A Control Theoretic Analysis of RED

C.V. Hollot, Vishal Misra, Don Towsley and Wei-Bo Gong

Abstract— **In this paper we use a previously developed nonlinear dynamic model of TCP to analyze and design Active Queue Management (AQM) control systems using RED. First, we linearize the interconnection of TCP and a bottlenecked queue and discuss its feedback properties in terms of network parameters such as link capacity, load and round-trip time. Using this model, we next design an AQM control system using the random early detection (RED) scheme by relating its free parameters such as the low-pass filter break point and loss probability profile to the network parameters. We present guidelines for designing linearly stable systems subject to network parameters like propogation delay and load level. Robustness to variations in system loads is a prime objective. We present ns simulations to support our analysis.**

I. INTRODUCTION

In [1] leading researches in the networking community have proposed implementation of RED in IP routers for Active Queue Management (AQM). It is believed that RED will alleviate problems related to synchronization of flows and also provide some notion of quality of service by intelligent dropping. The analysis of RED has generated several interesting papers. Tuning of RED parameters has been an inexact science for sometime now, so much so that some researchers have advocated against using RED because of this tuning difficulty [2], [3]. Numerous RED variants [4], [5], [6], [7] have also been proposed, perhaps motivated by the difficulty in understanding the dynamics of RED completely.

In [8], the authors investigated the issue of recommendations of RED parameters, and gave thumb rules and guidelines for choosing them. In this paper we investigate similar issues, however from a more formal, control theoretic standpoint. We use a previously developed model of TCP and RED dynamics [9] as a starting point to perform our analysis. The inherently non-linear model presented in that paper is converted to a linear system via the technique of linearization, and we subsequently apply the well developed tools in classical linear feedback control theory. We are able to give guidelines on designing linearly stable systems as well as provide metrics indicating stability and robustness of the linear system. Our analysis also reveals tradeoffs in various parameter choices. We support our analysis via nonlinear simulations using the well known `ns-2` simulator [10].

The rest of the paper is organized as follows. In Section II we develop the linearized model for the AQM control system. Section III deals with with application of the earlier developed

model to an AQM system implementing RED. We present design guidelines in this section. Next, in Section IV we present simulation results done using `ns-2` that verify our analysis and design recommendations. Finally, we present our conclusions in Section V.

II. MODEL

We begin our discussion of AQM by first introducing a dynamic model for TCP's congestion-avoidance mode.

A. A fluid-flow model of TCP behavior

In [9], a dynamic model of TCP behavior was developed using fluid-flow and stochastic differential equation analysis. Simulation results demonstrated that this model accurately captured the dynamics of TCP. In this paper we use a simplified version of that model which ignores the TCP timeout mechanism. This model relates the average value of key network variables and is described by the following coupled, nonlinear differential equations:

$$
\begin{aligned}
\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t - R(t))}{2R(t - R(t))}p(t - R(t)) \\
\dot{q}(t) &= \frac{W(t)}{R(t)}N(t) - C
\end{aligned} \tag{1}
$$

where $\dot{x}$ denotes the time-derivative of x and

$$
\begin{aligned}
W &\doteq \text{expected TCP window size (packets);} \\
q &\doteq \text{expected queue length (packets);} \\
R &\doteq \text{round-trip time} = \frac{q}{C} + T_p \text{ (secs);} \\
C &\doteq \text{link capacity (packets/sec);} \\
T_p &\doteq \text{propagation delay (secs);} \\
N &\doteq \text{load factor (number of TCP sessions);} \\
p &\doteq \text{probability of packet mark/drop.}
\end{aligned}
$$

The queue length q and window-size W are positive and bounded quantities; i.e., $q \in [0, \bar{q}]$ and $W \in [0, \bar{W}]$ where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size respectively. Also, the marking probability p takes value only in $[0, 1]$. We illustrate these differential equations in the block diagram of Figure 1 which highlights TCP window-control and queue dynamics. We now approximate these dynamics by their small-signal linearization about an operating point to gain insight for the purposes of feedback control (AQM).

B. Linearization

Taking (W, q) as the state and p as input, the operating point (W_0, q_0, p_0) is defined by $\dot{W} = 0$ and $\dot{q} = 0$ so that

$$
\begin{aligned}
\dot{W} = 0 &\Rightarrow W_0^2 p_0 = 2 \\
\dot{q} = 0 &\Rightarrow W_0 = \frac{R_0 C}{N}
\end{aligned} \tag{2}
$$

Fig. 1. Block-diagram of TCP's congestion-avoidance flow-control mode.

where

$$R_0 \doteq \frac{q_0}{C} + T_p.$$

Assuming $N(t) \equiv N$ and $R(t) \equiv R_0$ as constants,[1] we linearize (1) about the operating point to obtain (see Appendix I for details)

$$
\begin{aligned}
\delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - R_0)) \\
&\quad - \frac{R_0 C^2}{2N^2}\delta p(t - R_0) \\
\dot{\delta q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t)
\end{aligned}
\tag{3}
$$

where

$$
\begin{aligned}
\delta W &\doteq W - W_0; \\
\delta q &\doteq q - q_0; \\
\delta p &\doteq p - p_0.
\end{aligned}
$$

Performing a Laplace transform on the differential equations, the linearized dynamics are illustrated in a block diagram form in Figure 2.

Fig. 2. Block-diagram of the linearized TCP connection.

Remarks 1:

[1]this assumption will be in force for the duration.

1. In [11] a model for TCP's window-control mechanism was developed and argued that it contained a Smith regulator structure; see [12]. However, our model in (3) and Figure 2 does not support this claim. Indeed, for the TCP window control in Figure 2 to act like a Smith regulator structure, the term $\frac{N}{R_0^2 C}e^{-sR_0}$ should be replaced with $-\frac{C^2}{2N}\frac{1}{s+\frac{1}{R_0}}e^{-sR_0}$.

2. In [13] a model for the TCP window-control mechanism was developed which is similar to that in (3). However, that model does not contain a queue dynamic. The dynamical system considered is thus a little different from ours, and hence their analysis and results do not agree with the conclusions we arrive in this paper.

3. In Appendix II we show that the delay term e^{-sR_0} in the TCP window-control dynamic in Figure 2 is not significant when

$$\frac{N}{R_0^2 C} \ll \frac{1}{R_0}.$$

Since

$$\frac{N}{R_0^2 C} = \frac{1}{W_0 R_0}$$

this delay term can be ignored if $W_0 \gg 1$. For typical network conditions, $W_0 \gg 1$ is a reasonable assumption and hence, for the remainder of the paper, we will ignore this delay term and consider the simplified dynamics

$$
\begin{aligned}
\delta\dot{W}(t) &= -\frac{2N}{R_0^2 C}\delta W(t) - \frac{R_0 C^2}{2N^2}\delta p(t - R_0) \\
\dot{\delta q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t)
\end{aligned}
\tag{4}
$$

as illustrated in the block diagram of Figure 3.

