

RICE UNIVERSITY

Congestion Control and Complexity Reduction of Large-Scale Networks

by

Shahab Sanayei

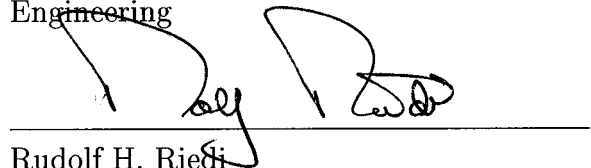
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

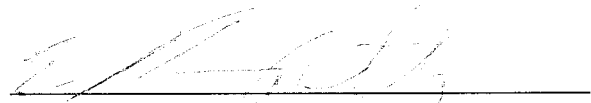
APPROVED, THESIS COMMITTEE:



A.C. Antoulas, Chair
Professor of Electrical and Computer
Engineering



Rudolf H. Riechi
Faculty Fellow of Electrical and
Computer Engineering



Edward W. Knightly
Associate Professor of Electrical and
Computer Engineering

Houston, Texas

August, 2002

UMI Number: 1414293

UMI[®]

UMI Microform 1414293

Copyright 2003 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

Congestion Control and Complexity Reduction of Large-Scale Networks

by

Shahab Sanayei

The current version of Transport Control Protocol (TCP) does not meet the high demands of the exponentially growing Internet. The packet loss is one of the major limiting factors on the performance and the quality of services over Internet (especially for multimedia applications). Also, other improved versions of TCP (Vegas) cannot be deployed in the heterogeneous environment of current Internet. In this paper we propose a new protocol which not only enhances the network performance, but also is deployable in the large scale networks. We study the stability and the fairness of the proposed protocol in the framework of dynamical systems and finally verify the results by simulation.

Acknowledgments

I would like to thank my advisors, professor Antoulas and Dr. Riedi, for their cordial support throughout this research. Also I thank professors Nowak and Knightly for their encouragements and comments on this work.

Contents

Abstract	ii
Acknowledgments	iii
List of Illustrations	vi
List of Tables	vii
1 Introduction	1
2 A New Protocol	4
2.1 AIMD Strategy:	4
2.2 TCP Vegas:	5
2.3 The Key Idea:	6
3 Analysis (Homogeneous Case)	8
3.1 Stability Analysis	8
3.2 Robustness	11
4 Analysis (Heterogeneous Case)	13
4.1 Stability Analysis	13
4.2 Tuning the Parameters	16
4.3 Compatibility and Competitiveness:	18
5 Fairness	20

6 Simulation Results	22
6.1 Homogeneous Case:	22
6.2 Heterogeneous Case:	25
7 Conclusion and Future Avenue	30
Bibliography	32

Illustrations

2.1	Bang Bang Controllers: a)without dead-band b)with dead-band	7
3.1	step response of a second order system for different values of damping ratio	11
6.1	Reno vs. Rice comparison a) Queue length b) Window size	23
6.2	Reno vs. Rice comparison a) Queue length b) Window size	24
6.3	Reno vs. Vegas comparison a) Queue length b) Window size	24
6.4	Vegas: a) Queue length b) Window size c) Rate d) RTT	26
6.5	Rice: a) Queue length b) Window size c) Rate d) RTT	27
6.6	a) Reno: Queue length($P_{loss} = 0.002$) b) Window size c) RTT	28
6.7	Rice: a) Queue length ($P_{loss} = 6.75e - 4$) b) Window size c)RTT . . .	29
6.8	a) Queue length ($P_{loss} = 0.0015$) b) Window size c)RTT	29

Tables

6.1	Comparison of rates, achieved by Vegas and Rice	25
-----	---	----

Chapter 1

Introduction

The Internet congestion control is implemented at the edges of the network where the end users stand. Thus the protocol that controls the data transfer over the network is referred to as an *end-to-end protocol*. The motivation for this idea is that, nowadays the processing units at user level can perform complex tasks at a cheap price, so it is more reasonable to place the data transfer control protocol at the edges. Another reason is that for upgrading the network there is no need to change the structure of the network which is very costly, and that can be done by software upgrades at user level. Also, by pushing the complexities to the edges, network structure is kept simple thus the network can scale in a very convenient way. TCP is known to be the most widely used end-to-end protocol for data transfer on Internet.

As addressed in [11] it is a well-known fact that TCP can oscillate wildly and there is little chance to reduce these oscillations by tuning RED parameters. The AIMD strategy employed by TCP-Reno creates a highly bursty traffic with relatively high loss rate. This is because of the fact that in order to estimate and grab the available bandwidth, TCP-Reno uses loss events as a congestion indicator to control the window size. Also recent models [5], [8] imply that oscillatory behavior of Reno is an inevitable outcome of the protocol itself. In 1994 TCP Vegas was invented as a remedy for all the aforementioned problems, also it was shown that it improves throughput about 40% to 70%. Eventually, however, Vegas was found to suffer in the current heterogeneous Internet in which most of the connection are running the

aggressive TCP-Reno. So there is still a need for a new protocol that can improve the performance and at the same time can compete with Reno which is aggressive by its nature. There are certain concerns in designing an end-to-end protocol for data transfer:

Stability: Stability is an important issue for a protocol. If the protocol is not stable by its nature congestion collapse may happen.

Robustness and Scalability: The designed protocol should also be robust to changes in the network. In a realistic network number of sources and also the topology of the network may change, so the protocol should not fail or have degraded performance because of these changes.

Fairness: It is also desirable for the designed protocol to be fair. In other words, each source should take an equal share from the available bandwidth. There are different notions of fairness like *max-min fairness* or *proportional fairness* (as discussed in [9]) and the designed protocol should satisfy at least one of the known fairness criteria.

Compatibility: The designed protocol should finally be implemented in a real network, therefore it should be compatible with the current network structures. In other words it should be able to compete with current protocol(s) running on the Internet.

This thesis is organized as follows. In chapter 2, we first take a quick look at Vegas. Then we propose a new protocol for congestion control which is basically an altered version of TCP Vegas. In chapter 3 we investigate the important issue of stability for the new protocol when all sources have the same round trip times. Also we examine the robustness of the protocol to changes in the network's edge which explains how well the protocol can scale. In chapter 4 we do a similar analysis for

heterogeneous case and then propose a method for tuning the parameters of the new protocol. In chapter 5, we take a look at the fairness problem and show that under this new protocol, proportional fairness can be achieved at steady state. Chapter 6 covers the simulation results. At the end, we draw the conclusion and propose the avenue for further research.

Chapter 2

A New Protocol

In this chapter, we propose a new protocol for congestion control and we believe this protocol has superiority over the existing versions of TCP. But before that, we first revisit the well-known AIMD strategy and also take a look at TCP-Vegas, because the new protocol is basically an enhanced version of Vegas. Finally, we introduce the key idea of our protocol.

