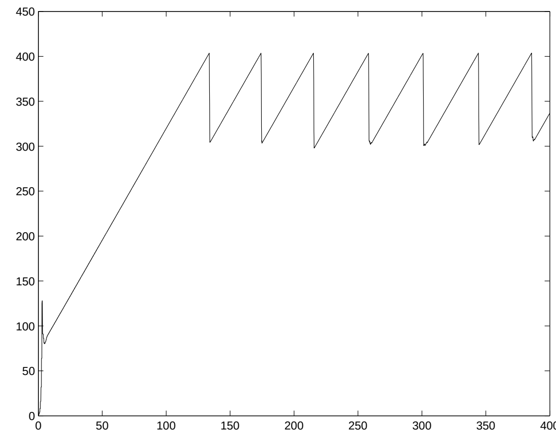


(b) Window trace

Fig. 8. TCP, $K_p = 0.0001$, $R = 200$ms



(a) Queue length



(b) Window trace

Fig. 9. TCP, $K_p = 0.0001$, $R = 400$ms

REFERENCES

[1] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

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

[3] S. H. Low, F. Paganini and J. C. Doyle, "Internet Congestion Control," *IEEE Control Systems Magazine*, 22(1):28-43, Feb. 2002.

[4] D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," *Journal of Computer Networks and ISDN,* Vol. 17, No. 1, pp. 1-14, June 1989.

[5] K. B. Kim and S. H. Low, "Analysis and Design of AQM for Stabilizing TCP," *Caltech Technical Report* caltechCSTR:2002.009, March, 2002.

[6] Steven H. Low, "A duality model of TCP and Queue management algorithms," *IEEE Trans. on Networking*, Vol. 11, No. 4, Aug. 2003.

[7] V. Misra, W. B. Gong, D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows," *Proc. of ACM/SIGCOMM*, pp. 151–160, 2000.

[8] C. V. Hollot, V. Misra, D. Towsley, W. B. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. on Automatic Control*, Vol. 47, No. 6, pp. 945-959, June 2002.

[9] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," *Proc. of IEEE INFOCOM*, Anchorage, pp. 1726–1734, April 2001.

[10] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoretic analysis of RED," *Proc. of IEEE INFOCOM*, Anchorage, pp. 1510–1519, April 2001.

[11] C. V. Hollot and Y. Chait, "Non-linear stability analysis of a class of TCP/AQM networks," *Proc. of IEEE Conference on Decision and Control*, Orlando, Florida USA, Dec. 2001.

[12] Y. Chait, C. V. Hollot, V. Misra, S. Oldak, D. Towsley, and W. B. Gong, "Fixed and adaptive model-based controllers for Active Queue Management," *Proc. of ACC*, 2001.

[13] S. Floyd, M. Handley, and J. Padhye, A comparison of equation-based and AIMD congestion control, May 2000. Available `http://www.aciri.org/tfrc/tcp- friendly.TR.ps` .

[14] Y. R. Yang and S. S. Lam, General AIMD congestion control. Technical Report TR-2000-09, University of Texas, May 2000. A shorter version appeared in Proceedings of ICNP'00, Osaka, Japan, November 2000.

[15] L. Cai, X. Shen, J. Pan, and J. W. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *IEEE Trans. on Multimedia*, Vol. 7, No. 2, pp. 339–355, April 2005.

[16] L. Cai, X. Shen, J. W. Mark, and J. Pan, "QoS support for multimedia traffic in wireless/wired networks using TCP-friendly AIMD protocols," *IEEE Trans. on Wireless Communications*, Vol. 5, issue 2, pp. 469–480, February 2006.

[17] J. E. Slotine and W. Li, *Applied Nonlinear Control,* Prentice Hall, Englewood Cliffs, NJ, 1991.