Fig. 3. Block-diagram of the linearized TCP connection when $W_0 \gg 1$.

4. The eigenvalues of the linearized TCP and queue dynamics (4) are, respectively,

$$-\frac{2N}{R_0^2 C} \left(\text{or } \frac{-2}{W_0 R_0}\right) \quad \text{and} \quad -\frac{1}{R_0}.$$

Since all the network parameters are positive quantities, these negative eigenvalues indicate that the equilibrium state of the nonlinear dynamics is locally asymptotically stable. An interpretation of the TCP window-control time constant $\frac{W_0 R_0}{2}$ comes from expressing the linearization of the $\delta\dot{W}$ equation above as:

$$\delta\dot{W}(t) = -\lambda_0 \delta W(t) - \frac{R_0 C^2}{2N^2}\delta p(t - R_0)$$

where λ_0 is the equilibrium packet-marking rate as discussed in [9]. Therefore, the window-control time constant can be equivalently expressed as $\frac{1}{\lambda_0}$. In equilibrium, $\dot{W} = 0$ implies that the *multiplicative* decrease in window size $\frac{1}{2}W_0\lambda_0$ balances its *additive* increase $\frac{1}{R_0}$. Consequently, $\lambda_0 = \frac{2}{W_0 R_0}$. It can be loosely interpreted as the average frequency of the TCP congestion-avoidance cycle.

5. Finally, it is interesting to note that linearization of the queue dynamic does not yield a pure integrator, as one may expect and as one sees in the literature (for example, [11]) but produces a leaky integrator with time constant R_0. This can be partially explained in noting that the flow into the queue is a function of the queue length. That is, this flow is $\frac{NW}{R_0}$ where part of the round-trip time R_0 is due to the queuing delay $\frac{q}{C}$.

III. THE AQM CONTROL PROBLEM

The objective of this section is to analyze the TCP dynamic described in (4) in terms of network parameters such as TCP load N, round-trip time R_0 and queue capacity C, and in terms of the feedback nature of AQM. We will also discuss performance objectives for AQM.

Using the linearized TCP model (4) an AQM control system can be modeled as in the block diagram of Figure 5 [2]. In this diagram P_{tcp} denotes the transfer function from loss probability δp to window size δW and P_{queue} relates δW to queue length q. The term e^{-sR_0} is the Laplace transform of the time-delay in the delayed loss probability $\delta p(t - R_0)$. In control-system language, we refer to the AQM Control Law block as the "controller" or "compensator" and the rest of the (uncompensated) system as the "plant". The goal of the compensator design is to provide a "stable" closed-loop system. However, there are concerns beyond stability which impact control design. Firstly, the system must have an acceptable transient response. Secondly, the compensator design should be robust to variations in model parameters and modeling errors. Hence, the goal of control engineers is to design systems with a margin of safety. These margins are called *stability margins*. There are two classical metrics to measure this relative stability. The first of those is the *gain margin*, which is the factor by which the open loop gain of a stable system must be changed to make the system unstable. If we look at Figure 1, the gain margin is roughly the uncertainty in the load level N that the design can tolerate. The second of those measures is the *phase margin*. The definition of the phase margin is a little bit more complex, but in the context of AQM we can interpret the phase margin as the amount of uncertainty in the round trip delay a design can sustain without becoming unstable. Stability margins of a system can be readily deduced from *Bode* plots. A Bode plot is the frequency response plot of the open-loop system. The magnitude and phase response of the system are plotted on a double log scale. The gain margin of a system is equal to the magnitude response of the system at the point where the phase response is -180°. The phase margin ϕ_m is defined as $\omega_{pm} - 180$ where ω_{pm} is the phase response at the

[2]This linearized control system assumes an infinite queue-length and allows queue length to take on negative values. While our subsequent analysis and design are based on this linear model, they are verified in nonlinear simulations which include these nonlinear constraints.

frequency where the magnitude response is unity (or 0 dB). The two quantities are shown in Figure 4. Intuitively, if we don't have positive margins, then the feedback control system starts behaving like a positive feedback system, i.e. one where the error gets amplified in the loop, leading to divergent and unstable behavior.
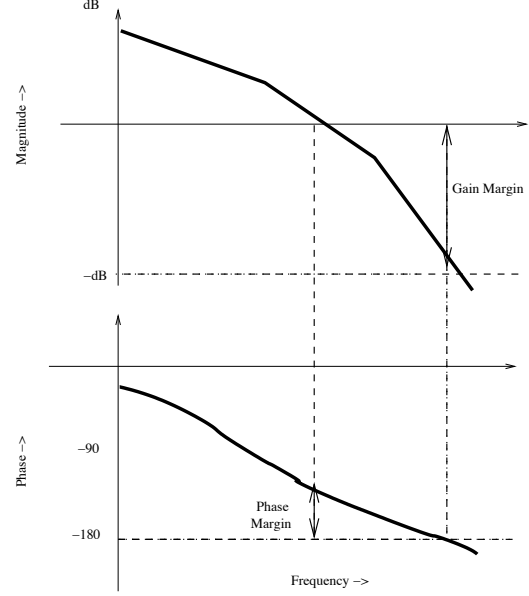

Fig. 4. Stability margins on the Bode plot

A. Plant dynamics

In Figure 5 we give a feedback control system depiction of AQM. The action of an AQM control law is to mark packets (with probability p) as a function of measured queue length q.
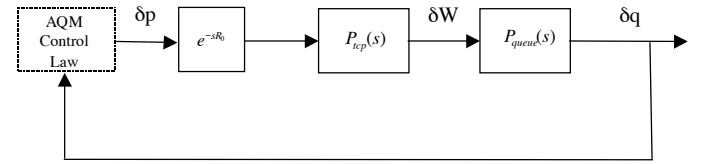

Fig. 5. AQM as feedback control.

From Figure 5, the plant transfer function, $P(s) = P_{tcp}(s)P_{queue}(s)$, can be expressed in terms of network parameters yielding:

$$P_{tcp}(s) = \frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0^2 C}};$$

$$P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}. \tag{5}$$

We refer to the two poles $-2N/(R_0^2 C)$ and $-1/R_0$ as p_{tcp} and p_{queue} respectively.