2.1 AIMD Strategy:

After a series of congestion collapses in late 80's, Jacobson et al proposed the so-called additive increase multiplicative decrease (AIMD) strategy to resolve this problem. TCP-Reno uses this mechanism for congestion control. A source using AIMD, increases the window size by one over the window size per each ACK signal, thus after one round trip time, the window size is increased by one. If a packet is lost, the ACK signals received afterwards carry the same number of the lost packet. After third duplicate ACK, the source infers that a loss has happened and then the window size is reduced by half. This mechanism causes an oscillation of window size around the equilibrium point. Also at queue, where all traffic from different sources are being aggregated, these oscillations are noticeable and this is one of the down-sides of using AIMD for congestion control.

2.2 TCP Vegas:

TCP-Vegas was proposed in 1994 by Brakmo et al [3] as an alternative to TCP-Reno. The congestion avoidance mechanism employed in Reno corrects the oscillatory behavior of Reno and this is accomplished by using round trip time measurements as an indication for congestion rather than using loss events as congestion indicators like what Reno does. In fact TCP-Reno has to induce loss in the network in order to be able to explore the available bandwidth and this leads to the oscillatory behavior, almost full queues and bursty traffic which is a drawback of deploying Reno in current Internet. The window adjustment algorithm in Vegas is based on measuring the difference between *the expected rate* and *the actual rate* based on the following formula:

$$\dot{W} = \frac{1}{RTT(t)} \text{sgn}\left(\alpha - \frac{W}{RTT_{base}} + \frac{W}{RTT(t)}\right) \quad (2.1)$$

Where RTT_{base} is the constant round trip time and $RTT(t) = RTT_{base} + \frac{q(t)}{C}$ ($q(t)$ is the queue length at time t and C is the bottle-neck link capacity). and the sign function is defines as:

$$\text{sgn}(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

In control theory this kind of control law is known as *bang-bang controller*. In other versions of Vegas a parameter β is introduced that provides a dead-band in which the window size remains unchanged.

2.3 The Key Idea:

The bang-bang controller which employs a sign function to control the window size is not very effective, a better alternative is a simple *proportional controller* so the window adjustment formula will be:

$$\dot{W} = K\left(\alpha - \frac{W}{RTT_{base}} + \frac{W}{RTT(t)}\right)$$

in which K is the controller gain. For the sake of convenience we refer to this new protocol as **TCP-Rice**. The idea of using a proportional control for congestion control is not new, in [9] a rate-based and in [15] a window-based congestion control mechanism has been proposed. However, in none of these works the problem of stability has been discussed in details, also no method or even insight for tuning the parameters of controller has been given. In the next chapters, we would like to study the properties of our new protocol such as stability and fairness. We first try to carry on the analysis for homogeneous case and then extend it to the heterogeneous case in which round trip times are not equal.

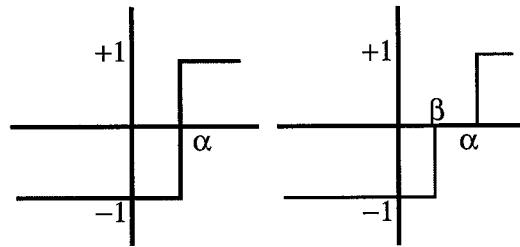


Figure 2.1 : Bang Bang Controllers: a)without dead-band b)with dead-band

Chapter 3

Analysis (Homogeneous Case)

In this chapter we analyze the stability of the closed loop system with N sources running TCP-Rice (N can be large). We assume that there is only one bottle-neck queue receiving packets from N individual sources. The other assumption is that the queue length can be infinite, in other words the hard nonlinearity of the queue does not play a role in our analysis. Furthermore we assume that the traffic is one-way (i.e., the acknowledgments are not queued) and we assume all the sources run a persistent FTP session (elephant). Since our protocol is a window-based protocol we can use a fluid model to describe the behavior of the system. Also, our analysis only covers the congestion avoidance phase and it does not include the slow-start phase which is only significant when we have short-time connections. In the following chapter, for the sake of simplicity, we have assumed that all the sources have the same round-trip delay, τ .

3.1 Stability Analysis

The following equations, describe the dynamics of the protocol, for N sources, sharing the same bottle-neck queue (η is *the bandwidth-delay product*):

$$\begin{cases} \dot{w}_i = K \left(\alpha - \frac{w_i}{\tau} + \frac{w_i}{\tau+q/C} \right) & \text{for } i = 1, \dots, N \\ \dot{q} = \frac{\sum w_j}{\tau+q/C} - C \end{cases} \quad (3.1)$$

The above equations, is a $N + 1$ dimensional dynamical system. In order to check

the stability of this system, we first need to find the equilibrium point:

$$\dot{w}_i = 0 \implies \alpha = \frac{w_i^*}{\tau} - \frac{w_i^*}{\tau + q^*/C} \implies \frac{Cw_i^*q^*}{\eta(\eta + q^*)} = \alpha \implies w_i^* = \frac{\alpha\eta(\eta + q^*)}{Cq^*} \quad (3.2)$$

on the other hand:

$$\dot{q} = 0 \implies \sum_{i=1}^N w_i^* = \eta + q^* \implies w_i^* = \frac{\eta + q^*}{N} \quad (3.3)$$

from 3.2,3.3 we have:

$$\begin{cases} q^* = \frac{N\alpha\eta}{C} \\ w_i^* = \frac{\eta}{N} \left(1 + \frac{N\alpha}{C}\right) \end{cases} \quad (3.4)$$

To check the stability of this equilibrium point we first linearize the non-linear dynamics around the equilibrium point (w_1^*, \dots, w_N^*, q) . Let's define the state vector $X := [w_1, \dots, w_N, q]^T$ also:

$$\begin{cases} f_i(X) := K \left(\alpha - \frac{w_i}{\tau} + \frac{w_i}{\tau + q/C} \right) & \text{for } i = 1, \dots, N \\ g(X) := \frac{\sum w_j}{\tau + q/C} - C \end{cases} \quad (3.5)$$

and $F := [f_1, f_2, \dots, f_N, g]^T$, then the equations (3.1) can be expressed as:

$$\dot{X} = F(X) \quad (3.6)$$

and the equilibrium X^* can be obtained from (3.4). Now to linearize (3.1) we use the Taylor expansion of $F(\cdot)$, hence the linearized system is of the form:

$$\delta\dot{X} = \mathbf{A}\delta\mathbf{X} \quad , \quad \mathbf{A} = \begin{pmatrix} \frac{\partial f_1}{\partial w_1} & 0 & \dots & 0 & \frac{\partial g}{\partial w_1} \\ 0 & \frac{\partial f_2}{\partial w_2} & \dots & \vdots & \frac{\partial g}{\partial w_2} \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \dots & \frac{\partial f_N}{\partial w_N} & \frac{\partial g}{\partial w_N} \\ \frac{\partial f_1}{\partial q} & \frac{\partial f_2}{\partial q} & \dots & \frac{\partial f_N}{\partial q} & \frac{\partial g}{\partial q} \end{pmatrix} \quad (\text{evaluated at } X^*)$$