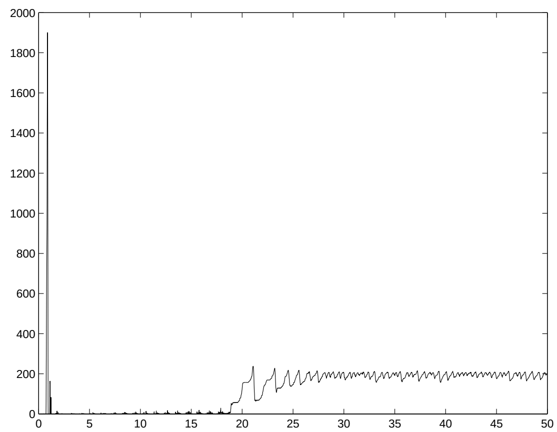
[18] S. Floyd and S. McCanne, "Network Simulator," LBNL public domain software. Available via ftp from ftp.ee.lbl.gov. NS-2 is available in http://www.isi.edu/nsnam/ns/.

[19] V. Firoiu and M. Borden, "A Study of Active Queue Management for Congestion Control," *In Proceedings of Infocom*, 2000.

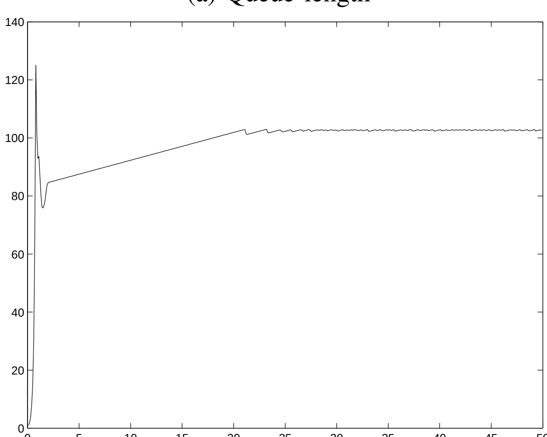
[20] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, Vol.1, August 1997.

[21] D. Lin and R. Morris, "Dynamics of Random Early Detection," *Proceedings of ACM/SIGCOMM*, 1997.

[22] H. K. Khalil, *Nonlinear Systems,* Upper Saddle River, N.J.: Prentice Hall, c2002.

