A Control Theoretic Analysis of RED*

Chris Hollot[1] Vishal Misra[2†] Don Towsley[2] Wei-Bo Gong[1]

[1]Department of ECE, [2]Department of Computer Science

University of Massachusetts, Amherst MA 01003

{hollot,gong}@ecs.umass.edu, {misra,towsley}@cs.umass.edu

CMPSCI Technical Report TR 00-41
July 2000

Abstract

In this report 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.

*This work is supported in part by the National Science Foundation under Grants ANI-9873328 and by DARPA under Contract DOD F30602-00-0554.

†Corresponding author

1 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] have also been proposed, perhaps motivated by the difficulty in understanding the dynamics of RED completely.

In [7], the authors investigated the issue of recommendations of RED parameters, and gave thumb rules and guidelines for choosing them. In this report we investigate similar issues, however from a more formal, control theoretic standpoint. We use a previously developed model of TCP and RED dynamics [8] 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 non-linear simulations using `ns`.

The rest of the report is organized as follows. In Section 2 we develop the linearized model for the AQM control system. Section 3 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 4 we present simulation results done using `ns` which verify our analysis and design recommendations. Finally, we present our conclusions in Section 5.

2 Model

2.1 TCP model

In [8], a dynamic model of a TCP connection through a congested AQM router was developed using fluid-flow analysis. Simulation results presented in the paper demonstrated that the model captured the dynamics of TCP with a fair degree of accuracy. We use a simplified version of that model, ignoring the timeout mechanism. This model can be described by the coupled differential equations

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

where:

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

We illustrate these differential equations through the block diagram in Figure 1. This figure highlights TCP window-control, TCP load and queue dynamic.To gain insight into both its behavior and feedback control we approximate these dynamics by their small-signal linearization about an operating point. This allows us to take advantage of well-developed techniques of linear systems analysis and control. Details of the technique can be found in any advanced text of control systems

design, e.g. [9]. The basic idea is to write equations for small perturbations about the operating point and then ignoring second order terms, thereby obtaining a linear model.

2.2 Linearization

Taking (W, q) as the system state and p as input, the equilibrium (W_0, q_0, p_0), defined by $\dot{W} = 0$ and $\dot{q} = 0$, is given by

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

where

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

Linearizing (1) about this equilibrium state (see Appendix) yields

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

where

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

represent perturbations in W, q and p from their equilibrium values. The eigenvalues of the linearized TCP and queue dynamics 3 are respectively

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

Since all the network parameters are positive quantities, these negative eigenvalues indicate that the equilibrium state of the non-linear dynamics is locally asymptotically stable. That is, for $p = p_0$, all responses starting "close" to (W_0, q_0) will

asymptotically converge to (W_0, q_0). An interpretation of the TCP's time constant $W_0R_0/2$ comes from expressing the linearization of the $\dot{W}$ equation above as:

$$\delta\dot{W}(t) = -\lambda_0\delta W(t) - \frac{R_0C^2}{2N^2}\delta p(t - R_0)$$

where λ is the *p*acket loss rate as discussed in [8]. Therefore, the time constant is $\frac{1}{\lambda_0}$. In the steady state ($\dot{W} = 0$), the decrease in window size $\frac{1}{2}W_0\lambda_0$ must balance the increase in TCP window size $\frac{1}{R_0}$. Consequently, $\lambda_0 = \frac{2}{W_0R_0}$. Finally, it is interesting to note that the linearization of the queue dynamic is not a pure integrator, as one may expect, but a first-order lag with time constant R_0. This can be explained by noting that the queue input is a function of the queue length by virtue of the $\frac{1}{R}$ factor in Figure 1.

2.3 The AQM Control Problem

Using the linearized TCP model (3) an AQM control system can be modeled as in the block diagram of Figure 2 [1]. In this diagram P_{tcp} denotes the transfer function

[1]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.

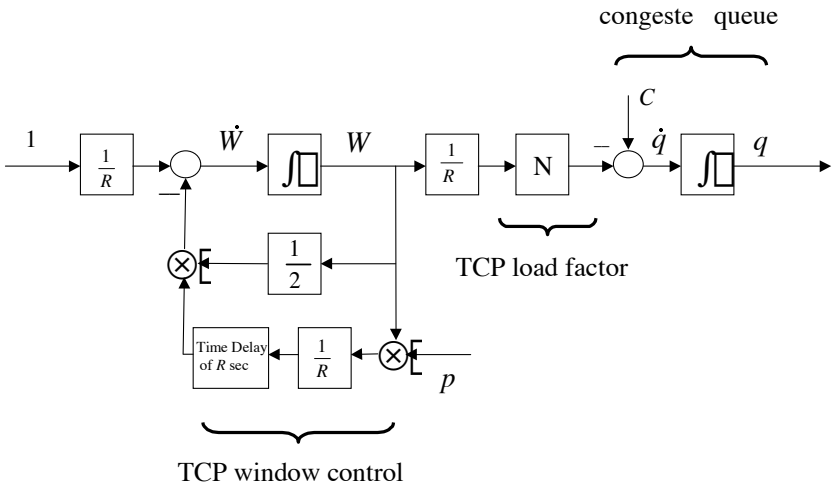

Figure 1: Block-diagram of a TCP connection.

from loss probability δp to window size δW and P_{queue} relates δW to queue length q. The term e^{-sR} is the Laplace transform of the time-delay in the delayed loss probability $\delta p(t - R)$. 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^o. The phase margin ϕ_m is defined as $\omega_{pm} - 180$ where ω_{pm} is the phase response at the frequency where the magnitude response is unity (or 0 dB). The two quantities are shown in Figure 3. 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.

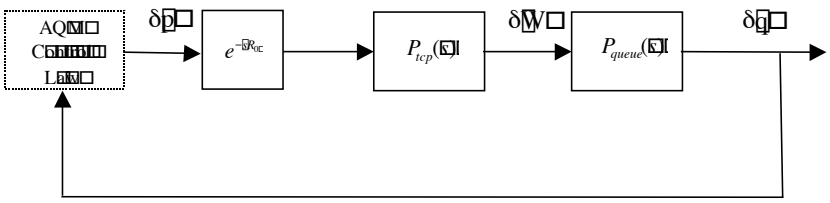

Figure 2: Block diagram illustrating AQM control.

2.3.1 Plant Dynamics

From Figure 2, the plant transfer function, $P(s) = P_{tcp}(s)P_{queue}(s)$, can be expressed in terms of network parameters yielding:

$$\begin{aligned} P_{tcp}(s) &= \frac{\frac{R_0C^2}{2N^2}}{s + \frac{2N}{R_0^2C}}; \\ P_{queue}(s) &= \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}. \end{aligned} \tag{4}$$

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

Example 1: Suppose $q_0 = 175$ packets, $T_p = 0.2$ secs and $C = 3750$ packets/sec. Then, for a load of $N = 60$ TCP sessions, we have $W_0 = 15$ packets; $p_0 = 0.008$;

$$\begin{aligned} P_{tcp}(s) &= \frac{482}{s + 0.53}; \quad P_{queue} = \frac{243}{s + 4.1}; \\ P^{60}(s) &\doteq P_{tcp}(s)P_{queue}(s) = \frac{117,126}{(s + 0.53)(s + 4.1)}. \end{aligned}$$

