uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Yang Hong
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Adaptive Controller Design for Internet Traffic: A Control Theoretical Approach

TITRE DE LA THÈSE / TITLE OF THESIS

Oliver Yang
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Peter Galko

Dimitrios Makrakis

Jörg Liebeherr

Peter Liu (absent)

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Adaptive Controller Design for Internet Traffic: A Control Theoretical Approach

by

Yang Hong

A thesis submitted to
the Faculty of Postdoctoral and Graduate Studies
in partial fulfillment of requirements for the degree of

Doctor of Philosophy

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario
Canada

May 2006

Your file Votre référence
ISBN: 978-0-494-18584-1
Our file Notre référence
ISBN: 978-0-494-18584-1

Abstract

This thesis presents adaptive control techniques for the prevention of end-to-end traffic congestion in the Internet. Instead of using heuristics to solve the congestion problem, we have systematically applied the classical control theory in our controller design in order to always guarantee the stability of the Internet control systems, and instead of using time-domain approaches, we have employed the frequency domain techniques in our control design.

To support AQM(Active Queue Management) in the router, we have designed the adaptive P (Proportional) and the adaptive PI (Proportional-Integral) window-based router controllers for the TCP (Transmission Control Protocol) traffic. Our adaptive controllers self-tune only when significant changes in network parameters have caused the phase margin or the gain margin of the AQM control system to drift outside a specified interval.

Using the same control theoretical framework, we have provided different controllers to support best-effort streaming media traffic under different network models that we want to study. We first design an adaptive PI rate-based controller in a typical network with SISO (Single-Input Single-Output) IP (Internet Protocol) routers. We then employ the DNA (Direct Nyquist Array) method to design an adaptive multi-loop PI rate-based controller under a network with MIMO (Multiple-Input Multiple-Output) IP routers.

In addition to window-based and rate-based router control algorithms, we finally design an adaptive utility-based source controller based on pole-placement technique that can provide a relatively smooth source transmission rate.

Acknowledgements

The greatest gratitude should be extended to Professor Oliver W.W. Yang for his academic supervision. Without his great guidance on my work, I cannot finish this thesis.

I would like to thank the members of my PhD advisory committee for their guidance at various stages of my PhD program, and to the many members in our computer communication network research group for many helpful discussions.

I am deeply appreciated by the constant love and support from my parents.

The financial support from NSERC Postgraduate Scholarship, University of Ottawa National Scholarship and NSERC Discovery Grant (#RGPIN42878) are also acknowledged.

Table of Contents

List of Figures

List of Tables

Table of Acronyms and Abbreviation

		Section of 1st Appearance
AIMD	Additive-Increase Multiplicative-Decrease	1.1
AQM	Active Queue Management	1.1
ARED	Adaptive RED	4.3
AVQ	Adaptive Virtual Queue	1.1
DNA	Direct Nyquist Array	1.1
ECN	Explicit Congestion Notification	1.1
ERD	Early Random Drop	1.1
GM	Gain Margin	2.1
IETF	Internet Engineering Task Force	1.1
IP	Internet Protocol	2.1
MIMO	Multiple Input Multiple Output	1.1
P	Proportional	1.1
PI	Proportional-Integral	1.1
PID	Proportional-Integral-Derivative	1.1
PM	Phase Margin	2.1
QoS	Quality of Service	1.1
RAP	Rate Adaptation Protocol	1.1
RED	Random Early Detection	1.1
REF_QS	Reference Queue Size	3.4
REM	Random Exponential Marking	1.1
SISO	Single Input Single Output	1.1
TCP	Transmission Control Protocol	1.1
TFRC	TCP Friendly Rate Control	1.1

Table of Notations and Symbols

		Section of 1st Appearance
A_m	Gain margin	2.5
A_{Mi}	Gain margin of the dominant loop $g_{ii}(s)$ of MIMO AQM control system	6.2
A'_{Mi}	Gain margin of the dominant Loop $g_{ii}(s)$ shaped by the Gershgorin circle	6.2
$C(s)$	Transfer function of the router controller	2.2
$j\|i$	Controlled source j that transmits IP packets routed through the input port i	6.1
K_I	Integral gain of PI router controller for the AQM	3.4
K_{Ii}	Integral gain of multiloop PI rate rate-based router controller for output port i	6.2
K_P	Proportional gain of PI router controller for the AQM	3.2
K_{Pi}	Proportional gain of multiloop PI rate-based router controller for output port i	6.2
$N(t)$	Load factor (Number of active long-lived TCP sessions)	3.1
$p(t)$	Packet drop probability	3.1
$q_i(t)$	Source transmission rate for controlled source i	5.1
$q_{ij}(t)$	Source transmission rate for controlled source $j\|i$	6.1
$q_r(t)$	Source transmission rate for utility-based controlled source r	2.3
$W(t)$	Average TCP window size (packets)	Apndx A
w_r	Weight assigned to the user r according to the price paid by the user r	2.3
$x(t)$	Instantaneous queue size	2.1
$x_i(t)$	Instantaneous queue length in the output port i of the MIMO IP router	6.1
α_{ij}	Traffic distribution ratio from the jth input port to the ith output port	6.1
ϕ_m	Phase margin	2.5
$\tau(t)$	Average Round-trip time = $x/\mu+\tau_p$ (secs)	2.1
τ_p	Propagation delay (secs)	2.1
τ	Average round trip time for all the controlled source nodes i	5.2
τ_i	Round trip time for controlled source i	5.2
τ_{bij}	Feedback time delay from the output port i of the MIMO IP router back to the controlled source node $j\|i$ via the destination	6.2
τ_{fij}	Forward time delay from the controlled source node $j\|i$ to the input port i of the MIMO IP router	6.1
τ_{ij}	Round trip time between the controlled source node $j\|i$ (which transmits the packets into the input port i) and its destination	6.2
$\bar{\tau}_i$	Average round trip time for all the controlled source nodes $j\|i$ that transmit the packets into the input port i	6.2
μ	Service rate of the router	2.1
$\mu(t)$	Link capacity (packets/sec)	2.1
μ_i	Service rate of the output port i in the MIMO IP router	6.1
$\upsilon(t)$	Input rate of guaranteed service traffic	5.1
ω_g	Gain crossover frequency	2.5
ω_p	Phase crossover frequency	2.5

Chapter 1 Introduction

The end-to-end congestion control has become an increasingly important problem in the robustness of the Internet due to ever-growing network bandwidth and intensive network applications e.g. [FlFa99]. In order to make maximum use of network resources, Internet may want to deliver packets at its maximum capacity. Unfortunately, without the feedback control mechanism applied, this usually results in congestion, leading to undesirable low throughput, excessive delays of packet delivery, and even possible network collapse. Thus, congestion control plays an important role in the effective and stable operation of Internet. The objectives of the congestion control are to regulate the source transmission rates based on the network congestion information and to prevent the network from the congestion. It is important for the effective and stable operation of Internet.

During the infancy of Internet, TCP/IP (Transmission Control Protocol/Internet Protocol) protocol as a congestion control is adequate for simple applications like non-real-time file transfer. However, as the Internet grows to support diverse applications like E-mail and multimedia communication, more stringent requirements have arisen in order to guarantee the network performance. In general, the flow control protocol in the TCP/IP suite is an end-to-end window-based protocol. The source transmission rate (via the window size) is halved whenever a packet is dropped in an overflowed router buffer. This leads to undesirable queuing delays and packet loss rates at congested links.

1.1 Literature Review

There are two general classification methods for the existing work done on Internet traffic control so far:

1) a bandwidth-allocation classification according to whether the control algorithm is rate-based or window-based.
2) a location-based classification according to whether the control algorithm is located in the router or in the source node [MaSe97].

We shall first use the first criteria for classification while using the second to sub-classify further in order to expose the shortcomings, deficiencies or further improvement to be desired in the existing work.

1.1.1 Window-based Control

In window-based control, a source is given a maximum number of credits (i.e., the window size) to transmit its packets. Once the credits are exhausted, the source has to withhold its transmission until acknowledgments are returned [Stal03]. All the window-based control methods can be either router-based or source-based, or a combination.

Source-based control attempts to control the packet transmission rate in each source based on the current network congestion situation. This is the basis of the popular flow control protocol in the original TCP/IP suite. Using an end-to-end window-based protocol, the source transmission rate (via the window size) is halved whenever a packet is dropped in an overflowed router buffer. This leads to undesirable queuing delays and packet loss rates at congested links. This type of flow control mechanism is called the AIMD (Additive-Increase Multiplicative-Decrease) [Jaco88]. It is very effective in the conventional IP networks where the propagation delays are short compared to the source transmission rates. However, under the network environment with long round trip delays and dynamically changing available bandwidth, the AIMD control mechanism becomes inefficient and often produces severe sending rate fluctuation in the sources and oscillation of queuing sizes in the routers [KaHa02, LoPa02]. Such fluctuation may greatly degrade the network performance, which makes AIMD control unsuitable for streaming media transmission [HuXu03, HaFl03].

Since it is unreasonable to exclusively rely on sources to perform end-to-end congestion control (i.e., AIMD source control), the IETF (Internet Engineering Task Force) has proposed the deployment of a class of router-based control algorithms [BrCl98] to improve the performance of TCP congestion control.

Called the AQM (Active Queue Management) mechanisms, this type of router-based control uses congestion avoidance to participate in the network traffic control. Unlike the drop-tail method expected in the original TCP protocol, random-drop is proposed for managing the queue length in an outbound buffer. Packets are dropped whenever congestion is anticipated, instead of waiting until the queue is full. The router adjusts packet drop/marking probability directly, thus regulating the source transmission rate indirectly.

As a first generation of window-based AQM, ERD (Early Random Drop) [Hash89] is proposed to be a congestion recovery mechanism. When the queue length exceeds a threshold, packets are dropped with a fixed probability. Simulations in [Zhan89] show that ERD cannot accommodate bursty traffic nor it can provide protection from misbehaving flows. To avoid the shortcoming of ERD, RED (Random Early Detection) [FlJa93] attempts to drop packets with a probability that may vary according to the average queue size, whenever the average queue size exceeds a preset threshold. However, it has been found that it is very difficult to tune RED parameters in order to perform well under different traffic conditions. One common problem is the severe queue fluctuation. Furthermore, the flow throughput of RED is sensitive to the traffic load and the parameter setting; the throughput will decrease dramatically when the average queue size is larger than the maximum queue threshold [MaBo99, FlGu01].

There has been a lot of research effort in window-based router control to improve AQM performance over RED. For example, BLUE [FeSh02] is a self-configuring (adaptive) AQM mechanism, which uses the packet loss and the link under-utilization events to adjust the rate of congestion notification. REM (Random Exponential Marking) [AtLi01] attempts to stabilize both the input rate around the link capacity and the queue around a small target at the steady state, regardless of the number of users sharing the link. AVQ (Adaptive Virtual Queue) scheme [KuSr04] tries to maintain the small queue size even during high congestion, and adopts the token bucket algorithm to limits the virtual queue capacity to be less than the actual link capacity. Other AQM algorithms such as FRED [LiMo97], SRED [OtLa99], DRED [OtLa99], CHOKe [PaPr99] and ARED [FlGu01] also try to adjust the packet drop probability to prevent network congestion. All of the above have succeeded to alleviate the RED performance problem but to certain extent only. From the control theory point of view, RED, BLUE and other AQM algorithms mentioned above are considered to be an on-off control strategy. It is well known from control theory that such an on-off mechanism can often lead to oscillations (limit-cycles) and exhibit complex and chaotic behavior [AsGo95]. Such oscillation may deteriorate which in turn degrade the network performance [HoMi02]. As an additional point, most of these protocols are heuristics without strong theoretical backing. Therefore, they cannot reveal the underlying relationship among the network and control parameters that can required to give a good (or poor) network performance in a more stringent environment (e.g., great changes in network traffic).

Strictly speaking, source control and router control usually co-exist in the network traffic control in order to maximize the utilization of the network resources. For example, the AIMD control [Jaco88] and the RED [FlJa93] cooperate to control Internet traffic control in to certain extent. Nevertheless, one can see from the discussion above that they all have much improvement to be desired of their performance.

1.1.2 Rate-based Control

Unlike the window-based control mechanism, rate-based control allows an end system to adjust its sending rate to support best-effort service traffic and to make the maximum use of network resources. Thus it offers the most effective solution under a network environment with long round trip delays and dynamical changes in available bandwidth where the AIMD control performs poorly in terms of sending rate fluctuation and queue oscillation (Ref. Section 1.1.1). Originally proposed for ATM networks e.g., [ZhYa97, BlCi02, CaGe04], this approach is very desirable in achieving a high QoS (Quality of Service) for streaming media transmission in the Internet [HuXu03, HaFl03]. Similar to the window-based control, the algorithms can also be sub-classified as either router-based or source-based.

In the router-based control, the rate-based method also uses AQM to allow the router to participate in the network traffic control. Basically, the router calculates the advertised source transmission rate directly and converts it into the advertised source window size, thus regulating the source transmission rate "directly" [GeLo02, KaHa02, HoYa05b]. A rate-based router control scheme was presented for transparently augmenting the end-to-end TCP performance by controlling the sending rate of a source in [KaKa00]. By controlling the sending rate of a source, the rate-based method of [KaKa00] is capable to augment the end-to-end TCP performance. Another rate-based router control scheme used the mismatch between measured incoming TCP traffic rates and available service rate to calculate the effective window size [AwOu02]. A two-state adaptive rate-based router control mechanism was proposed for streaming media in [HuXu03].

Rate-based router controller not only achieves fair allocation of network bandwidth and high utilization of network resource, but is also shown [GeLo02] to provide relatively smooth source transmission rates. However, this performance improvement does not come for free. First, an IP router must calculate a desirable advertised source transmission rate for each packet. Secondly, in a high-speed network with long propagation delays, IP routers must have huge buffer sizes (e.g., to accommodate the number of packets equivalent to at least the round-trip delay-bandwidth product for a PI rate-based router controller) so that the sources have enough time to regulate their transmission rates to avoid excessive buffer overflow [GeWe00]. Finally, like their counter parts in window-based control, most of these methods are heuristics without theoretical backing of their performance claim.

Rate-based control can also be source-based. Some methods have been proposed recently for AQM in IP router supporting best effort service traffic (e.g. streaming media such as audio and video over IP). For example, TFRC (TCP-Friendly Rate Control) proposal was presented for equation-based congestion control for unicast traffic in [FlHa00, HaFl03]. RAP (Rate Adaptation Protocol) employed an improved AIMD source control scheme for real-time stream transmission [ReHa99].

The utility-based control [KeMa98, JiWe04, HoYa05a] is a variation of the rate-based source control method that has emerged recently. The source in the IP networks adjusts its transmission rate based on the feedback network congestion information and its own utility function, which is called utility-based control. Utility function is applied in the controller design in order to make the optimal use of the network resources [KeMa98]. A framework model is used to analyze the stability and fairness of the rate control algorithm, and a primal algorithm is provided to adjust the users' sending rates to converge at an equilibrium state. An optimal solution to the resource allocation problem is then obtained under the assumption that there are no communication delays between the end users. Unfortunately, there always exists the communication delay over the physical channel especially in high-speed networks. In particular, the propagation delay may have an adverse impact on network performance due to the dramatic queuing fluctuation in the bottleneck. Such queuing oscillation may cause data flow fluctuation and reduce the bandwidth utilization of the network.

1.1.3 Control theory and Adaptive Algorithms

Control theory has found applications in many branches of engineering. For example, the PID (Proportional-Integral-Derivative) controllers have found wide acceptance and application in the process industries, e.g., [AsHa92, AsHa95, HoHo03]. Since the PID controller is not necessarily optimal, it is estimated that present sophisticated controllers would not have more economic effect in 80% of control loops [Bris86]. In view of the fact that PI controllers are becoming widely used for industrial process control e.g. [FuWa98, HoHo03], and will continue to be so, any new insights that lead to the effective use of this class of controllers will certainly bring about much benefit. Many sophisticated theories have evolved over the years. For example, the DNA (Direct Nyquist Array) design method has been presented for MIMO (Multiple-Input

Multiple Output) control system [Rose70], and the corresponding DNA Stability Theorem [Maci89] finds its application in the industrial process control [Desh89] by shaping a band of circles (called the Gershgorin band). However, their shaping is only done by trial and error manually [Maci89].

Classical control theory first found its application in computer networking in the 90's such as the traffic control for ATM networks, e.g., [ZhYa97]. Much AQM research work for the Internet traffic has followed after. Of all the existing work on traffic controller, they can be classified as whether

1) they use time-domain method as opposed to the frequency-domain method;

2) they use an adaptive control method as opposed to a fixed control method.

Time-domain analysis can be found in most of the early traffic-control works are time-based, e.g., [ZhYa97, BlCi02] and there are some recently, e.g., [ZhCo03, CaGe04]. A rate-based router control method is presented in co-operation with the RED scheme to control best-effort traffic in an environment of packet loss and varying round-trip delays [ZhCo03]. A PI rate-based source controller is designed for the realization of best-effort service traffic using Jury's stability test criterion in [HoYa03]. Time-based method may be easy for direct implementation but has inherent shortcomings of not able to provide a direct relationship between the controller parameters and system variables, thus not able to facilitate the implementation of real-time algorithm as discussed later. Therefore, much of the latest work appears to focus on the frequency domain method.

Frequency domain method allows one to understand the control system behavior better. For example, a PI window-based router control is designed for the TCP traffic based on pole placement [ChYa04]. The closed-loop pole locations can determine the stability of the control system, and can be shown to have a direct impact on system transient response characteristics such as rise time, settling time, and transient oscillations. Pole placement technique has been widely applied in the control system design. On the other hand, it is also typical to measure the stability robustness of the control system using gain and phase margins [HoLe01]. Based on these margins, a PI window-based router controller has been designed for AQM router to support TCP flows in [HoMi01a, HoMi02]. It can calculate a continuous and smooth packet drop probability based on the instantaneous queue size, thus avoiding possible queue oscillation in earlier window-based router-control design.

The papers by [HoMi01a, HoMi02] are perhaps the first papers to provide theoretical basis to back the performance of their controller design. Using a frequency domain approach, they clearly relate the network performance to the network parameters. However, there are several shortcomings. First, there is an assumption of $2N << \tau\mu$ (N, τ and μ will be defined later on) for calculating the parameters of PI window-based router controller in [HoMi01a]. Then the analytical method in [HoMi02] to derive the PI window-based router controller is very complicated, and difficult for real-time implementation. Finally, these two papers use a fixed solution for PI window-based router controller for TCP traffic, thus allowing no flexibility to react to great variations in the Internet traffic, which in turn may result in slow response to the traffic change and cause possible excessive buffer overflow.

There are also recent work in self-tuning window-based router controllers in [DeYi03, WuRe01, ZhHo03], but these controllers always self-tune unnecessarily even when the traffic load changes slightly.

A new type of control theoretical method for rate-based source controller is the utility-based congestion control scheme [KeMa98]. It can be divided into two classes: primal algorithms and dual algorithms. In primal algorithms, the users regulate their source sending rates dynamically based on the route prices, and the links select a static law to calculate their link prices directly from the arrival rates at the links. On the other hand, in dual algorithms such as [LoLa99, YaMa00], the links adjust their link prices dynamically based on the link rates, and the users select a static law to compute their source sending rates directly from the route prices and the source parameters [LiBa03]. Many attempts have been made to implement utility-based control for Internet traffic control recently. For example, a simple framework using deterministic fluid model was proposed, on which utility functions, random losses and ECN (Explicit Congestion Notification) are exploited on the design of end-to-end congestion controllers [KuSr03]. Linearization approach has been adopted to solve the optimization problem and to make the source transmission rates converge to an equilibrium point (i.e., a steady-state level), where the local stability of the primal algorithm can be achieved e.g. [JoTa01, Mass02, Vinn02, LiBa03, WaTa04]. A utility-based controller is designed to realize the global stability of the network under the case of a single source and a single link [DeSr03]. However, these papers have very limited performance evaluation on the transient response of control system, which is an important measure of the control system performance. Moreover, these papers are only providing a fixed solution for the controllers, thus having no flexibility to adapt to great variations in the traffic load level of the Internet, and in turn might result in source sending rate fluctuation and efficiency reduction of link utilization. It is interesting to note that the time-domain methods used by these papers might have contributed to some of the above shortcomings.

1.2 Motivations

Since most of the existing Internet traffic control algorithms are heuristics with no theoretical basis, they cannot always guarantee their system stability. We are therefore very much inclined to use control theories to revisit all the combinations of control algorithms based on the source/router and window/rate-based classifications.

As reviewed before, algorithms with an on-off mechanism have a non-continuous packet drop probability and exhibit oscillatory behavior in queue size and queuing delay. So we would like to propose PI window-based router controller to obtain a continuous packet drop probability in order to clamp the queue size around a reference value.

On the other hand, there is still the source-control problem of the TCP/AIMD mechanism that is now understood to be unsuitable for streaming media transmission. This is further supported by our initial OPNET [OPNE00] simulations over very high-capacity and large-delay links. As observed, the TCP/AIMD source control mechanism cannot avoid the large fluctuations in source transmission rates no matter what kind of window-based router controller is used for the AQM. So we would like to find a controller that can provide a relatively smooth source transmission rate. Our theoretical rate-based method appears to be a good candidate to achieve this.

In addition to providing relatively smooth source transmission rates, rate-based router controllers are also expected to achieve a fair allocation of network bandwidth and a high utilization of network resource. However, this kind of performance improvement comes with a price (more overhead, larger buffer size and more CPU time-consuming) as reviewed before. Therefore, to stabilize the source transmission rates by applying source control instead of router control has been a promising research topic in Internet traffic control. The utility-based source control appears to be an interesting candidate. Since no frequency-domain analysis on such controller has been performed before, we would also like to study the transient network performance of such control.

Since Internet environment is constantly changing, we would like to find a controller that does not depend on a set of fixed parameter values, whether it is for the window-based controller or rate-based controller. In other words, we would like to introduce adaptive control. As reviewed, adaptive control has found limited applications in network traffic control so far. Thus, it would be interesting to see how applicable is adaptive control theory in the Internet congestion control design.

From the control theory viewpoint, all the current rate-based router control themes only consider the SISO model for an IP router. In practice, the inputs and outputs of a router quite often interact with each other, and therefore a MIMO model should be considered. Furthermore, the unused link capacity of some output ports by the high priority traffic can be shared with the incoming traffic from other input ports. For example, most enterprise users normally surf the Internet in the daytime, while most residential users do it in the evening. Thus, different types of users can share their slack bandwidth with each other by using an MIMO IP router. This will make MIMO router important in maximizing the bandwidth utilization for the Internet service providers. Therefore, we would like to propose a network model for a typical MIMO IP router in the Internet, and we would like to design a multi-loop adaptive PI rate-based router controller for AQM in the router. The complexity of the network and its analysis is a challenge, probably being the reason why no related work has been done in this area.

1.3 Objectives

We would like to study controllers that can adapt to network environment changes. Specifically, we would like to use control theory (not heuristics) to

1) design an adaptive P (Proportional) window-based router controller for TCP traffic;
2) design an adaptive PI window-based router controller for TCP traffic;
3) establish a network model for a typical SISO IP router supporting best-effort streaming media traffic, and then develop an adaptive PI rate-based router controller accordingly;
4) establish a network model for a typical MIMO IP router supporting best-effort streaming media traffic, and then develop an adaptive multiloop PI rate-based router controller accordingly;
5) develop an adaptive utility-based source controller for Internet traffic.

1.4 Approaches and Methodologies

In order to implement both the source-based control and the router-based control, we would like to consider window-based router control or rate-based router control.

To explore window-based router control schemes, we shall adopt the linearized TCP/AQM model described in [HoMi01b]. This network model consists of both long-lived and short-lived TCP sources (ftp and http sources) and their respective destinations, as well as AQM routers, and network links for TCP connections.

In our work on rate-based router control schemes, we shall adopt the differential equation approach to describe the dynamic behavior between the controlled sources and the rate-based AQM router. Under the system parameter approximation, we want to obtain a transfer function for the AQM control system with the rate-based router controller.

We will use a proportional controller (K_P) to reduce the rise time and the steady-state error. We would further introduce an integral controller (K_I) because it can eliminate the steady-state error. In practice, considering the stochastic characteristic of the network and the complicated implementation of PID controller, we would use the P or the PI controller in the network traffic control, because with the derivative term applied, small amounts of noise (e.g. the short-lived TCP flows) can cause large amount of change in the output (e.g. great fluctuation of queue size in the Router), which may cause excessive buffer overflow and undesirable flow throughput in the network. Even though a derivative controller (K_D) has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

Instead of the time domain analysis method, we shall select the frequency domain analysis for control system. We would like to employ the following 2 design techniques:

(1) gain and phase margin specifications: we make use of these 2 specifications, because they are important measures to the stability robustness of the control system [Ogat02, HoLe01]. The controller design based on phase margin has comparable performance as the controller design based on gain margin. The combine phase margin and gain margin design has better performance over the sole phase margin design or gain margin design [Ogat02], but it requires more complicated design procedure and real-time implementation.