The plant dynamics, denoted by the transfer function $P(s)$, then relates how this packet-marking probability dynamically

affects the queue length. From (4) and Figure 3 we have

$$P(s) = \frac{(\frac{C^2}{2N})e^{-sR_0}}{(s + \frac{2N}{R_0^2C})(s + \frac{1}{R_0})}. \tag{6}$$

Remarks 2:
1. The high-frequency plant gain of $P(s)$ in (6) is $\frac{C^2}{2N}$. The variation in this gain as a function of TCP load N should be a concern in the design of AQM control schemes since it has direct bearing on stability, transient response and steady-state performance. Indeed, a small TCP load N increases the hi-frequency gain leading to decreased stability margins and increased oscillatory response. Conversely, larger TCP loads will tend to damp the closed-loop transient response.
2. Stable AQM in the face of the time-delay R_0 can place a hard limit on the closed-loop control bandwidth and consequently, on the achievable speed of transient response. Indeed, for stable behavior, closed-loop time constants are approximately bounded by $R_0/2$ seconds.

B. AQM performance objectives

As in any control system design, a first step is to pose performance objectives. For AQM, performance objectives include efficient queue utilization, regulated queuing delay and robustness.
1. *efficient queue utilization:* For efficient use, the queue should avoid overflow or emptiness. The former situation results in lost packets and undesired retransmissions, while an empty buffer under-utilizes the link. Both of these extremes should be avoided in both transient and steady-state operation.
2. *queuing delay:* The time required for a data packet to be serviced by the routing queue is called the queuing delay and is equal to $\frac{q}{C}$. This time, together with the propagation delay T_p, accounts for the network delay and it is desirable to keep small both the queuing delay and its variations. This calls for regulating to small queue lengths; however, doing so may result in link underutilization and this limitation presents a fundamental tradeoff to AQM design.
3. *robustness:* AQM schemes need to maintain closed-loop performance in face of varying network conditions. These conditions include variations in the number of TCP sessions N, variations in the propagation delay T_p and the introduction of short-lived into the queue.

C. Designing RED

An active queue-management (AQM) system can be modeled as the feedback control system shown in Figure 6. Here $P(s)e^{-sR_0}$ denotes the previously derived small-signal linearization of TCP-queue dynamics (linearized about queue-length q_0). $P(s)$ is $P_{tcp}(s)P_{queue}(s)$ derived previously. δp and δq denote perturbations in the *loss probability* and *queue length* respectively. In Figure 6 the transfer function $C(s)$ denotes an AQM control strategy such as *tail-drop* or *RED*.

Tail-drop is an on-off control strategy. In terms of our set-up in Figure 6, tail-drop amounts to the on-off action $\delta p \in \{0, 1\}$. It is known in control theory that such an on-off mechanism[3]

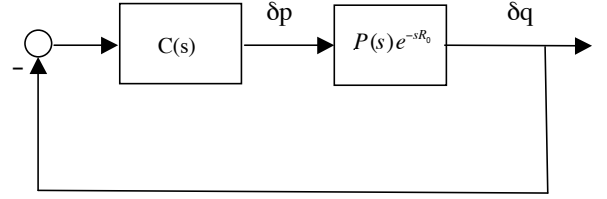[3]also referred to as *relay control* in the feedback literature.

Fig. 6. Block diagram of a linearized AQM control system

leads to oscillations (limit-cycles) that can exhibit complex and chaotic behavior; e.g., see [14]. Such oscillations may be undesirable in queue management, and RED was introduced to stabilize them.

A transfer-function model for RED is:

$$C(s) = C_{red}(s) = \frac{L_{red}}{s/K + 1}, \tag{7}$$

where

$$L_{red} = \frac{p_{max}}{max_{th} - min_{th}}; \quad K = \frac{\log_e(1 - \alpha)}{\delta},$$

$\alpha > 0$ is the *queue averaging* parameter and δ is the sample time; see [9]. In designing $C_{red}(s)$ to stabilize the AQM control system, variations in both the number of TCP sessions N and round-trip time R_0 should be taken into account. The variations in R_0 are due to a variable propagation time T_p where

$$R_0 = \frac{q_0}{C} + T_p.$$

Let's assume a range for the number of TCP sessions, say $N \geq N^-$, and the round-trip time, $R_0 \leq R^+$. The objective is to select RED parameters L_{red} and K in (7) to stabilize the linear control system in Figure 6 for all these N and R_0. The linear feedback control system in Figure 6 is stable if bounded exogenous inputs produce only bounded outputs. This in turn implies that responses to initial conditions will be bounded and converge exponentially to zero. Under this definition of stability, we give the following two propositions:

Proposition 1: *Let L_{red} and K satisfy:*

$$\frac{L_{red}(R^+C)^3}{(2N^-)^2} \leq \sqrt{\frac{\omega_g^2}{K^2} + 1} \tag{8}$$

where

$$\omega_g = 0.1 \min\left\{\frac{2N^-}{(R^+)^2C}, \frac{1}{R^+}\right\}. \tag{9}$$

Then, the linear feedback control system in Figure 6 using $C(s) = C_{red}(s)$ in (7) is stable for all $N \geq N^-$ and all $R_0 \leq R^+$.

Proof: Consider the frequency response of the compensated loop transfer function

$$
\begin{aligned}
L(j\omega) &\doteq C_{red}(j\omega)P(j\omega)e^{-j\omega R_0} \\
&= \frac{L_{red}\frac{(R_0C)^3}{(2N)^2}e^{-j\omega R_0}}{(\frac{j\omega}{K} + 1)(\frac{j\omega}{\frac{2N}{R_0^2C}} + 1)(\frac{j\omega}{\frac{1}{R_0}} + 1)}.
\end{aligned}
$$

From this and (9) we have:

$$L(j\omega) \approx \frac{L_{red}\frac{(R_0C)^3}{(2N)^2}e^{-j\omega R_0}}{\frac{j\omega}{K}+1}, \quad \forall \omega \in [0, \omega_g].$$

Now, given any $N \geq N^-$ and any $R_0 \leq R^+$,

$$|L(j\omega_g)| \leq \frac{L_{red}\frac{(R^+C)^3}{(2N^-)^2}}{\sqrt{\frac{\omega_g^2}{K^2}+1}}.$$

From this and (8) it follows that $|L(j\omega_g)| \leq 1$ for all $N \geq N^-$ and for all $R_0 \leq R^+$. Thus, the unity-gain crossover frequency is bounded above by ω_g. To establish closed-loop stability, we invoke the Nyquist stability criterion [12] and show that $\angle L(j\omega_g) > -180°$. To this end, we again use (9) to obtain

$$\angle L(j\omega_g) \geq \angle \frac{K_{red}\frac{(R^+C)^3}{(2N^-)^2}}{\frac{j\omega_g}{p_{red}}+1} - \omega_g R_0 \geq -90° - 0.1\frac{180°}{\pi} > -180°.$$

$\square$

Remarks 3:

1. The rationale behind this choice of parameters is to force $C_{red}(s)$ to dominate closed-loop behavior. This is done by making the closed-loop time constant ($\approx 1/\omega_g$) greater than either the TCP time-constant $\frac{(R^+)^2C}{2N^-}$ or the queue time-constant R^+.