now we can compute the elements of \mathbf{A} using (3.4) :

$$\frac{\partial f_i}{\partial w_i} = -\frac{KCq^*}{\eta(\eta + q^*)} = -\frac{KCN\alpha}{\eta(C + N\alpha)} \quad (3.7)$$

$$\frac{\partial f_i}{\partial q} = -\frac{KCw_i^*}{(\eta + q^*)^2} = -\frac{KC}{N\eta(1 + \frac{N\alpha}{C})} \quad (3.8)$$

$$\frac{\partial g}{\partial w_i} = \frac{1}{\tau + q^*/C} = \frac{C}{\eta(1 + \frac{N\alpha}{C})} \quad (3.9)$$

$$\frac{\partial g}{\partial q} = -\frac{\sum w_i^*/C}{(\tau + q^*/C)^2} = -\frac{C}{\eta(1 + \frac{N\alpha}{C})} \quad (3.10)$$

thus the matrix \mathbf{A} is of the form:

$$A = \begin{pmatrix} -\beta & 0 & \dots & 0 & \gamma \\ 0 & -\beta & & \vdots & \gamma \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \dots & 0 & -\beta & \gamma \\ -\delta & -\delta & \dots & -\delta & -\delta \end{pmatrix}$$

where:

$$\beta = \frac{KCN\alpha}{\eta(C + N\alpha)} \quad , \quad \gamma = \frac{KC}{N\eta(1 + \frac{N\alpha}{C})} \quad , \quad \delta = \frac{C}{\eta(1 + \frac{N\alpha}{C})}$$

now we can compute the poles of the linearized dynamical system:

$$|sI - A| = (s + \beta)^{N-1}(s^2 + (\beta + \delta)s + \delta(\beta + \gamma)) \quad (3.11)$$

$$\Re\{s_{1,2}\} = -(\beta + \delta) \quad , \quad \Re\{s_{3,\dots,N+1}\} = -\beta$$

As we can see all the poles belong to left half plane (LHP), thus the closed loop dynamics is **stable** in other words, the equilibrium point is an **attractor**. The dominant part of the dynamics is the second order part. We also notice that since the equilibrium is an attractor, starting from zero, the window size is always positive.

3.2 Robustness

In this section we study the sensitivity of the dynamics of closed loop network which runs our new protocol. As mentioned in the previous section the dominant part of the closed loop dynamics is the second order part. We already know that the response of a second order system depends on a parameter called *the damping ratio* (denoted by ζ). In figure [3.2], one can see the change in step-response of a second order system when the damping ratio takes different values:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

ζ : Damping Ratio ω_n : Natural Frequency

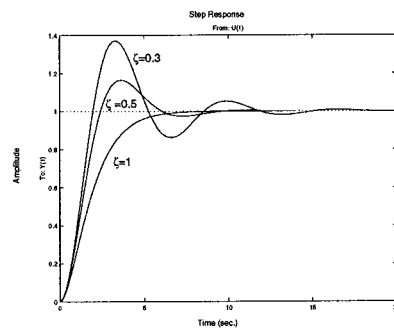


Figure 3.1 : step response of a second order system for different values of damping ratio

From (3.11) we can compute the damping ratio of the dominant second order dynamics:

$$\zeta = \frac{1}{2} \sqrt{\frac{KC \sqrt{N^2\alpha + C}}{N(KN\alpha + C)}} \quad (3.12)$$

now we examine the change in the damping ratio of the dominant second order dynamics when a single source is entering or leaving the network.

$$\frac{\partial \zeta}{\partial N} = -\frac{\zeta}{2N} + \frac{(N-K)\alpha C \zeta}{(N^2\alpha + C)(KN\alpha + C)} \quad (3.13)$$

for $N \gg 1$, $|\delta N| = 1$ and $K \ll \frac{C}{N\alpha}$:

$$\left| \frac{\delta \zeta}{\zeta} \right| \approx \frac{1}{2N} \quad (3.14)$$

in other words, the change of dynamics is negligible for large N when a single user enters or leaves the network.

Chapter 4

Analysis (Heterogeneous Case)

4.1 Stability Analysis

In this section we develop the stability analysis for the heterogeneous case in which the round trip delays, also the congestion control parameters of each individual source (K and α) are different. The framework for the analysis is the same, we find the equilibrium point of the closed loop dynamical system and then check the stability of the linearized model. So we start from the original dynamical system:

$$\begin{cases} \dot{w}_i = K_i \left(\alpha_i - \frac{w_i}{\tau_i} + \frac{w_i}{\tau_i + q/C} \right) & \text{for } i = 1, \dots, N \\ \dot{q} = \sum_{j=1}^N \frac{w_j}{\tau_j + q/C} - C \end{cases} \quad (4.1)$$

now we find the equilibrium point:

$$\dot{w}_i = 0 \implies \alpha_i = \frac{w_i^*}{\tau_i} - \frac{w_i^*}{\tau_i + q^*/C} \implies \frac{Cw_i^*q^*}{\eta_i(\eta_i + q^*)} = \alpha_i \implies w_i^* = \frac{\alpha_i \eta_i (\eta_i + q^*)}{Cq^*} \quad (4.2)$$

On the other hand:

$$\dot{q} = 0 \implies \sum \frac{w_i^*}{\tau_i + q/C} = C \implies \sum \frac{w_i^*}{\eta_i + q} = 1 \quad (4.3)$$

where $\eta_i := \tau_i C$, now from (4.2) and (4.3) we have:

$$\begin{cases} q^* = \frac{\alpha_i \eta_i}{C \rho_i} \\ w_i^* = \eta_i \left(\rho_i + \frac{\alpha_i}{C} \right) \\ x_i^* = C \rho_i \end{cases} \quad (4.4)$$

where $\rho_i := \frac{\alpha_i \eta_i}{\sum_{j=1}^N \alpha_j \eta_j}$ and $x_i^* = \frac{w_i^*}{\tau_i + q^*/C}$ is the steady state rate of the source i .