For a load of $N = 120$ TCP sessions, we have $W_0 = 7.7$; $p_0 = 0.034$;

$$\begin{aligned} P_{tcp}(s) &= \frac{120}{s + 1.05}; \quad P_{queue} = \frac{486}{s + 4.1}; \\ P^{120}(s) &\doteq P_{tcp}(s)P_{queue}(s) = \frac{58,320}{(s + 1.05)(s + 4.1)}. \end{aligned}$$

Notice that the P^{60} dynamic has larger time constants $(\frac{1}{0.53}, \frac{1}{4.1})$ vs. $(\frac{1}{1.05}, \frac{1}{4.1})$ and larger DC gain[2], $P^{60}(0)$ and $P^{120}(0)$, $(53,901$ vs. $13,547)$. Expressing this plant

[2]the DC gain of a system is simply it's transfer function evaluated at $s = 0$, i.e., at 0 (DC) frequency

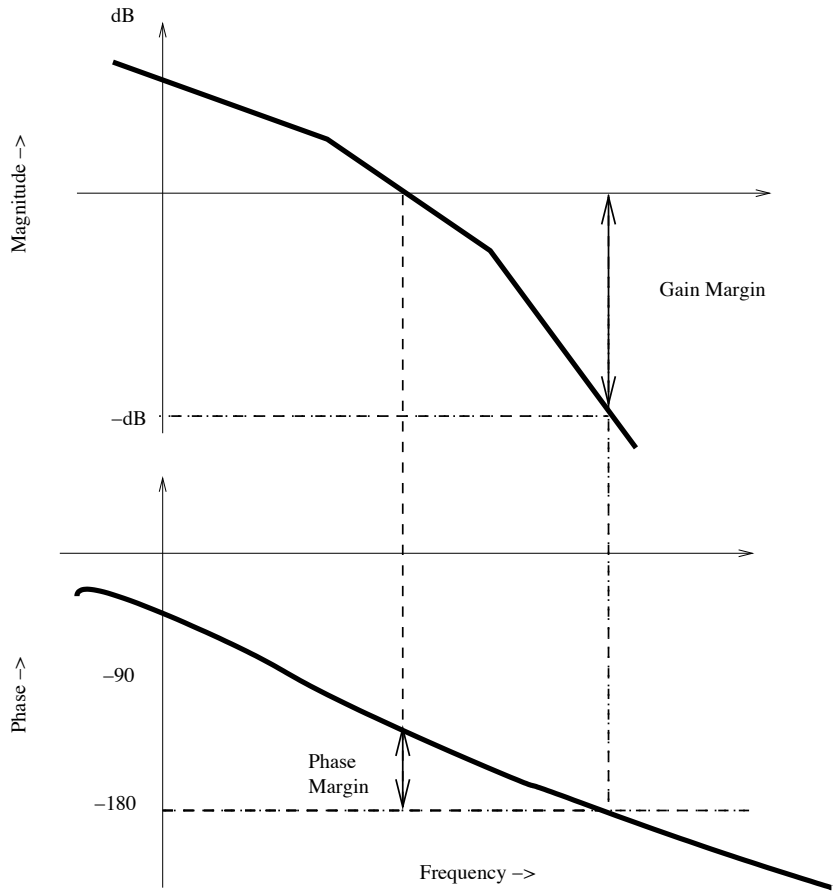

Figure 3: Stability margins on the Bode plot

gain in terms of network parameters shows its dependency on the TCP load factor N explicitly:

$$\text{DC plant gain} \doteq P_{tcp}(0)P_{queue}(0) = \frac{(RC)^3}{(2N)^2}. \tag{5}$$

The Bode plots for $P^{60}(s)$ and $P^{120}(s)$ are shown in Figure 4. One of the goals of the controller should be to introduce positive stability margins in the Bode plots.

Remarks 1:

1. Roughly speaking, an increase in the DC plant gain (5) can affect AQM control in two competing ways. First, such an increase reduces the sensi-

tivity of queue length on network parameters such as load N. Secondly, increasing plant gain can decrease stability margins; i.e., render closed-loop response more oscillatory. From (5) we see that *increasing capacity* C and *decreasing TCP load* N both lead to larger DC plant gain. The variation in this gain as a function of network parameters should be a main concern in the design of AQM control schemes.

2. From the block diagram in Figure 1, it seems counter-intuitive that the DC plant gain is inversely proportional to the load N. From (4) we see that the DC gain of the queue dynamic $\frac{N}{R}$ is indeed proportional to N; this is intuitive. However, the DC gain of the TCP dynamic $\frac{RC^2}{2N^2}$ is inversely proportional to N^2. This can be seen from the $\dot{W}$ equation of (1) where the loss probability p is weighted by W^2 and where the equilibrium value of the TCP window size is itself inversely proportional to N.

3. The (round-trip) time-delay e^{-sR} ($R \approx T_p$) places an upper limit on the speed of responses achievable in AQM control. Closed-loop time constants of AQM's are limited to $\approx T_p/2$ seconds[3]

2.3.2 AQM Performance Objectives

One objective of an AQM control system is to improve throughput by regulating fluctuations in queue length to avoid either buffer overflow (lost packets) or an empty buffer (link underutilization). The other objective is to remove synchronization amongst flows by "spreading out" losses. Ironically, an unstable AQM system often leads to oscillations and strong synchronizations amongst flows. We thus take as our performance objectives the regulation of both *transient* and *steady-state* queue length.

[3]The sinusoidal phase lag due to a time delay element e^{-sT_p} is ωT_P. Feedback systems are usually designed so that time delays do not contribute more than 90° of phase lag at control bandwidth. This requires $\omega < \frac{\pi}{2T_p}$ and hence closed-loop time constants greater than $\frac{2T_p}{\pi}$.

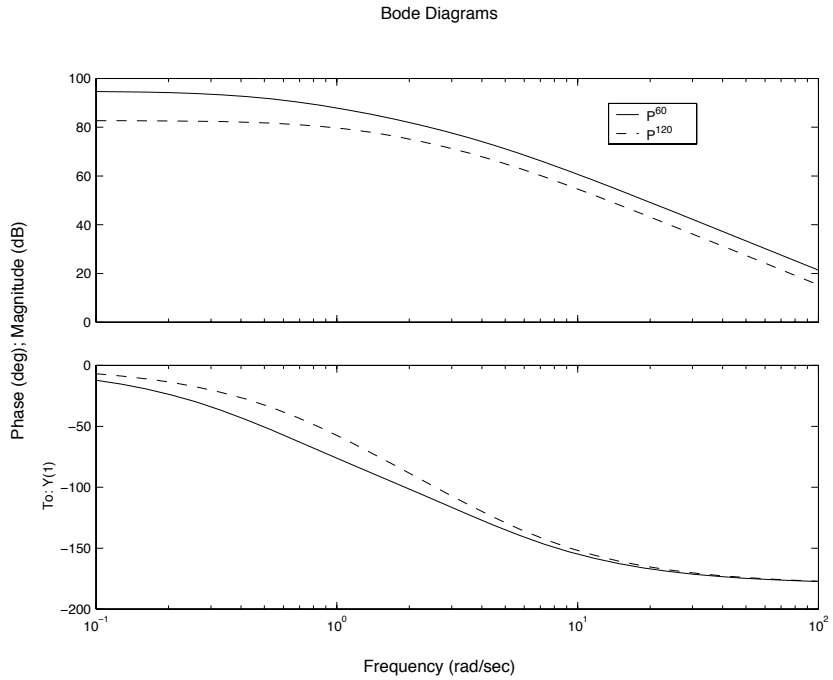

Figure 4: Bode plots of $P(s) = P_{tcp}(s)P_{queue}(s)$ for TCP loads of $N = 60$ and $N = 120$.