2. Different choices of (L_{red}, K) satisfy the condition above. For example, Figure 7 illustrates a region of admissible parameters[4] when $R^+ = 0.25$ secs, $N^- = 40, 60$ and 80 flows and $C = 3750$ packets/sec.
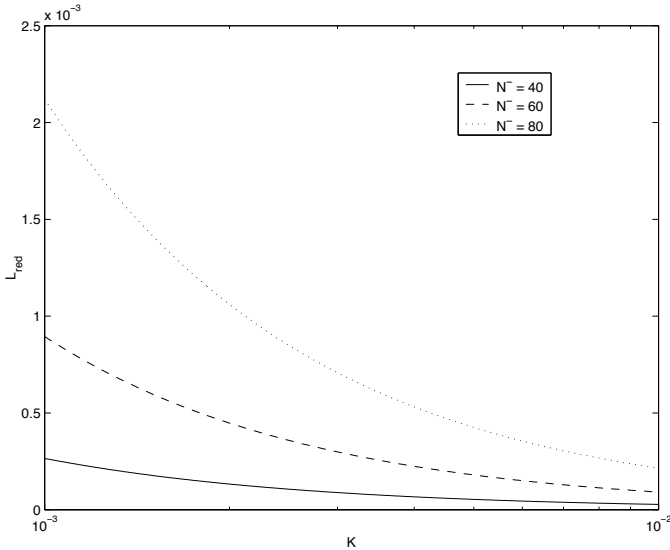
Fig. 7. Stabilizing RED parameters L_{red} and K.

3. This RED design is linearly robust to the network parameter variations $N \geq N^-$ and $R_0 \leq R^+$. The extent to which this feedback control system is stable to further variations in these parameters is characterized in Proposition 2 below.

[4]values of admissible pairs (L_{red}, K) lie below the graphs.

4. The multiplicative factor 0.1 in the choice of ω_g is the one which provides stability margins. If we choose a higher value than 0.1, we produce a controller with lower stability margins. The benefit of the more aggressive design is that it gives faster response times (due to an increase in ω_g).

5. It seems counter intuitive that the system is stable for all load levels *greater* than N^-. In fact, the system may oscillate if the load level makes the system go into a region where the operating point lies in the discontinuity region of the loss profile. This was studied in [8]. However, the *gentle_* mechanism recommended in [15] removes the instability related to the discontinuity.

6. At high load levels, the loss probability becomes sufficiently high to cause some flows to go into timeouts. We have ignored timeouts in our model and analysis. Timeouts should not impact stability from our analysis; indeed, they tend to make the system less oscillatory.

7. The analysis we presented considered a bound on the round trip time R^+. However, it does not follow that if the round trip time of some flows exceed this bound then the system becomes unstable. In fact, in the presence of heterogenous round trip times for the flows, this bound should be interpreted for what we call the *equivalent* round trip time of the flows. For simple (single bottleneck) cases, the *equivalent* round trip time is the harmonic mean of the individual round trip times of the flows. Consider a scenario with N flows having heterogenous round trip times R_i. The harmonic mean (R_{eq}) of the round trip times is given by

$$\frac{1}{R_{eq}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{R_i}$$

Now, at the bottlenecked router, the capacity is shared by these different flows. Thus, at equilibrium, ignoring timeouts and using the simplified $\sqrt{p}$ formula for throughput ([16], [17]) we have

$$C = \sum_{i=1}^{N}\frac{\sqrt{2}}{\sqrt{p_0}R_i}$$
$$= \sqrt{\frac{2}{p_0}}\sum_{i=1}^{N}\left(\frac{1}{R_i}\right)$$
$$= \sqrt{\frac{2}{p_0}}\left(\frac{N}{R_{eq}}\right).$$

Hence the system behaves in the mean as a system with N flows each having an identical *equivalent* round trip time that is R_{eq}.

Proposition 2: *Consider any RED controller $C_{red}(s)$ satisfying conditions (8) and (9) in Proposition 1. Then, the gain margin (GM) and phase margin (PM) of the linear control system in Figure 6 satisfy:*

$$GM \geq 5\pi; \quad PM \geq 85°.$$

Consequently, this linear control system will remain stable if either $R_0 < 15R^+$ or $N > \frac{1}{5\pi}N^-$.

Proof: Sharpening a phase computation made in the proof of Proposition 1 gives

$$\angle L(j\omega_g) \geq \angle \frac{K_{red}\frac{(R^+C)^3}{(2N^-)^2}}{\frac{j\omega_g}{p_{red}}+1} - \omega_g R_0 \geq -90° - 0.1\frac{180°}{\pi} \approx -95°.$$

Thus, $PM = 180° + \angle L(j\omega_g) \geq 85°$. The phase lag due to additional round-trip time delay ΔR is:

$$\angle e^{-j\omega_g \Delta R} = -\omega_g \Delta R.$$

From (9), $\omega_g \leq \frac{0.1}{R_0}$. Using this and $\omega_g \Delta R = 85(\frac{\pi}{180})$ gives $\Delta R \leq 14.8R$. For the gain margin computation we recall from the proof of Proposition 1 that

$$P(j\omega) \approx \frac{L_{red}\frac{(R_0C)^3}{(2N)^2}}{\frac{j\omega}{K}+1}, \quad \forall \omega \in [0, \omega_g].$$

Consequently,

$$\angle P(j\omega_g) \geq -90°.$$

Since

$$\angle e^{-j\omega R_0} = -90°$$

then $\angle L(j\frac{\pi}{2R_0}) \geq -180°$. Because $|L(j\omega_g)| \leq 1$, then $1/|L(j\frac{\pi}{2R_0})| \approx \frac{\frac{\pi}{2R_0}}{\omega_g}$ gives a lower-bound to the gain margin. Since $\omega_g \leq 0.1R_0$ then $GM \geq 5\pi$. $\square$

Example 1: Consider the case of network parameters: $C = 3750$ packets/sec [5], $N^- = 60$ and $R^+ = 0.2$ sec. From (9),

$$\omega_g = 0.1 \min\{0.5259, 4.0541\} = 0.053 \text{ rad/sec}.$$

For $K = 0.005$, we compute from (8):

$$L_{red} \leq \frac{(2N^-)^2}{(R^+C)^3}\sqrt{\frac{\omega_g^2}{K^2}+1} = 1.86(10)^{-4}.$$

Thus, one choice for C_{red} is

$$C_{red}(s) = \frac{1.86(10)^{-4}}{\frac{s}{0.005}+1}.$$

In terms of implementation, we can break this $C_{red}(s)$ down as

$$L_{red} = 1.86(10)^{-4}; \; K = 0.005$$

Now, for a link capacity of 3750 $packets/sec$, $\delta = 2.66(10)^{-4}$, yielding α, the averaging weight, as $1.33(10)^{-6}$. $L_{red} = p_{max}/(max_{th} - min_{th})$. Thus, if we choose p_{max} as 0.1, then the dynamic range of the average queue size is approximately 540 packets.