Now similar to what we did in section 3.1, using (4.4) we can form the matrix A by computing its elements:

$$\beta_i = -\frac{\partial f_i}{\partial w_i} \Big|_{(w_i^*, q^*)} = \frac{K_i \alpha_i}{\eta_i (\rho_i + \alpha_i / C)} \quad (4.5)$$

$$\gamma_i = \frac{\partial g}{\partial w_i} \Big|_{(w_i^*, q^*)} = \frac{C \rho_i}{\eta_i (\rho_i + \alpha_i / C)} = \left(\frac{C \rho_i}{K_i \alpha_i} \right) \beta_i \quad (4.6)$$

$$\delta_i = -\frac{\partial f_i}{\partial q} \Big|_{(w_i^*, q^*)} = \frac{C K_i \rho_i^2}{\eta_i (\rho_i + \alpha_i / C)} = K_i \rho_i \gamma_i \quad (4.7)$$

$$\epsilon = -\frac{\partial w}{\partial q} \Big|_{(w_i^*, q^*)} = \sum_{i=1}^N \frac{C K_i \rho_i^2}{\eta_i (\rho_i + \alpha_i / C)} = \sum_{i=1}^N \delta_i \quad (4.8)$$

and the matrix A is of the form:

$$A = \begin{pmatrix} -\beta_1 & 0 & \dots & 0 & \gamma_1 \\ 0 & -\beta_2 & & \vdots & \gamma_2 \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \dots & 0 & -\beta_N & \gamma_N \\ -\delta_1 & -\delta_2 & \dots & -\delta_N & -\epsilon \end{pmatrix}$$

Note that $\beta_i, \gamma_i, \delta_i, \epsilon > 0$. In order to check the stability, we should compute the characteristic polynomial of A , by direct computation we obtain the following recursive formula:

$$D_N = |sI - A| = (s + \beta_1)D_{N-1} + (-1)^{N+1} \gamma_1 Q_{N-1} \quad (4.9)$$

$$Q_{N-1} = \begin{vmatrix} 0 & s + \beta_2 & 0 & \dots & 0 \\ 0 & 0 & s + \beta_3 & & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & s + \beta_N \\ \delta_1 & \delta_2 & \dots & \delta_{N-1} & \delta_N \end{vmatrix}$$

by induction one can easily verify that:

$$Q_{N-1} = (-1)^{N-1} \delta_N \prod_{i=1}^{N-1} (s + \beta_i) \quad (4.10)$$

now by substituting (4.10) in (4.9) and going through recursion we obtain the following closed form formula for D_N :

$$D_N(s) = P_N(s)F_N(s) \quad (4.11)$$

where $P_N(s) = \prod_{i=1}^N (s + \beta_i)$ and $F_N(s) = (s + \epsilon) + \sum_{i=1}^N \frac{\gamma_i \delta_i}{s + \beta_i}$.

Now to show the stability of the equilibrium point, we have to solve the equation $D_N(s) = 0$ and check the zeros, if all the zeros of the function $D_N(s)$ have negative real parts, then the equilibrium point is **stable**.

Proposition: All the zeros of $D_N(s)$ have negative real parts.

proof: We only need to show that the zeros of $F_N(s)$ have negative real parts. We know that $\frac{\gamma_i \delta_i}{s + \beta_i}$ is an SPR (strictly positive real) function. Thus $F_N(s)$ is the sum of SPR functions, hence $F_N(s)$ is SPR. Obviously, $F_N(s)$ can not have a zero with positive real part hence we conclude that $D_N(s)$ can not have a zero with positive

real part. ■

Corollary: The linearized dynamical system (4.1) is dissipative, i.e., $A + A^T < 0$.

proof: One can easily verify that the characteristic polynomial of $A + A^T$ is:

$$G_N(s) = \left(\prod_{i=1}^N (s + 2\beta_i) \right) \left((s + \epsilon) + \sum_{i=1}^N \frac{(\gamma_i - \delta_i)^2}{s + 2\beta_i} \right)$$

with a proof similar to the above proposition we can conclude that all eigen values of the matrix $A + A^T$ are negative thus $A + A^T < 0$. ■

This explains that the system around the equilibrium point is not only stable, but also passive, meaning that it only dissipates energy and from the system theory we know that the interconnection of passive systems is still passive, hence we can argue that TCP-Rice is a scalable protocol.

4.2 Tuning the Parameters

In this section we propose a method to tune the congestion control parameters K and α . Tuning the parameters is one of the most important problems because careless choice of the parameters may cause degraded performance or even instability in the network. Also, parameter tuning might not be an easy task (for example, it is a well known fact that tuning RED parameters is very difficult, [14], [17]). Now we propose a method to choose the parameters α and K in a way that both stability and performance conditions are satisfied. First, we try to choose K_i in a way that stability is preserved. In order to do this, we have to find a *Lyapunov function* and then, set K in a way that the time derivative of this function is strictly negative. Now

let's rewrite the window variation law of our protocol (4.1) in the following form:

$$\dot{w}_i = \frac{K_i}{\tau_i} \left(\Delta - \left(\frac{q}{C} \right) x_i \right) \quad (4.12)$$

where the term $\Delta - \left(\frac{q}{C} \right) x_i$ represents difference between the ideal and the actual share of the source i in the total backlog. $G_i = \frac{K_i}{\tau_i}$ defines the gain and as can be seen it is inversely proportional to the round trip delay. This is a requirement for stability as mentioned in [12]. One can also see that by choosing $K_i = \frac{\kappa}{RTT_i w_i}$ (where κ is a global constant) our algorithm reduces to the congestion control algorithm proposed in [15]. We have tried different Lyapunov candidates to find the optimal solution for K_i but still unsuccessful. So the solution to this problem in an adaptive optimal way remains a challenging problem for our future research.

To set the parameter α for each source, we use a rather intuitive way by choosing α_i to be:

$$\alpha_i = \frac{\Delta}{\tau_i} \quad (4.13)$$

in which Δ is a constant parameter for all sources. The motivation for choosing α_i as in (4.13) is to obtain a fair share of bandwidth for each source at steady state. This can be seen by computing the steady state values for window size, queue length and rate. Using (4.4) and (4.13) we have:

$$\begin{cases} w_i^* = \frac{\eta_i}{N} + \Delta \\ q^* = N\Delta \\ x_i^* = \frac{C}{N} \end{cases} \quad (4.14)$$

as can be seen, the steady state rate is the same for each source so a fair allocation of bandwidth is achieved. The parameter Δ can be interpreted as the share of each

source in building up the total backlog. Choosing Δ in an optimal way is also a striking question because it is directly related to fairness issue. In an ideal scenario, Δ should tend to zero (empty queues). One suggestion is to use the ECN bit to tune α in an exponential way:

$$\Delta = \Delta_0 \phi^{-\mathbb{E}[ECN]} \quad (4.15)$$

for some globally constant ϕ . The expected value can be computed as the average of ECN bit over an appropriate time scale. We also notice that the expected value of ECN bit is equal to the probability of Δ being greater than the marking threshold. This technique is more or less similar to REM technique as discussed in [2]. but the optimal solution to this problem is also a question which requires further study.