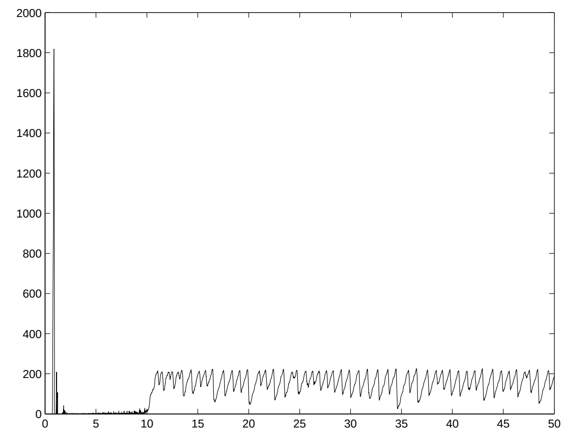


(a) Queue length

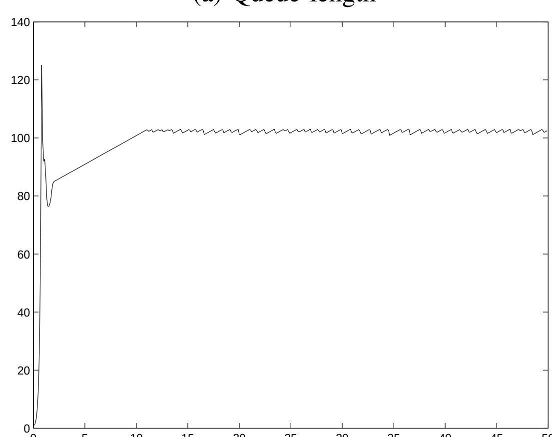


(b) Window trace

Fig. 10. AIMD$(0.09677, 0.9375)$, $K_p = 0.0001$, $R = 100$ms

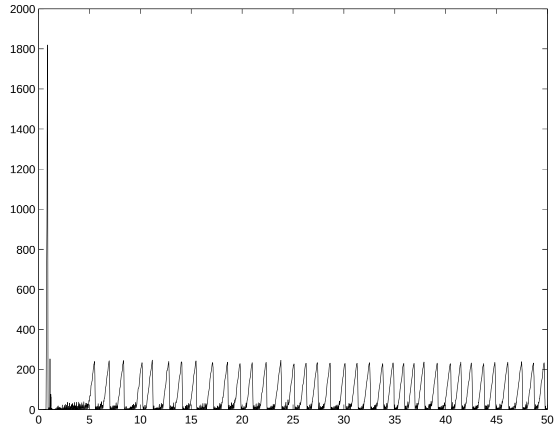


(a) Queue length

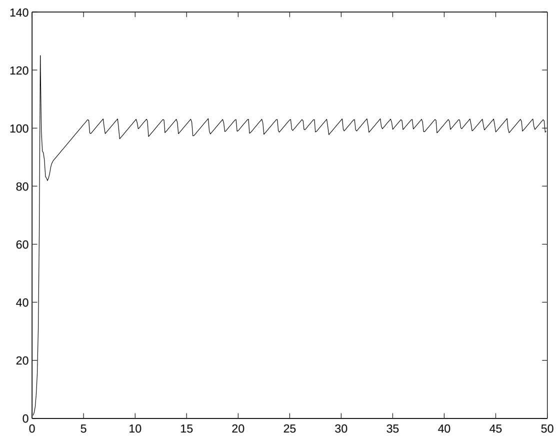


(b) Window trace

Fig. 11. AIMD$(0.2, 0.875)$, $K_p = 0.0001$, $R = 100$ms



(a) Queue length



(b) Window trace

Fig. 12. AIMD$(0.4286, 0.75)$, $K_p = 0.0001$, $R = 100$ms

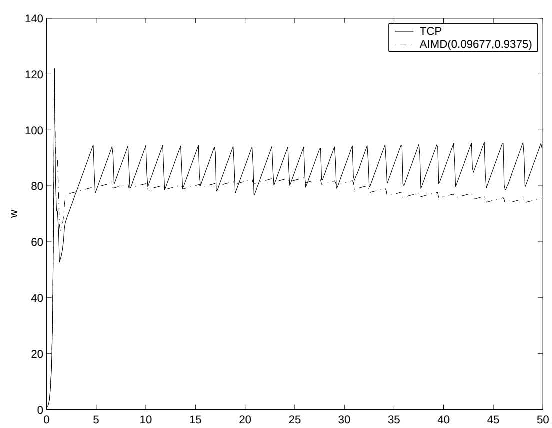


Fig. 13. TCP and AIMD$(0.09677, 0.9375)$ window trace

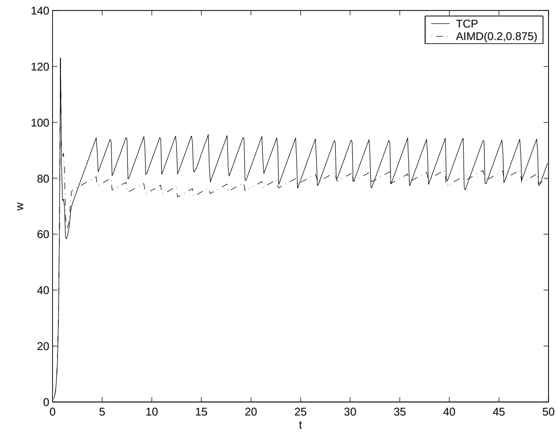


Fig. 14. TCP and AIMD$(0.2, 0.875)$ window trace

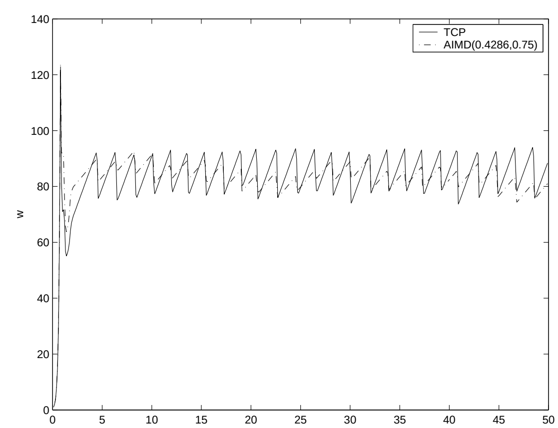


Fig. 15. TCP and AIMD$(0.4286, 0.75)$ window trace