(2) pole placement technique using damping ratio: we would like to use this technique in our utility-based source controller design based on transient response specification, because the

locations of the closed-loop poles can determine transient characteristics of the control system performance [Ogat02].

Furthermore, we will combine the adaptive control theory with the above two techniques in our network traffic controller design. We shall select a monitoring parameter and specify an interval to monitor the network environment. The adaptive controller will self-tune when a dramatic change of network traffic drifts the monitoring parameter outside its specified interval.

Unlike all the current rate-based router control themes that only consider the SISO model for an IP router, we would explore a MIMO model in order to account for the interaction among the inputs and outputs of a router. We shall investigate how to maximize the link utilization by sharing the unused link capacity of some output ports with the incoming traffic from other input ports. We shall find out how the MIMO IP router may benefit in making maximum use of the limited network bandwidth (especially for wireless networks) in return for its complicated structure. As mentioned earlier, the particular challenge is the analytical shaping of the Gershgorin band [Maci89] instead of manual trial and error done before. Therefore, we would like to extend the DNA method [Rose70] to our multi-loop rate-based router controller design by shaping the Gershgorin band analytically (instead of manually).

We shall use the OPNET [OPNE00] network simulation tool to test the capability of our controllers. OPNET is chosen because it is a very popular commercial tool to provide hierarchical representation for network model. OPNET provides a pseudo-C language for defining network objects, algorithms and messages in an event-driven message-exchanging environment, where protocols are implemented inside finite state machines [ElDe03].

Note that the controllers we shall design are based on the linearization of the TCP window operation mentioned above. However, no such approximation will be used in the simulation of the real TCP operation. In this way, our test results will add to the confidence of the capability of our controllers. We shall create behavioral models of TCP source nodes and run simulations on window-based router control and SISO rate-based router control. We shall examine the performance of our controller such as instantaneous queue length and packet loss rate, and from which we can analyze and compare with other algorithms in terms of measures such as response time and stability limits. Since we are dealing with stochastic processes in our evaluation, and oscillations are expected, we also show the average queue performance curves at some selected sections (e.g., Section 4.3.1 and Appendix E) in order for readers to appreciate our observations that the queues in many cases (such as our designs) do have steady-state mean values. These are in general omitted for the rest of this thesis for clarity reason.

We also use the MATLAB [HaLi04] tool for the implementation of MIMO rate-based router control and utility-based source control. The language of technical computing MATLAB integrates mathematical computing, visualization, and a powerful language to provide a flexible environment for technical computing. The fact that it is using an open architecture, makes MATLAB and its companion products easy to explore data, create algorithms, and create custom tools that provide early insights and competitive advantages.

1.5 Contributions

Instead of using heuristics as done in most of the controller proposed in the literature, we make use of the control theory to establish our controllers that can always guarantee the stability of the Internet control systems, and instead of using time-domain approaches, we employ the frequency domain techniques in the control design. They can be summarized in the following:

1) Adaptive P and PI window-based router controllers for AQM in the IP router to support TCP traffic in the Internet. When compared to other current window-based router control algorithms, our adaptive PI window-based router controller can provide a better AQM performance in the presence of significant changes in the network traffic.
2) An adaptive PI rate-based router controller for SISO IP router supporting streaming media traffic, which can make the controlled source nodes transmit the maximum allowed streaming media traffic so as to catch the spare bandwidth left over by the guaranteed service traffic.
3) An adaptive multi-loop PI rate-based router controller for MIMO IP router supporting streaming media traffic with the same capability as the controller in (2).
4) An adaptive utility-based source controller for Internet traffic, which can provide a relatively smooth source transmission rate based on the aggregate congestion information of the network.

1.6 Thesis Organization

In Chapter 2, we introduce general network model for the Internet, then we propose a general network model for typical IP core router. Next, we present a basic concept for AQM control system. Nyquist diagram is present to introduce phase margin and gain margin and the pole placement method is summarized. Then we extend Kelly's network utility model into a general network model for the Internet with delay. Finally, we introduce our general simulation model, traffic model and some specific assumptions in our control algorithms.

In Chapter 3, fixed P and PI window-based route controllers for AQM in the IP router are proposed. We run simulations to study the performance of these two algorithms. In Chapter 4, adaptive P and PI window-based router controllers for IP router are proposed. Equations for monitoring the change of network environment are derived. We run simulations to make performance comparison with some fixed window-based control algorithms.

In Chapter 5, we present an adaptive PI rate-based router controller for AQM in the SISO IP router to support best-effort streaming media traffic based on phase margin. We use simulation to demonstrate the stability robustness of AQM control system with our adaptive PI rate-based router controller. In Chapter 6, an adaptive multiloop PI rate-based router controller is proposed for AQM in the MIMO IP router to support streaming media traffic in the Internet. The gain margin technique is employed in the controller design.

In Chapter 7, an adaptive utility-based source controller is proposed to provide a relatively smooth source transmission rate based on the feedback network congestion information. Chapter 8 concludes our research work and makes proposals for future works.

1.7 Publication List

A good portion of this thesis work has appeared in the following publications.

Journal Publications:

1. Yang Hong, Oliver W.W. Yang, "Controlling TCP Traffic Using a Self-Tuning AQM Controller With Specification on Gain Margin", Accepted for publication in *IEE Proceedings on Communications*.
2. Yang Hong, O. W.W. Yang, "Design of an Adaptive PI Rate Controller for Streaming Media Traffic Based on Gain and Phase Margins", *IEE Proceedings on Communications*, Volume 153, Issue 1, February 2006, pp. 5-14.
3. Yang Hong, Oliver W.W. Yang, "Adaptive AQM Controllers for IP Routers With a Heuristic Monitor on TCP Flows", *Wiley International Journal of Communication Systems*, Volume 19, Issue 1, February 2006, pp. 17-38.
4. Yang Hong, Oliver W.W. Yang, "Self-Tuning PI Rate Controller for Best-Effort Traffic With Interval Phase Margin Assignment", *Inderscience International Journal of Internet Protocol Technology*, Vol.1, No.2, June 2005, pp.86-93.
5. Yang Hong, Oliver W.W. Yang "An End-to-End PI Congestion Controller for Best-Effort Traffic: Single Source Case", 39(5), *IEE Electronics Letters, March 2003,* pp 480-482.

6 Yang Hong, Oliver W.W. Yang "Design of Adaptive PI Rate Controller for Best-Effort Traffic in the Internet Based on Phase Margin", submitted to *IEEE Transactions on Parallel and Distributed Systems* (With major revision for 2nd round of review).
7 Yang Hong, Oliver W.W. Yang "Self-Tuning PI Controller for AQM Router Supporting TCP Traffic", submitted to *Kluwer Journal of Telecommunication Systems.*
8 Yang Hong, Oliver W.W. Yang, "Adaptive Proportional and PI Controllers for AQM/TCP Router With Interval Phase and Gain Margin Assignment", submitted to *Elsevier Performance Evaluation Journal.*
9 Yang Hong, Oliver W.W. Yang "Adaptive Utility-Based Controller for Internet Traffic", submitted to *IEEE Transactions on Automatic Control.*
10 Yang Hong, Oliver W.W. Yang, "Adaptive Multiloop PI Rate Controller For a MIMO IP Router With Specification on Phase Margin", submitted to *Elsevier Computer Networks Journal.*

Conference Publications:

1 Yang Hong, Oliver W.W. Yang "Adaptive Multiloop PI Rate-based Controller for a MIMO IP Router", *Proceedings of IEEE Global Telecommunications Conference* (Globecom 2005), St. Louis, U.S.A, November 2005, pp. 1070-1074.
2 Yang Hong, Oliver W.W. Yang "Self-Tuning PI Rate Controller for AQM Router Supporting Best-Effort Traffic", *Proceedings of IEEE International Conference on Communication* (ICC 2005), Seoul, Korea, May 2005, pp1539-1543.
3 Yang Hong, Oliver W.W. Yang "Self-Tuning Multiloop PI Rate Controller for an MIMO AQM Router With Interval Gain Margin Assignment", *Proceedings of IEEE workshop on High Performance Switching and Routing* (HPSR 2005), Hong Kong, China, May 2005, pp.401-405.
4 Yang Hong, Oliver W.W. Yang "Design of Utility-Based Congestion Controller for Internet Traffic Based on Pole Placement", *Proceedings of 39th annual conference on Information Sciences and Systems* (CISS 2005), Johns Hopkins University, U.S.A, March 2005.
5 Yang Hong, Oliver W.W. Yang and ChangCheng Huang "Self-Tuning PI TCP Flow Controller for AQM Routers With Interval Gain and Phase Margin Assignment", *Proceedings of IEEE Global Telecommunications Conference* (Globecom 2004), Dallas, U.S.A, November 2004, pp. 1324-1328.
6 Yang Hong, Oliver W.W. Yang "Design of Self-Tuning Proportional TCP Flow Controller for AQM Routers Based on Gain Margin Specification", *Proceedings of the 22nd Biennial Symposium on Communications* (QBSC 2004), Kingston, Canada, June 2004, pp. 280-282.
7 Yang Hong, Oliver W.W. Yang "Design of TCP Traffic Controllers for AQM Routers Based on Phase Margin Specification", *Proceedings of IEEE workshop on High Performance Switching and Routing* (HPSR 2004), Phoenix, U.S.A, April 2004, pp. 314-318.
8 Yang Hong, Oliver W.W. Yang "Self-Tuning PI Controller for AQM Routers With Interval Gain Margin Assignment", *Proceedings of 38th annual conference on Information Sciences and Systems* (CISS 2004), Princeton University, U.S.A, March 2004, pp. 455-460.
9 Yang Hong, Oliver W.W. Yang "Design of Multiloop PI Controller for Active Queue Management of Switching Node in High Speed Network", *Proceedings of 38th annual conference on Information Sciences and Systems* (CISS 2004), Princeton University, U.S.A, March 2004, pp. 461-466.
10 Yang Hong, Oliver W.W. Yang "Self-Tuning Proportional Controller for AQM Routers Supporting TCP Flows", *Proceedings of OPNETWORK 2003*, Washington D.C., U.S.A, August 2003.
11 Yang Hong, Qiang Chen and Oliver W.W. Yang "Design of TCP Traffic Controllers for AQM Routers Based on Gain Margin Specification", *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems* (SPECTS 2003), Montreal, Canada, July 2003, pp 452-457.

12 Yang Hong, Qiang Chen and Oliver W.W. Yang "A PI Controller for AQM Routers Supporting TCP Flows", *Proceedings of IASTED International Conference on Applied Modeling and Simulation* (AMS 2002), MIT, Cambridge, U.S.A, November 2002, pp157-162.
13 Yang Hong, Qiang Chen and Oliver W.W. Yang "A Proportional Controller for AQM Routers Supporting TCP Flows With Simulation Using OPNET Modeler", *Proceedings of OPNETWORK 2002*, Washington D.C., U.S.A, August 2002.
14 Yang Hong, Oliver W.W. Yang "Multiloop PI Controller Design Based on the Direct Nyquist Array Method", *Proceedings of IASTED International Conference on Control and Applications*, Cancun, Mexico, May 2002, pp 161-120.
15 Yang Hong, Oliver W.W. Yang "A PI Controller for the Stabilization of End-to-End Congestion: Single Source Case", *Proceedings of 36th annual conference on Information Sciences and Systems* (CISS 2002), Princeton University, U.S.A., March 2002, pp367-371.

Chapter 2 Modeling and Assumptions

We first describe the common models of the network we are going to investigate. Then we shall review some pertinent control theory in order to summarize the terminology and concepts that will be used in our controller design. The common simulation models for our performance evaluation on our control algorithms are then described. Finally, we state the few common assumptions used in this thesis.

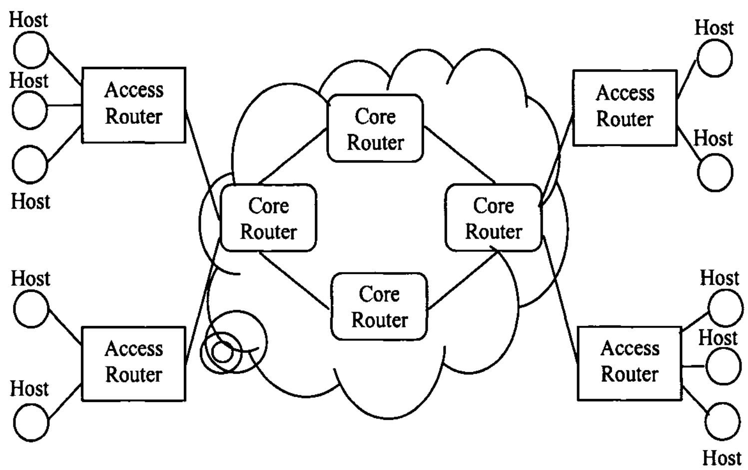

Figure 2.1: General Network Model

2.1 The Network Model

Figure 2.1 illustrates the topology for a general TCP/IP network in our investigation. Users/hosts are attached to the access routers that are interconnected to a backbone network consisting of core routers. The access and core routers cooperate to send data packets from host to host, thus enabling end-to-end communications. This commonly used network model as well as the TCP protocol can be found in many networking textbooks, e.g., [Stal03].

Congestion occurs when many connections traverse and interact adversely to overflow a core router at the same time. Congestion may occur in one or more routers along an end-to-end data path. Therefore, we need to implement our AQM controllers (window-based or rate-based) in the core routers in order to participate in the network traffic control, thus preventing the end-to-end congestion in the Internet.

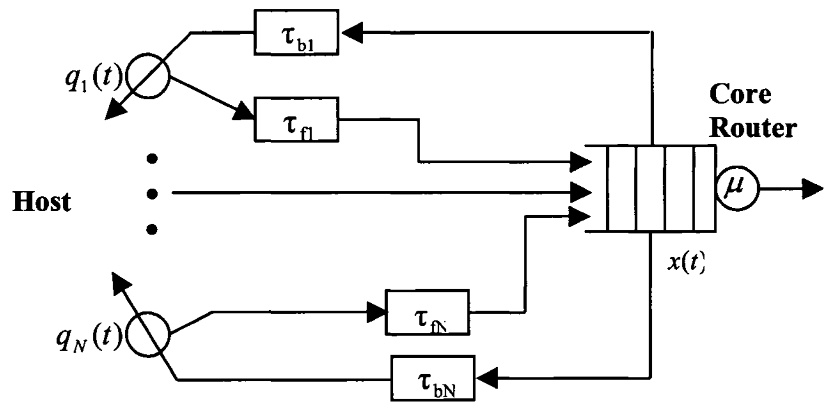

Figure 2.2: Network Model for IP Core Router

2.1.1 Core Router

Figure 2.2 shows the general network model as seen by an IP core router. The source controller is located at a host (i.e. source). The router controller is located at an IP core router for AQM. Based on the instantaneous queue length $x(t)$ of the buffer, the router controller can calculate the packet drop probability or advertised source transmission rate to participate in adjusting the source transmission rate $q_i(t)$, as we discussed in Section 1.1. In Figure 2.2, τ_{fi} and τ_{bi} represent the forward time delay and feedback time delay between the host and the core router, by which we can obtain the round trip time $\tau_i = \tau_{fi} + \tau_{bi}$. μ represents the service rate (i.e. total output link capacity) of the core router. In the linearized TCP/AQM model [HoMi01a], we shall estimate the round trip delay by its mean, that is, $\tau \approx \frac{1}{N}\sum_{i=1}^{N}\tau_i$ and the link capacity μ.

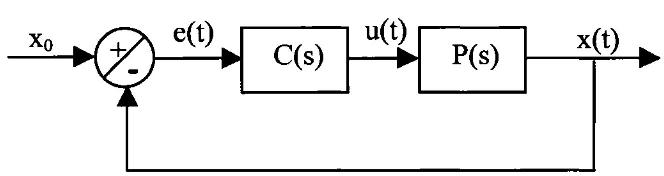

Figure 2.3: Block Diagram of AQM Control System

2.2 The Router-Based Control System

Figure 2.3 shows the block diagram of an AQM Control system. The transfer function *C(s)* represents the router controller for AQM, i.e. window-based router controller (in Chapters 3 and 4), or rate-based router controller (in Chapters 5 and 6) to be discussed later. The transfer function *P(s)* represents an IP router with multiple controlled source nodes (e.g. TCP sources [Jaco88], utility-based controlled sources [KeMa98, HoYa05a]). The output $x(t)$ of the AQM control system represents the queue size of the IP router, while the input x_0 of the AQM control system represents the target buffer occupancy of the IP router. The router controller $C(s)$ will regulate its output $u(t)$ based on the difference between $x(t)$ and x_0 in order to make $x(t)$ track x_0. The control signal $u(t)$ represents the packet drop probability calculated by window-based router controller or the advertised source transmission rate calculated by rate-based router controller.

2.3 The Source-Based Control System

As reviewed in Chapter 1, the original TCP suite is a source-based control system. The source is located in the host. It is an AIMD controller that controls the network load by adjusting the sources' window sizes based on the level of congestion in the network [Jaco88, Stev97]. Each source uses its own congestion window to limit the number of unacknowledged packets. Upon receiving successful acknowledgments, the source increases its congestion window size so as to increase its packet transmission rate. Then it calculates how many packets are still unacknowledged and decides how many new packets should be sent out, according to the new congestion window size. Thus, the total transmission rate from all the sources eventually exceeds the network bandwidth, which will cause queues to build up in the routers. If the source receives duplicate ACK (acknowledgment) packets for a given packet, the source considers that the packet is lost. Then the source decreases its packet transmission rate by reducing its congestion window size. If the source does not receive any ACK packet for a period of time, the source gets into a time-out period by triggering a retransmission timer. The window size reduces to half in earlier size and the lost packet is retransmitted. A summary of TCP window-based source control model is provided in Appendix A.

Other variations of source-based control exists, one of which we study in this research work is the Utility-Based Source Control System described in the following.

2.3.1 Utility-Based Source Control System

Consider a network with a finite set J of AQM routers (i.e. resources), and let μ_j be the finite link capacity (i.e. service rate) of router j that can be shared by a finite set of sources (i.e. users), indexed by r. Let R_r be a route starting from source r and routed through a non-empty set of J, which consists of two paths: (1) the data path through which the source r transmits its data to its destination; (2) the control path through which the destination sends the ACK packets back to source r. The ACK packets contain network congestion information (i.e. route price) so that the utility-based controller can regulate the source transmission rate to prevent the network from congestion. Let $R=\{R_r\}$ be the set of all possible routes. For any route R_r, define a 0-1 routing matrix $A=(A_{jr}, j\in J, R_r\in R)$ such that for any $R_r\in R$, we have

$$A_{jr} = \begin{cases} 1, & \text{if} \quad j \in R_r \\ 0, & \text{otherwise} \end{cases} .$$

A summary of the utility network model without communication delay [KeMa98] is provided in Appendix D and on which we have extended to utility-based control system that has the non-negligible round trip delays. Let τ_{jr}^{f} be the forward delay from the utility-based controlled source r to the AQM router j and τ_{jr}^{b} be the feedback delay which is the sum of delay from the AQM router j to the destination and from the destination back to the source r. Let $\tau_r = \tau_{jr}^{f} + \tau_{jr}^{b}$ be the round trip time between the source r and its destination. Then the optimal utility-based control algorithm in Equation (D.4) has to be rewritten as follows:

$$\dot{q}_r(t) = K_r\left(w_r - q_r(t-\tau_r)\sum_{j\in R_r} p_j\left(\sum_{s:j\in R_s} A_{js} q_s(t-\tau_s)\right)\right) . \tag{2.1}$$

In this thesis, we have adopted the RED and ECN marking strategy to mark the packets by a probability p_j in the router j. We have used the method proposed by [HaFl03] to estimate the round trip time τ_r.

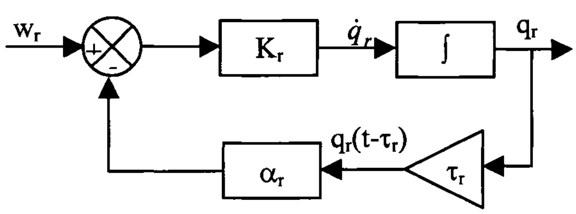

Figure 2.4: Block Diagram of State Feedback Utility-Based Control System

To simplify the controller design, we can let $\alpha_r(t) = \sum_{j\in R_r} p_j\left(\sum_{s:j\in R_s} A_{js} q_s(t-\tau_s)\right)$ so that we can rewrite Equation (2.1) as

$$\dot{q}_r(t) = K_r\left(w_r - q_r(t-\tau_r)\alpha_r(t)\right) . \tag{2.2}$$

We can treat $q_r(t)$ as a state variable and $\alpha_r(t)$ as a varying state feedback gain for the utility-based control system [Ogat02], as shown in Figure 2.4. Therefore we can achieve the good transient network performance by adjusting the dynamics of the transmission rate $q_r(t)$ of the source r. The proportional gain K_r is the focus of our controller design in Chapter 7.

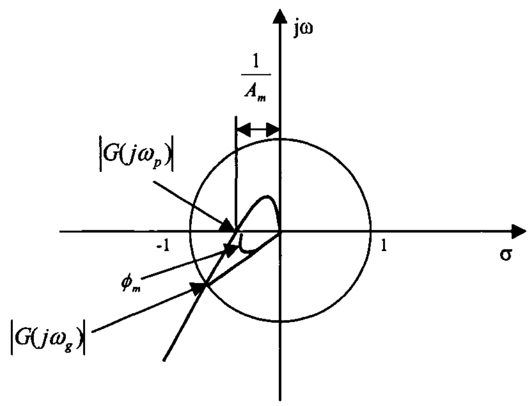

Figure 2.5: Nyquist Diagram of AQM Control System $G(s)$

2.4 Nyquist Diagram with Phase Margin and Gain Margin

Figure 2.5 shows the Nyquist diagram of AQM control system $G(s)$ [Ogat02] and its gain margin A_m defined at the phase crossover frequency ω_p, i.e. where $\angle G(j\omega_p)=-180^0$ when $|G(j\omega_p)|=1/A_m$, while its phase margin ϕ_m defined at the gain crossover frequency ω_g, i.e. where $\angle G(j\omega_g)=-180^0+\phi_m$ when $|G(j\omega_g)|=1$. It is a common practice in modern control that we normally specify the gain margin A_m in the interval of $2\leq A_m\leq 4$ and the phase margin ϕ_m in the interval of $30^0\leq\phi_m\leq 60^0$. This is done in order to get a good response [HoLe01], because too small a gain margin or a phase margin will make the AQM control system sensitive to large load changes and cause instability, while too large a gain margin or a phase margin will result in sluggish response to load changes. Gain margin of more than unit ($A_m>1$) and positive phase margin ($\phi_m>0$) can guarantee the stability of AQM control system, in accordance with Nyquist Stability Theorem [Ogat02]. Because the AQM control system in the IP-based Internet can be treated as a linear system with a delay, positive phase margin ($\phi_m>0$) can always guarantee gain margin of more than unit ($A_m>1$) and vice versa from the control practice [Ogat02]. This has also been verified by our performance evaluation and simulations later on. Therefore, for the rest of this thesis, sometimes we do not mention both.

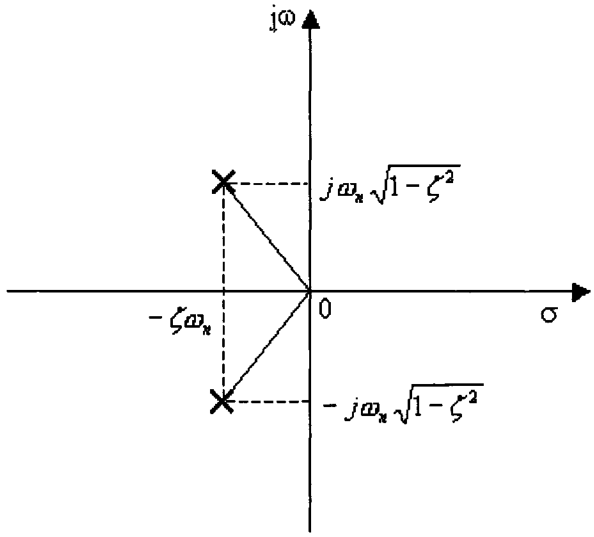

Figure 2.6: Complex Conjugate Poles of Closed-loop Control System $G(s)$

2.5 Control Theoretical Background for Pole Placement Method

We summarize here a second-order network control system with a closed-loop transfer function in order to introduce the control theoretical background and its terminology in Chapter 7. A second-order system in the standard form can be expressed as

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad , \tag{2.3}$$

where ζ and ω_n are the damping ratio ζ and the natural frequency ω_n of the network control system respectively. These parameters can be used to describe the dynamic behavior of the second-order network control system [Ogat02]. In the under-damped case ($0<\zeta<1$), the transient response is oscillatory and the closed-loop poles are a pair of complex conjugates ($s_1, s_2 = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$) in the left-half s plane (with real axis σ and imaginary axis $j\omega$), as shown in Figure 2.6. The network control system is called undamped if $\zeta=0$, and the transient response does not die out. The system is called critically damped if $\zeta=1$, and overdamped system if $\zeta>1$ [Ogat02]. The undamped case is considered to be unstable while the under-damped cases are scenarios we want to alleviate as they represent improper resource utilization as demonstrated later. All the closed-loop poles must be in the left-half s plane in order to guarantee the stability of the network control system [Ogat02], which is a basis of the design rule for our utility-based source controller.