4.3 Compatibility and Competitiveness:

One important issue about designing a protocol is the compatibility and competitiveness of the protocol. Our protocol should be able to be implemented in the current network in which the aggressive TCP-Reno is ruling. Vegas was not successful in achieving this goal. What we propose here is a remedy for this. At steady state the rate formulate for our protocol is:

$$\alpha_i = \frac{\bar{q}\bar{x}_i}{\eta_i} \quad (4.16)$$

now we would like the steady state rate to be compatible with that of a Reno source. so we replace \bar{x}_i by the well-known rate for TCP-Reno first mentioned in [13]:

$$\bar{x}_i = \frac{1}{RTT_i} \sqrt{\frac{3}{2p}}$$

where p is the probability of loss. Hence we have:

$$\alpha_i = \frac{\bar{d}}{\tau_i} \frac{1}{RTT_i} \sqrt{\frac{3}{2p}} \quad (4.17)$$

where $\bar{d} = \bar{q}/C$ is the average queuing delay. In fact, in (4.17) two network-centric prices, i.e. loss probability (Reno) and average queue delay (Vegas), are employed for congestion avoidance. When routers deploy the RED algorithm, then $p = f(\bar{q})$. In order to estimate the probability of loss, as proposed in [6], we can use the inter-arrival time between losses.

Chapter 5

Fairness

In this chapter, we investigate the fairness of the new protocol. It was first addressed in [9] that the fairness and the stability of networks are two closely related concepts. While fairness has been considered as an economic issue, evaluated by utility functions, stability is traditionally regarded as an engineering issue which is analyzed by well-known mathematical techniques such as the Lyapunov method. In fact in [9],[10] Kelly showed that the problem of fair allocation of bandwidth among different sources which is formulated as a convex constrained optimization problem, can be solved by treating the aggregate utility function as a Lyapunov function that guarantees stability. Thus the equilibrium point which is asymptotically reached by the system, provides a solution to fairness problem. Depending on the type of the utility function, there are different types of fairness. The utility function is related to the dynamics of the protocol. For example, the utility function of TCP-Reno is of the form $U_s(x_s) = \frac{\sqrt{2}}{D_s} \alpha_s \tan^{-1} \left(\frac{x_s D_s}{\sqrt{2}} \right)$, whereas that of TCP-Vegas is of the form $U_s(x_s) = \alpha_s d_s \log(x_s)$, where x_s , D_s and d_s are the sending rate, round trip time and propagation delay of source s , respectively. The utility function of TCP-Vegas defines a different fairness criteria known as *weighted proportional fairness*, defined as follows. Let $x^* = (x_1^*, \dots, x_N^*)$ be a feasible rate vector, namely, $x_i^* > 0$ and $\sum_{i=1}^N x_i^* \leq C$, then x^* is a *weighted proportionally fair* rate vector if for all other feasible rate vectors x we have:

$$\sum_{i=1}^N \omega_i \frac{x_i - x_i^*}{x_i^*} \leq 0 \quad (5.1)$$

Now we claim that the asymptotically stable equilibrium point in (4.4) provides a weighted proportionally fair allocation of bandwidth.

Proof: From (4.4) we have $x_i^* = C\rho_i$ and $\rho_i = \frac{\alpha_i\eta_i}{\sum_{j=1}^N \alpha_j\eta_j}$. Since x_i is a feasible rate, we have:

$$\sum_{i=1}^N x_i^* = C \sum_{i=1}^N \rho_i = C \implies \sum_{i=1}^N x_i \leq \sum_{i=1}^N x_i^* \implies \sum_{i=1}^N \alpha_i\eta_i \left(\frac{x_i - x_i^*}{\alpha_i\eta_i} \right) \leq 0$$

and hence,

$$\sum_{i=1}^N \alpha_i\tau_i \left(\frac{x_i - x_i^*}{x_i^*} \right) \leq 0$$

and this explains that in steady state, like TCP-Vegas, our protocol also leads to a weighted proportionally fair allocation of bandwidth. However, we should mention that if we tune the parameters of TCP-rice to compete with Reno, then the proportional fairness result does not hold any more and like Reno, TCP-Rice also shows bias towards connections with smaller round-trip time. However, if we tune TCP-Rice to compete with Reno, the fairness result does not hold anymore and like Reno, TCP-Rice shows bias towards connections with smaller delays.

Chapter 6

Simulation Results

In this chapter we examine the new protocol by software simulation. We have used our own discrete event simulator written in Matlab. Since we are only interested in long term behavior (connections are elephants), we have not considered the slow-start phase in the simulation. We have carried on the simulation for both homogeneous and heterogeneous cases as follows.

6.1 Homogeneous Case:

In the first experiment, we would like to compare the behavior of the new protocol to Reno (which is the current protocol running on Internet) and Vegas and then see the behavior of the protocol in a mixed scenario in which half of the sources run TCP-Reno whereas the other half run TCP-Rice. In order to be able to compare to Vegas, we also repeat the same experiment by mixing up Vegas and Reno sessions together. We have assumed the round-trip delays (τ) are all equal to 10 *millisecond* (homogeneous case) and $N = 10$, $K = .05$ and $C = 50,000\text{pkt}/\text{sec}$. Another assumption is that all sources send packets in a synchronized way. We notice that this is a worst case scenario and it happens when the network goes through *global synchronization*. In figure (6.1-b) we can see the individual behavior of TCP-Reno and TCP-Rice. As can be seen, Reno sources increase their window size (dashed line) up to where the queue is full and then they experience a loss and suddenly reduce their window size, this causes an oscillatory behavior for Reno which is well-known in the literature [11],

(also known as *global synchronization*). As can be seen in figure (6.1-a) this oscillatory window size variations are induced to the queue variations (dashed line) which is not desirable. Compared to Reno, the 10 sessions of TCP-Rice behave differently, in figure (6.1-b) the window size (solid line) after a short transient behavior) reaches a constant equilibrium which verifies our previous stability results and as can be seen the value of this equilibrium is considerably less than the maximum window size in our experiment with Reno. This result leads to much less queue size, as can be seen in figure (6.1-a) (solid line). Also one can recognize the second order dynamic behavior in the variations of queue which again verifies our analytic results.