[5]corresponds to a 15 Mb/s link with average packet size 500 Bytes.

Remarks 4:
1. From the viewpoint of *steady-state regulation* it is desirable to select L_{red} in (8) as large as possible. By steady state regulation we mean that δq should go down to zero in steady state. However, under the RED mechanism the steady state value of the queue length (for a stable system) depends upon network conditions. Thus δq in our linear model never goes to zero which is not a desirable feature. We can reduce this steady state error by decreasing K. From (8), $K \to 0$ allows $L_{red} \to \infty$. In this limiting case we have

$$C_{red} = \frac{K}{s}$$

which corresponds to classical *integral* compensation.
2. A drawback in using RED for stabilizing queue length is that it has a low control-bandwidth ω_g which, from (9), must be less than the bandwidth of either the queue or TCP dynamic. Consequently, closed-loop responses are commensurately slow[6]. This can be improved by introducing lead compensation into RED. This results in a classical proportional-integral (PI) compensator

$$C_{PI} = K_{PI}\frac{(s/z+1)}{s}.$$

The design of such a compensator is discussed in a separate companion paper [18].

IV. SIMULATIONS

We verify our propositions via simulations using the ns simulator. Although our analysis was carried out with a linearized model, the simulations are non-linear in nature. We look at a single bottlenecked router running RED. In addition to infinite duration, greedy flows such as the one we model, we introduce short lived, http flows into the router, to generate a more realistic traffic scenario. The http flows were simulated using the http module provided with ns. The effect of flows which are very short lived is essentially that of introducing noise to the queue. The objective of the control system is to achieve full utilization of the bandwidth in the presence of these short lived flows. In all our plots we depict the time evolution of the instantaneous queue length, with the unit of the time axis being seconds.

A. Experiment 1

In the first experiment, we look at a queue with 60 ftp (greedy) flows, and 180 http sessions. The link bandwidth is 15 Mb/s, and the propagation delays for the flows range uniformly between 160 and 240 ms. We attempt to control the queue to provide a queueing delay of around 50-70 ms, and hence set the min_{th} and max_{th} of the queue as 200 and 250 respectively, with average packet size being 500 Bytes. The averaging weight and p_{max} is retained as "vanilla", i.e. the values which are the default in ns. The buffer has a maximum capacity of 800 packets. We set the *gentle_* parameter in RED as "on". The instantaneous queue length is shown in Figure 8. Observe the oscillating nature of the queue. It frequently goes down to zero, thereby

[6]In Example 1, the bandwidth was approximately 0.053 rad/sec. Closed-loop responses are dominated by the associated 20-second time constant.

under-utilizing the link. The large oscillations also add considerable jitter to the round trip times of the packets.
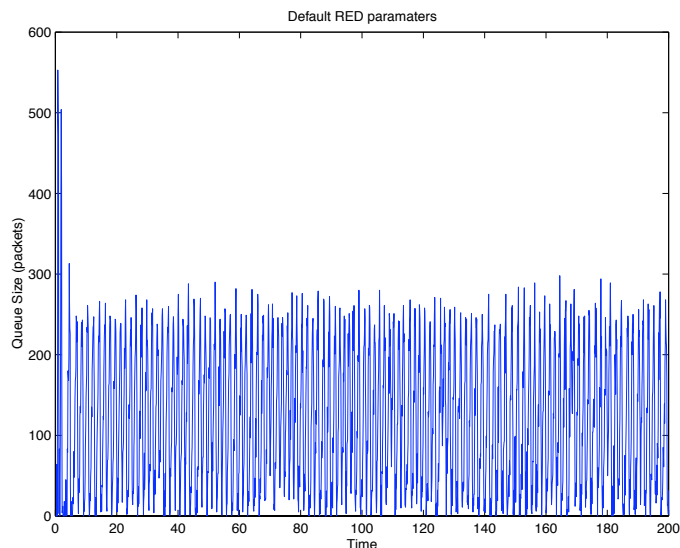

Fig. 8. Experiment 1

B. Experiment 2

Now we use the design as derived in Example 2. Thus, we set the averaging weight to be $1.33e - 6$, p_{max} to 0.1 and the dynamic range (min_{th}, max_{th}) to 150-700 packets. This should yield a stable mode of operation. The results are plotted in Figure 9. We indeed see that the system is stable, with small fluctuations about an operating level of the queue. The deterministic oscillations which were observed in the previous experiment are absent in this configuration. A point to note however, is that it takes a long time to "settle" to the operating point. This initialization is a major "disturbance" and one doesn't expect to encounter it in the normal mode of operation. However, it is related to the responsiveness of the control system to change in operating conditions. This slow response is related to a low value of ω_g that we use. [7] We can be more aggressive in our choice of ω_g, to get a faster response, however that will lead to lower stability margins. In the next experiment, we test a design which has a faster response time.

C. Experiment 3

We increase ω_g to 0.2 from 0.05. Recall that $1/\omega_g$ is approximately the time constant of the feedback loop. Thus, increasing ω_g should yield a faster response time. There are a number of ways of incorporating the effect of the increased ω_g. We look back at Proposition 1, and evaluate the effect of increasing ω_g on L_{red}. We could either maintain a constant ω_g/K ratio, thereby increasing K (which in turn means increasing α) and maintain L_{red}, or we could retain the value of K and increase L_{red} correspondingly. The latter can be achieved in two different ways - by

[7] The settling time is also increased due to the non-linear effects of the tail drop phenomena happening as the queue size reaches 800. The clamping at 800 results in a longer time for the average to "grow" to a value which can start providing loss feedbacks
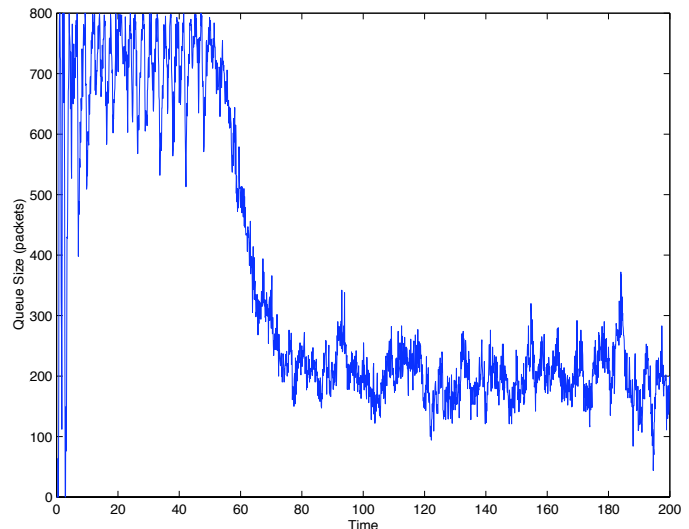