The transient response of a practical control system often exhibits damped oscillations before reaching the steady state. For the second-order closed-loop network control system as expressed in Equation (2.3), the transient specifications, such as maximum percent overshoot and settling time, are determined by ζ and ω_n. For example, the maximum percent overshoot enlarges as the damping ratio ζ reduces under the range of $0<\zeta<1$, while for a fixed value of ζ, the settling time increases as the natural frequency ω_n decreases [Ogat02].

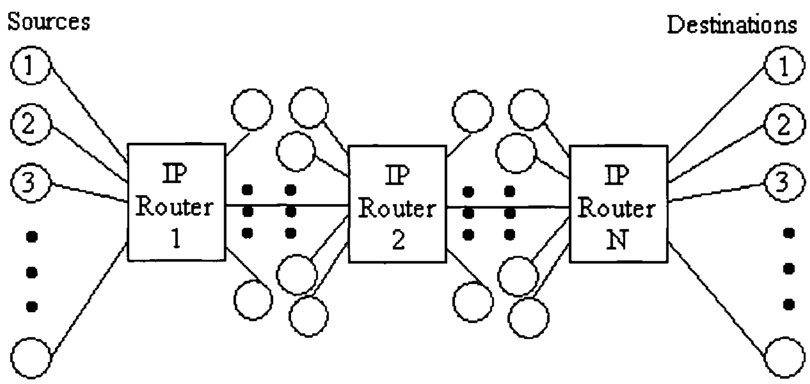

Figure 2.7: Network Simulation Topology for Performance Evaluation

2.6 Simulation Model

We use simulations to verify the effectiveness of our Internet traffic controllers (i.e. window-based router control, rate-based router controller and utility-based source controller). We run our simulation on the network topology shown in Figure 2.7 in all the experiments for all the Internet traffic control algorithms. This simulation model allows us to consider a network system with multiple controlled sources and destinations. The sources send the data packets to their destinations through a series IP routers in the Internet. A router controller is located in the every IP router, and a source controller is located in each host. Both router controller and source controller cooperate for network traffic control to prevent the end-to-end congestion in the Internet.

In the window-based router control scheme, we adopt TCP-Reno sources [MoLa99] that use the Fast-Retransmit and Fast-Recovery mechanisms. We provide greedy ftp (long-lived TCP) flows (i.e. the ftp sources always have data to send as long as their congestion windows allow) and http (short-lived TCP) flows (generated by the pattern suggested by [CrBe97]) as the data packets transmitted by the source in the OPNET environment.

For the rate-based SISO router control scheme, we use streaming media as traffic source to generate data packets in the OPNET environment.

For the rate-based MIMO router control scheme, we use streaming media as traffic source to generate data packets in the Matlab environment.

For the utility-based source control scheme, we do simulations in Matlab environment.

Although N routers are shown in Figure 2.7, we usually observe the behavior of our router control algorithms in the most congested controller in Chapters 3, 4, 5 and 6, while we apply RED and ECN to provide link congestion information of multiple routers in Chapter 7.

2.7 Assumptions

The following assumptions are used for the remainder of this thesis:

(1) Only the queue size in the most congested router is observed and studied although the router control algorithms are implemented in all IP routers in the networks. This is because the transmission rate of a source node is regulated "directly" (by calculating the advertised source window size) or indirectly (by adjusting the packet drop probability) from all the congested routers. Therefore, the most congested router (usually with the largest queue size and therefore resulting in the smallest source transmission rate) can also prevent congestion in all the less congested router.

(2) The link capacity μ, the number N of active controlled sources and the average round trip time τ can be estimated from measurements made locally at the congested router. These estimation methods have been widely adopted in classical control system design e.g., [AsHa95], and recently in network traffic control e.g., [HoMi01b, KaHa02].

(3) The average round trip time τ is dominated by the propagation delay and the queuing delay, a reasonable assumption for the current and future high capacity links.

(4) The TCP/AQM performance can be linearized around an operating point.

(5) The destination's advertised window size is set sufficiently large so that TCP connections are not constrained at the destination.

(6) The "significant" (or "great" or "dramatic") changes of the network traffic means that there is more than *100%* change in the traffic.

Chapter 3 Fixed P and PI Window-Based Router Controllers

In this chapter, we shall provide the design of fixed P (Proportional) and PI (Proportional-Integral) window-based router controllers for AQM in the IP router in the Internet. Classical control theory is applied in the controller design to achieve good AQM performance by specifying proper phase and gain margins.

3.1 Linearized Network Model for TCP/AQM Control System

Consider the AQM control system model shown in Figure 2.3, where the transfer function $C(s)$ is the window-based router controller (e.g., our P and PI window-based router controllers), and the transfer function $P(s) = P'(s)e^{-s\tau}$ is the "plant" of AQM control system [HoMi011a]. The parameter τ is average round trip time, which causes a delay in the feedback of losses. The parameter $x(t)$ indicates the instantaneous queue size in the IP router , while $u(t)$ indicates the instantaneous packet drop probability.

The behavior of TCP/AIMD/AQM is generally non-linear. Its dynamic model has been developed by [MiGo00] and summarized in Appendix A (Similar network model can also be found in [HiOh03]). This has been linearized [HoMi01b] and will be used as a network model for our window-based router controller design. Mathematically, it can be expressed as $P'(s)$ that is a product of $P_{tcp}(s)$ and $P_{queue}(s)$, where

$$P_{tcp}(s) = \left(\frac{\tau\mu^2}{2N^2}\right) \Big/ \left(s + \frac{2N}{\tau^2\mu}\right) \quad ,$$

$$P_{queue}(s) = \left(\frac{N}{\tau}\right) \Big/ \left(s + \frac{1}{\tau}\right) \quad , \tag{3.1}$$

$P_{tcp}(s)$ describes the TCP source window size control mechanism, while $P_{queue}(s)$ describes the queueing dynamical behavior in the routers. We have a constant traffic load factor $N(t)=N$, a constant round trip delay $\tau(t)=\tau$ and a constant link capacity $\mu(t)=\mu$ at the operating point. The number of active long-lived TCP flows N can be estimated using a simple form of flow cache (called the *zombie list*). The zombie list is equivalent to a list of N recently seen flows augmented with a *count* and *timestamp* field [OtWo99]. The link capacity μ can be estimated by the service rate of the local congested router. [ZhHo03]. The round trip time τ can be estimated by the TCP throughput equation $\tau = \frac{\sqrt{2}N}{\mu\sqrt{p_0}}$, while p_0 is the equilibrium packet dropping probability which is also the output of window-based router controller (i.e., AQM controller) [ZhHo03]. Since it is not easy to obtain the equilibrium packet dropping probability p_0 upon the dramatic change of the network environment, we use instead the average value of the previous packet drop probability and current packet drop probability (i.e., $p_a = (p_{previous} + p_{cuurent})/2$) to approximate p_0. Therefore round-trip time can be estimated by $\tau = \frac{\sqrt{2}N}{\mu\sqrt{p_a}}$ in the real-time implementation.

3.2 Fixed-P Window-Based Router Controller Design Based on Phase Margin

In a fixed P control, the feedback signal is simply the regulated output (queue size) multiplied by a gain factor K_P. Thus the transfer function of the P controller is $C(s)=K_P$. To obtain the unknown proportional gain K_P, we only need the value of an extra parameter, which can be either a phase margin or a gain margin. We shall first design a P window-based router controller based on the phase margin in this section.

For the design of the P window-based router controller based on phase margin, we shall specify an interval of the phase margin. Given a nominal phase margin ϕ_m, and from the condition on $\angle G(j\omega_g)$, where $|G(j\omega_g)|=1$ [Ogat02], we have the following equation to obtain ω_g and K_P:

$$-\arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right) - \arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right) - \omega_g\tau = -\pi + \phi_m \tag{3.2}$$

with
$$K_P \frac{\frac{\tau\mu^2}{2N^2}}{\sqrt{\omega_g^2 + (\frac{2N}{\tau^2\mu})^2}} \frac{\frac{N}{\tau}}{\sqrt{\omega_g^2 + (\frac{1}{\tau})^2}} = 1 \tag{3.3}$$

It is not difficult to show that Equation (3.2) holds true only when $0 \le \omega_g\tau < 3\pi/4$. Furthermore, based on the approximation of $\tan(x) \approx x$ when $0 \le x \le \pi/4$, we can simplify Equation (3.2) for 3 different cases:

Case (1): By taking the tangent function in Equation (3.2), expanding the identity for $\tan(x \pm y)$ and simplifying for the range of $\omega_g\tau \le \pi/4$, we obtain

$$\frac{\frac{\omega_g}{\frac{2N}{\tau^2\mu}}+\frac{\omega_g}{\frac{1}{\tau}}}{1-\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\frac{\omega_g}{\frac{1}{\tau}}}=\frac{\tan(\pi-\phi_m)-\tan(\omega_g\tau)}{1+\tan(\pi-\phi_m)\tan(\omega_g\tau)}=\frac{\tan(\pi-\phi_m)-\omega_g\tau}{1+\tan(\pi-\phi_m)\omega_g\tau},$$

$$\omega_g^3-\frac{2\tan(\pi-\phi_m)}{\tau}\left(1+\frac{N}{\tau\mu}\right)\omega_g^2-\left(\frac{1}{\tau^2}+\frac{4N}{\tau^3\mu}\right)\omega_g+\frac{2N}{\tau^4\mu}\tan(\pi-\phi_m)=0\quad . \tag{3.4}$$

Case (2): By applying the same approach as in Case (1) on Equation (3.2) for the range of $\pi/4<\omega_g\tau<\pi/2$, we obtain

$$\begin{aligned}&\omega_g^3-\frac{1}{\tau}\left(\frac{\pi}{4}+2\tan(\frac{3\pi}{4}-\phi_m)\left(1+\frac{N}{\tau\mu}\right)\right)\omega_g^2-\left(\left[1-\frac{\pi}{4}\tan(\frac{3\pi}{4}-\phi_m)\right]\left[\frac{1}{\tau^2}+\frac{2N}{\tau^3\mu}\right]+\frac{2N}{\tau^3\mu}\right)\omega_g\\&+\frac{2N}{\tau^4\mu}\left[\tan(\frac{3\pi}{4}-\phi_m)+\frac{\pi}{4}\right]=0\end{aligned} \tag{3.5}$$

Case (3): By applying the same approach as in Case (1) on Equation (3.2) for the range of $\omega_g\tau\geq\pi/2$, we obtain

$$\begin{aligned}&\omega_g^3-\frac{1}{\tau}\left(\frac{\pi}{2}+2\tan(\frac{\pi}{2}-\phi_m)\left(1+\frac{N}{\tau\mu}\right)\right)\omega_g^2-\left(\left[1-\frac{\pi}{2}\tan(\frac{\pi}{2}-\phi_m)\right]\left[\frac{1}{\tau^2}+\frac{2N}{\tau^3\mu}\right]+\frac{2N}{\tau^3\mu}\right)\omega_g\\&+\frac{2N}{\tau^4\mu}\left[\tan(\frac{\pi}{2}-\phi_m)+\frac{\pi}{2}\right]=0\end{aligned} \tag{3.6}$$

It is a common practice in the modern control that we normally specify $30^0\leq\phi_m\leq60^0$ for a good response, as shown in Figure 3.2. Within this range of ϕ_m, one can show that the smallest positive real root of Equations (3.4) to (3.6) on the three cases will be used as the default value of bandwidth ω_g of the control system.

This provides us a design procedure for the P window-based router controller as follows:

Step 1: Specify the phase margin ϕ_m.

Step 2: Estimate the network parameters τ, μ and N using the method presented in Section 3.1.

Step 3: Obtain ω_g using Equation (3.4).

Step 4: Calculate the value of $\omega_g\tau$.

(4a) If we find that $\omega_g\tau\leq\pi/4$, then go to step 7.

(4b) If we find that $\pi/4<\omega_g\tau<\pi/2$, then go to step 6.

Step 5: Recalculate ω_g by Equation (3.6), then go to step 7.

Step 6: Recalculate ω_g by Equation (3.5).

Step 7: Obtain the proportional gain K_P using Equation (3.3).

For further performance analysis and evaluation, we consider a network system with TCP-Reno sources that use the Fast-Retransmit and Fast-Recovery mechanisms. We use $N=100$ greedy ftp sources and *20* http sources in the beginning of all the following experiments on window-based router control algorithms. The propagation delays of between TCP sources and TCP destinations will be uniformly distributed between *0.05* sec and *0.13* sec with the mean of *0.09* sec, i.e., the average end-to-end propagation delay is *0.09* second. Considering the queuing delay, the upper limit of round trip time (RTT) is set as $\tau=0.25$ second. The link bandwidth is T_3 *(44,736,000*bps), with the average packet size being 536 bytes. It means that the link capacity is $\mu=44736000/536/8=10433$packets/second.

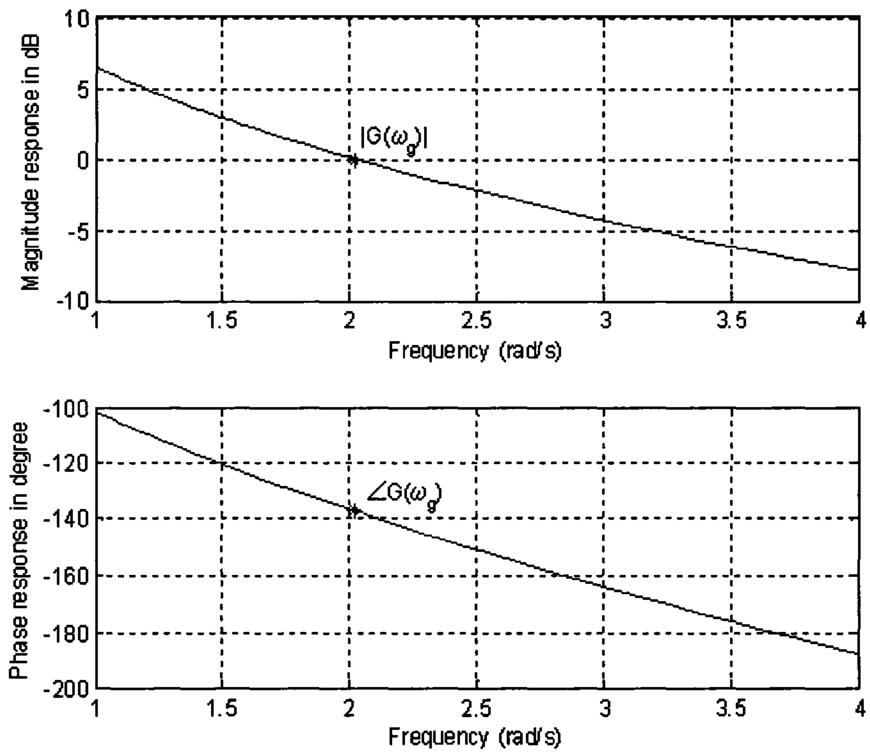

Figure 3.1: Frequency Response for AQM Control System with P Window-Based Router Controller (Phase Margin = 45°)

<u>Example 3.1</u>:

Using the network system described above, we specify the phase margin ϕ_m as 45°. From the given network parameters N, τ and μ, we obtain $\omega_g\approx 2.022$rad/sec from Equation (3.4) and find that $\omega_g \tau \leq \pi/4$. Then from Equation (3.3), we obtain our fixed P window-based router controller $C(s)=K_P=1.684\times10^{-5}$.

Figure 3.1 is the Bode plot for the frequency response of the AQM control system in the frequency range of $1\leq\omega\leq4$. Both the magnitude $|G(s)|$ and phase $\angle G(s)$ are decreasing function of ω as expected. It is shown that the phase margin of AQM control system with the fixed P window-based router controller has been pegged at $\phi_m=180^0+\angle G(j\omega_g)\approx45^0$ as desired.

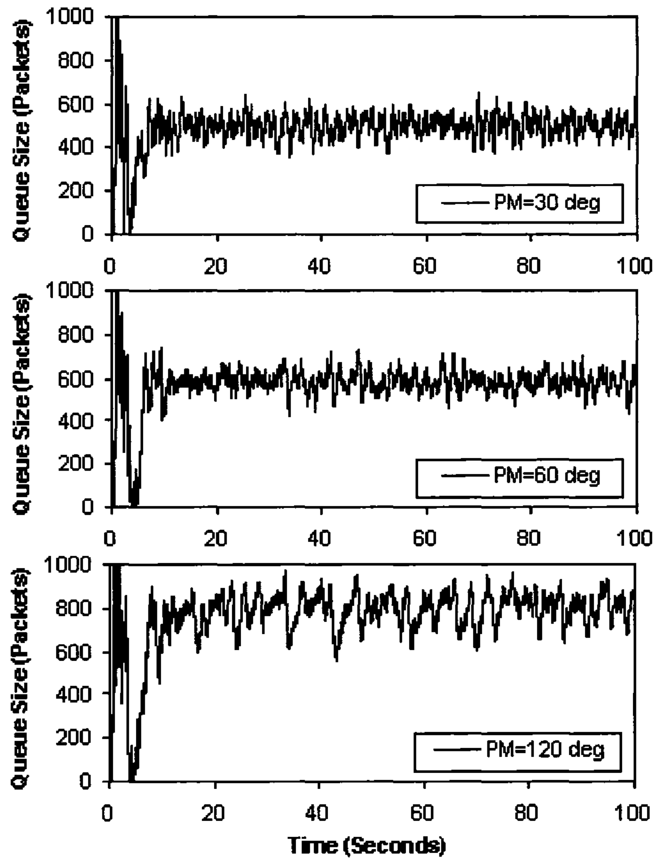

Figure 3.2: Instantaneous Queue Size Comparison of Fixed P Window-Based Router Controller with Different Phase Margins ($PM=30°, 60°, 120°$)

We verify the P window-based router controller via simulations using OPNET Modeler [OPNE00] based on the network simulation topology shown in Figure 2.7 with $N=2$ IP routers. We simulated the case of long-lived sources (i.e., greedy ftp sources), which always have IP packets to send as long as their congestion windows allow; and the short-lived sources (i.e., http sources), which enter the network after an undetermined think time, send a file of an undetermined length, and then wait for another think time period. The destination's advertised window size is set sufficiently large so that TCP connections are not constrained at the destination. The buffer size is *1000* packets based on the different bandwidth-delay product [HoMi01a]. The target buffer occupancy is specified as 400 packets. In all the graphs shown subsequently in this chapter, we depict the time evolution of the instantaneous queue size, with the time axis drawn in seconds. The notation *PM* means *phase margin*.

Figure 3.2 shows the instantaneous queue size of our fixed P window-based router controller with different phase margins. As phase margin is increasing from $\phi_m=30°$ to $\phi_m=120°$, we can see that the fluctuation of queue size increases. Also evident from Figure 3.2, a larger phase margin results in a slower settling time and a larger average queue size. Thus too large phase margin may cause buffer overflow due to the limitation of the buffer size. Note that the instantaneous queue size in this figure (and the only figure in the thesis) is based on the ensemble (statistical) average of several simulation runs as explained in Appendix E.

3.3 Fixed P Window-Based Router Controller Design Based on Gain Margin

We have also designed the P window-based router controller based on the gain margin. This is a design parallel to Section 3.2. Our study shows that it has comparable performance with phase margin design. It is omitted for clarity purpose, but the details can be found in [HoCh03].

3.4 Fixed PI Window-Based Router Controller Design Based on Phase Margin

From the experiment in Section 3.2, the P window-based router controller has a steady state error that is dependent on the network parameters, and such error can sometimes be larger than the buffer size, which might cause instability. Therefore the PI controller design is proposed to clamp the queue size to a specified reference value q_{ref}, which is independent of the load level.

Our fixed PI window-based router controller has a transfer function $C(s) = K_P + \frac{K_I}{s}$, where K_P is the proportional gain and K_I is integral gain. For the controller design based on phase margin, we shall specify a phase margin as ϕ_m. From the condition on $\angle G(j\omega_g)$ where $|G(j\omega_g)|=1$ [Ogat02], we have

$$-\frac{\pi}{2} + \arctan\left[\frac{K_P\omega_g}{K_I}\right] - \arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right) - \arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right) - \omega_g\tau = -\pi + \phi_m \quad , \tag{3.7}$$

with $$\frac{\sqrt{K_P^2\omega_g^2+K_I^2}}{\omega_g}\frac{\frac{\tau\mu^2}{2N^2}}{\sqrt{\omega_g^2+(\frac{2N}{\tau^2\mu})^2}}\frac{\frac{N}{\tau}}{\sqrt{\omega_g^2+(\frac{1}{\tau})^2}}=1 \quad . \tag{3.8}$$

To obtain two unknown parameters (the proportional gain K_P and the integral gain K_I) based on a known parameter (phase margin ϕ_m), we need to set up a proper relationship between K_P and K_I in order to simplify the controller design. We can set $K_P=K_I/\omega_g$ to make $\arctan(K_P\omega_g/K_I)=\arctan(1)=\pi/4$, as suggested by [HoMi01a]. Therefore we can simplify Equations (3.7) and (3.8) as

$$-\frac{\pi}{4}-\arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right)-\omega_g\tau=-\pi+\phi_m \quad , \tag{3.9}$$

with $$\frac{\sqrt{2}K_I}{\omega_g}\frac{\frac{\tau\mu^2}{2N^2}}{\sqrt{\omega_g^2+(\frac{2N}{\tau^2\mu})^2}}\frac{\frac{N}{\tau}}{\sqrt{\omega_g^2+(\frac{1}{\tau})^2}}=1 \quad . \tag{3.10}$$

It is not difficult to show that Equation (3.9) holds true only when $0\le\omega_g\tau<\pi/2$. Furthermore, based on the approximation of $\tan(x)\approx x$ when $0\le x\le\pi/4$, we can simplify Equation (3.9) in two different cases.

Case (1) $\omega_g\tau\le\pi/4$:

$$\omega_g^3-\frac{2\tan(3\pi/4-\phi_m)}{\tau}\left(1+\frac{N}{\tau\mu}\right)\omega_g^2-\left(\frac{1}{\tau^2}+\frac{4N}{\tau^3\mu}\right)\omega_g+\frac{2N}{\tau^4\mu}\tan(\frac{3\pi}{4}-\phi_m)=0 \tag{3.11}$$

Case (2) $\pi/4<\omega_g\tau<\pi/2$:

$$\omega_g^3-\frac{1}{\tau}\left(\frac{\pi}{4}+2\tan(\frac{\pi}{2}-\phi_m)\left(1+\frac{N}{\tau\mu}\right)\right)\omega_g^2-\left(\left[1-\frac{\pi}{4}\tan(\frac{\pi}{2}-\phi_m)\right]\left[\frac{1}{\tau^2}+\frac{2N}{\tau^3\mu}\right]+\frac{2N}{\tau^3\mu}\right)\omega_g$$
$$+\frac{2N}{\tau^4\mu}\left[\tan(\frac{\pi}{2}-\phi_m)+\frac{\pi}{4}\right]=0 \tag{3.12}$$

The smallest positive real root of Equations (3.11) or (3.12) on the two cases will be used as the default value of bandwidth ω_g of the control system. This provides us a design procedure as follows from known system parameters.

Step 1: Obtain ω_g using Equation (3.11).

Step 2: Calculate the value of $\omega_g\tau$.

(2a) If we find that $\omega_g \tau \leq \pi/4$, then go to step 4.

Step 3: Recalculate ω_g by Equation (3.12).

Step 4: Obtain the integral gain K_I using Equation (3.10).

Example 3.2:

Consider the same network example as in Section 3.2. We specify a phase margin as $\phi_m=45^0$. Then the network parameters N, τ and μ are estimated, from which we obtain $\omega_g \approx 0.769$rad/sec by Equation (3.11). Since we have $\omega_g \tau \leq \pi/4$, we obtain $K_P=4.38\times 10^{-6}$ and $K_I=3.366\times 10^{-6}$ for our PI window-based router controller by Equation (3.10).