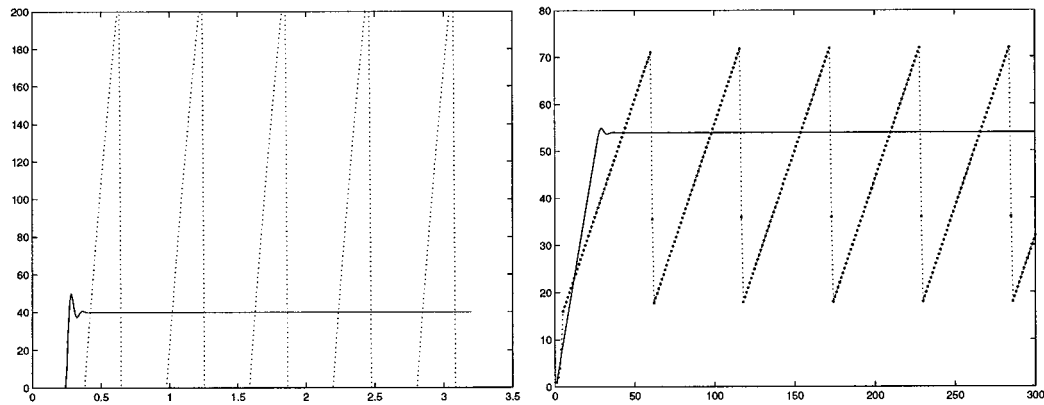


Figure 6.1 : Reno vs. Rice comparison a) Queue length b) Window size

In another experiment, we choose 5 Rice sources to compete with 5 sources running Reno. The result is shown in figure (6.1). As can be seen, sources running Rice increase their window size much faster than the sluggish Reno sessions (multiplicative increase) but as Reno sources keep increasing their window size and make the queue almost full, Rice sources multiplicatively decrease their window size to compensate for the irresponsible behavior of Reno, but since Reno sources keep increasing their window

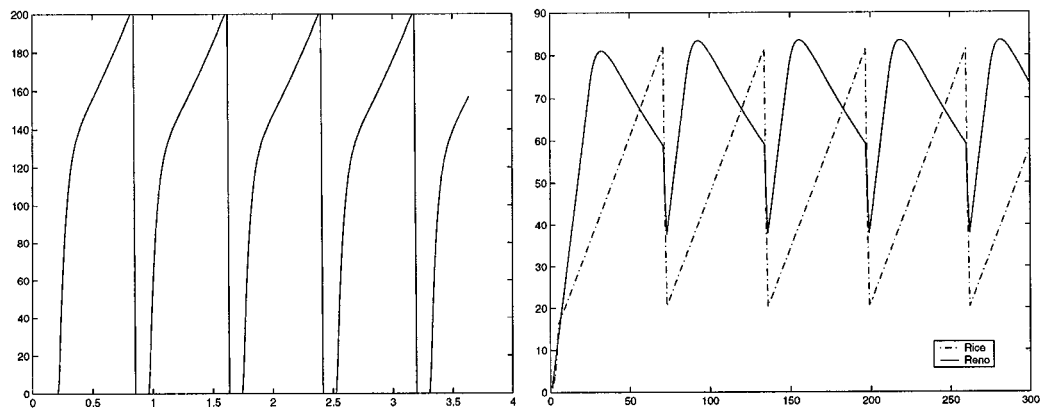


Figure 6.2 : Reno vs. Rice comparison a) Queue length b) Window size

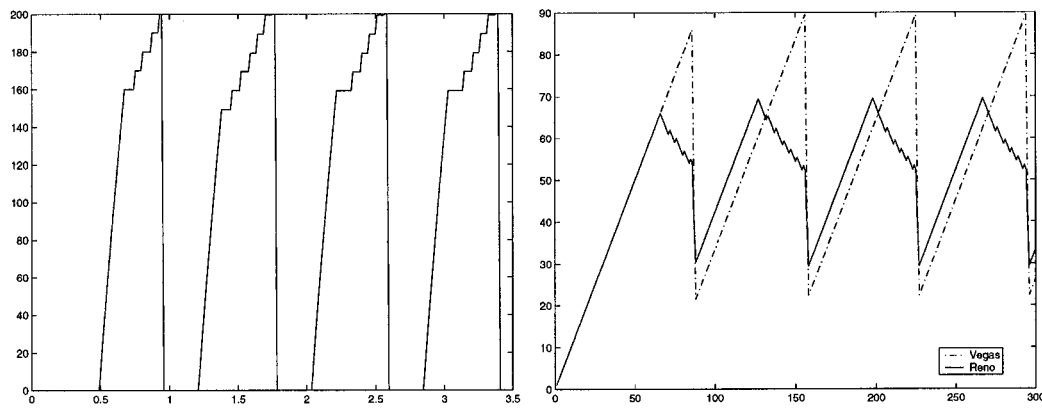


Figure 6.3 : Reno vs. Vegas comparison a) Queue length b) Window size

size, the queue becomes full and loss happens, then both Reno and Rice halve their window size. On average one can see that Rice sources' throughput is more than Reno's and moreover, Rice sources do not cause loss also the queue experiences a better utilization in this case.

We also did the same experiment with Vegas, and as can be seen in figure (6.1) the sources running Vegas are beaten by Reno, because as the Reno sources increase their window size, the queue enlarges and this cause the Vegas sources to decrease their window size and since the increase rate of Vegas and Reno are the same (one per round-trip time) on average, Vegas sources receive less share of bandwidth.

6.2 Heterogeneous Case:

In this case, we performed the simulation when users have different propagation delays and also they are not synchronized. We picked 10 users whose round trip delays range from 50 to 140 milli-second. The bottle-neck link capacity is assumed to be $C = 2000 \text{ pkt/sec}$ and the maximum queue length is 200 pkts (Round Trip Delays (in milli-seconds): {50, 62, 70, 80, 66, 72, 56, 140, 72, 50}).

In order to compare, we carried on two separate experiments both with Vegas and Rice. In both cases we have assumed $\Delta = 15$. In table (6.2) we can see the rate values for the first 5 sources of each experiment. As can be seen, like Vegas, Rice sessions receive a fair share of bandwidth.

RTT (ms)	50	62	70	80	66
Rate vector (Vegas)	180.1	197.3	218.7	207.2	201.3
Rate vector (Rice)	200.9	205.4	209.9	187.0	211.1