3 AQM with RED

Random early detection (RED) is an AQM feedback scheme introduced in [10]. In this section we use the previously developed linearized model of AQM to suggest rules for tuning a RED-based controller.

3.1 Description of RED

The RED active queue management control law computes loss probability p as a function of measured queue length q as depicted by the AQM control law in the block diagram of Figure 2. Specifically, RED consists of a low-pass filter (LPF) and nonlinear gain map as shown in Figure 5. The form of the LPF was derived in [8]. The pole K is equal to $\log_e(1-\alpha)/\delta$, where α is the averaging weight and δ is the sampling frequency. Normally RED updates it's moving average on every packet arrival, and hence δ is $1/C$, where C is the queue capacity in packets/sec. At high load levels this sampling frequency exceeds C, whereas at low load levels it falls below C. On an average however, under the assumption of a stable congested queue, the sampling frequency is C. Tuning RED amounts to selection of the low-pass filter pole K, threshold q_{min}, level p_{max} and gain L_{red}. A point to be noted here is that the role of a low-pass filter in AQM is unclear. A low-pass filter provides both attenuation and phase lag at frequencies past its corner frequency; $\omega = K$. At such frequencies, the introduction of a phase lag reduces phase margins and worsens transient performance, and is often the cause of oscillations seen in RED systems. The benefit of a low-pass filter lies in its magnitude attenuation at these higher frequencies. This property gives it an ability to attenuate the effects of high-frequency sensor noise on the measured variable q. For example, a low-pass filter in an AQM control law could be useful in attenuating sensor noise n on the regulated output q. However, the range of frequencies over which this occurs must be sufficiently removed from the control loop's bandwidth.

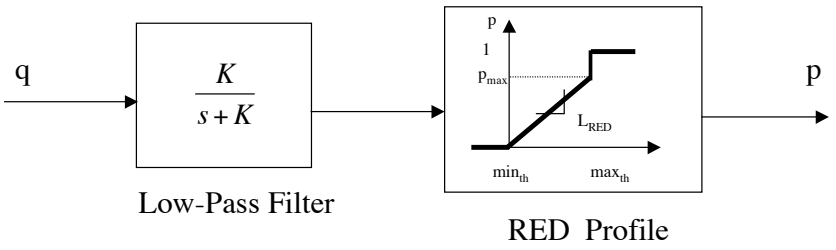

Figure 5: RED as a cascade of low-pass filter and nonlinear gain element.

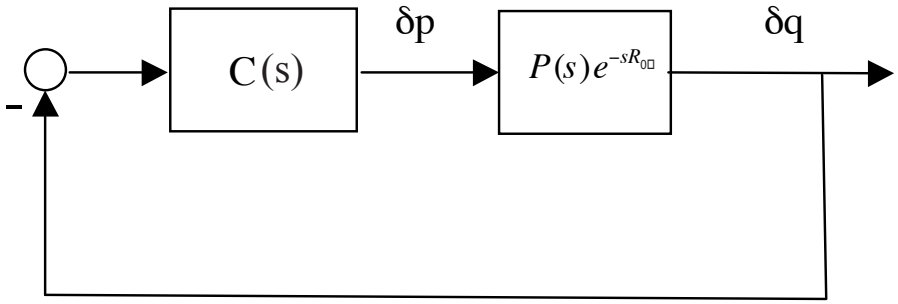

Figure 6: Block diagram of a linearized AQM control system

3.2 Designing RED

An active queue-management (AQM) system can be modeled as the feedback control system shown in Figure 6. Here $P(s)e^{-sR}$ 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[4] leads to oscillations (limit-cycles) that can exhibit complex and chaotic behavior; e.g., see [11]. Such oscillations may be undesirable in queue management, and RED was introduced to stabilize them.

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

A transfer-function model for RED is:

$$C(s) = C_{red}(s) = \frac{L_{red}}{s/K + 1}, \tag{6}$$

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 [8]. 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 should be taken into account. The variations in R are due to a variable propagation time T_p where

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

Let's assume a range for the number of TCP sessions, say $N \geq N^-$, and the round-trip time, $R \leq R^+$. The objective is to select RED parameters L_{red} and K in (6) to stabilize the linear control system in Figure 6 for all these N and R. 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:*

$$\frac{L_{red}(R^+C)^3}{(2N^-)^2} \leq \sqrt{\frac{\omega_g^2}{K^2} + 1} \tag{7}$$

where

$$\omega_g = 0.1 \min\left\{\frac{2N^-}{(R^+)^2C}, \frac{1}{R^+}\right\}. \tag{8}$$

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

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

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

From this and (8) we have:

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

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

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

From this and (7) it follows that $|L(j\omega_g)| \leq 1$ for all $N \geq N^-$ and for all $R \leq R^+$. Thus, the unity-gain crossover frequency is bounded above by ω_g. To establish closed-loop stability, we invoke the Nyquist stability criterion [9] and show that $\angle L(j\omega_g) > -180^\circ$. To this end, we again use (8) to obtain

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

□

Remarks 2:

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[5] when $R^+ = 0.25$ secs, $N^- = 40, 60$ and 80 flows and $C = 3750$ packets/sec.

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

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

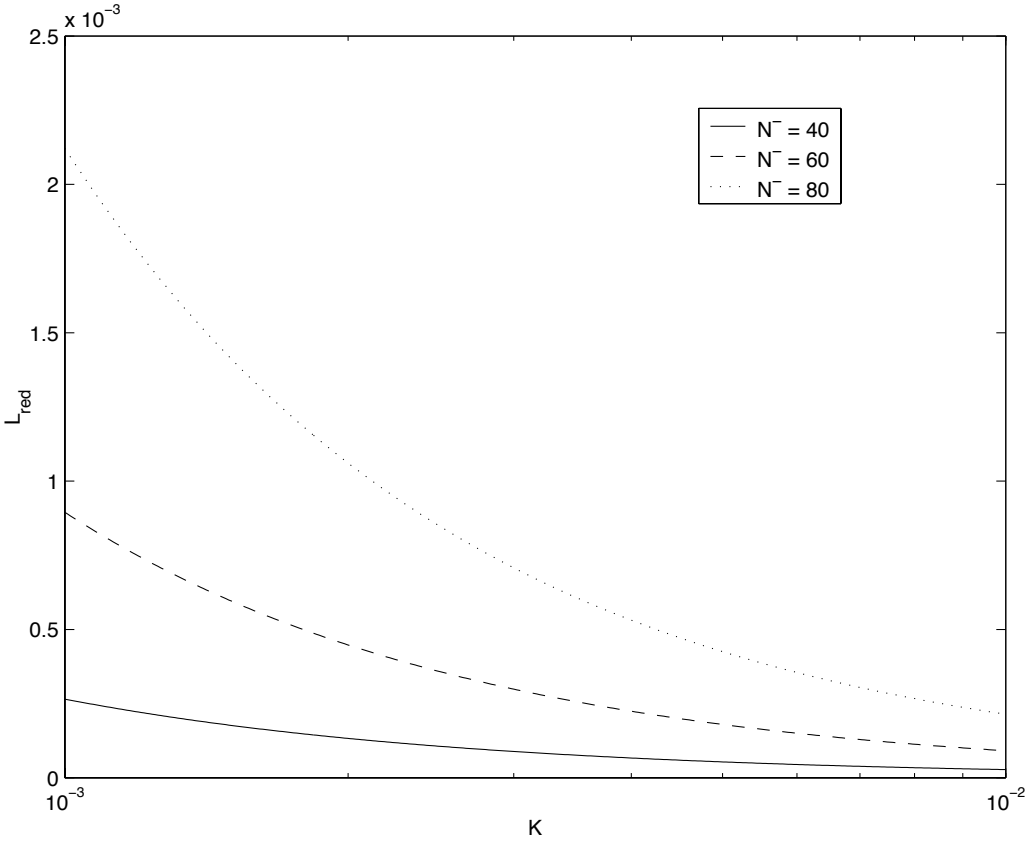

Figure 7: Stabilizing RED parameters L_{red} and K.