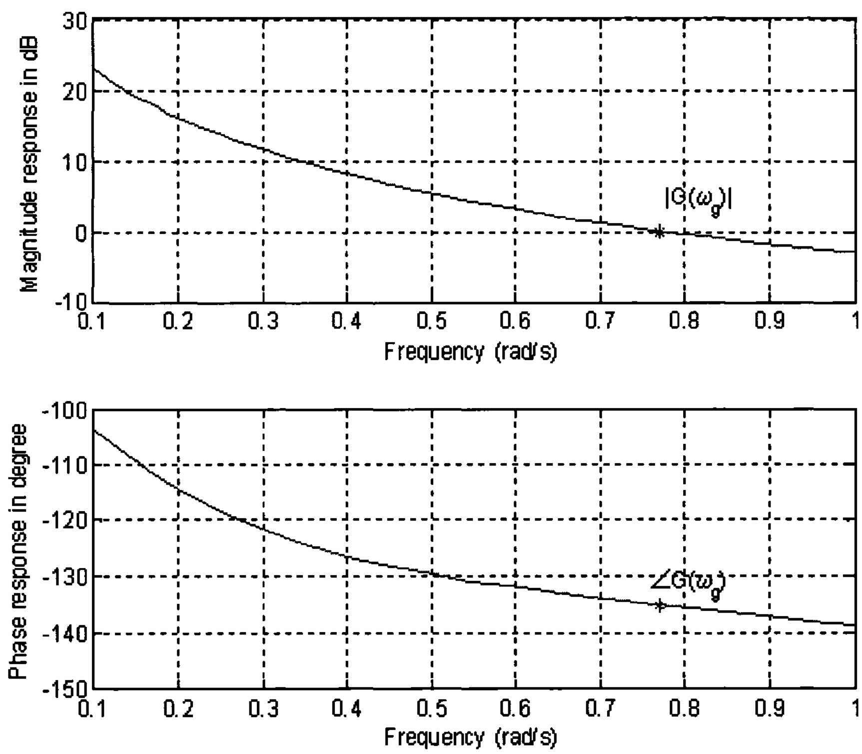

Figure 3.3: Frequency Response for AQM Control System with PI Window-Based Router Controller (Phase Margin = 45^0)

Figure 3.3 is Bode plot for the frequency response of the AQM control system in the frequency range of $0.1 \leq \omega \leq 1$. Both the magnitude $|G(s)|$ and the phase $\angle G(s)$ are decreasing function of ω as expected. It is also demonstrated that the phase margin of our AQM control system with a fixed PI window-based router controller can be pegged at $\phi_m=180^0+\angle G(j\omega_g)\approx 45^0$ as desired.

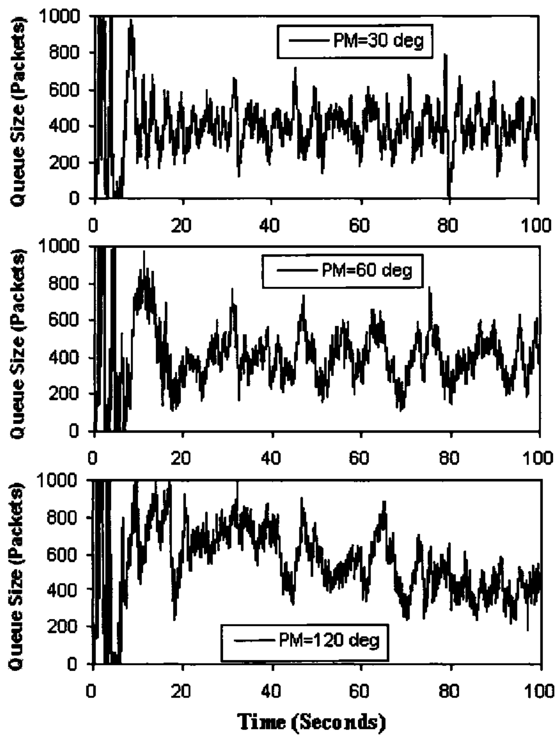

Figure 3.4: Instantaneous Queue Size Comparison of Fixed PI Window-Based Router Controller with Different Phase Margins (*PM=30°, 60°, 120°*)

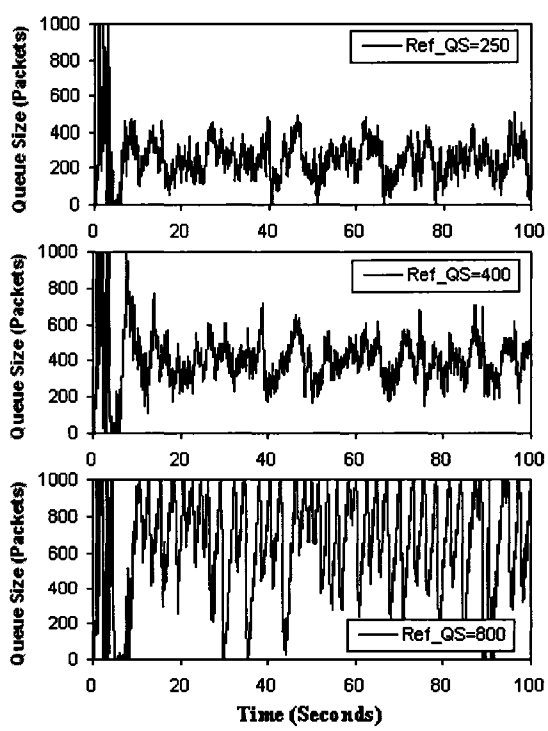

Figure 3.5: Instantaneous Queue Size Comparison of Fixed PI Window-based Router Controller with Different Reference Queue Size (*REF_QS=250, 400, 600*)

We have verified our fixed PI window-based router controller via simulations using OPNET Modeler [OPNE00]. The simulation scenario in the Section 3.2 is reused. The reference queue size is specified as *400*packets. Note the notation *REF_QS* means *reference queue size* (target buffer occupancy).

Figure 3.4 shows the instantaneous queue size of PI window-based router controller with different phase margins. As is evident from Figure 3.4, when the phase margin increases, the settling time increases and the fluctuations of queue size turns larger. If the phase margin is too large (e.g., 120°), the settling time for the queue size to reach a mean value of *400* packets (reference queue size) is extremely long when compared to those with smaller phase margins.

Figure 3.5 gives the instantaneous queue size comparison using different reference queue sizes in the PI window-based router controller. When the reference queue size increases, the settling time becomes longer. If reference queue size is too large, due to the limitation of the buffer size, the instantaneous queue size may have a large fluctuation and cause some buffer overflow as shown in Figure 3.5.

3.5 Fixed PI Window-Based Router Controller Design Based on Gain Margin

We have also designed the PI window-based router controller based on the gain margin. This is a design parallel to Section 3.4. Our study shows that it has comparable performance with phase margin design. It is omitted here for clarity purpose, but the details can be found in [HoCh03]. In fact, our finding and experience also motivate us to examine the combined design of phase margin and gain margin of the PI window-based router controller in the next section.

3.6 Fixed PI Window-Based Router Controller Design Based on Both Phase and Gain Margins

As we discussed in Section 3.4, the transfer function of PI controller is given by $C(s) = K_P + \frac{K_I}{s}$, where K_P is the proportional gain and K_I is integral gain. To obtain the two unknown parameters – K_P and K_I, we will use two equations derived based on the definition of phase margin ϕ_m and gain margin A_m.

Let ω_p be the phase crossover frequency and ω_g be the gain crossover frequency. Then from the definition of the gain margin and phase margin [Ogat02], we can obtain

$$P(j\omega_p)\left[K_P - j\frac{K_I}{\omega_p}\right] = -\frac{1}{A_m} \quad , \tag{3.13}$$

$$P(j\omega_g)\left[K_P - j\frac{K_I}{\omega_g}\right] = -e^{j\phi_m} \quad . \tag{3.14}$$

where $P(s) = P'(s)e^{-s\tau}$ represents the "plant" of AQM control system in Figure 2.3.

It is noted that there are four unknowns altogether, namely, K_P, K_I, ω_p, ω_g in Equation (3.13) and (3.14). Observing that both equations are complex, they can be broken down into four real equations. Since the number of unknowns equals the number of real equations, we can obtain a unique solution. We adopt the method proposed by [FuWa98] to obtain the parameters of PI controller $C(s) = K_P + \frac{K_I}{s}$. Splitting Equations (3.13) and (3.14) into their respective real and imaginary parts yields

$$K_P = \mathrm{Re}\left[\frac{-1}{A_m P(j\omega_p)}\right] = \mathrm{Re}\left[\frac{-e^{j\phi_m}}{P(j\omega_g)}\right] \quad , \tag{3.15}$$

$$K_I = \omega_p \,\mathrm{Im}\left[\frac{1}{A_m P(j\omega_p)}\right] = \omega_g \,\mathrm{Im}\left[\frac{e^{j\phi_m}}{P(j\omega_g)}\right] \quad . \tag{3.16}$$

Since the frequency points ω_p and ω_g are unknown, we define the following two complex functions:

$$f_P(\omega) = \mathrm{Re}\left[\frac{-1}{A_m P(j\omega)}\right] + j\omega \,\mathrm{Im}\left[\frac{1}{A_m P(j\omega)}\right] \quad -\frac{\pi}{2} < \angle P(j\omega) < -\pi \quad , \tag{3.17}$$

$$f_I(\omega) = \mathrm{Re}\left[\frac{-e^{j\phi_m}}{P(j\omega)}\right] + j\omega \,\mathrm{Im}\left[\frac{e^{j\phi_m}}{P(j\omega)}\right] \quad -\frac{\pi}{2} + \phi_m < \angle P(j\omega) < -\pi + \phi_m \quad . \tag{3.18}$$

The functions are plotted in the same complex plane. An intersection of these two functions means that they have the same real and imaginary parts, and thus both equations (3.15) and (3.16) are satisfied. The intersection is therefore a solution to equations (3.13) and (3.14) that provides a set of values for K_P and K_I, and the corresponding ω_p and ω_g, and no solution exists if they do not intersect. The solution will exist if we choose the proper phase margin and gain margin within the range of $30^0 \le \phi_m \le 60^0$ and $2 \le A_m \le 4$.

This provides us a design procedure for the fixed PI window-based router controller based on phase margin and gain margin as follows:

Step 1: Specify a phase margin ϕ_m and a gain margin A_m.

Step 2: Estimate network parameters N, τ and μ using the method presented in Section 3.1.

Step 3: Obtain the proportional gain K_P and integral gain K_I for fixed PI controller using Equations (3.15), (3.16), (3.17) and (3.18).

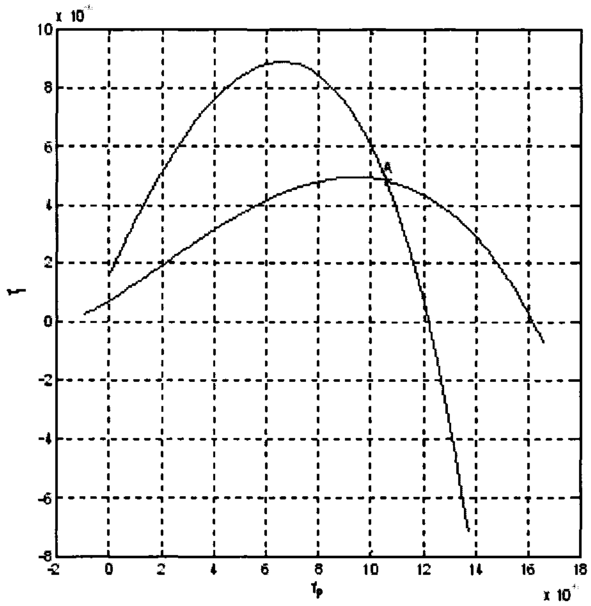

Figure 3.6: The Graph of f_P and f_I

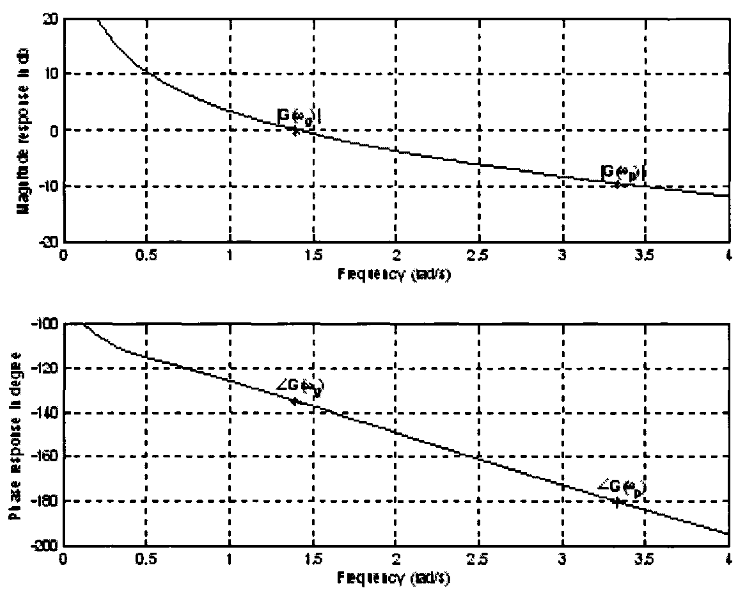

Figure 3.7: Frequency Response for AQM Control System with Fixed PI Window-Based Router Controller (Gain Margin = *3* and Phase Margin = 45^0)

Example 3.3:

Using the same network system described in Section 3.2, we specify a phase margin $\phi_m = 45^{\circ}$ and a gain margin $A_m = 3$ (within the range of $30^0 \leq \phi_m \leq 60^0$ and $2 \leq A_m \leq 4$). Next, we estimate the network parameters N, τ and μ. Then from the equations (3.15) to (3.18), we can draw a plot, as shown in Figure 3.6. From the Figure 3.6 (Point A), we can obtain the PI controller $C(s) = 1.053\times10^{-5} + \frac{4.847\times10^{-6}}{s}$.

Figure 3.7 is the Bode plot for the frequency response of the AQM control system in the frequency range of $0.1 \le \omega \le 4$. Both the magnitude $|G(s)|$ and the phase $\angle G(s)$ are decreasing function of ω as expected. It is shown that the nominal gain margin and phase margin of AQM control system with the fixed PI window-based router controller have been pegged at $A_m=3$ and $\phi_m=45^\circ$ as desired.

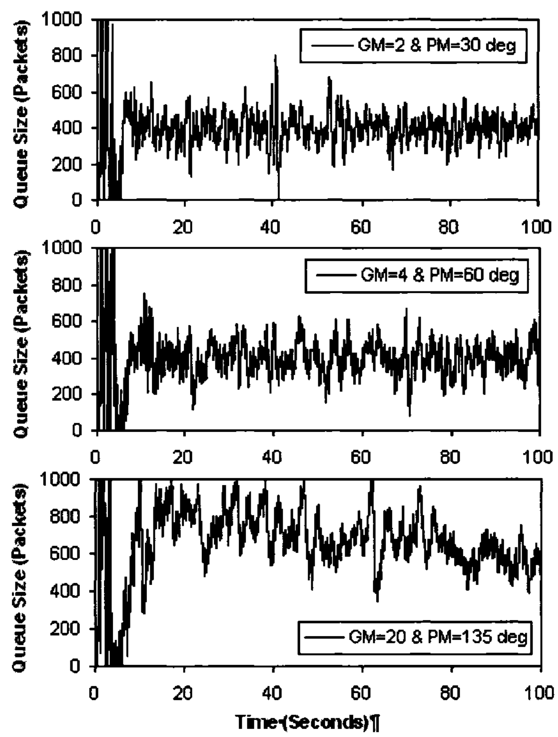

Figure 3.8: Instantaneous Queue Size Comparison Among Different Gain and Phase Margins of Fixed PI Window-Based Router Controller (REF_QS=*400*)

We have verified our fixed PI window-based router controller via simulations using OPNET Modeler [OPNE00]. The simulation scenario in Section 3.2 is reused. The reference queue size (target buffer occupancy) is *400* packets. We set the gain margin to *3* and the phase margin to 45°. Note the notation *GM* means *gain margin.*

Figure 3.8 shows the instantaneous queue size of fixed PI window-based router controller with different gain and phase margins. When the gain margin (and therefore phase margin) increases, the settling time increases and the fluctuations of queue size becomes larger. If both the gain and the phase margins are too large (e.g., $GM=20, PM=135°$), the settling time becomes so long that the queue size reaches the buffer limit before settling down to the target buffer occupancy slowly.

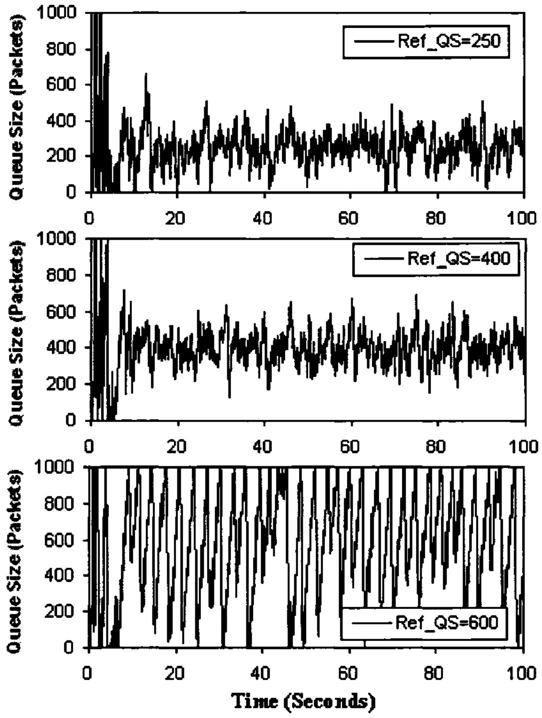

Figure 3.9: Instantaneous Queue Size Comparison Among Different Reference Queue Size of Fixed PI Window-Based Router Controllers (REF_QS=*250*, *400*, *600*, *800*)

Figure 3.9 gives the comparison of our fixed PI window-based router controller with different reference queue sizes. When the reference queue size increases, the settling time becomes longer. If the reference queue size is too large, the instantaneous queue size may become unstable and cause some buffer overflow due to the finite buffer size.

3.7 Stability Robustness of PI Window-Based Router Controller

To demonstrate the stability robustness of our PI window-based router controller designed based on the phase margin and the gain margin, we want to observe the AQM performance upon changes in the network environment during the simulations. However, in order to guarantee the stability of the AQM control system, we shall restrict the range of traffic load changes in the network during the simulations. According to Nyquist stability theorem [Ogat02], the gain margin should be more than unit ($A_m>1$) and the phase margin should be positive ($\phi_m>0$). We have done simulations upon the changes of active TCP flows and round-trip time respectively. In the following simulations, the link bandwidth is also T_3 with a link capacity of $\mu=10433$ packets/second. The reference queue size is also *400* packets. We set the gain margin to be *3* and the nominal phase margin to be 45^0.

Figure 3.10 shows the instantaneous queue size of our fixed PI window-based router controller upon varying active TCP flows. The average round trip time is around *250*ms. At time $t=0$, $N=100$ greedy ftp sources and *20* http sources are used. Following the controller design procedure in Section 3.6, the fixed PI window-based router controller is given as $C(s)=K_P+\frac{K_I}{s}=1.053\times10^{-5}+\frac{4.847\times10^{-6}}{s}$. Simulation results show that the queue size starts to approach the target queue length at time $t=20$s. At time $t=100$s, the number of active TCP flows decreases, (that is, only $N=50$ greedy ftp sources and *10* http sources are used), and the queue size decreases till the buffer is almost empty. Then after the time $t=105$s, the queue size starts to reach the target queue length (*400* packets). At the time $t=200$s, more TCP flows are added, that is, $N=200$ greedy ftp sources and *40* http sources are used. The queue size increases till nearly buffer overflow, then decreases and approaches the target queue length (*400* packets).

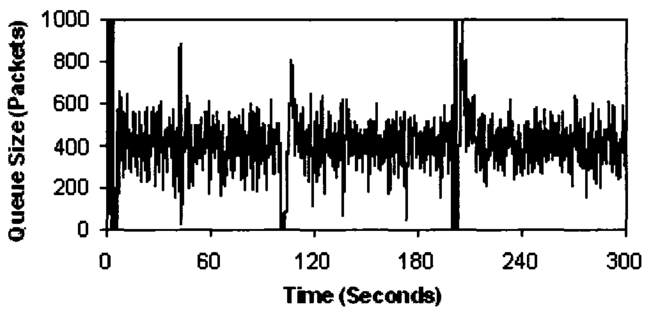

Figure 3.10: Instantaneous Queue Size of Fixed PI Window-Based Router Controller upon Varying Active TCP Flows (*N=100*, *50* and *200*)

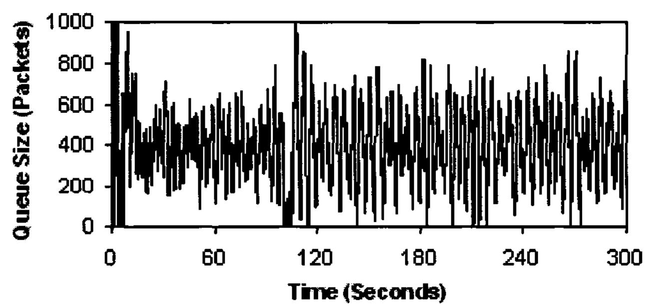

Figure 3.11: Instantaneous Queue Size of "Fixed" Adaptive PI Controller upon Varying Round Trip Time (RTT=0.5s, 0.25s and 0.75s)

Figure 3.11 shows the instantaneous queue size of fixed PI window-based router controller upon varying the round trip time. During the simulation, $N=400$ greedy ftp sources and *80* http sources are used. At time $t=0$s, the average round trip time is around *500*ms. Following the controller design procedure, we obtain the fixed PI window-based router controller $C(s) = K_P + \frac{K_I}{s} = 1.045\times10^{-5} + \frac{4.079\times10^{-6}}{s}$. It can be seen that the queue size starts to approach the target queue length at the time $t=60$s. Between the time $t=100$s and $t=200$s, the average round trip time decreases to about *250*ms. The queue size decreases to nearly empty at time $t=105$s, then increases and starts oscillation around target buffer occupancy (*400* packets). At the time $t=200$s, average round trip time increases to around 750ms, the queue size continues oscillation along the target queue length (*400* packets).

In summary, our fixed PI window-based router controller designed based on gain and phase margins can provide AQM control system with good stability robustness. However, our fixed PI controller cannot adapt very well to the great changes in network environment. This has motivated us to design an adaptive PI window-based router controller, instead of the fixed PI window-based router controller, to guarantee the stability of the network in the next chapter, with the purpose of avoiding excessive buffer overflow and preventing the Internet from congestion.

3.8 Concluding Remarks

We have applied classical control theory to develop the fixed P and PI window-based router controllers for AQM routers in the Internet. Phase and gain margins are used to measure the stability robustness of the AQM control system. We can achieve good performance of AQM by assigning a proper phase margin and/or gain margin. Equations for designing P and PI window-based router controllers have been derived. Based on the knowledge of the queue size in the router, the PI window-based router controller can regulate the TCP source window size by adjusting packet drop probability in order to clamp the steady value of queue size around the specified target buffer occupancy.

As a final remark, we reiterate that our controllers are developed based on control theory which, as demonstrated, can always guarantee the stability of our AQM control system.

Chapter 4 Adaptive P and PI Window-Based Router Controllers

In this chapter, instead of specifying a fixed value of phase margin and gain margin, we will design the adaptive P (Proportional) and PI (Proportional-Integral) window-based router controllers by assigning a proper interval of phase margin and gain margin. We propose equations for real-time monitoring the change of the network environment. These adaptive P and PI window-based router controllers will self-tune only when big changes in the network environment drift the phase margin or the gain margin outside its specified interval. In this way, adaptive P and PI controller need not react to any small traffic load change.

4.1 Adaptive P Window-Based Router Controller Design Based on Phase Margin

To design an adaptive P window-based router controller based on phase margin, we need to specify a phase margin interval (ϕ_1, ϕ_2). From such interval, we determine a nominal phase margin ϕ_m (e.g., $\phi_m=(\phi_1+\phi_2)/2$), then we can use Equations (3.2) and (3.3) to obtain (or design) adaptive P window-based router controller and develop the monitoring equations (4.1) and (4.2) below.

$$\omega_g^4+\left(\left((\frac{2N}{\tau^2\mu})^2\right)+\left((\frac{1}{\tau})^2\right)\right)\omega_g^2+\left(\frac{2N}{\tau^3\mu}\right)^2-\left(\frac{K_p\mu^2}{2N}\right)^2=0 \quad , \tag{4.1}$$

$$\phi_m=\pi-\arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right)-\omega_g\tau \quad , \tag{4.2}$$

where the smallest positive real root of Equation (4.1) will be the default value of the bandwidth ω_g to calculate the phase margin of the AQM control system.