Table 6.1 : Comparison of rates, achieved by Vegas and Rice

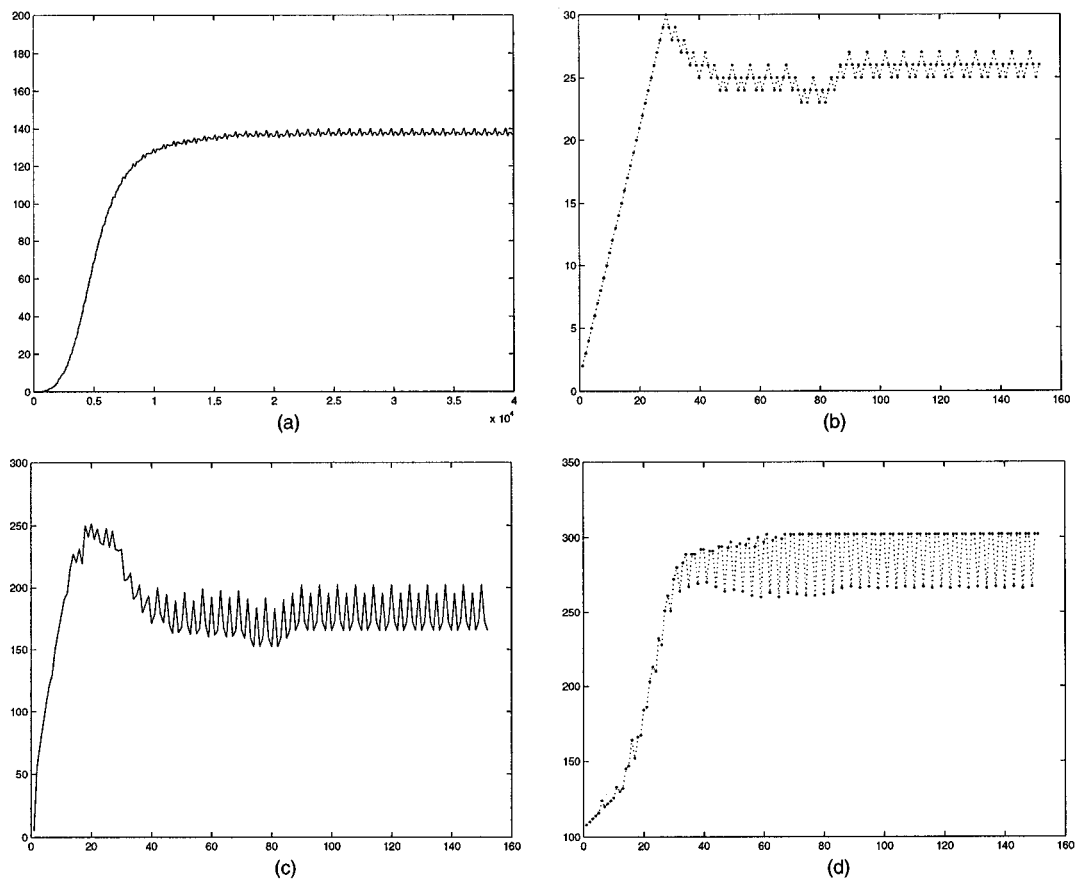


Figure 6.4 : Vegas: a) Queue length b) Window size c) Rate d) RTT

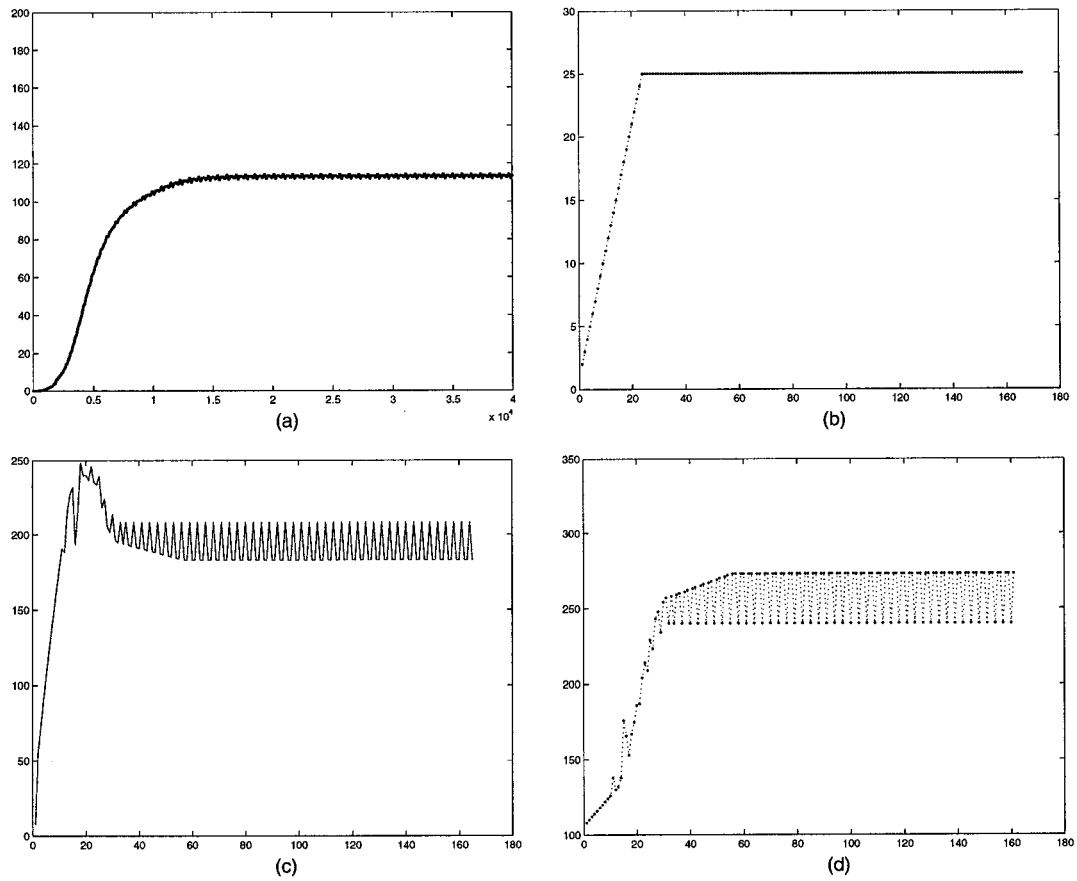


Figure 6.5 : Rice: a) Queue length b) Window size c) Rate d) RTT

Figure (6.2) and (6.2) depict the result of our experiment with Vegas and Rice, respectively. TCP-Rice lead to a smoother window size, smaller queue thus smaller delay.

We also carried on another experiment to see if TCP-Rice can survive together with aggressive Reno sources. We have used the results obtained in section 4.3 to update the parameter α . Again we assume there are 10 sources and the bottle-neck link capacity is $C = 2000$ pkt/sec. In figure (6.6) we see the results when all the sources run Reno. As can be seen the traffic is bursty (the probability of loss is 0.002) and round trip times have a lot of variations. In figure (6.7) we did the same simulation with 10 sessions of Rice and the results show that in this case, queue length exhibits smoother variations (the probability of loss is $6.75e - 4$) and the window size does not oscillate like Reno and round-trip times have smaller variance. Finally we mixed the two experiments together, to have 5 sessions of Rice competing with 5 Reno sources. The results are depicted in figure (6.8). They suggest that TCP-Rice can successfully compete with Reno at a lesser price (the probability of loss is 0.0016 in this case).