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 [7]. However, the *gentle_* mechanism recommended in [12] 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.

Proposition 2: *Consider any RED controller $C_{red}(s)$ satisfying conditions (7) and (8) in Proposition 1. Then, the gain margin (GM) and phase margin (PM) of the linear control system in Figure 6 satisfy:*

$$GM \geq 5\pi; \quad PM \geq 85^\circ.$$

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

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

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

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

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

From (8), $\omega_g \leq \frac{0.1}{R}$. 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

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

Consequently,

$$\angle P(j\omega_g) \geq -90^\circ.$$

Since

$$\angle e^{-j\omega R} = -90^\circ$$

then $\angle L(j\frac{\pi}{2R_{tt0}}) \geq -180^\circ$. Because $|L(j\omega_g)| \leq 1$, then $1/|L(j\frac{\pi}{2R_{tt0}})| \approx \frac{\frac{\pi}{2R_{tt0}}}{\omega_g}$ gives a lower-bound to the gain margin. Since $\omega_g \leq 0.1R_{tt0}$ then $GM \geq 5\pi$.

□

Example 2: Consider the case of network parameters: $C = 3750$ packets/sec [6], $N^- = 60$ and $R^+ = 0.2$ sec. From (8),

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

For $K = 0.005$, we compute from (7):

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

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

$$L_{red} = 1.86\text{e-}4; \; K = 0.005$$

Now, for a link capacity of 3750 $packets/sec$, $\delta = 2.66e-04$, yielding α, the averaging weight, as $1.33e-6$. $L_{red} = p_{max}/max_{th} - min_{th}$. Thus, if we choose p_{max} as 0.1, then the dynamic range of the average queue size is approximately

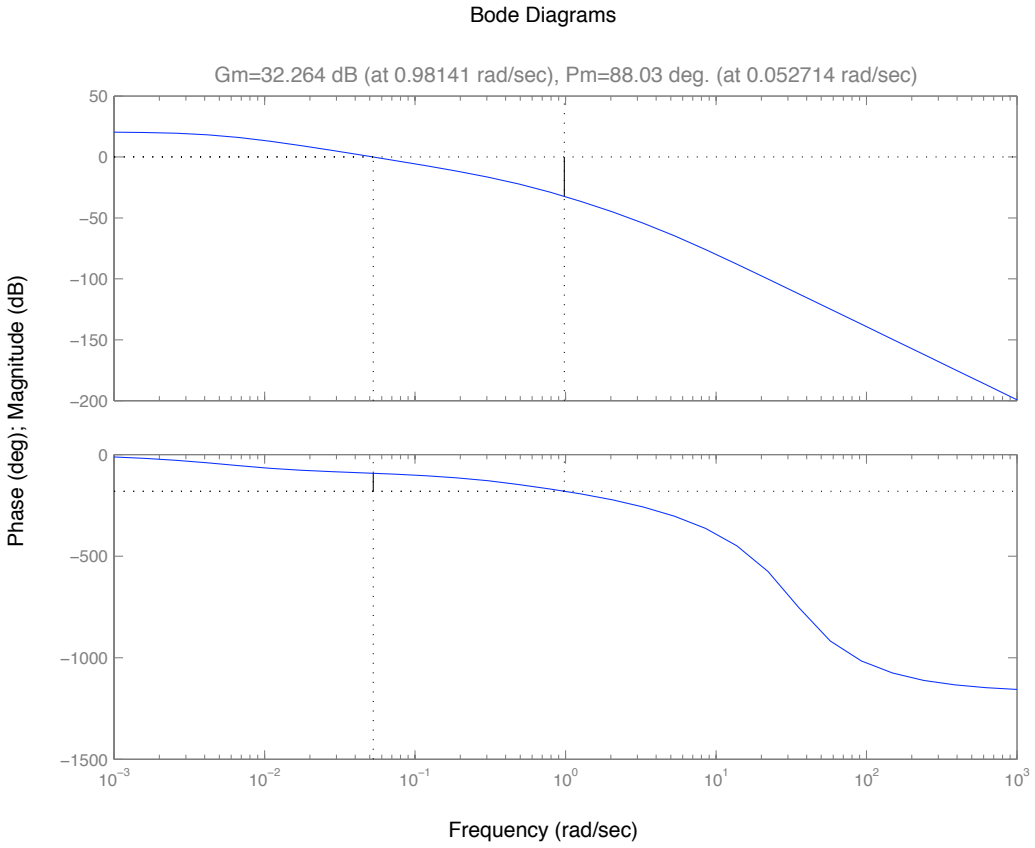

Figure 8: Frequency response for loop using $C_{red} = \frac{9(10)^{-5}}{\frac{s}{0.01}+1}$.

540 packets. In Figure 8 we give the Bode plot for $L(j\omega)$ for $N = N^-$ and $R = R^{tt0^+}$. The phase margin = 83° and the gain margin is 79 $(38dB)$.

Remarks 3:

1. From the viewpoint of *steady-state regulation* it is desirable to select L_{red} in (7) 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 (7), $K \to 0$ 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 (8), must be less than the bandwidth of either the queue or TCP dynamic. Consequently, closed-loop responses are commensurately slow[7]. 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 [13].

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

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

4 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.

4.1 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 9. Observe the oscillating nature of the queue. It frequently goes down to zero, thereby under-utilizing the link. The large oscillations also add considerable jitter to the round trip times of the packets.