Should the parameters N, τ and μ of the network change so significantly that the system phase margin falls outside the interval [ϕ_1, ϕ_2], the adaptive P window-based router controller will self-tune and make the phase margin return to its nominal value ϕ_m. To achieve this, we use Equations (4.1) and (4.2) to monitor the phase margin of the AQM control system online. We also choose $\phi_1=30^0$ and $\phi_2=60^0$ as discussed in Section 2.5 because numerous simulations (as done later on) confirm the suitability of this range. With these choices, we are able to provide a design procedure for the adaptive P controller as follows:

Step 1: Specify the interval of phase margin as $30^0 \leq \phi_m \leq 60^0$ and nominal phase margin $\phi_m = 45^0$.

Step 2: Estimate network parameters N, τ and μ using the method presented in Section 3.1.

Step 3: Obtain ω_g using Equation (3.4).

Step 4: Calculate the value of $\omega_g \tau$.

(4a) If we find that $\omega_g \tau \leq \pi/4$, then go to step 7.

(4b) If we find that $\pi/4 < \omega_g \tau < \pi/2$, then go to step 6.

Step 5: Recalculate ω_g by Equation (3.6), then go to step 7.

Step 6: Recalculate ω_g by Equation (3.5).

Step 7: Obtain the proportional gain K_P using Equation (3.3).

Real-time monitoring:

Step 8: Estimate the current network parameters N, τ and μ using the method presented in Section 3.1, then compute the phase margin using Equations (4.1) and (4.2).

Step 9: (9a) If the phase margin falls outside the specified interval, go to Step 3.

(9b) If the phase margin is still inside the interval, P controller remains unchanged, go to Step 8.

4.1.1 Performance Comparison of Adaptive and Fixed P Window-Based Router Controller

We have verified our adaptive P window-based router controller via simulations using OPNET Modeler [OPNE00]. The simulation scenario in Section 3.2 is reused. In order to demonstrate the importance of adaptive control in the network traffic control, we have run the RED algorithm [FlJa93] and the fixed P window-based router controller in [HoMi01a] for comparison. To allow further comparison, we use similar RED parameters from [HoMi01a] in this thesis. That is, maximum drop probability $p_{\max}$ is 0.1; the maximum queue threshold max_{th} is 700; the minimum queue threshold min_{th} is 150; the queue weight w_q is 1.33×10^{-6}.

Figure 4.1 compares the instantaneous queue size of RED, the fixed P window-based router controller in [HoMi01a] and our adaptive P window-based router controller with a nominal margin = 45°. At time $t=0$s, $N=100$ greedy ftp sources and 20 http sources are used, it is obvious that the adaptive P window-based router controller shows a better response than RED. Its settling time is much shorter than RED. Also the response time of the adaptive P window-based router controller is shorter than the fixed P window-based router controller. Furthermore our method has less oscillation. At time *t=100*s, another *300* greedy ftp sources and *60* http sources are added. The instantaneous queue sizes of both RED and fixed P window-based router controller exhibit sluggish response. Moreover, the queue size of the fixed P window-based router controller has increased to more than *800* packets and some buffer overflow has occurred. Since the phase margin of the AQM control system falls outside the interval, our adaptive P window-based router controller self-tunes and gives a new $\phi_m=45°$. It can be seen that the instantaneous queue size of our adaptive P window-based router controller then reaches a new steady state value very quickly and avoids excessive buffer overflow. However, both the fixed P window-based router controller and the adaptive P window-based router controller cannot make the instantaneous queue size approach the target buffer occupancy (*400* packets).

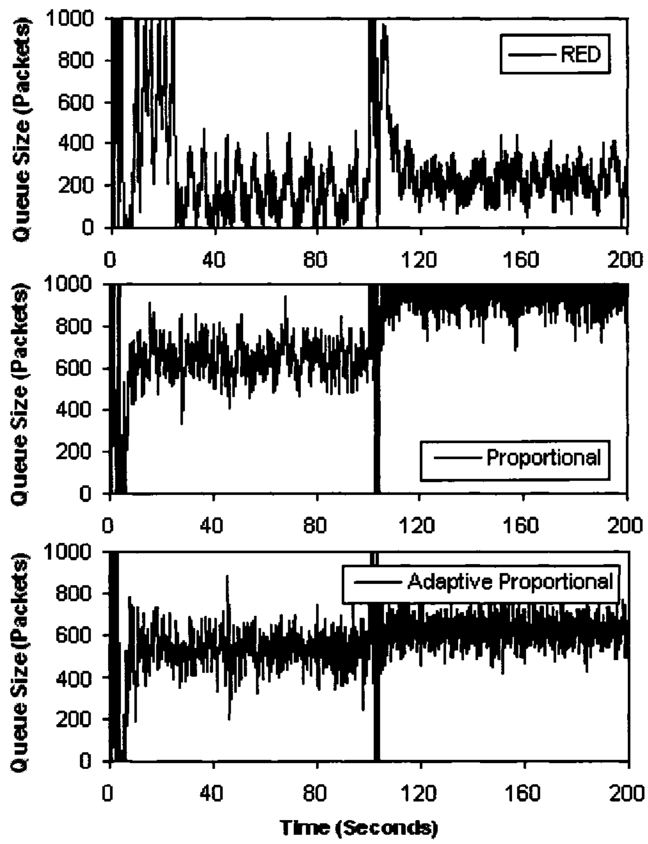

Figure 4.1: Instantaneous Queue Size Comparison among RED, Fixed P Window-Based Router Controller, Adaptive P Window-Based Router Controller (Nominal PM= *45°*)

4.2 Adaptive P Window-Based Router Controller Design Based on Gain Margin

We have also designed the adaptive P window-based router controller based on the gain margin. This is a parallel design of Section 4.1. Our study shows that it has comparable performance with the phase margin design. It is omitted here for clarity purpose, but the details can be found in [HoYa04b].

4.3 Adaptive PI Window-Based Router Controller Design Based on Both Phase and Gain Margins

To design of an adaptive PI window-based router controller based on both the phase margin and the gain margin, we shall specify a phase margin interval (ϕ_1, ϕ_2) and a gain margin interval (A_1, A_2). From such intervals, we determine a nominal phase margin ϕ_m (e.g., $\phi_m=(\phi_1+\phi_2)/2$) and a nominal gain margin A_m, (e.g., $A_m=(A_1+A_2)/2$), then we can obtain (or design) an adaptive PI window-based router controller using Equations (3.15), (3.16), (3.17) and (3.18).

Should the parameters N, τ and μ of the network change so significantly that the system phase margin or gain margin falls outside its specified interval, the PI window-based router controller need to self-tune and make both phase margin and gain margin return to their nominal values ϕ_m and A_m. This is done by real-time monitoring the phase and gain margins of the AQM control system using the following Equations (4.3), (4.4), (4.5) and (4.6).

From the definition of phase margin [Ogat02], we have

$$-\frac{\pi}{2}+\arctan\left[\frac{K_P\omega_g}{K_I}\right]-\arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right)-\omega_g\tau=-\pi+\phi_m \quad ,$$

when $$\frac{\sqrt{K_P^2\omega_g^2+K_I^2}}{\omega_g}\frac{\frac{\tau\mu^2}{2N^2}}{\sqrt{\omega_g^2+(\frac{2N}{\tau^2\mu})^2}}\frac{\frac{N}{\tau}}{\sqrt{\omega_g^2+(\frac{1}{\tau})^2}}=1 \quad ,$$

which can be simplified as

$$\phi_m=\frac{\pi}{2}+\arctan\left[\frac{K_P\omega_g}{K_I}\right]-\arctan\left(\frac{\omega_g}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_g}{\frac{1}{\tau}}\right)-\omega_g\tau \quad , \tag{4.3}$$

when $$\omega_g^6+\left(\left((\frac{2N}{\tau^2\mu})^2\right)+\left((\frac{1}{\tau})^2\right)\right)\omega_g^4+\left[(\frac{2N}{\tau^3\mu})^2-\frac{K_P^2\mu^4}{4N^2}\right]\omega_g^2-\frac{K_I^2\mu^4}{4N^2}=0 \tag{4.4}$$

The smallest positive real root of the Equation (4.4) will be the default value of gain crossover frequency ω_g to calculate the phase margin of the AQM control system by Equation (4.3).

Similarly from the definition of gain margin [Ogat02], we have

$$\frac{\sqrt{K_P^2\omega_p^2+K_I^2}}{\omega_p}\frac{\frac{\tau\mu^2}{2N^2}}{\sqrt{\omega_p^2+(\frac{2N}{\tau^2\mu})^2}}\frac{\frac{N}{\tau}}{\sqrt{\omega_p^2+(\frac{1}{\tau})^2}}=\frac{1}{A_m} ,$$

when $-\frac{\pi}{2}+\arctan\left[\frac{K_P\omega_p}{K_I}\right]-\arctan\left(\frac{\omega_p}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_p}{\frac{1}{\tau}}\right)-\omega_p\tau=-\pi$,

which can be simplified as

$$A_m=\frac{2N\omega_p\sqrt{\omega_p^2+(\frac{2N}{\tau^2\mu})^2}\sqrt{\omega_p^2+(\frac{1}{\tau})^2}}{\mu^2\sqrt{K_P^2\omega_p^2+K_I^2}} , \tag{4.5}$$

when $-\arctan\left(\frac{\omega_p}{\frac{2N}{\tau^2\mu}}\right)-\arctan\left(\frac{\omega_p}{\frac{1}{\tau}}\right)-\omega_p\tau=-\frac{\pi}{2}-\arctan\left[\frac{K_P\omega_p}{K_I}\right]$. (4.6)

It is not difficult to show that Equation (4.6) holds true only when $0\le\omega_p\tau<\pi/2$. Based on the approximation that $\tan(x)\approx x$ when $0\le x\le\pi/4$, we can simplify Equation (4.6) in 2 different cases:

Case (1) when $\omega_p\tau\le\pi/4$: by taking tangent function on the Equation (4.6), we can obtain

$$\frac{\frac{\omega_p}{\frac{2N}{\tau^2\mu}}+\frac{\omega_p}{\frac{1}{\tau}}}{1-\frac{\omega_p}{\frac{2N}{\tau^2\mu}}\frac{\omega_p}{\frac{1}{\tau}}}=\frac{\tan\left(\frac{\pi}{2}+\arctan(\frac{K_P\omega_p}{K_I})\right)-\omega_p\tau}{1+\tan\left(\frac{\pi}{2}+\arctan(\frac{K_P\omega_p}{K_I})\right)\omega_p\tau} ,$$

$$\omega_p^4+\left(\frac{2K_I}{\tau K_P}\left(1+\frac{N}{\tau\mu}\right)-\frac{1}{\tau^2}\left(1+\frac{4N}{\tau\mu}\right)\right)\omega_p^2-\frac{2NK_I}{\tau^4\mu K_P}=0 . \tag{4.7}$$

Case (2) when $\pi/4<\omega_pR_0<\pi/2$: by applying the tangent function on both sides of Equation (4.6), we can obtain

$$\omega_p^4 + \left(\frac{2N}{\tau^2\mu} + \left(2-\frac{\pi}{4}\right)\frac{1}{\tau} - \frac{K_I}{K_P}\right)\omega_p^3 - \frac{1}{\tau}\left(\left[2+\frac{\pi}{4}\right]\frac{2N}{\tau^2\mu} + \left[1+\frac{\pi}{4} - \frac{2NK_I}{\mu K_P}\right]\frac{1}{\tau} - \left[2+\frac{\pi}{4}\right]\frac{K_I}{K_P}\right)\omega_p^2$$
$$-\frac{1}{\tau^2}\left(\left[1-\frac{\pi}{4}\right]\frac{2N}{\tau^2\mu} - \left[2-\frac{\pi}{4}\right]\frac{2NK_I}{\tau\mu K_P} - \left[1-\frac{\pi}{4}\right]\frac{K_I}{K_P}\right)\omega_p - \frac{2NK_I}{\tau^4\mu K_P}\left(1+\frac{\pi}{4}\right) = 0 \qquad (4.8)$$

The smallest positive real root of Equation (4.7) or (4.8) will be the default value of the phase crossover frequency ω_p which will be used to calculate the gain margin of the control system through Equation (4.5). We first use Equation (4.7) to obtain ω_p. Then we can calculate the value of $\omega_p\tau$. If we find $\pi/4<\omega_p\tau<\pi/2$, then we recalculate ω_p by Equation (4.8). This has provided us a design procedure for the adaptive PI window-based router controller as follows:

Step 1: Specify the interval of gain margin and nominal gain margin A_m, the interval of phase margin and nominal phase margin ϕ_m.

Step 2: Estimate the network parameters N, τ and μ using the method presented in Section 3.2.

Step 3: Obtain the proportional gain K_P and integral gain K_I for adaptive PI controller using Equations (3.15), (3.16), (3.17) and (3.18).

Real-time monitoring:

Step 4: Estimate the current network parameters N, τ and μ using the method presented in Section 3.1.

Step 5: Compute the gain margin and phase margin using Equations (4.3), (4.4), (4.5) and (4.6).

Step 6: (6a) If the gain margin or phase margin falls outside the specified interval, go to Step 3.

(6b) If both the gain margin and the phase margin are still inside the interval, the PI window-based router controller remains unchanged. Then go to Step 4.

4.3.1 Performance Comparison of Adaptive and Fixed PI Window-Based Router Controller

We verify the adaptive PI window-based router controller via OPNET simulations [OPNE00]. The scenario in Section 3.2 is reused. We run simulations for the adaptive PI window-based router controller under large traffic change and small traffic change respectively. We compare the RED algorithm [FlJa93] and the fixed PI window-based router controller in [HoMi01a]. We use the same RED parameters as in Section 4.1.1.

Since the Adaptive RED (ARED) [FlGu01] can also clamp the instantaneous queue size around the target buffer occupancy, we also run the ARED algorithm for comparison under the same OPNET simulation environment. We adopt the ARED parameters suggested by [FlGu01] in this thesis. That is, the adaptive maximum drop probability max_p is confined to the range [0.01, 0.5], the maximum queue threshold max_{th} is 650, and the minimum queue threshold min_{th} is 150. The target buffer occupancy is set to 400 packets, which is the mean of max_{th} and min_{th}. The queue weight w_q is set to 0.00026. The increment parameter α is min(0.01, $max_p/4$) and the decrease factor β is 0.9.

4.3.1.1 Case with Large Traffic Change

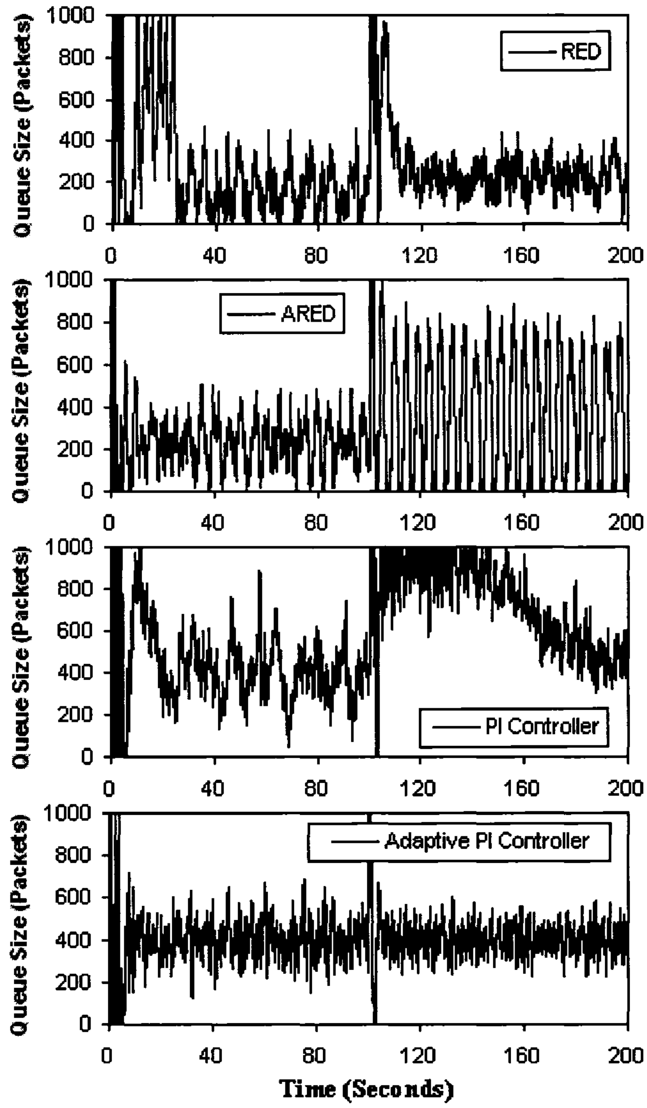

Figure 4.2: Instantaneous Queue Size Comparison among RED, ARED, Fixed PI Window-Based Router Controller, Adaptive PI Window-Based Router Controller (REF_QS=400)

To produce large traffic changes, $N=100$ greedy ftp sources and *20* http sources are used between the time $t=0$s and $t=100$s, while $N=400$ greedy ftp sources and *80* http sources are used between the time $t=100$s and $t=200$s.

Figure 4.2 shows the instantaneous queue size of RED, ARED, the fixed PI window-based router controller in [HoMi01a] and the adaptive PI window-based router controller with a reference queue size of *400* packets. At time $t=0$s, it is obvious that the adaptive PI window-based router controller and ARED show a better response than RED. Their settling time is much shorter than RED. Also the settling times of our adaptive PI window-based router controller and ARED are shorter than the fixed PI window-based router controller. Due to the lower limit of the maximum packet drop probability, the queue size of ARED does not approach the target buffer occupancy (*400* packets), while both the adaptive PI window-based router controller and the fixed PI window-based router controller reach the target buffer occupancy (*400* packets). At time $t=100$s, *300* greedy ftp sources and *60* http sources were added. The instantaneous queue sizes of both RED and the fixed PI window-based router controller exhibit sluggish response. The instantaneous queue size of the fixed PI window-based router controller increases to more than *800* packets and some buffer overflow occurs between $t=100$s and $t=140$s. The queue size of ARED adapts to the great traffic change and has great oscillations continuously around the target buffer occupancy. Both the gain margin and the phase margin of the adaptive PI window-based router control system has gone beyond their specified intervals, so the router controller self-tunes and gives a new $A_m=3$ and $\phi_m=45°$. It can be seen that the instantaneous queue size of adaptive PI window-based router controller still reaches around the target buffer occupancy (*400* packets) very quickly.

Figure 4.3 uses the average queue size to show more clearly the comparison of RED, ARED, the fixed PI window-based router controller and our adaptive PI window-based router controller. We can see that the average queue size of both RED and the fixed PI controller still cannot reach the steady state value at time $t=100$sec, while the average queue size of our adaptive PI controller has already reached the target steady state value (i.e. reference queue size *400* packets) at time $t=40$sec. Although ARED reaches its steady state value at $t=30$sec, it still cannot reach the target queue length as anticipated due to the lower limit of the maximum packet drop probability. For the big traffic load level change at time $t=100$sec, the average queue size of the adaptive PI controller does not appear to be sensitive to such change, and it reaches the steady state value (*400* packets) at time $t=110$sec. The average queue size of ARED increases slowly and will reach the target queue length (*400* packets) beyond $t=200$sec (although not shown here). On the other hand, RED and especially the fixed PI controller have reacted unfavorably – both failed to reach the steady state even at time $t=200sec$. We make the observation that the change of the average queue size in general corresponds well to the change of instantaneous queue size.

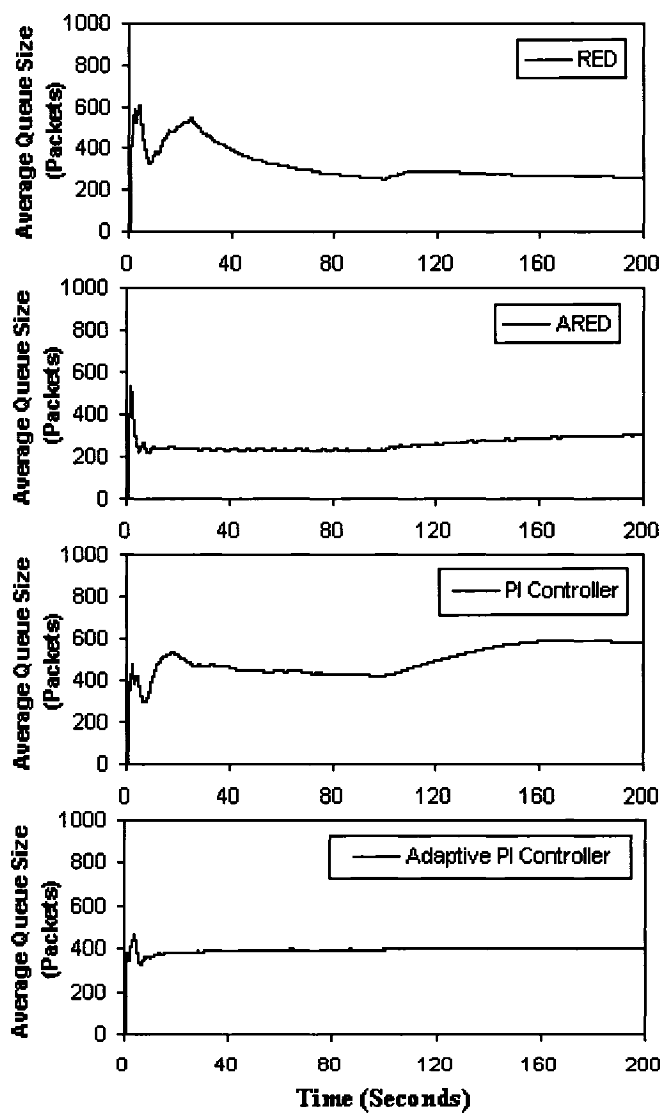

Figure 4.3: Average Queue Size Comparison among RED, ARED, Fixed PI Window-Based Router Controller, Adaptive PI Window-Based Router Controller (REF_QS=*400*)

4.3.1.2 Case with Small Traffic Change

In this case study, we keep the traffic fluctuation to within *30%* of the original traffic in the beginning of the experiment. We first use *N=100* greedy ftp sources and *20* http sources between the time t=0sec and *t=100*s, and then use *N=70* greedy ftp sources and 2*0* http sources between the time *t=100*s and *t=200*s. Finally, we use *N=130* greedy ftp sources and *30* http sources between the time *t=200*s and *t=300*s.

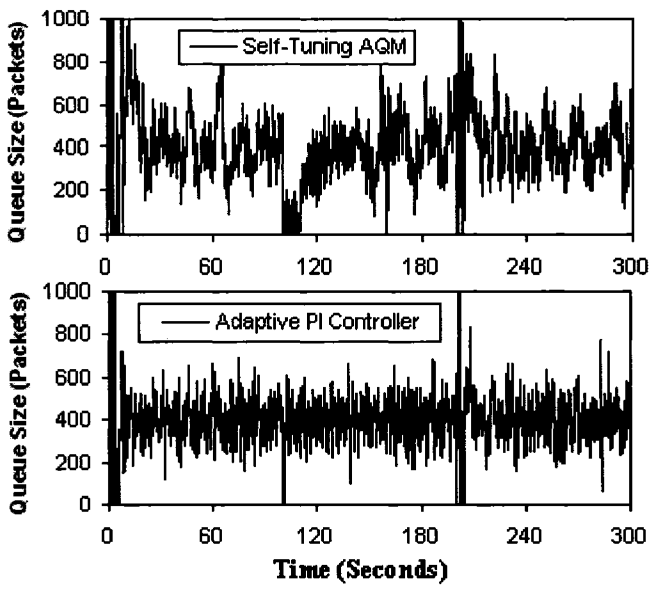

Figure 4.4: Instantaneous Queue Size Comparison of Adaptive PI Window-Based Router Controller and Self-Tuning AQM Controller [ZhHo03]

Figure 4.4 shows the instantaneous queue size of our adaptive PI window-based router controller and the self-tuning AQM controller in [ZhHo03] upon a small traffic change. At time *t=0*s, the response time of our adaptive PI window-based router controller is shorter than the self-tuning AQM controller. At time *t=100*s, the number of active TCP flows has decreased. Since both the gain margin and the phase margin are still within their respective intervals, our adaptive PI window-based router controller remains unchanged. The queue size of our adaptive PI window-based router controller does have a small spike upon this slight traffic change. On the other hand, the self-tuning AQM controller in [ZhHo03] always self-tunes to give a new AQM controller on any change of network environment. This can be seen clearly from the figure where the queue size of the self-tuning AQM controller decreases till the buffer is nearly empty and then increases to reach the target buffer occupancy (*400* packets). Continuing on with our observation. At time *t=200*s, the number of active TCP flows increases. Both the gain margin and the phase margin are still within their respective specified intervals, so our adaptive PI window-based router controller remains unchanged except for a small spike again for its queue size. The self-tuning AQM controller also produces a spike, but otherwise continues to self-tune (with larger oscillations than our adaptive PI window-based router controller) before it reaches the target buffer occupancy.