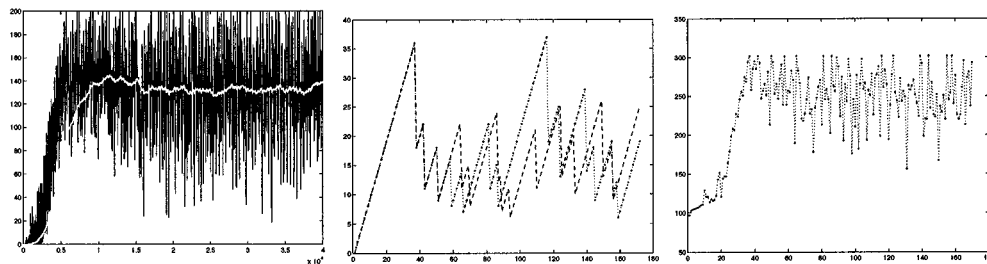


Figure 6.6 : a) Reno: Queue length($P_{loss} = 0.002$) b) Window size c) RTT

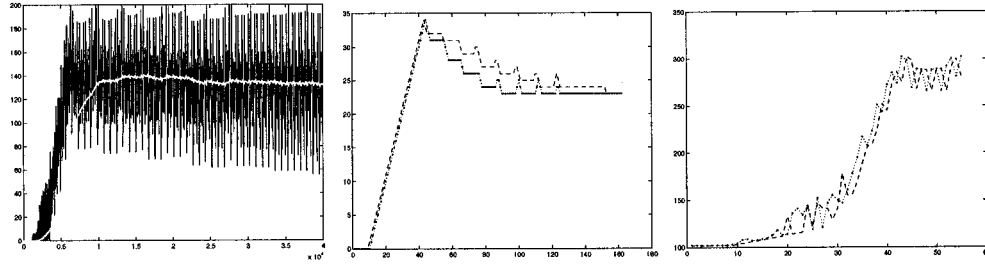


Figure 6.7 : Rice: a) Queue length ($P_{loss} = 6.75e - 4$) b) Window size c)RTT

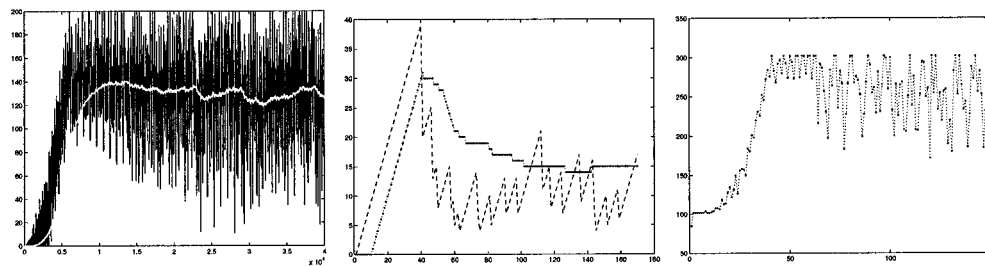


Figure 6.8 : a) Queue length ($P_{loss} = 0.0015$) b) Window size c)RTT

Chapter 7

Conclusion and Future Avenue

In this thesis, the idea of designing a new protocol with better performance (in terms of higher utilization and less loss) over the existing data transfer protocols on the Internet has been revisited. We proposed a new protocol which is basically an enhanced version of Vegas. Unlike previous works, our approach for analysis is in the framework of large-scale dynamical systems. We proved the stability of the protocol (when arbitrary number of sources are involved) for both homogeneous and heterogeneous cases. Also we showed that the stable equilibrium leads to a fair share of bandwidth for each source. Consequently, we proposed a method to tune the gain parameter K (to conserve the Lyapunov stability) and the rate parameter α (to be compatible and competitive with TCP-Reno).

Simulation results show that this new protocol possesses all the good features of Vegas (less loss, high utilization, stability and fairness) and at the same time it can be deployed in the heterogeneous environment of the current Internet in which TCP-Reno is ruling.

There are a plenty of open questions to answer for further research. One interesting problem is how to tune the parameters of each source in a hybrid scenario where some of the hosts use other protocols. Also using ECN and marking to tune the parameters is an interesting problem to study. For the new protocol, measuring the burstiness of the queue and studying the behavior with mice background mice can be a another topic for research. Another interesting research avenue is the complexity reduction.

In [11] it was shown that the complex closed loop dynamic of TCP/RED can be modeled by a high order LTI system. We also showed in chapter 3 and 4 that the closed loop dynamics of our protocol can be approximated by a high order linear dynamics. There is a well-developed theory for model reduction of linear systems [1], so one interesting idea is to use these techniques to approximate the dynamics of data transfer protocols with low-order linear systems which are computationally manageable. High-speed software tools for network simulation is a possible outcome of this endeavor.

Bibliography

- [1] A. C. Antoulas *Lectures on the approximation of large-scale dynamical systems*, SIAM Book Series: Advances in Design and Control.
- [2] A. Athuraliya, V. H. Li, S. H. Low and Q. Yin. *REM: Active queue management*, IEEE Network, May/June 2001.
- [3] L. S. Brakmo, S. W. O'Malley, and L. Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In Proc. of ACM SIGCOMM '94, pages 24-35, London, Oct 1994.
- [4] L. S. Brakmo and L. Peterson. Performance Problems in BSD4.4 TCP. Computer Communication Review, 25(5):69-86, Oct 1995.
- [5] V. Firoiu and M. Borden, *A study of active queue managements for congestion control*, In Proceedings of IEEE INFOCOM, March 2000.
- [6] S. Floyd, M. Handley, J. Padhye, J. Widmer, *Equation-based congestion control for unicast applications*, Proc. of SIGCOMM 2000.
- [7] U. Hengartner, J. Bolliger, Th. Gross, *TCP Vegas revisited*. In the Proc. of IEEE INFOCOMM 2000.
- [8] C. Hollot, V. Misra, D. Towsley, and W. B. Gong, *A control theoretic analysis of RED* in Proc. of IEEE Infocom April 2001.

- [9] F. P. Kelly, A. K. Maulloo and D. Tan, *Rate control for communication networks: shadow prices, proportional fairness and stability*, J. of Operations Research Society, 49, March, pp:237-252.
- [10] F. P. Kelly *Mathematical modeling of the Internet* In Proc. of 4th International Congress On Industrial and Applied Mathematics, 1999.
- [11] S. H. Low, F. Paganini, J. Wang, S. Adlakha, J. C. Doyle, *Dynamics of TCP/RED and Scalable Control*, Proc. of IEEE INFOCOM, JUNE 2002.
- [12] S. Low, F. Paganini, J. Doyle, *Internet Congestion Control: An Analytical Perspective*, IEEE Control Systems Magazine , Jan. 2002.
- [13] J. Mahdavi and S. Floyd, *TCP-friendliness unicast rate-based flow control*. Note sent to end2end-interest mailing list, Jan 1997.
- [14] M. May, T. Bonald, J. C. Bolot *An analytic evaluation of RED performance*, in Proc. of IEEE Infocom, Mar. 2000.
- [15] J. Mo and J. Walrand, *Fair end-to-end window-based congestion control*, IEEE Trans. on Networking, Vol 8, No. 5, Oct 2000.
- [16] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, *Modeling TCP Throughput: a simple model and its empirical validation*, Proc. of SIGCOMM 1998.
- [17] M. Christiansen, K. Jeffay, D. Ott, F.D. Smith, *Tuning RED for web traffic*, in Proc. of ACM Sigcomm, 2000.