Fig. 9. Experiment 2

shrinking the dynamic range (recall that $L_{red} = \frac{p_{max}}{max_{th} - min_{th}}$) or by increasing p_{max}. In our first approach to obtain a faster re-
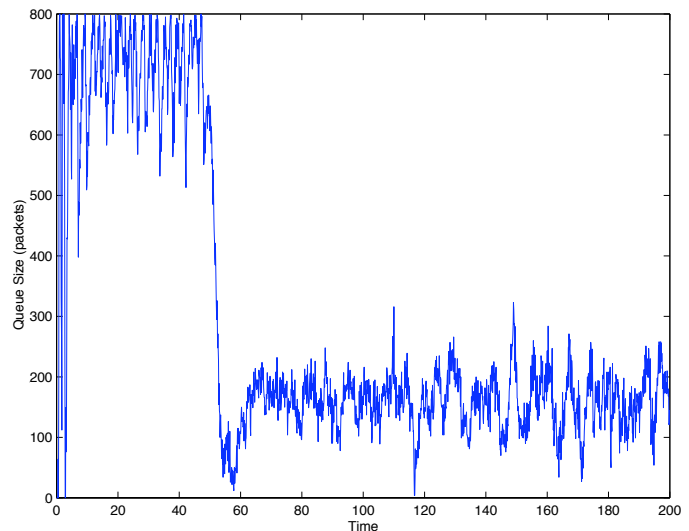

Fig. 10. Experiment 3

sponse time, we make the dynamic range shorter, reducing it to 150-250, from the earlier 150-700. The queue size for this scenario is plotted in Figure 10. As we observe, the queue settles to around the operating point after 60 seconds , compared to 80 seconds in Experiment 2. Notice also the somewhat large deviations in the queue size around the 100-160 second range in the simulation. This is because our aggressive design has reduced the stability margins. The presence of http flows introduces a stochastic element in the load level and hence we can expect to see those larger variations with lower stability margins.

The non-linear effects of tail drop result in a higher settling time than predicted by linear analysis. We can reduce the non-linear effects by moving the dynamic range down, however that leaves us with a lower margin of error as far as underutilization of the queue goes. We show the effects of lowering the dynamic
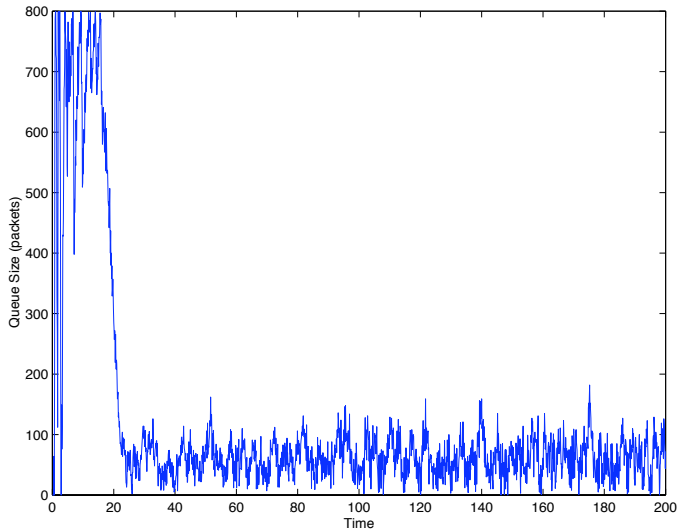
Fig. 11. Experiment 3a

range to 50-150 in Figure 11 (Experiment 3a). The settling time has indeed gone down, however the queue length stays closer to zero and the link is much more likely to be underutilized. Next we try the alternative approach of increasing K and retaining the original dynamic range of 150-700. This is shown in Figure 12 (Experiment 3b). As we can see, the settling time has come down and there is also better margin at the lower end of the queue. Thus, choosing a larger value of K appears to be a better option than lowering min_{th}.
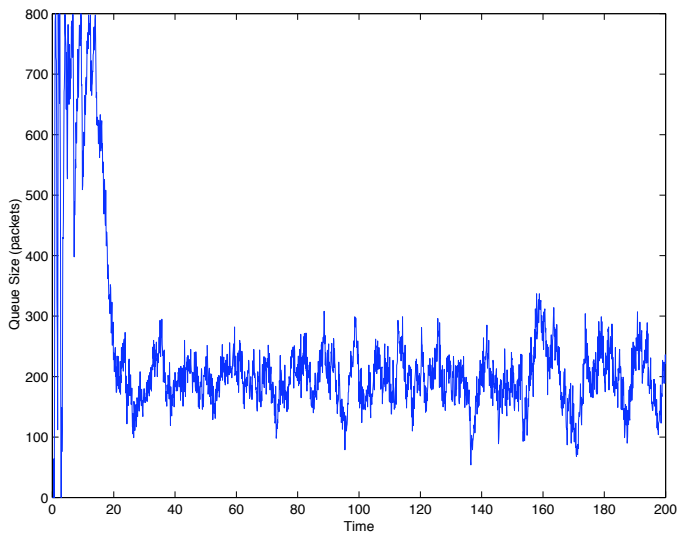
Fig. 12. Experiment 3b

D. Experiment 4

Next we begin to investigate the relative stability of the design, and look at issues relating to "over" designing or being too conservative. In Experiment 4, we take the base stable design, but double the round trip times of one fourth of the flows. The results are plotted in Figure 13. The system remains stable and the longer round trip times of the fraction of the flows don't

affect things too much.
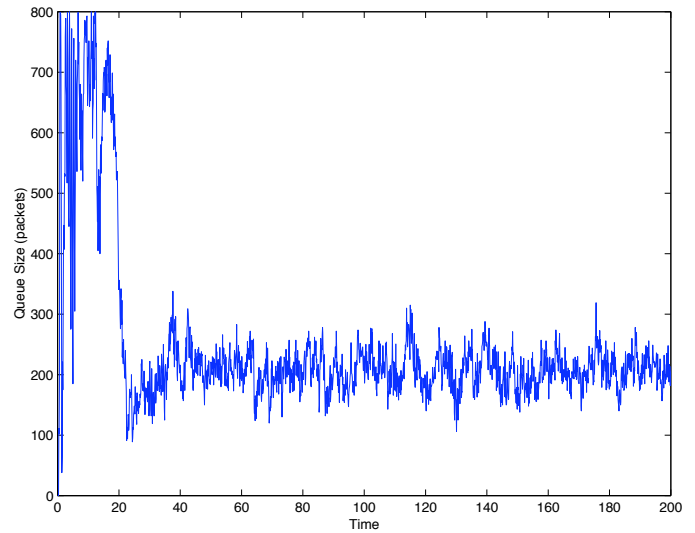
Fig. 13. Experiment 4