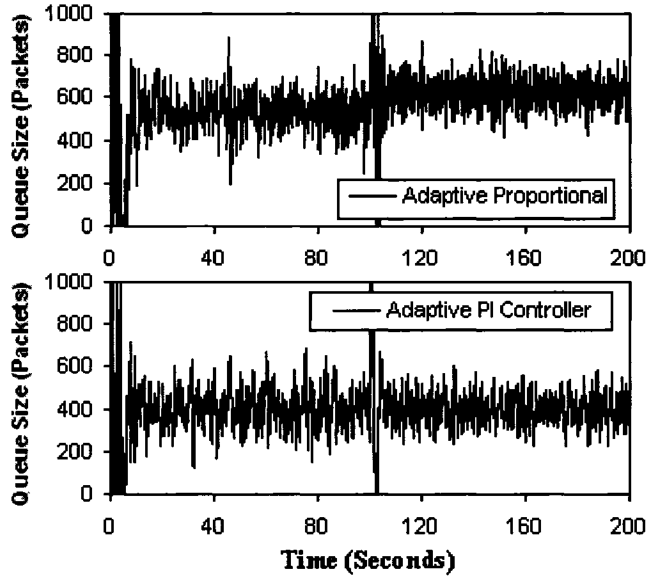

Figure 4.5: Instantaneous Queue Size Comparison of Adaptive P and PI Window-Based Router Controller

4.3.2 Performance Comparison of Adaptive P and PI Window-Based Router Controller

To confirm the advantage of the PI window-based router controller over the P window-based router controller, we have run simulations under the same network scenario as in Section 3.1. Figure 4.5 shows the instantaneous queue size comparison of our adaptive P window-based router controller and our adaptive PI window-based router controller. It can be seen that the average queue size of the adaptive P window-based router controller is around *550* packets, which is higher than the target buffer occupancy (*400* packets). Especially after time $t=100$s when the number of active TCP flows increases, its average queue size increases to be approximately *630* packets. On the other hand, the adaptive PI window-based router controller always clamps the instantaneous queue size around the reference queue size (*400* packets) regardless of the dramatic change of Internet traffic.

4.4 Adaptive PI Window-Based Router Controller with Heuristic Monitoring on TCP Traffic

Since the method to calculate the gain margin and phase margin of the AQM control system in Section 4.3 is complicated, it may not be easy for real-time implementation. Therefore, we would also suggest another approach in this section. We will provide a simpler design by specifying a proper phase margin ϕ_m only. Then we can obtain adaptive PI window-based router controller, as discussed in Section 3.4.

To achieve this, we first examine Equation (3.9), and select a heuristic parameter $\alpha=\tau^2\mu/N$ to monitor the changes of network environment. When the parameters N, τ and μ of the network change so greatly that result in $|\alpha_{new}-\alpha_{old}|>\min\{\alpha_{new}, \alpha_{old}\}$ (that is $\alpha_{new}>2\alpha_{old}$ or $\alpha_{new}<1/2\alpha_{old}$), our adaptive PI window-based router controller will self-tune based on Equations (3.9) and (3.10) and update $\alpha_{old}=\alpha_{new}$. This has provided us the following design procedure:

Step 1: Estimate network parameters N, τ and μ, calculate the initial online heuristic monitoring parameter $\alpha_0=\tau^2\mu/N$ and specify the phase margin ϕ_m.

Step 2: Obtain ω_g using Equation (3.11).

Step 3: Calculate the value of $\omega_g\tau$.

(3a) If we find that $\omega_g\tau\leq\pi/4$, then go to step 5.

Step 4: Recalculate ω_g by Equation (3.12).

Step 5: Obtain the integral gain K_I using Equation (3.10).

Step 6: Use $\alpha=\tau^2\mu/N$ to monitor the change of network environment. Once we find $|\alpha-\alpha_0|>\min\{\alpha_0,\alpha\}$, then update $\alpha_0=\alpha$ and go to Step 2. Otherwise, PI window-based router controller remains unchanged.

4.4.1 Performance Comparison of Adaptive and Fixed PI Window-Based Router Controller with Heuristic Monitor

We verify our adaptive PI window-based router controller via simulations using OPNET Modeler [OPNE00]. The simulation scenario in Section 3.2 is reused. We compare the RED algorithm [FlJa93] and the fixed PI window-based router controller in [HoMi01a].

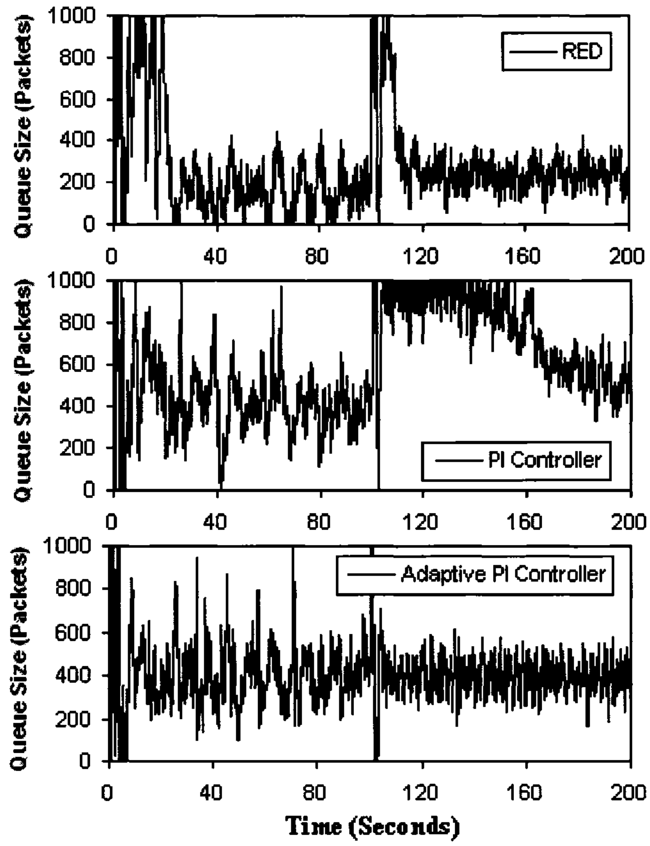

Figure 4.6: Instantaneous Queue Size Comparison among RED, Fixed PI Window-Based Router Controller, Adaptive PI window-Based Router Controller (REF_QS=*400*)

Figure 4.6 shows the instantaneous queue size of RED, fixed PI window-based router controller in [HoMi01a] and the adaptive PI window-based router controller with a 45° phase margin. At time $t=0$s, $N=100$ greedy ftp sources and *20* http sources are used. It is obvious that the adaptive PI window-based router controller shows a better response than RED. Its settling time is much shorter than RED. Also the response time of the adaptive PI window-based router controller is shorter than the fixed PI window-based router controller. Furthermore our method shows less oscillation. At time $t=100$s, *300* greedy ftp sources and *60* http sources are added. The instantaneous queue sizes of both RED and the fixed PI window-based router controller exhibit sluggish response. The instantaneous queue size of the fixed PI window-based router controller increases to more than *800* packets and some buffer overflow occurs between $t=100$s and $t=140$s. Since the monitoring inequality $|\alpha-\alpha_0|>\min\{\alpha,\alpha_0\}$ is satisfied, our adaptive PI window-based router controller self-tunes and gives a new $\phi_m=45^{\circ}$. It can be seen that the instantaneous queue size of adaptive PI window-based router controller still reaches around the steady state value quickly.

4.5 The Tradeoff between Delay and Utilization

Both queuing delay and link utilization are very important performance considerations in designing AQM control system. Intuitively, a larger buffer leads to a higher link utilization, but it also results in a longer queuing delay. While the RED controller cannot control the target queue length x_0, the queuing delay with the PI window-based router controller applied is essentially controllable with respect to the target queue length x_0, as shown in Figure 4.8.

Under the same simulation environment as in Section 3.2, we have performed experiments to study the tradeoff among the mean queuing delay and the link utilization with respect to various target queue lengths. Two scenarios are considered: (1) a network with pure 100 ftp (long-lived) flows; (2) a network with a mixture of 100 ftp and 500 http (short-lived) flows.

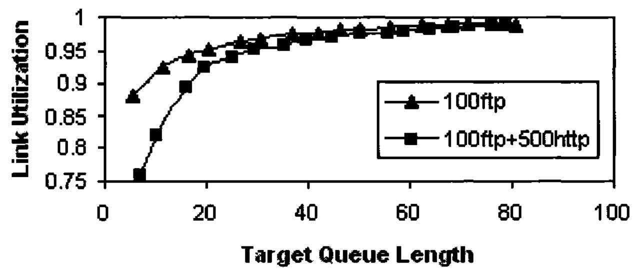

Figure 4.7: Link Utilization versus Target Queue Length

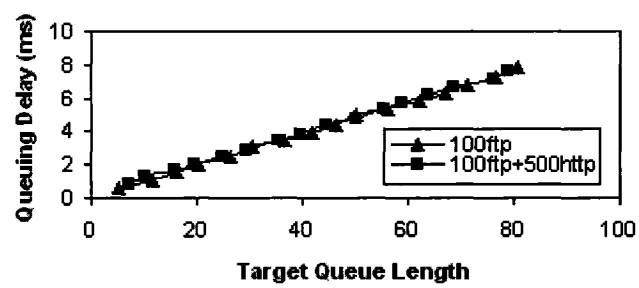

Figure 4.8: Queuing Delay versus Target Queue Length

In Figure 4.7, we plot the link utilization curve vs the target queue length for two scenarios mentioned above. In both scenarios, the link utilization is an increasing function of x_0. It is seen that a small target queue length x_0 (e.g., $x_0=20$) can already yield an extremely high link utilization (e.g., 0.95) in Scenario 1, whereas a larger x_0 is needed to reach the same link utilization when both ftp and http flows exist in Scenario 2. Beyond $x_0=40$, there is not much difference between the two scenarios. As shown in Figure 4.8, the queuing delay τ_q has a nearly linear relationship with the target queue length x_0, which confirms the formula $\tau_q = x_0/\mu$ (where μ is constant link capacity here). It is shown that larger values of x_0 give longer queuing delay and higher link utilization.

4.6 Concluding Remarks

We have applied the classical control theory to develop our adaptive P and PI window-based router controllers for AQM in the IP routers in the Internet. Considering the tradeoff between the response time and the stability of the control system, we can achieve better AQM performance by selecting a proper interval of phase margin and/or gain margin (i.e., $30^0 \leq \phi_m \leq 60^0$ and $2 \leq A_m \leq 4$). The equations for monitoring the phase margin and gain margin of AQM control system have been derived. We also can select a heuristic parameter and assign an interval to monitor the change of the Internet environment online. Our adaptive P and PI window-based router controllers self-tune only when the load changes in the Internet cause the phase margin, the gain margin, or the heuristic monitoring parameter to drift outside the specified interval. Otherwise, the controllers remain unchanged. OPNET simulations have demonstrated that with our adaptive PI window-based router controllers applied, the AQM control system can adapt very well to large fluctuations of Internet traffic, thus providing the network with a good transient behavior.

Chapter 5 Adaptive PI Rate-Based Router Controller for SISO IP Router

Having designed the window-based SISO (Single-Input Single-Output) router controllers, we now change our direction to study the rate-based router controller that can provide relatively smooth source transmission rate for streaming media traffic. We propose an adaptive PI rate-based router controller for AQM in the IP router supporting best-effort traffic in the Internet. We can again achieve good AQM performance by assigning a proper interval of phase margin. We can also select a heuristic parameter and assign an interval to monitor the change of the Internet environment online. Similar to the controller in Chapter 4, once the changes of network environment cause the monitoring parameter to fall outside the specified interval, our adaptive PI rate-based router controller will self-tune. Otherwise, adaptive PI rate-based router controller remains unchanged.

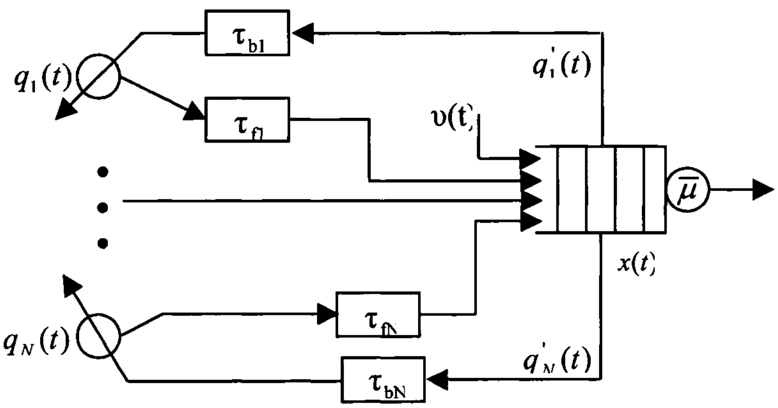

Figure 5.1: System Model of SISO IP Router

5.1 Network Model for SISO IP Router Supporting Streaming Media Traffic

The SISO IP router under consideration accumulates multiple best-effort service traffic streams and one guaranteed service traffic stream, as shown in Figure 5.1. There are N controlled source nodes that transmit the streaming media traffic through IP the packets routed through SISO IP router. The SISO IP router has a finite buffer space K to store the incoming packets and an output link to serve them at a constant data rate of μ. The buffer in the router is organized into two logical queues for the best-effort service traffic and the guaranteed service traffic respectively. It is assumed that the service for each queue is FIFO (First-Input First-Output) and both queues share the same bandwidth of the output link. The queue for the guaranteed service traffic has a higher priority for the service than the queue for the best-effort service traffic. There are two kinds of time delays: (1) the varying forward time delay τ_{fi} from the controlled source node i to the router; which includes the propagation delay, the queuing delay and the processing delay; (2) the varying feedback time delay τ_{bi} from the router to the controlled source node i via the destination, which also includes the propagation delay, the queuing delay and the processing delay. The sum $\tau_i=\tau_{fi}+\tau_{bi}$ of these two kinds of time delay is what we call the varying round trip time of the controlled source node i. The PI rate-based router controller is located in the router and can calculate a desirable source window size (i.e., source transmission rate $q_i^{'}(t)$) based on the instantaneous queue length of the buffer in the router. The controller then advertises it to the source through the IP packet and the ACK packet, so that the source can regulate its current transmission rate $q_i(t)$ to provide the best-effort service traffic. There is also an uncontrolled guaranteed service traffic flows (both local and from the upstream router) with input rate $\upsilon(t)$ into the router.

From the model of the SISO IP router with a PI rate-based router controller in Figure 5.1, the change of the queue length in the buffer of the router is the sum of the input rates of N best-effort service traffic streams and the guaranteed service traffic stream minus the service rate. It can be written as

$$\dot{x}(t)=\sum_{i=1}^{N} q_i(t-\tau_{fi})+\upsilon(t)-\mu \quad . \tag{5.1}$$

5.2 Adaptive PI Rate-Based Router Controller Design

We propose an adaptive PI rate-based router controller that can achieve a zero queue deviation in the IP router and avoid excessive buffer overflow. Every controlled source node is allowed to send the packets into the network at its maximum allowed transmission rate in order to utilize the slack bandwidth not used by the guarantee service traffic. Let x_0 be the target buffer occupancy of the buffer in the IP router. Let K_P and K_I be the proportional gain and the integral gain, two important parameters in our PI rate-based router controller design. Based on the instantaneous queue length $x(t)$ of the buffer, the advertised transmission rate $q_i^{'}(t)$ for the controlled source node i can be obtained by the following PI rate-based router control algorithm:

$$
\begin{aligned}
q_i'(t) &= K_P e(t) + K_I \int_0^t e(\tau) d\tau \\
&= K_P (x_0 - x(t)) + K_I \int_0^t (x_0 - x(\tau)) d\tau
\end{aligned} \quad . \tag{5.2}
$$

It can be easily shown that the transfer function between the advertised source transmission rate $q_i'(t)$ and the queue deviation $e(t)$ is

$$
C(s) = K_P + \frac{K_I}{s} \quad . \tag{5.3}
$$

It can also be seen from the control structure through Equation (5.2) that the advertised source rate $q_i'(t)$ is calculated based on the instantaneous queue length of the buffer. Since the advertised source transmission rate $q_i'(t)$ becomes the source rate $q_i(t)$ after a feedback time delay of τ_{bi}, we can write

$$
q_i(t) = q_i'(t - \tau_{bi}) \quad . \tag{5.4}
$$

By substituting Equation (5.4) into Equation (5.1), we can obtain the dynamic of the router as

$$
\dot{x}(t) = \sum_{i=1}^{N} q_i'(t - \tau_{bi} - \tau_{fi}) + \upsilon(t) - \mu = \sum_{i=1}^{N} q_i'(t - \tau_i) + \upsilon(t) - \mu \quad . \tag{5.5}
$$

Therefore, the transfer function between the instantaneous queue length $x(t)$ and the current advertised source transmission rate $q_i'(t)$ is given by

$$
P(s) = \sum_{i=1}^{N} \frac{e^{-s\tau_i}}{s} \quad . \tag{5.6}
$$

To simplify the controller design, we can approximate the standardized round trip time by the average value τ of τ_i, i.e., $\tau = \sum_{i=1}^{N} \tau_i / N$. Such control plant approximation has been widely adopted in the industrial process control system design e.g., [AsWi95, AsHa95, AsHa92, HoHo03], which also has found its application in network traffic control recently e.g., [KaHa02, LoAn05, HoMi01b]. The validity of this approximation has been verified by our performance evaluation later on. Therefore, under this approximation, we can update Equation (5.6) as

$$
P(s) = \frac{Ne^{-s\tau}}{s} \quad . \tag{5.7}
$$

5.2.1 PI Rate-Based Router Controller Design Based on Phase Margin

We will design our adaptive PI rate-based router controller based on its phase margin. Figure 2.3 is the AQM control system we consider for our PI rate-based router controller design. In this AQM control system, the transfer function $P(s)$ is the AQM control plant (a SISO IP router with

N controlled source nodes) and the transfer function $C(s)$ is the PI rate-based router controller for AQM. We can define the open-loop transfer function of the AQM control system as

$$G(s) = P(s)C(s) = \frac{Ne^{-s\tau}}{s}\left(\mathrm{K_P} + \frac{\mathrm{K_I}}{\mathrm{s}}\right) \quad . \tag{5.8}$$

From the definition on the phase margin of $G(s)$ [Ogat02], we can obtain

$$-\frac{\pi}{2} + \arctan\left[\frac{K_P \omega_g}{K_I}\right] - \frac{\pi}{2} - \omega_g \tau = -\pi + \phi_m \quad , \tag{5.9}$$

with
$$\frac{\sqrt{K_P^2 \omega_g^2 + K_I^2}}{\omega_g} \frac{N}{\omega_g} = 1 \quad . \tag{5.10}$$

In order to compute the two unknown parameters, the proportional gain K_P and the integral gain K_I of the PI rate-based router controller, we first determine the relationship between the two unknown parameters based on the known phase margin ϕ_m. From our many simulation results (some of which shown in Figure 5.6 later on), the relationship $K_P = (2+\sqrt{3})K_I / \omega_g$ (that is, $\arctan[K_P\omega_g/K_I]=5\pi/12$) is a good choice to achieve good AQM performance, where ω_g is the gain crossover frequency of the AQM control system. Thus we can rewrite Equation (5.9) and (5.10) as

$$-\pi + \arctan\left[\frac{(2+\sqrt{3})K_I \omega_g / \omega_g}{K_I}\right] - \omega_g \tau = -\pi + \phi_m \quad , \tag{5.11}$$

with
$$\frac{2\sqrt{2+\sqrt{3}}NK_I}{\omega_g^2} = 1 \quad . \tag{5.12}$$

Using the relationship $K_P = (2+\sqrt{3})K_I / \omega_g$, we can obtain K_P and K_I from Equations (5.11) and (5.12) as

$$K_P = \frac{\sqrt{2+\sqrt{3}}(5\pi/12 - \phi_m)}{2N\tau} \quad , \tag{5.13}$$

$$K_I = \frac{(5\pi/12 - \phi_m)^2}{2\sqrt{2+\sqrt{3}}N\tau^2} \quad . \tag{5.14}$$

When the network parameters (N and τ) change while keeping the two controller parameters K_P and K_I unchanged, we have to force the phase margin to change. Using Equations (5.9) and (5.10), we obtain ϕ_m and ω_g based on the current parameters K_P and K_I as

$$\phi_m = \arctan\left[\frac{K_P\omega_g}{K_I}\right] - \omega_g\tau \quad , \tag{5.15}$$

with $\quad \omega_g^4 - N^2K_P^2\omega_g^2 - N^2K_I^2 = 0 \quad ,$ (5.16)

where the smallest positive real root of Equation (5.16) is the default value of the gain crossover frequency ω_g which is then used to calculate the phase margin of the AQM control system by Equation (5.15), i.e.,

$$\omega_g = \sqrt{\frac{N^2K_P^2 + N\sqrt{N^2K_P^4 + 4K_I^2}}{2}} \quad . \tag{5.17}$$

We will apply Equations (5.15) and (5.17) to monitor the phase margin of AQM control system. When the network parameters (i.e., the number N of active controlled sources and the average round-trip time τ) change so significantly that the system phase margin falls outside its specified interval, the PI rate-based router controller need to self-tune to give a new ϕ_m value.

Another method to monitor the network environment

Since the above method to calculate the phase margin of the AQM control system is complicated, it may be time consuming for real-time implementation. Therefore, we would also suggest another approach. From Equation (5.15), we can see that the phase margin ϕ_m is a function of τ and ω_g, while ω_g is a function of N according to Equation (5.17). Based on Lemma 5.1 below, the phase margin would decrease when the average round trip time τ increases. Also based on Lemma 5.2, we can verify that the phase margin decreases when the number N of the active controlled sources increases. Therefore, based on our practical experience from many simulation results (some of which shown in performance evaluation later on), we observe that we can choose a parameter $\alpha=N\tau$ to monitor N (the number of the active controlled sources) and τ (the average round trip time) instead of using the phase margin ϕ_m. When the network parameters N and τ change so greatly that causes the monitoring parameter to change such that $|\alpha_{new}-\alpha_{old}|>0.5\min\{\alpha_{new}, \alpha_{old}\}$ (that is $\alpha_{new}>1.5\alpha_{old}$ or $\alpha_{new}<0.6\alpha_{old}$), our adaptive PI rate-based router controller will self-tune based on Equations (5.13) and (5.14) ,and then update $\alpha_{old}=\alpha_{new}$. We will adopt this method to monitor the change of the network environment in our performance evaluation later on.

Lemma 5.1:

If the current average round trip time τ' is longer than the previous average round trip time τ (that is, $\tau' > \tau$), and the PI rate-based router controller is designed based on the previous average round trip time τ, ***then*** the AQM control system with the current average round trip time τ' will have less phase margin than that with the previous average round trip time τ (that is, $\phi_m' < \phi_m$).

Proof:

From Equations (5.13) and (5.14), we obtain the PI rate controller $C(s) = K_P + \frac{K_I}{s}$ for AQM control system based on the average round trip time τ. For the AQM control system with the average round trip time τ, the open-loop transfer function is $G(s) = \left(K_P + \frac{K_I}{s}\right)\frac{N}{s}e^{-s\tau}$. Then from the definition of the phase margin [Ogat02], we can obtain

$$-\frac{\pi}{2} + \arctan\left(\frac{K_P\omega_g}{K_I}\right) - \frac{\pi}{2} - \omega_g\tau = -\pi + \phi_m \quad , \tag{5.9}$$

which can be simplified as

$$\phi_m = \arctan\left(\frac{K_P\omega_g}{K_I}\right) - \omega_g\tau \quad , \tag{5.15}$$

$$\frac{\sqrt{K_P^2\omega_g^2 + K_I^2}}{\omega_g}\frac{N}{\omega_g}\left|e^{-j\omega_g\tau}\right| = \frac{\sqrt{K_P^2\omega_g^2 + K_I^2}}{\omega_g}\frac{N}{\omega_g} = 1 \quad . \tag{5.10}$$

Similarly, for the AQM control system with the average round trip time τ', we can also obtain

$$\phi_m' = \arctan\left(\frac{K_P\omega_g'}{K_I}\right) - \omega_g'\tau' \quad , \tag{5.18}$$

$$\frac{\sqrt{K_P^2\omega_g'^2 + K_I^2}}{\omega_g'}\frac{N}{\omega_g'}\left|e^{-j\omega_g'\tau'}\right| = \frac{\sqrt{K_P^2\omega_g'^2 + K_I^2}}{\omega_g'}\frac{N}{\omega_g'} = 1 \quad . \tag{5.19}$$

From the Equations (5.10) and (5.19), we can determine that $\omega_g' = \omega_g$. Therefore, based on Equations (5.15) and (5.18), we can obtain

$$\phi_m' - \phi_m = \omega_g\tau - \omega_g'\tau' = \omega_g\left(\tau - \tau'\right) < 0 \quad . \tag{5.20}$$

Lemma 5.2:

If the current number of active controlled source nodes are more than the previous active controlled source nodes (that is, $N' > N$) and the PI rate-based router controller is designed based on the previous N active controlled sources, ***then*** the AQM control system with the current N'

controlled source nodes will have less phase margin than that with the previous N controlled source nodes (that is, $\phi_m^{'} < \phi_m$).