4.2 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

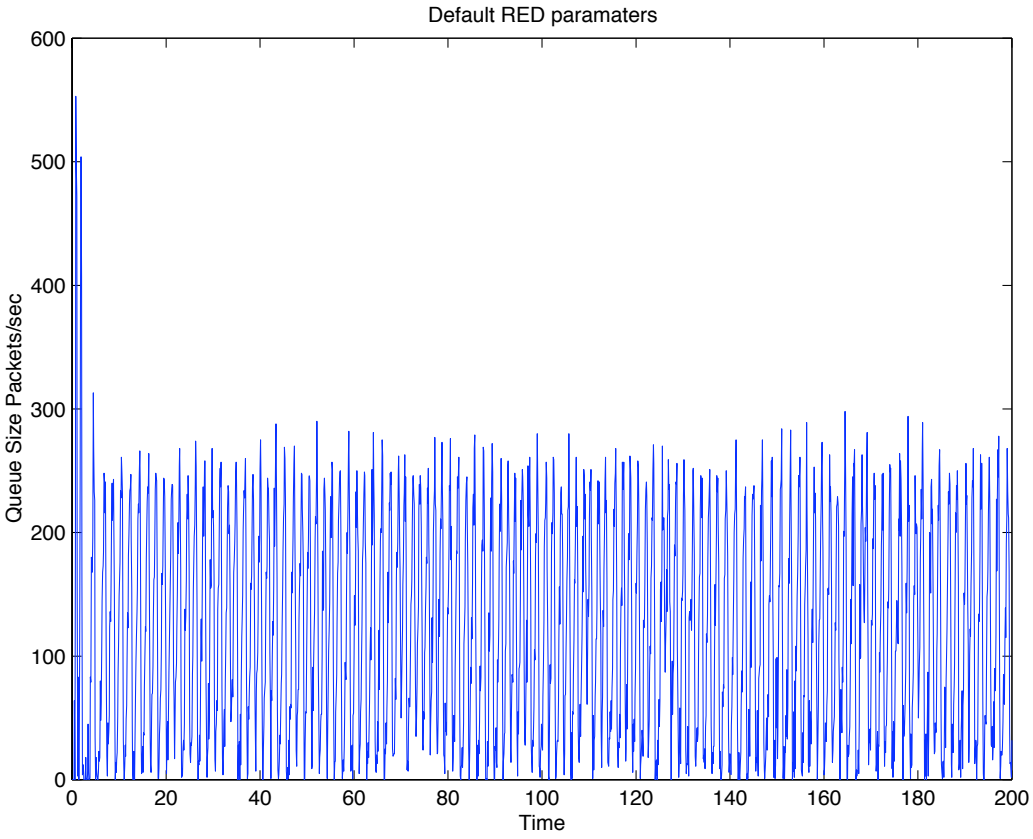

Figure 9: Experiment 1

in Figure 10. 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. [8] 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.

4.3 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 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 response 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 11. 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