E. Experiment 5

In this experiment, we double the round trip times of *all* the flows. Thus, our system was designed for a much lower round trip time and it should show instability. The plot is shown in Figure 14. Observe the large oscillations. One thing to take from this experiment is that the phase margins for the non-linear system seem to be a little lower than the ones we derived for the linear system.
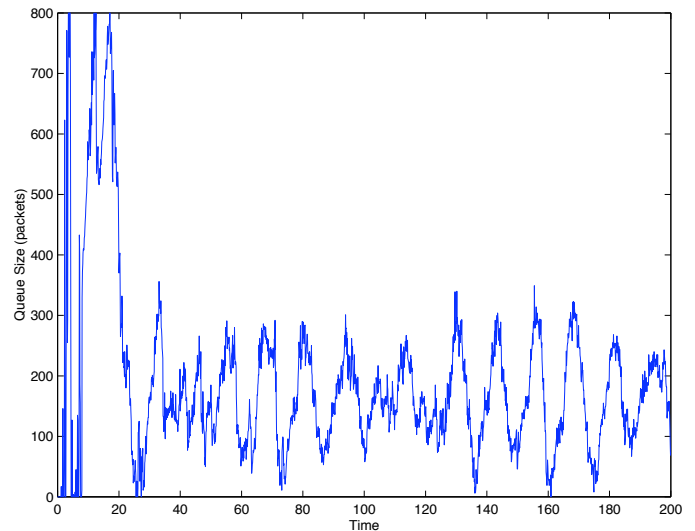

Fig. 14. Experiment 5

F. Experiment 6

Now we retain the round trip time, but vary the load level. First, we reduce the number of ftp flows to 8. This should reduce the stability margin according to Proposition 2. The plot in Figure 15 reveals some oscillations but the system remains rela-

tively stable. The gain margins in the non-linear system seem to be retained from the linear model.
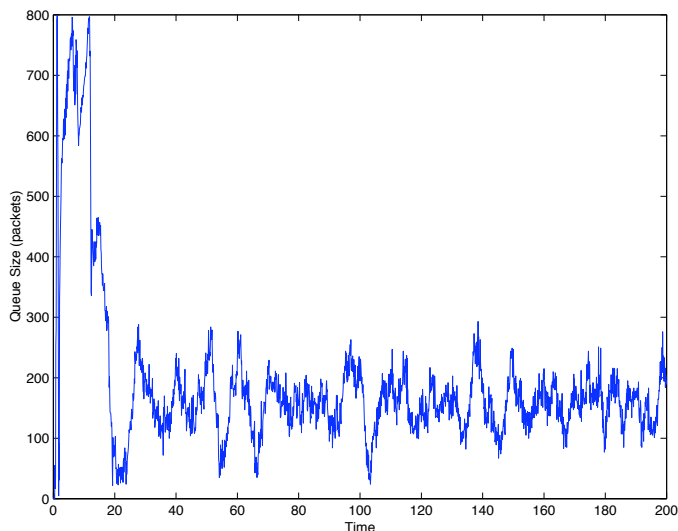

Fig. 15. Experiment 6

G. Experiment 7

Now we increase the load level. According to our analysis, the system should remain stable, however since we have been too conservative in our design, the performance of the compensator should be slower. Figure 16 exhibits the phenomena, with the queue length taking a longer time to settle.
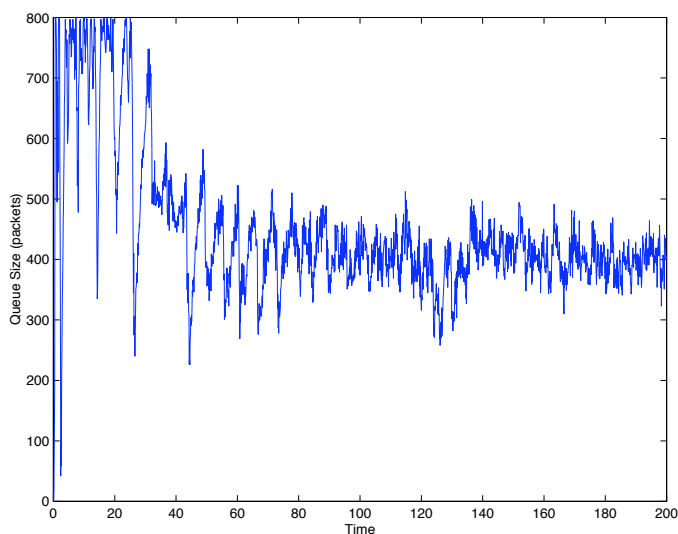

Fig. 16. Experiment 7

H. Experiment 8

Finally, we retain the load level of 60, but reduce the propagation delay to 50ms. This again should have no effect on stability, but performance should be degraded. Figure 17 displays the slow response.
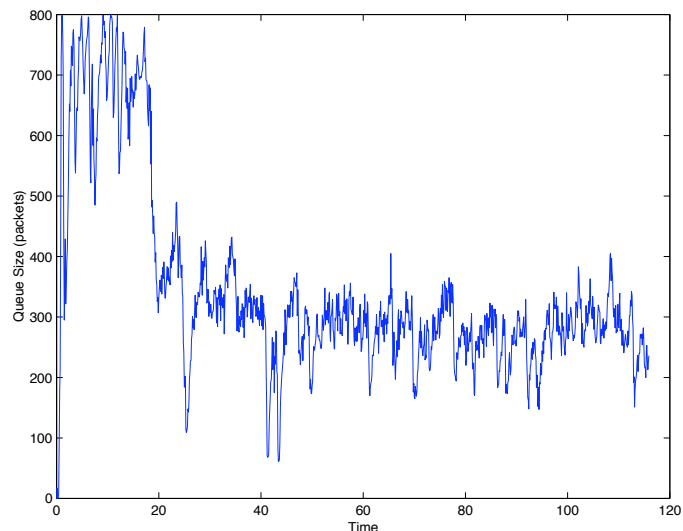

Fig. 17. Experiment 8

V. CONCLUSIONS

In this paper we analyzed a combined TCP and AQM model from a control theoretic standpoint. We used linearization to analyze a previously developed non-linear model of the system. We performed the analysis on an AQM system implementing RED. We are able to present design guidelines for choosing parameters that lead to stable operation of the linear feedback control system. We are also able to derive expressions for the relative stability of the system so designed. We performed non-linear simulations using `ns` that verified our analysis. We are also able to make some comments on tradeoffs of various parameter choices for RED.