Proof:

As discussed in Lemma 5.1, for the AQM control system with N controlled source nodes, and from the definition of the phase margin [Ogat02], we can obtain Equations (5.10) and (5.15). Then we can obtain the gain crossover frequency ω_g of the AQM control system, which is the smallest positive real root of Equation (5.10), i.e.,

$$\omega_g = \sqrt{\frac{N^2 K_P^2 + N\sqrt{N^2 K_P^4 + 4K_I^2}}{2}} \quad . \tag{5.17}$$

From the definition of the gain margin [Ogat02], we can obtain

$$-\frac{\pi}{2} + \arctan\left(\frac{K_P \omega_p}{K_I}\right) - \frac{\pi}{2} - \omega_p \tau = -\pi \quad ,$$

which can be simplified as

$$-\pi + \arctan\left(\frac{K_P \omega_p}{K_I}\right) - \omega_p \tau = -\pi \quad , \tag{5.21}$$

where ω_p is the phase crossover frequency of the AQM control system.

Similarly, for the AQM control system with $N^{'}$ controlled source nodes, from the definition of the phase margin [Ogat02], we can obtain

$$\phi_m^{'} = \arctan\left(\frac{K_P \omega_g^{'}}{K_I}\right) - \omega_g^{'} \tau \quad , \tag{5.22}$$

$$\frac{\sqrt{K_P^2 \omega_g^{'2} + K_I^2}}{\omega_g^{'}} \frac{N^{'}}{\omega_g^{'}} \left| e^{-j\omega_g^{'}\tau} \right| = \frac{\sqrt{K_P^2 \omega_g^{'2} + K_I^2}}{\omega_g^{'}} \frac{N^{'}}{\omega_g^{'}} = 1 \quad . \tag{5.23}$$

We can also obtain the gain crossover frequency $\omega_g^{'}$, which is the smallest positive real root of Equation (5.23), i.e.,

$$\omega_g^{'} = \sqrt{\frac{N^{'2} K_P^2 + N^{'}\sqrt{N^{'2} K_P^4 + 4K_I^2}}{2}} \quad . \tag{5.24}$$

From the definition of the gain margin [Ogat02], we can also obtain

$$-\pi + \arctan\left(\frac{K_P \omega_p^{'}}{K_I}\right) - \omega_p^{'} \tau = -\pi \quad . \tag{5.25}$$

Finally, from the Equations (5.21) and (5.25), we can show that $\omega_p^{'} = \omega_p$.

By drawing the Bode plot of the transfer function $G(s)$, we can find the following useful Relationship 5.1 analytically.

Relationship 5.1:

If there exists the frequencies $\omega_g \le \omega_1 < \omega_2 \le \omega_p$, ***then*** we will have $-\pi = \arg(G(j\omega_p)) \le \arg(G(j\omega_2)) < \arg(G(j\omega_1)) \le \arg(G(j\omega_g)) = -\pi + \phi_m$ and $\frac{1}{A_m} = |G(j\omega_p)| \le |G(j\omega_2)| < |G(j\omega_1)| \le |G(j\omega_g)| = 1$, and vice versa. This will be shown in Figure 5.2 later on.

Let ω_g and ω_p be the gain crossover frequency and the phase crossover frequency of the AQM control system with N controlled source nodes respectively. In order to find the relationship between the phase margins ϕ_m' and ϕ_m, we let $\omega_2 = \omega_g' < \omega_p' = \omega_p$ and $\omega_1 = \omega_g$ as two different frequency points. Then from Equations (5.17) and (5.24), we can show that $\omega_g = \omega_1 < \omega_2 = \omega_g' < \omega_p' = \omega_p$. Applying the Relationship 5.1, we can obtain $-\pi = \arg(G(j\omega_p)) < \arg(G(j\omega_g')) = \arg(G(j\omega_2)) < \arg(G(j\omega_1)) = \arg(G(j\omega_g))$, i.e.,

$-\pi + \phi_m' = \arg(G(j\omega_g')) < \arg(G(j\omega_g)) = -\pi + \phi_m$. In other words, we have $\phi_m' < \phi_m$.

5.2.2 Fairness among Rate-Based Router Controlled Sources

Most congestion control algorithms show that the fairness can be achieved when the system reaches its steady state as $t \to \infty$ [KeMa98]. We define fairness here to be the condition that all the controlled sources have the same steady transmission rate. From the PI rate-based router control algorithm in Equations (5.2) and (5.4), we can see that every controlled source node i always tries to transmit the data into the network at the maximum allowed rate as advertised by the router. When the queue length of the IP router reaches its steady state, the fairness among the rate-based router controlled source nodes is considered to be achieved, as shown in our performance evaluation of Figure 5.7 and Figure 5.8 later on.

5.2.3 Implementation of Rate-Based Router Control Protocol for SISO IP router

To be compatible with the current dominant TCP/IP protocol in the Internet, we convert the advertised source transmission rate into an advertised window size for the controlled source nodes. Similar methods have been well accepted in the analysis of TCP/IP networks e.g., [KaHa02, KaKa00, LoAn05]. Thus the advertised window size $win_i'(t)$ for the controlled source node i is computed by

$$win_i'(t) = \tau_i \cdot q_i'(t) \quad . \tag{5.26}$$

Considering Equation (5.4), we can obtain the window size $win_i(t)$ for the controlled source node i as

$$win_i(t) = \tau_i \cdot q_i(t) = \tau_i \cdot q_i^{'}(t - \tau_{bi}) = win_i^{'}(t - \tau_{bi}) \quad , \tag{5.27}$$

where the window size must be an integer confined to be between 1 and the maximum window size win_i^{max} allowed for the controlled source node i, i.e., $1 \le win_i(t) \le win_i^{max}$.

Two important mechanisms of our PI rate-based router controller design are to estimate N (the number of the active controlled source nodes) and to estimate τ (the average round trip time) based on the round trip time τ_i between the controlled source i and its destination. This is done as follows:

(1) We use the specific address queue to store the source addresses of the incoming IP packets from the controlled sources:

  (1a) the source address of each IP packet under processing will enter as a "packet" into the address queue per packet interval (i.e., equal to a packet processing time). However, this is ignored if the same source address is found;

  (1b) the address queue will drop the "packet" at the head of the queue per average round trip time in order to evict the inactive controlled source address "packet" from the address queue. Therefore, the source address queue size can approximate the number N of the controlled source nodes.

(2) We compute the average round trip time τ using the averaging filter method proposed by [FlJa93], i.e.,

$$\tau(t) = (1 - w_\tau)\tau(t-1) + w_\tau \tau_i \qquad i = 1, \cdots, N \quad . \tag{5.28}$$

where we recommend the averaging weight w_τ to be $w_\tau = 1/N$ based on Relationship 5.2 in the following.

Relationship 5.2:

If (N-1) controlled source nodes transmit IP packets to their destinations through the router, then the average round trip time is $\tau^{'} = \sum_{i=1}^{N-1} \tau_i \big/ (N-1)$. The *Nth* controlled source starts to transmit the IP packets routed through the router, so the new average round trip time becomes $\tau = \sum_{i=1}^{N} \tau_i \big/ N = \left((N-1)\tau^{'} + \tau_N\right)/ N = (1 - 1/N)\tau^{'} + \tau_N / N$.

As a whole, to implement the PI rate-based router control algorithm in the IP networks, we need to define two fields in the IP packet header:

(1) the round trip time τ_i: this can be estimated by the source using the method proposed by [Stev94];

(2) the advertised window size $win_i^{'}(t)$: the original value is set to be win_i^{max}.

Normally an IP packet sent from source i will be routed through a series of AQM routers before it reaches the destination. In order to prevent all the routers from congestion in the Internet, the router will compare the current advertised window size with the advertised window size field on the IP packet, and then record the smaller value in the header of IP packet. After an IP packet reaches its destination successfully, the advertised window size is extracted from the header of IP packet and then integrated into its ACK packet. When the source receives the ACK packet, it extracts the advertised window size $win_i^{'}$ from the ACK packet and replaces the current window size $win_i(t)$.

5.3 Adaptive PI Rate-Based Router Control Algorithm

We will design an adaptive PI rate-based router controller for AQM in the SISO IP router in the Internet and regulate the source transmission rate according to the following procedure:

Router Behavior:

Step 1: Specify a proper phase margin in the range of $30^0 \leq \phi_m \leq 60^0$ for the AQM control system.

Step 2: Estimate the number N of the controlled ftp sources based on the source address queue size, extract the round trip time τ_i from the header of the IP packet and then calculate the average round trip time τ based on Equation (5.28);

Step 3: Calculate the initial value $\alpha_0 = N\tau$ for monitoring the change of the network traffic.

Step 4: From Equations (5.13) and (5.14), obtain the proportional gain K_P and integral gain K_I of the PI rate controller.

PI rate-based router controller online implementation:

Step 5: Apply the PI rate-based router control algorithm in Equation (5.2) to calculate the advertised source transmission rate $q_{i\_new}^{'}$ based on the instantaneous queue length $x(t)$ of the buffer.

Step 6: Apply Equation (5.26) to calculate the advertised window size $win_{i\_new}^{'}$.

Step 7: Compare the current advertised window size $win_{i\_new}^{'}$ with the advertised window size field $win_{i\_old}^{'}$ of the IP header.

(7a) Set $win^{'}_{i\_old} = win^{'}_{i\_new}$ if we find that $win^{'}_{i\_new} < win^{'}_{i\_old}$;

(7b) Otherwise, the advertised window size field $win^{'}_{i\_old}$ remains unchanged.

Real-time monitoring the network environment:

Step 8: Calculate the current monitoring parameter $\alpha = N\tau$.

(8a) If we find that $|\alpha - \alpha_0| > 0.5\min\{\alpha_0, \alpha\}$, then update $\alpha_0 = \alpha$. Go to Step 4.

(8b) If we find that $|\alpha - \alpha_0| \le 0.5\min\{\alpha_0, \alpha\}$, PI rate controller would not self-tune, go to Step 5.

Destination Behavior:

Step 9: Extract the advertised window size $win^{'}_{i}$ from the header of the IP packet and then integrate it into the ACK packet when IP packet arrives in the destination.

Source Behavior:

Step 10: Extract the advertised window size $win^{'}_{i}$ from the ACK packet and then regulate the current source transmission rate after the source receives ACK packet.

5.3.1 An Example

To illustrate the above design procedure for an adaptive PI rate-based router controller, we will consider an AQM router with *100* controlled ftp source nodes (*i=1, ..., 100*) in the experiment in Section 5.4.1.

Table 5.1: Round Trip Propagation Delay (RTPD) Configuration for Section 5.4.1

Source ID	RTPD(ms)	Simulation Time t (sec)
ftp 1 to 20	80	0<t≤320
ftp 21 to 40	120	0<t≤320
ftp 41 to 60	160	0<t≤320
ftp 61 to 80	200	0<t≤320
ftp 81 to 100	240	0<t≤320
http 1 to 20	80	0<t≤320
http 21 to 40	120	0<t≤320
http 41 to 60	160	0<t≤320
http 61 to 80	200	0<t≤320
http 81 to 100	240	0<t≤320
uncontrolled ftp	160	0<t≤320

The round trip propagation delays between the controlled ftp source nodes and their destinations are shown in Table 5.1. Considering the queuing delay, we can approximate the average round-trip time τ for *100* controlled ftp sources as $\tau\approx 0.25$s. Therefore, the open-loop transfer function of the AQM control system can be represented by $G(s)=\frac{100e^{-0.25s}}{s}\left(K_P+\frac{K_I}{s}\right)$. Then we specify the phase margin as $\phi_m=45^0$. Based on Equations (5.13) and (5.14), we get the PI rate-based router controller $C(s)=0.0202+\frac{0.0114}{s}$. The adaptive PI rate-based router controller will now calculate the advertised source transmission rate and integrate the converted advertised source window size into the IP packets from the controlled sources so that each controlled source can regulate its transmission rate to provide the best-effort service traffic. We use a heuristic parameter $\alpha=N\tau$ to monitor the change of network parameters (N and τ). As soon as the monitoring parameter α goes outside its specified interval, our adaptive PI rate-based router controller will self-tune.

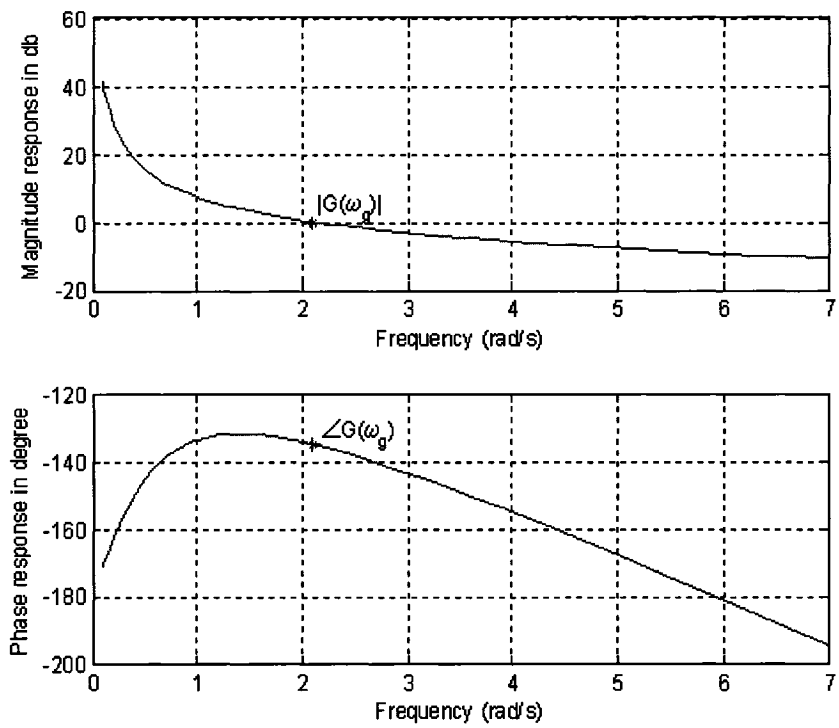

Figure 5.2: Frequency Response for the AQM Control System with Adaptive PI Rate-Based Router Controller

Figure 5.2 is Bode plot for the frequency response of the AQM control system in the frequency range of $0.1\leq\omega\leq 7$. Both the magnitude $|G(s)|$ and phase $\angle G(s)$ are decreasing function of ω in range of ω_g and ω_p as expected. It is shown that the phase margin of AQM control system with the adaptive PI rate-based router controller has been pegged at $\phi_m=180^0+\angle G(j\omega_g)=45^0$ as desired.

5.4 Performance Evaluation of PI Rate-Based Router Controller

Since our PI rate-based router controller can regulate the source transmission rate over time for AQM in the router and can provide best-effort service, we are particularly interested in analyzing the transient behaviors of the network. Transient behaviors such as the duration of the response time and the maximum transient queue length are the main concerns in the performance evaluation. We have verified our adaptive PI rate-based router controller using OPNET Modeler [OPNE00]. We consider two types of TCP Reno sources in the simulations – the long-lived sources (i.e., controlled ftp sources for best-effort service traffic and uncontrolled ftp source for guaranteed service traffic); and the short-lived sources (i.e., http sources), as described in Chapter 3.

We have conducted simulations for AQM control system with our adaptive PI rate-based router controller under the case of varying controlled source nodes (that is, N changes) and the case of constant controlled source nodes (that is, N does not change). To validate our PI rate-based router control algorithm, we have made performance comparison with the AIMD control using RED [FlJa93] and the AIMD control using PI-RED [HoMi01a] under the network environment comparable to [HoMi01a]. In the following simulations, all IP packets have the same size of *1024* bytes. The link bandwidth (i.e., service rate of AQM Router 1) is T3 (*44,736,000*bps, i.e., *5,461*packets/sec). The buffer size of the AQM Router is *1000* packets based on the different bandwidth-delay product [HoMi01a]. The maximum window size allowed for all the TCP sources is *1000* packets. The target buffer occupancy is *400* packets. The sampling time for PI rate-based router controller and PI-RED controller [HoMi01a] is chosen to be per packet interval (i.e., *0.183*ms) [FlJa93]. The phase margin for the AQM control system with an PI rate-based router controller is $\phi_m=45^0$.

5.4.1 Case With Constant Controlled Source Nodes

We first investigate the case where the number of controlled ftp source nodes is fixed. The window size of the uncontrolled source is different in different active period, as shown in Figure 5.3. Then we compare the RED algorithm [FlJa93] and the PI-RED controller in [HoMi01a]. Table 5.1 provides the round trip propagation delays and the active periods (i.e., simulation time) for *100* controlled ftp sources, *1* uncontrolled ftp source and *100* http sources. To allow further comparison, we adopt the comparable RED parameters as in [HoMi01a]. That is, the maximum drop probability $p_{\max}$ is *0.1*; the maximum queue threshold max_{th} is *700*; the minimum queue threshold min_{th} is *150*; the queue weight w_q is 1.33×10^{-6}. We adopt Equations (6) and (7) in [HoMi01a] to calculate the PI-RED controller based on the estimated parameters ($N=100$, τ =0.25s and $\mu=5{,}461$packets/sec) of the current network. Figures 5.3 to 5.5 below demonstrate the comparison of various instantaneous performance measure of the uncontrolled ftp source and that of the controlled ftp source node *41* (whose RTPD=*160*ms) among the RED controller, the PI-RED controller and the adaptive PI rate-based router controller as the uncontrolled traffic changes.

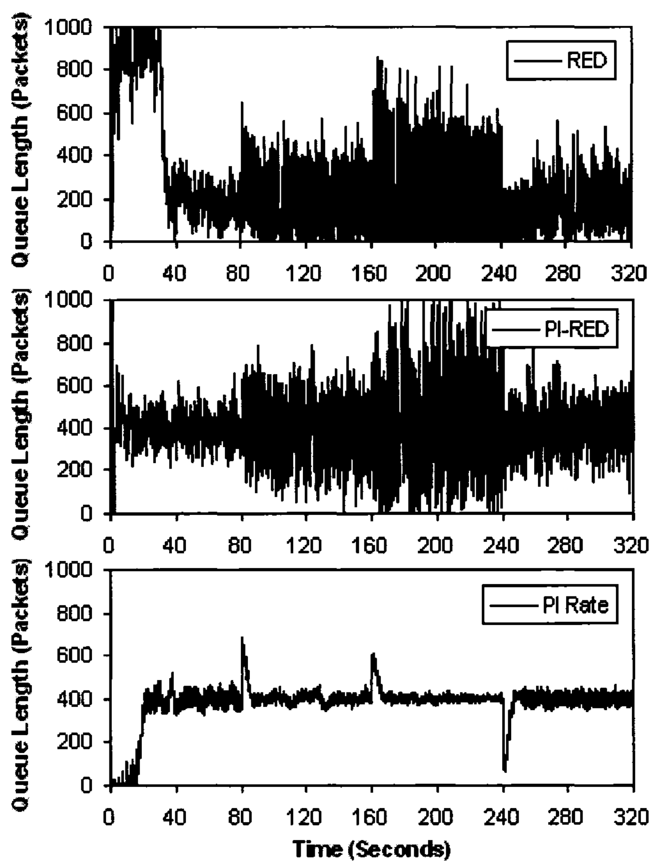

Figure 5.3: Instantaneous Queue Length Comparison among RED, PI-RED Controller and Adaptive PI Rate-based Router Controller

Figure 5.3 compares the instantaneous queue length of the RED controller, the PI-RED controller and the adaptive PI rate-based router controller as the uncontrolled traffic changes. It can be seen that the queue length in the RED router fluctuates with a large amplitude. The PI-

RED controller can clamp the queue length around the target buffer occupancy (*400* packets), but its queue length oscillation is still noticeable. With our adaptive PI rate-based router controller applied, the queue length remains around the target buffer occupancy, and is quite stable upon the changes of the uncontrolled traffic.

Figure 5.4 compares the instantaneous window size of the uncontrolled ftp source and that of the controlled ftp source node *41* (whose RTPD=*160*ms) for the RED controller, the PI-RED controller and the adaptive PI rate-based router controller as the uncontrolled traffic changes. It can be seen that both window sizes of the RED and the PI-RED controller have great fluctuation, while that of our adaptive PI rate-based router controller has slight oscillation.

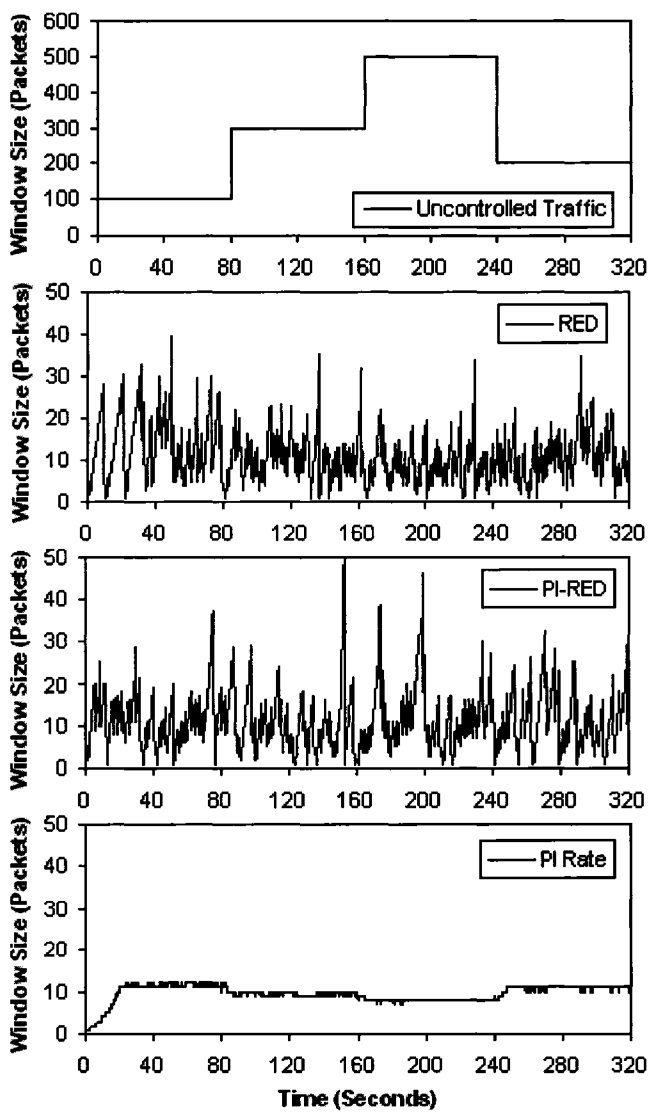

Figure 5.4: Instantaneous Window Size Comparison for Controlled ftp Source among RED, PI-RED Controller and Adaptive PI Rate-Based Router Controller

Figure 5.5 shows compares the instantaneous transmission rates of the controlled ftp source node *41* (whose *RTPD=160*ms) for RED, PI-RED controller and adaptive PI rate-based router controller as the uncontrolled traffic changes. Both transmission rates of the RED and the PI-RED controller exhibit large fluctuation. Our adaptive PI rate-based router controller provides a relatively smooth source transmission rate. Other controlled ftp source nodes have comparable transmission rates.

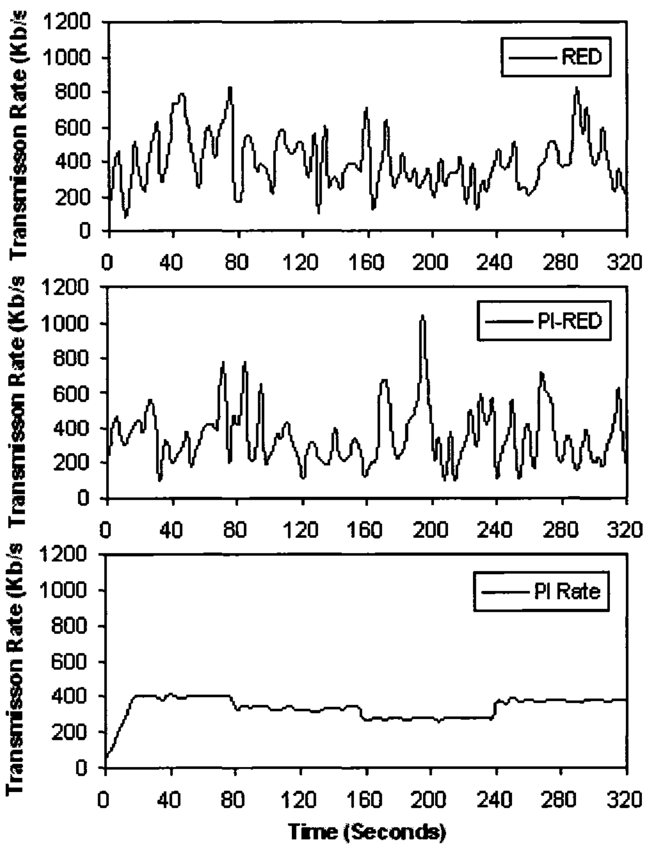

Figure 5.5: Instantaneous Transmission Rate Comparison for Controlled ftp Source among RED, PI-RED Controller and Adaptive PI Rate-based Router Controller

5.4.1.1 Impact of Phase Margin on AQM Performance of SISO IP Router

In order to demonstrate the impact of the phase margin on the transient response of our AQM control system, we have performed OPNET simulations for different phase margin respectively under the same network environment as Section 5.4.1 for the time between *t=0*s and *t=120*s. The uncontrolled guaranteed service traffic flows into the AQM router with a constant window size of *100* packets. We use the notation *PM* to stand for *phase margin*.