[8] 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

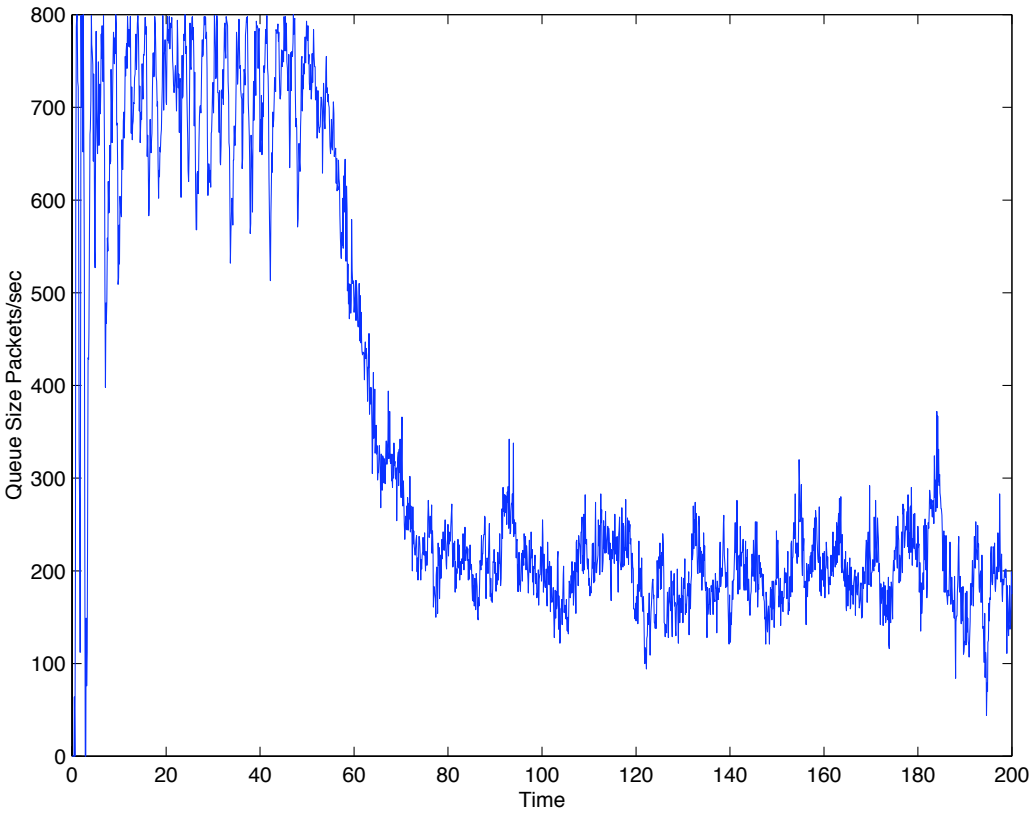

Figure 10: Experiment 2

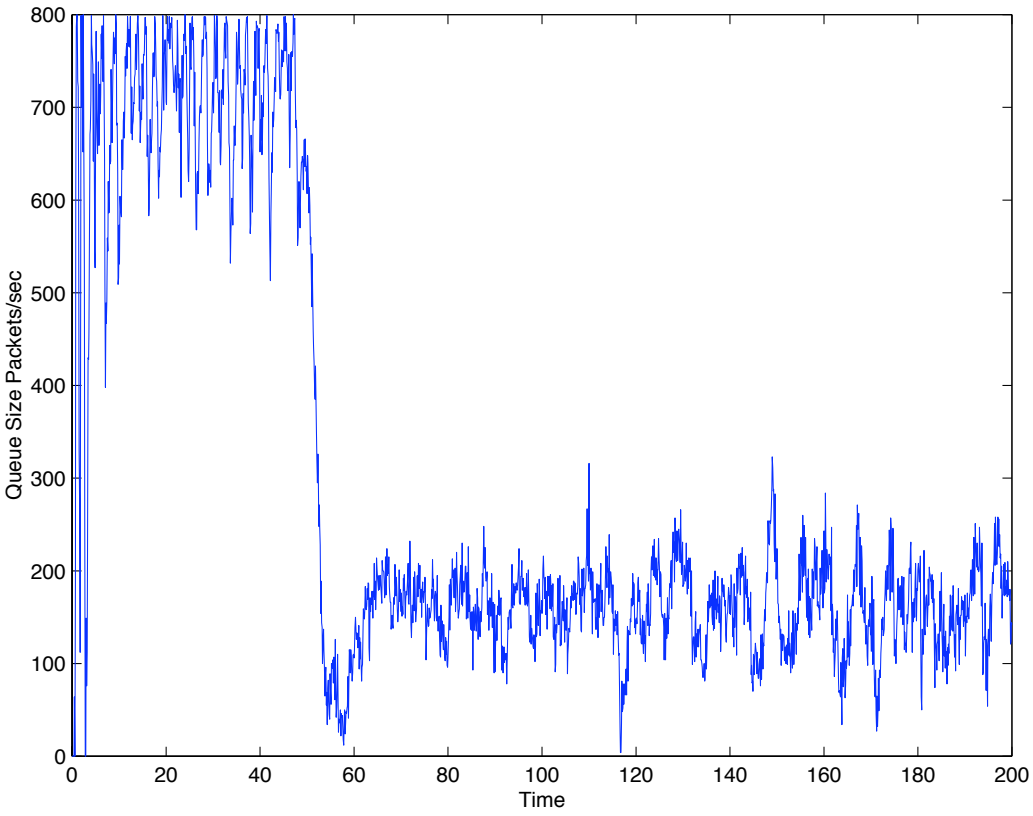

Figure 11: Experiment 3

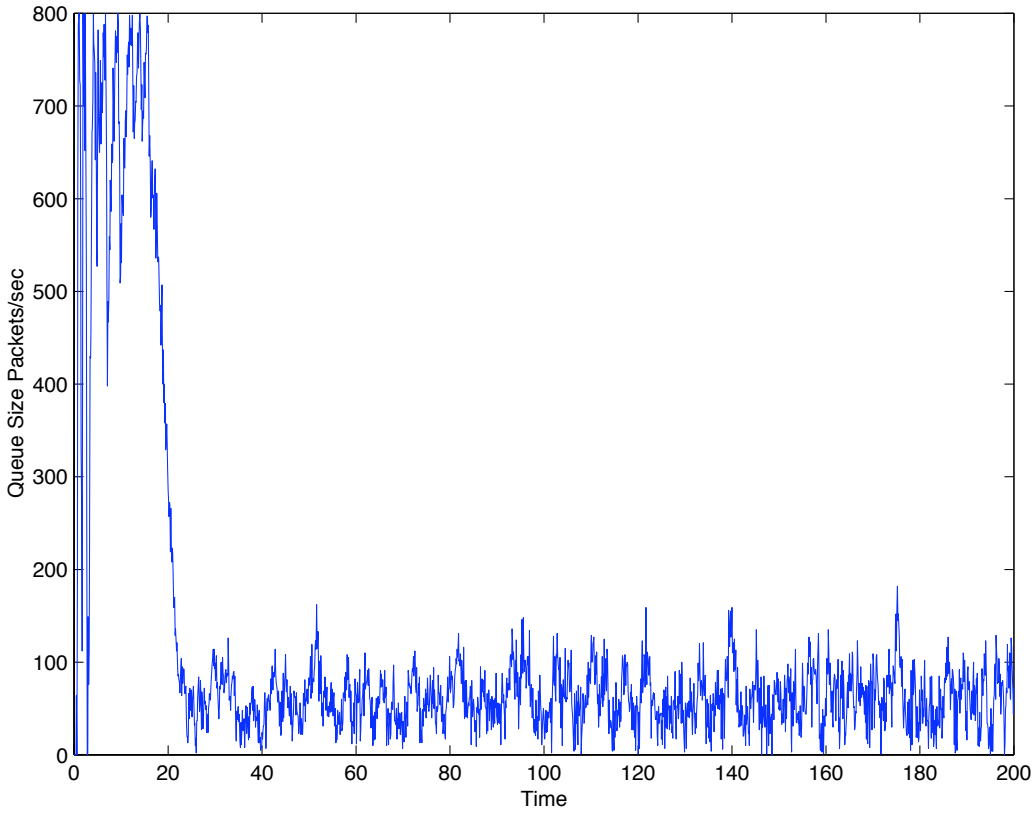

Figure 12: Experiment 3a

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 range to 50-150 in Figure 12 (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 13 (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

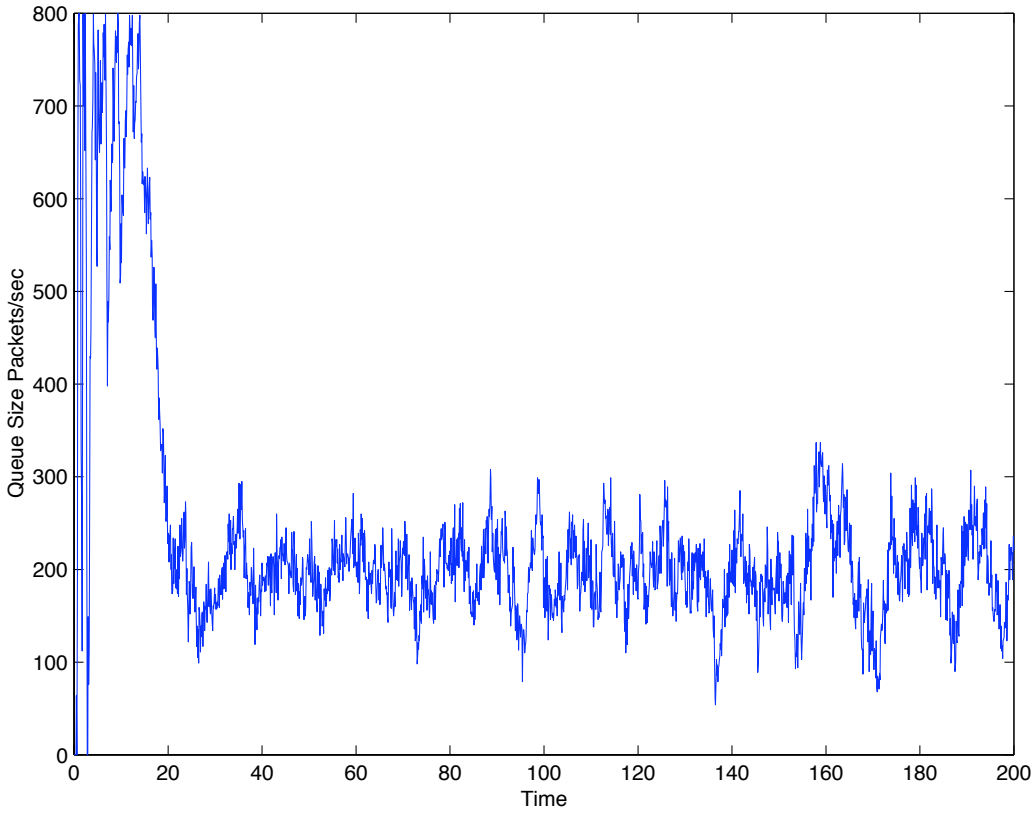

Figure 13: Experiment 3b

appears to be a better option than lowering min_{th}.

4.4 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 14. The system remains stable and the longer round trip times of the fraction of the flows don't affect things too much.