While in our design methodology we focused on the stability of the queue length as a means to achieve AQM performance objectives, this may not be the overriding goal of RED implementations. Indeed, stable designs are also slow designs, and more responsive systems can be obtained by trading stability for speed from our guidelines. However, the resulting faster systems may come with a price of large delay jitter, fluctuating loss levels and inefficient utilization of resources. The oscillations can also affect the performance of other protocols (for instance TCP-friendly schemes) that rely on efficient estimation of loss and delay. Another limitation of RED is the direct coupling between queue length and loss feedback. This results in load dependent queue levels. Higher load levels results in lower bandwidth per flow, but this coupling of load level and queue length in RED imposes an additional performance penalty in terms of the higher delay that can be avoided. The control theoretic model we developed points us in the direction of AQM schemes more suited for the particular application. There are well developed tools in classical linear system analysis that help in designing improved controllers for AQM. Our investigations in that direction are detailed in [18].

mous reviewers for their helpful comments.

APPENDIX

I. LINEARIZATION OF THE FLUID-FLOW MODEL

Assuming constant number of TCP flows N and round-trip times R, we first define the right-hand sides of the differential equations in (1) by:

$$f(W, W_R, q, p) \doteq \frac{1}{R} - \frac{W(t)W_R}{2R}p(t-R)$$

$$g(W, q) \doteq \frac{N(t)}{R}W(t) - C.$$

(10)

where $W_R(t) \doteq W(t-R)$. Recall the operating point relationships:

$$\dot{W} = 0 \Rightarrow W_0^2 p_0 = 2$$

$$\dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N}.$$

Evaluating partials at this operating point (W_0, q_0, p_0) in (2) gives:

$$\begin{aligned}
\frac{\partial f}{\partial W} = \frac{\partial f}{\partial W_R} &= -\frac{W_0}{2R_0}p_0 \\
&= -\frac{W_0}{2R_0}\frac{2}{W_0^2} \\
&= -\frac{1}{R_0 W_0} \\
&= -\frac{N}{R_0^2 C};
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f}{\partial q} &= \frac{\partial}{\partial q}\left(\frac{1}{\frac{q}{C}+T_p} - \frac{W^2 p}{2(\frac{q}{C}+T_p)}\right) \\
&= -\frac{1}{R_0^2 C} + \frac{W_0^2 p_0}{2R_0^2 C} \\
&= -\frac{1}{R_0^2 C} + \frac{2}{2R_0^2 C} = 0;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f}{\partial p} &= -\frac{W_0^2}{2R_0} \\
&= -\frac{\frac{R_0^2 C^2}{N^2}}{2R_0} \\
&= -\frac{R_0 C^2}{2N^2};
\end{aligned}$$

$$\begin{aligned}
\frac{\partial g}{\partial q} &= \frac{\partial}{\partial q}\frac{NW}{(\frac{q}{C}+T_p)} \\
&= -\frac{NW_0}{C(\frac{q_0}{C}+T_p)^2} \\
&= -\frac{1}{R_0};
\end{aligned}$$

$$\frac{\partial g}{\partial W} = \frac{N}{R_0}.$$

II. JUSTIFICATION OF (4) WHEN $W_0 \gg 1$

Consider the differential equation

$$\delta\dot{W}(t) = -\alpha(\delta W(t) + \delta W(t - R_0)) + u(t)$$

where $\alpha \doteq \frac{N}{R_0^2 C}$ and u is a ficticious input. Denote the transfer function from u to δW by $H(s)$. Then

$$\begin{aligned}
H(s) &= \frac{1}{s + \alpha(1 + e^{-sR_0})} \\
&= \frac{1}{(s+\alpha)(1 + \frac{\alpha e^{-sR_0}}{s+\alpha})}.
\end{aligned}$$

If the bandwidth of the transfer function $\frac{\alpha}{s+\alpha}$ is much less than $\frac{1}{R_0}$; i.e.,

$$\frac{N}{R_0^2 C} \ll \frac{1}{R_0},$$

or equivalently, $W_0 \gg 1$, then

$$\begin{aligned}
H(s) &\approx \frac{1}{(s+\alpha)(1 + \frac{\alpha}{s+\alpha})} \\
&= \frac{1}{s + 2\alpha}.
\end{aligned}$$

Thus, the approximation (4).

REFERENCES

[1] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the internet," RFC 2309, April 1998.

[2] M. Christiansen, K. Jeffay, D. Ott, and F.D. Smith, "Tuning RED for web traffic," in *Proceedings of ACM/SIGCOMM*, 2000.

[3] Martin May, Thomas Bonald, and Jean-Chrysostome Bolot, "Analytic Evaluation of RED Performance," in *Proceedings of IEEE/INFOCOM*, 2000.

[4] Teunis J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proceedings of IEEE/INFOCOM*, 1999.

[5] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Blue: A New Class of Active Queue Management Algorithms," Tech. Rep., UM CSE-TR-387-99, 1999.

[6] Dong Lin and Robert Morris, "Dynamics of random early detection," in *Proceedings of ACM/SIGCOMM*, 1997.

[7] W. Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin, "A Self-Configuring RED Gateway," in *Proceedings of IEEE/INFOCOM*, 1999.

[8] Victor Firoiu and Marty Borden, "A Study of Active Queue Management for Congestion Control," in *Proceedings of IEEE/INFOCOM*, 2000.

[9] Vishal Misra, Wei-Bo Gong, and Don Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," in *Proceedings of ACM/SIGCOMM*, 2000.

[10] "ns-2 Network Simulator," Obtain via http://www.isi.edu/nsnam/ns/.

[11] S. Mascolo, "Congestion control in high-speed communication networks," *Automatica*, vol. 35, pp. 1921–1935, March 1999.

[12] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, 1995.

[13] F. Kelly, "Mathematical modelling of the Internet," in *Mathematics Unlimited - 2001 and Beyond*, 2000.

[14] Karl J. Åström, "Oscillations in Systems with Relay Feedback," in *Adaptive control, filtering and signal processing, IMA Volume sin Mathematics and its Applications*, 1995, vol. 74, pp. 1–25.

[15] Sally Floyd, "Recommendation on using the "gentle_" variant of RED," http://www.aciri.org/floyd/red/gentle.html, March 2000.

[16] J. Mahdavi and Sally Floyd, "TCP-friendly unicast rate-based flow control," Note sent to end2end-interest mailing list, Jan 1997.

[17] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, no. 3, July 1997.

[18] C. V. Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows," in *Proceedings of IEEE/INFOCOM*, April 2001.