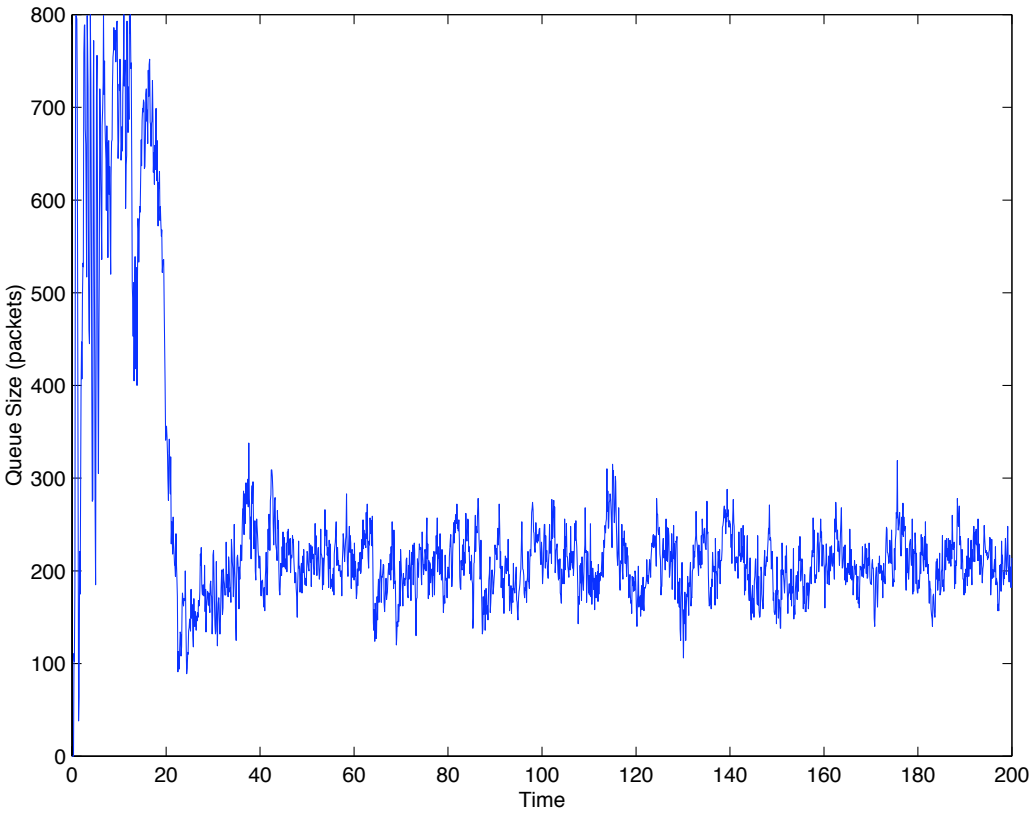

Figure 14: Experiment 4

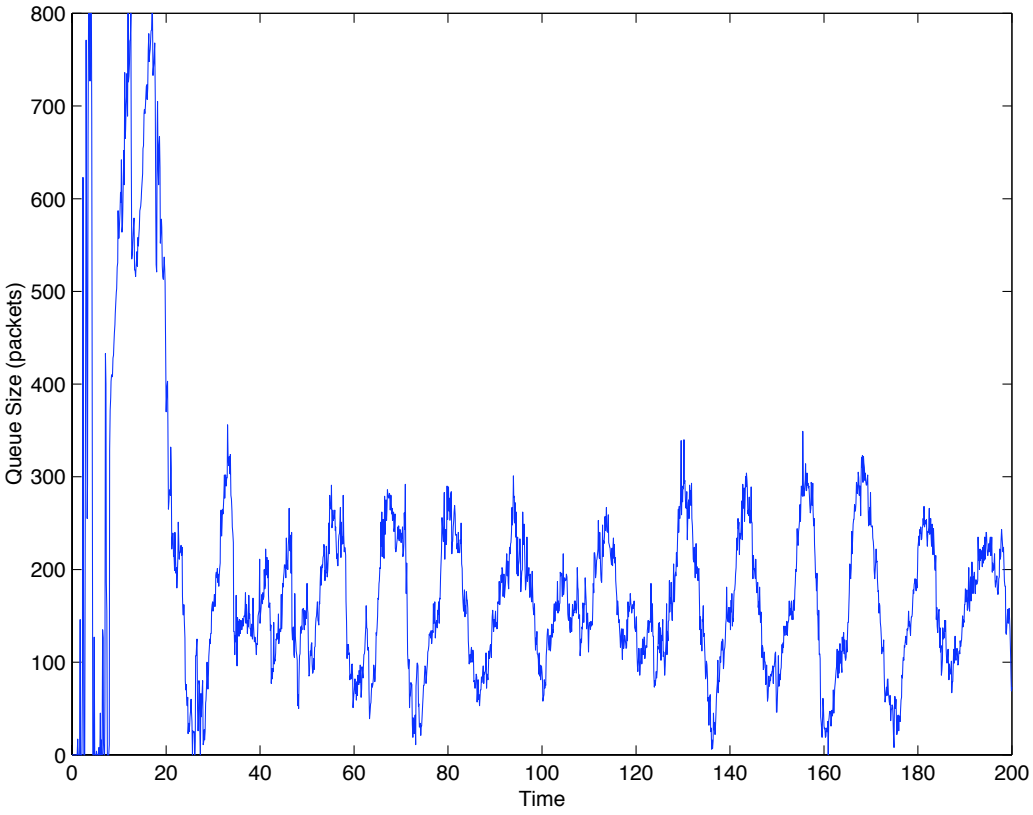

Figure 15: Experiment 5

4.5 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 15. 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.

4.6 Experiment 6

Now we retain the round trip time, but play around with the load level. First, we reduce the number of ftp flows to 8. This should reduce the stability margin

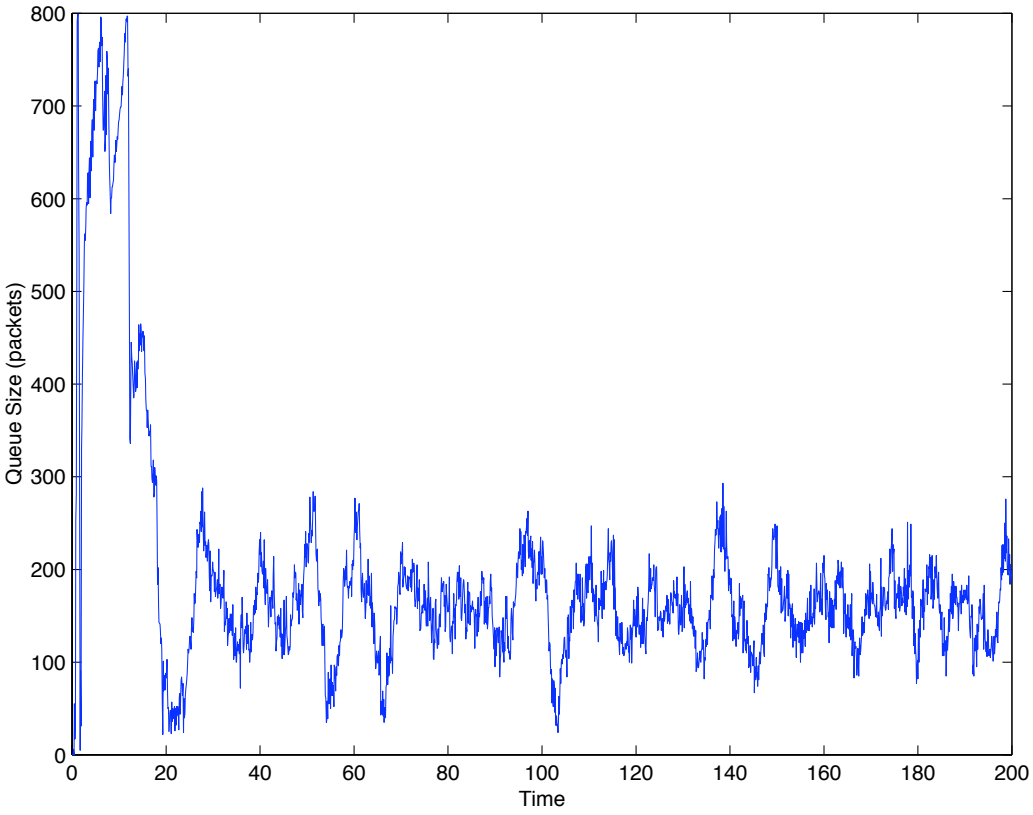

Figure 16: Experiment 6

according to Proposition 2. The plot in Figure 16 reveals some oscillations but the system remains relatively stable. The gain margins in the non-linear system seem to be retained from the linear model.

4.7 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 17 exhibits the phenomena, with the queue length taking a longer time to settle.

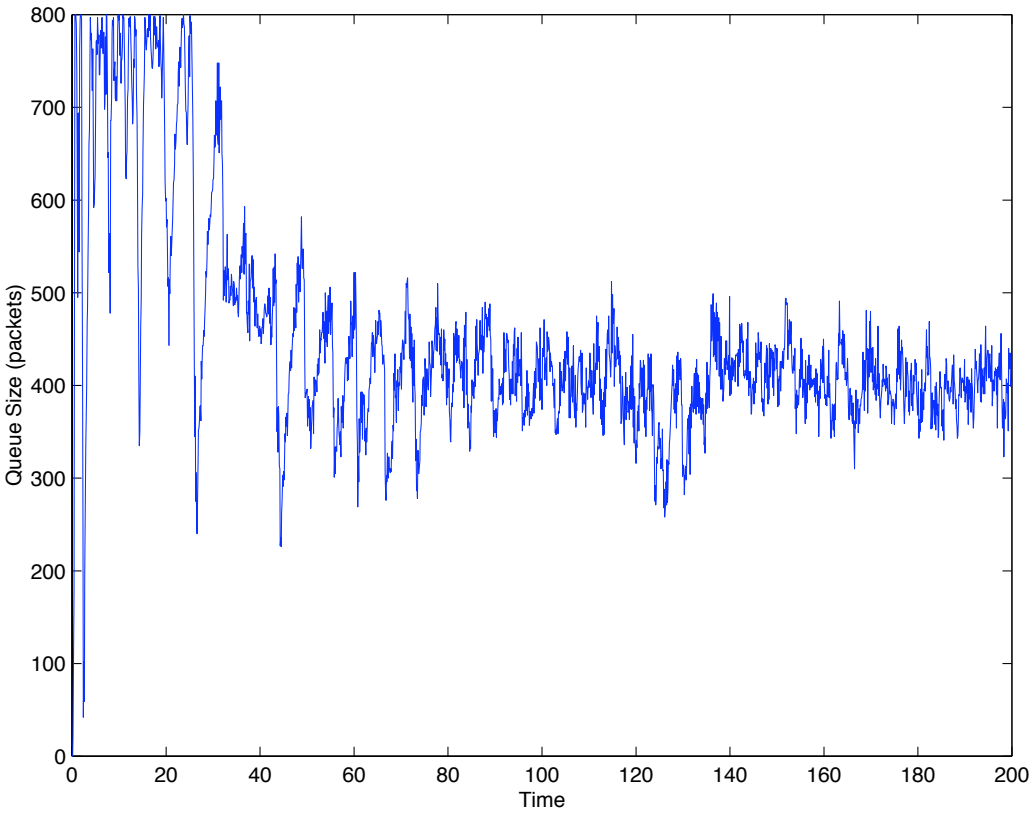

Figure 17: Experiment 7

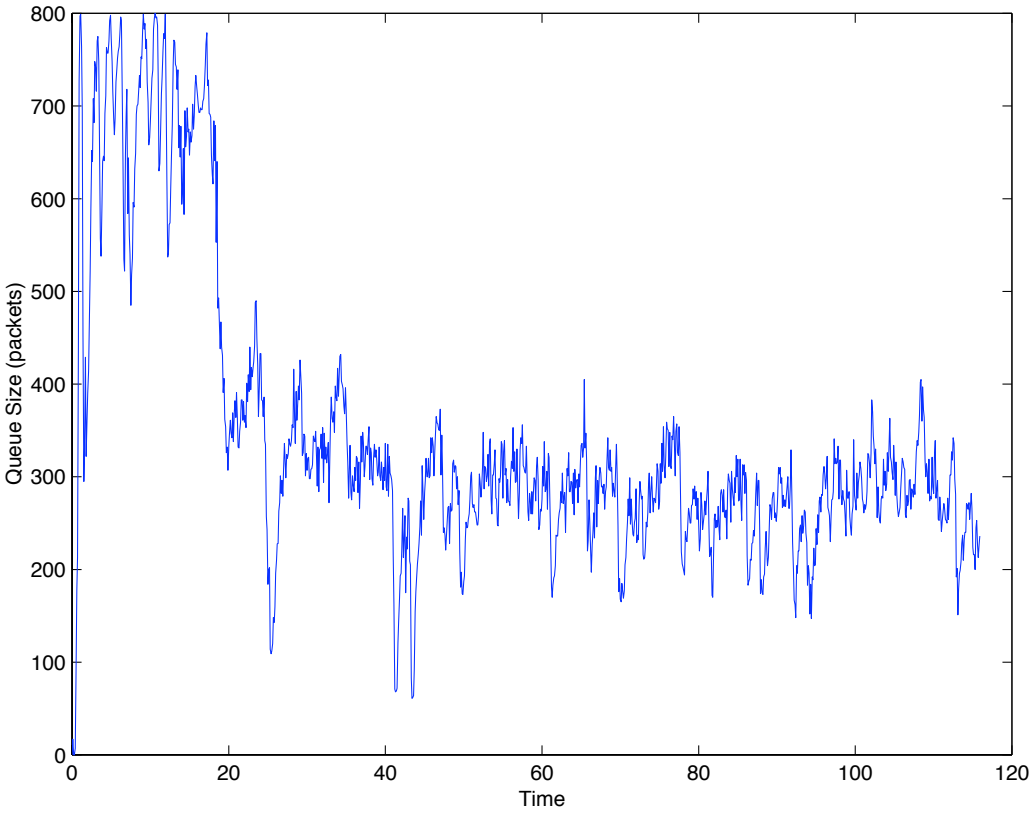

Figure 18: Experiment 8

4.8 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 18 displays the slow response.

5 Conclusions

In this report we analyzed a combined TCP and AQM model from a control theoretic standpoint. We used linearization to analyze a previously developed nonlinear model of the system. We performed the analysis on an AQM system imple-

menting RED. We are able to present design guidelines for choosing parameters which 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` which verified our analysis. We are also able to make some comments on tradeoffs of various parameter choices for RED. In doing the analysis, we uncovered some fundamental limitations of RED. The control theoretic model we developed points us in the direction of controllers more suited for the particular application. There are well developed tools in classical linear system analysis which help in designing improved controllers for AQM. We are investigating those designs and they are the subject of another paper of ours.

A Linearization of TCP Model

From (1) let

$$\begin{aligned} f(W,q,p) &\doteq \frac{1}{R_{tt}(t)} - \frac{W(t)W(t-R_{tt}(t))}{2R_{tt}(t)}p(t-R_{tt}(t)) \\ g(W,q) &\doteq \frac{N(t)}{R_{tt}(t)}W(t) - C. \end{aligned} \tag{9}$$

Recall the equilibrium values:

$$\begin{aligned} \dot{W} = 0 &\Rightarrow W_0^2 p_0 = 2 \\ \dot{q} = 0 &\Rightarrow W_0 = \frac{R_{tt0}C}{N}. \end{aligned}$$

Now, compute partials evaluated at the operating point (W_0, q_0, p_0) in (2)

$$\begin{aligned} \frac{\partial f}{\partial W} &= -\frac{W_0}{R_0}p_0 \\ &= -\frac{W_0}{R_0}\frac{2}{W_0^2} \\ &= -\frac{2}{R_0 W_0} \end{aligned}$$

$$= -\frac{2N}{R_0^2 C};$$

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

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

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

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

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. C. K. Jeffay, D. Ott, and F. Smith, "Tuning Red for web traffic," in *Proceedings of ACM/SIGCOMM*, 2000.

[3] M. May, T. Bonald, and J.-C. Bolot, "Analytic Evaluation of RED Performance," in *Proceedings of Infocom 2000*.

[4] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," in *Proceedings of 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] D. Lin and R. Morris, "Dynamics of random early detection," in *Proceedings of ACM/SIGCOMM*, 1997.

[7] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proceedings of Infocom 2000*.

[8] V. Misra, W. B. Gong, and D. Towsley, " Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," in *Proceedings of ACM/SIGCOMM*, 2000.

[9] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Addison-Wesley, 1995.

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

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

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

[13] C. Hollot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows." Submitted for review, ftp://gaia.cs.umass.edu/pub/Misra00-AQM-Controller.ps.gz, 2000.