Figure 5.6 shows the instantaneous queue length of the AQM router under different phase margins. As evident from Figure 5.6, when the phase margin increases, the settling time increases. If the phase margin is set to a negative value (e.g., *–10*degree), from the viewpoint of classic control theory, the queue length should become unstable, as demonstrated in the top figure of Figure 5.6.

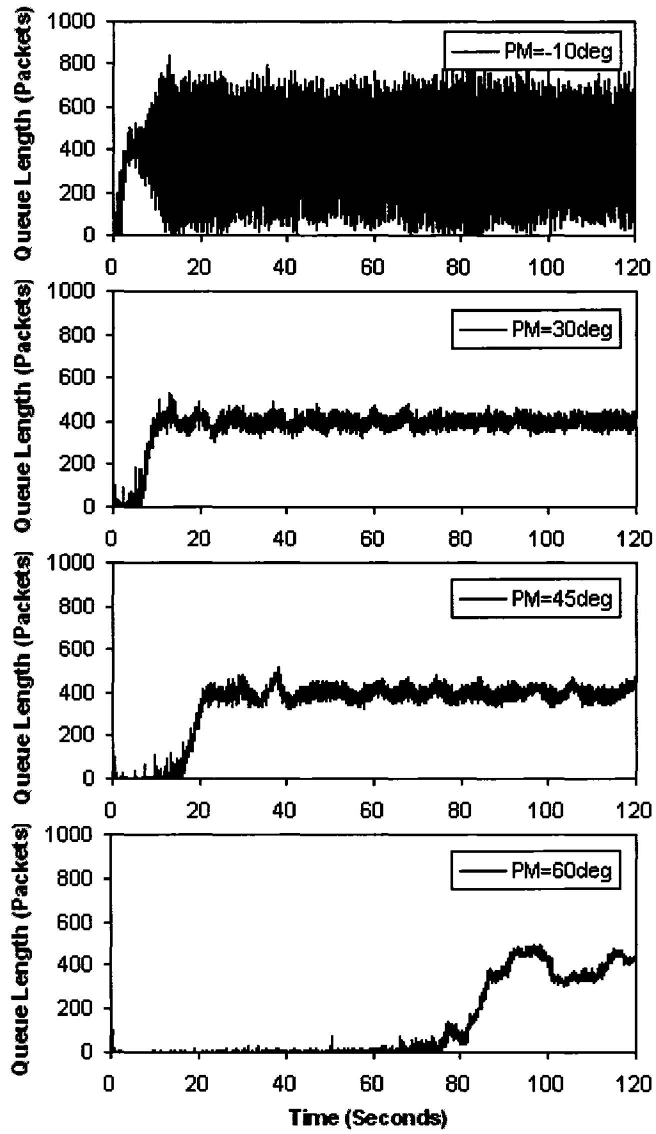

Figure 5.6: Instantaneous Queue Length of AQM Router upon Different Phase Margin

5.4.2 Case with Varying Controlled Source Nodes

In this case study, we allow the number of active controlled ftp source nodes to vary. There are *25* controlled ftp source nodes and one uncontrolled guaranteed ftp source node transmitting the IP packets through the router. At the same time, there are *25* http sources sending data through the router. Table 5.2 provides the round trip propagation delays and the active periods (i.e., simulation time) for the controlled ftp sources, uncontrolled ftp source and http sources. The window size of the uncontrolled source is fixed within different active periods, as will be shown in Figure 5.7.

Table 5.2: Round Trip Propagation Delay (RTPD) Configuration for Section 5.4.2

Source ID	RTPD(ms)	Simulation Time t (sec)
ftp 1 to 5	80	0<t≤320
ftp 6 to 10	160	0<t≤320
ftp 11 to 15	240	0<t≤320
ftp 16 to 20	280	160<t≤320
ftp 21 to 25	340	160<t≤320
http 1 to 5	80	0<t≤320
http 6 to 10	160	0<t≤320
http 11 to 15	240	0<t≤320
http 16 to20	280	0<t≤320
http 21 to 25	340	0<t≤320
uncontrolled ftp	160	0<t≤320

Figure 5.7 shows the instantaneous window size and transmission rate of the uncontrolled ftp source, the instantaneous transmission rate of the controlled ftp source 1 whose round-trip propagation delay (RTPD) is *80*ms, and the instantaneous queue length of the AQM Router 1 respectively. Figure 5.8 shows the instantaneous transmission rates of the controlled ftp source nodes ($i=6,11,16,21$) with different typical RTPD (as shown in Table 5.2).

At time t=0s, the 15 controlled ftp source nodes ($i=1,2,\ldots, 15$) start to transmit the packets into the network. The open-loop transfer function of the AQM control system is approximated by $G(s)=\frac{15e^{-0.25s}}{s}\left(K_P+\frac{K_I}{s}\right)$. Then we can obtain the PI rate controller $C(s)=0.1349+\frac{0.0757}{s}$. The controlled source node *1* starts to increase its transmission rate till it reaches a steady state of about *2.5*Mbps at the time around *20*s. The instantaneous queue length increases and approaches its target buffer occupancy (*400* packets). It starts to clamp its steady state around *400* packets at the time about *20*s, as shown in Figure 5.7. At time $t=80$s, uncontrolled ftp source increases its window size from *200* packets to *500* packets, the controlled source node *1* starts to decreases its transmission rate and starts to maintain a rate of around *1.78*Mbps at the time about *92*s. The instantaneous queue length increases till reaches its peak at the time about *82*s. Then it decreases and approaches to its target buffer occupancy (*400* packets). At time *t=160*s, another *10* controlled ftp sources (*i=16,17,..., 25*) start to send the packets to the destination through the router. The monitoring parameter fall outside its specified interval, so adaptive PI rate controller self-tunes. Both new proportional and integral gain of PI rate controller decrease, which causes all *15* controlled ftp source nodes (*i=1,2,..., 15*) to reduce their transmission rates sharply, as shown in Figure 5.7 and Figure 5.8. Therefore the instantaneous queue length decreases and then approaches the target buffer occupancy (*400* packets) very quickly. At time *t=240*s, the

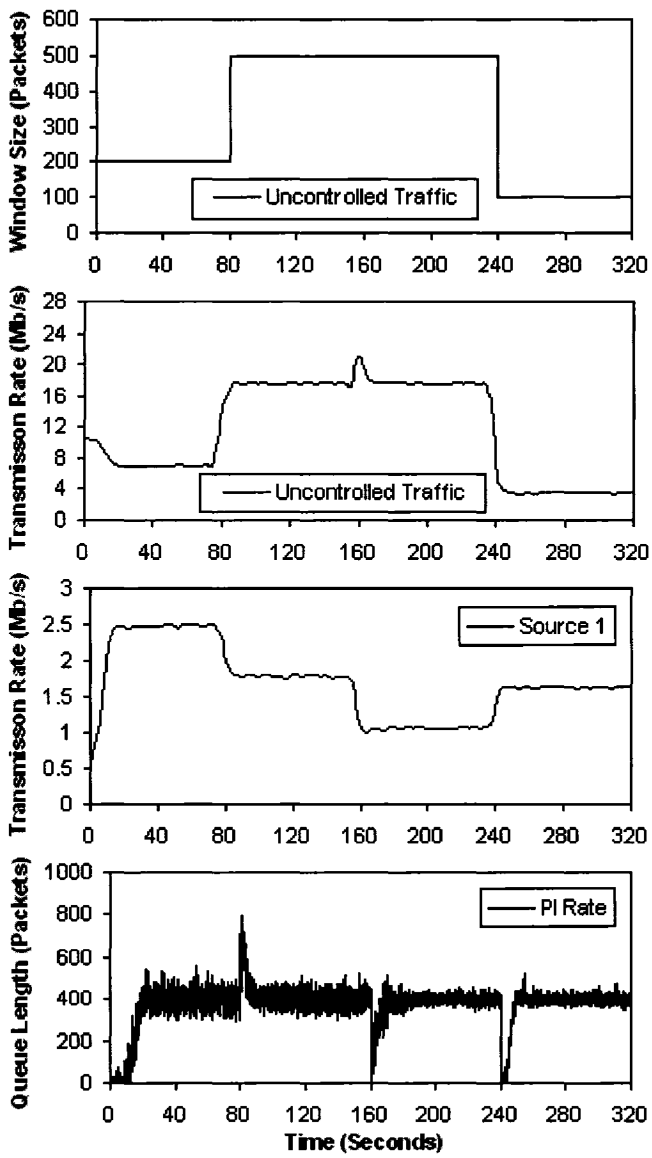

Figure 5.7: Transient Behavior of AQM Control System

uncontrolled guaranteed traffic decreases. The controlled source node *1* increases its transmission rate from *1.1*Mbps to *1.6*Mbps. The instantaneous queue length decreases and then approaches to the steady value $x_s = x_0 = 400\text{packets}$ at time *t=258*s. The transmission rates of the other controlled ftp source nodes (*i=6, 11, 16, 21*) have similar changes as the source node *1*, as shown in Figure 5.8. The controlled ftp source nodes with the same RTPD have the similar transmission rates.

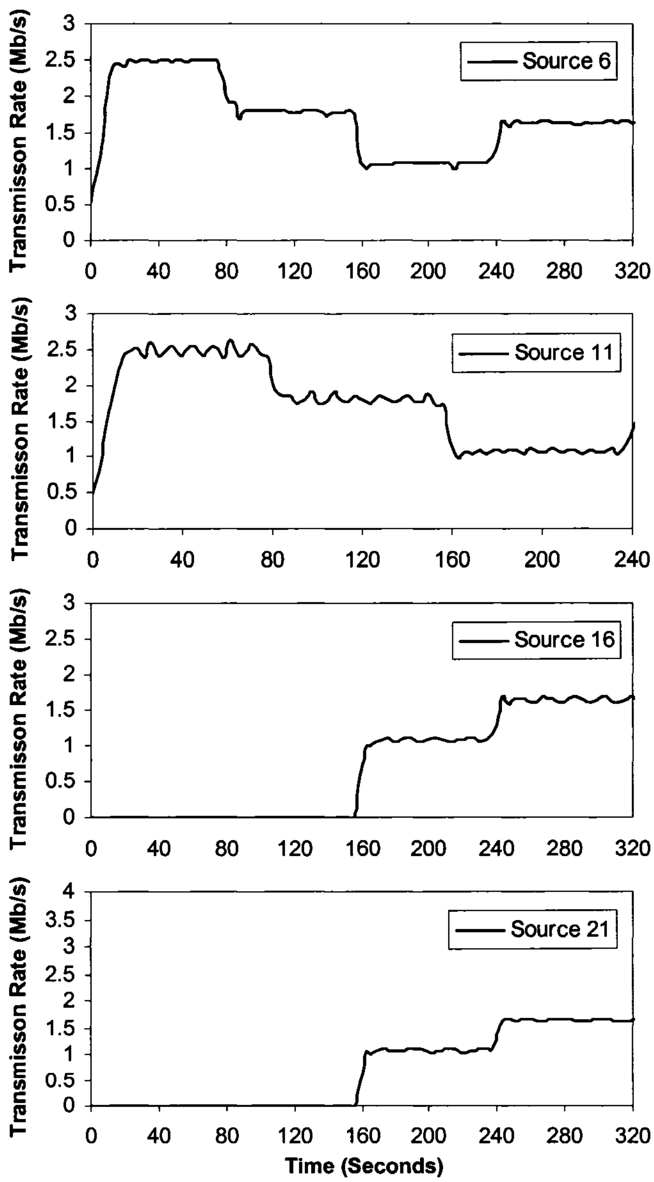

Figure 5.8: Transmission Rates of Multiple Controlled ftp Sources (*i=6,11,16,21*)

5.5 Concluding Remarks

We have developed an adaptive PI rate-based router controller for the AQM router supporting best-effort service traffic in the Internet. Considering the tradeoff between the response time and the stability condition, we can achieve a better AQM performance by selecting a proper phase margin. The PI rate-based router controller can clamp the steady-state value of the queue length around the specified target buffer occupancy. Every source node always tries to transmit the IP packets at a maximum allowed transmission rate to support the best-effort service traffic, so that is can use the unused bandwidth left behind by the guaranteed service traffic and therefore make maximize use of the network resources. We have demonstrated by OPNET simulations that with our adaptive PI rate-based router controller applied, the AQM control system can adapt very well to both the fluctuation of the uncontrolled guaranteed traffic and the dramatic changes of the network environment, thus providing the IP network with good stability robustness. The relatively smooth source transmission rate makes our adaptive PI rate-based router controller quite suitable for the rate control of streaming media transmission such as voice over IP in the Internet [HaFl03].

Again, we would like to reiterate that our controller is developed based on control theory which, as demonstrated, can always guarantee the stability of our AQM control system.

Chapter 6 Adaptive Multiloop PI Rate-Based Router Controller for MIMO IP Router

Unlike previous chapters on SISO routers, we shall explore the MIMO (Multiple-Input Multiple-Output) territory. We shall design an adaptive multiloop PI rate-based router controller for the MIMO IP router to support best-effort streaming media traffic in the Internet. Our controller design employs the DNA (Direct Nyquist Array) method that would allow users to achieve a good stability robustness of the control system through the gain and the phase margin specifications. The controller for the dominant loop *i* self-tunes only when the changes in network environment cause the monitoring parameter to drift outside the specified interval.

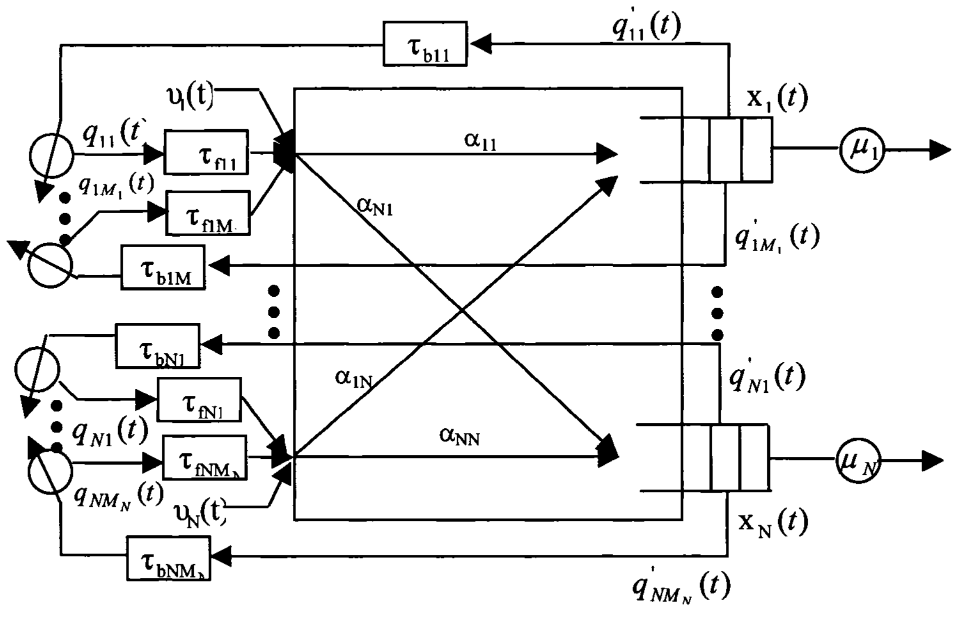

Figure 6.1: Multiple-Input Multi-Output Model of IP Router

6.1 Network Model for MIMO IP Router Supporting Streaming Media Traffic

The central square box and the output buffer in Figure 6.1 depict the MIMO model of an IP router in the Internet. There are *N* input ports and *N* output ports. The controlled source nodes are routed through a MIMO IP router consisting of *N* input ports and *N* output ports. Each input port *i* receives streaming media IP packets from M_i controlled sources. Each output port *i* of the router has a finite buffer space K_i to store the incoming packets and serves them at a constant data rate of μ_i. The buffer in every output port is organized into two logical queues for the best-effort service traffic and the guaranteed service traffic respectively. It is assumed that the service discipline for each queue is FIFO (First-Input First-Output), and both queues share the same bandwidth of the output link. The queue for the guaranteed service traffic has a higher priority for the service than the queue for the best-effort service traffic. In the following, we shall let "$j|i$" to represent controlled source j that transmits IP packets routed through the input port i of MIMO IP router.

It is assumed that the majority of the traffic into the input port i flows out of the output port i, and therefore only a small portion flows to other output ports. That is, $\sum_{i=1}^{N}\alpha_{ij}=1$ and $\alpha_{jj} > \sum_{i=1,i\neq j}^{N}\alpha_{ij}$, where α_{ij} is traffic distribution ratio from the jth input port to the ith output port. The internal processing time delays between the input ports and the output ports are so short that they can be neglected. Therefore we only consider two kinds of time delays: (1) the varying forward time delay τ_{fij} from the source node $j|i$ to the input port i of the router, which includes the propagation delay, the queuing delay and thye processing delay; (2) the varying feedback time delay τ_{bij} from the output port i of the router via the destination before returning back to the source node $j|i$ (that transmits the packets into the input port i) This also includes the propagation delay, the queuing delay and the processing delay. As in Chapter 5, the sum $\tau_i=\tau_{fi}+\tau_{bi}$ of these two kinds of time delay is what we call the varying round trip time between the output port i of the router and the source node $j|i$.

The PI rate-based router controller is located in every output port i of the MIMO IP router. Based on the instantaneous queue length $x_i(t)$ in the buffer of the output port i, the PI rate-based router controller can calculate a desirable advertised window size (i.e., advertised transmission rate $q'_{ij}(t)$) for the respective source node. The advertised window size is recorded in the IP packet going to the destination and then transferred to the ACK packet for the controlled source node $j|i$. After extracting the advertised window size from the ACK packet, the source node $j|i$ can regulate its sending rate $q_{ij}(t)$ to offer best-effort service traffic. There is also an uncontrolled guaranteed service traffic flows (both local and from the upstream router) with an input rate $\upsilon_i(t)$ into the input port i.

From the MIMO model of the router in Figure 6.1, the change of the total queue length in the output port i of the router is the sum of the input rates of both best-effort service traffic and guaranteed service traffic minus the service rate. It can be written as

$$\dot{x}_i(t)=\sum_{j=1}^{N}\alpha_{ij}\sum_{k=1}^{M_j}q_{jk}(t-\tau_{fjk})+\sum_{j=1}^{N}\alpha_{ij}\upsilon_j(t)-\mu_i \qquad i=1,2,\cdots,N \quad . \tag{6.1}$$

6.2 Adaptive Multiloop PI Rate-Based Router Controller Design

Our design is to support the best-effort service traffic while preventing congestion in the network. Every controlled source node is allowed to send the packets into the network at its maximum allowed transmission rate in order to utilize the slack bandwidth not used by the guarantee service traffic. Let x_{0i} be the target buffer occupancy of the output port i of the MIMO router. Let the parameters K_{Pi} and K_{Ii} be the proportional gain and the integral gain of PI rate-based router controller for the *ith* output port of MIMO IP router respectively. Based on the instantaneous queue length $x_i(t)$ of the buffer, the advertised transmission rate $q_{ij}^{'}(t)$ for the respective source node $j|i$ will be regulated by the following PI rate-based router control algorithm.

$$\begin{aligned} q_{ij}^{'}(t) &= K_{Pi}e_i(t)+K_{Ii}\int_0^t e_i(\tau)d\tau \\ &= K_{Pi}(x_{0i}-x_i(t))+K_{Ii}\int_0^t (x_{0i}-x_i(\tau))d\tau \end{aligned} \tag{6.2}$$

It can be easily shown that the transfer function between the advertised transmission rate $q_{ij}^{'}(t)$ and the queue deviation $e_i(t)$ is given as

$$c_{ii}(s)=K_{Pi}+\frac{K_{Ii}}{s} \quad , \tag{6.3}$$

which forms the diagonal multiloop PI rate-based controller transfer function matrix $C(s)$ as

$$C(s)=\left[c_{11}(s),c_{22}(s),\cdots,c_{NN}(s)\right] \quad . \tag{6.4}$$

Equation (6.2) says that the advertised source rate $q_{ij}^{'}(t)$ is calculated based on the instantaneous queue length $x_i(t)$ of the buffer. The advertised transmission rate $q_{ij}^{'}(t)$ becomes the source rate $q_{ij}(t)$ after the feedback time delay τ_{bij}, which can be expressed as

$$q_{ij}(t)=q_{ij}^{'}(t-\tau_{bij}) \quad . \tag{6.5}$$

Therefore, by substituting Equation (6.5) into Equation (6.1), we can obtain the dynamic of the router as

$$\begin{aligned}\dot{x}_i(t) &= \sum_{j=1}^{N}\alpha_{ij}\sum_{k=1}^{M_j} q'_{jk}(t-\tau_{bjk}-\tau_{fjk}) + \sum_{j=1}^{N}\alpha_{ij}\upsilon_j(t)-\mu_i \\ &= \sum_{j=1}^{N}\alpha_{ij}\sum_{k=1}^{M_j} q'_{jk}(t-\tau_{jk}) + \sum_{j=1}^{N}\alpha_{ij}\upsilon_j(t)-\mu_i \qquad i=1,2,\cdots,N\end{aligned} \quad . \tag{6.6}$$

Therefore, the transfer function between the queue length $x_i(t)$ (of the output port i) and the advertised transmission rate $q'_{ij}(t)$ (for the controlled source node $j|i$ that transmits the packets (best-effort service traffic) into the input port j) can then be obtained as

$$p_{ij} = \frac{\alpha_{ij}\sum_{k=1}^{M_j} e^{-\tau_{jk}s}}{s} \quad , \tag{6.7}$$

which forms the general open-loop transfer function matrix $P(s)$ for the stable N x N AQM control plant (i.e., our MIMO IP router shown in Figure 6.1) represented by

$$P(s) = \begin{bmatrix} p_{11}(s) & \cdots & p_{1N}(s) \\ \vdots & & \\ p_{N1}(s) & \cdots & p_{NN}(s)\end{bmatrix} \quad .$$

To simplify the controller design later on, we can adopt the approach of [KaHa02] to approximate the round trip time for the dominant loop i by the average value $\bar{\tau}_i$ of τ_{ij}, i.e., $\bar{\tau}_i = \frac{\sum_{j=1}^{M_i}\tau_{ij}}{M_i}$. Such approximation for control plant has been widely adopted in industrial process control system e.g., [AsHa92, AsHa95, HoHo03], and acceptable in network traffic control as well e.g., [KaHa02, LoAn05]. This has also been verified by our performance evaluation later on. Therefore, under this approximation, we can update Equation (6.7) as

$$p_{ij} = \frac{\alpha_{ij}M_j e^{-\bar{\tau}_j s}}{s} \quad , \tag{6.8}$$

which forms the general open-loop transfer function matrix $P(s)$ as

$$P(s) = \begin{bmatrix} \frac{\alpha_{11}M_1 e^{-\bar{\tau}_1 s}}{s} & \frac{\alpha_{12}M_2 e^{-\bar{\tau}_2 s}}{s} & \cdots & \frac{\alpha_{1N}M_N e^{-\bar{\tau}_N s}}{s} \\ \frac{\alpha_{21}M_1 e^{-\bar{\tau}_1 s}}{s} & \frac{\alpha_{22}M_2 e^{-\bar{\tau}_2 s}}{s} & \cdots & \frac{\alpha_{2N}M_N e^{-\bar{\tau}_N s}}{s} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\alpha_{N1}M_1 e^{-\bar{\tau}_1 s}}{s} & \frac{\alpha_{N2}M_2 e^{-\bar{\tau}_2 s}}{s} & \cdots & \frac{\alpha_{NN}M_N e^{-\bar{\tau}_N s}}{s}\end{bmatrix} \quad . \tag{6.9}$$

6.2.1 The DNA Design Method

We will make use of the DNA Stability Theorem (summarized in Appendix C) to design our adaptive multi-loop PI rate-based router controller. Unlike the trial and error approaches in the previous DNA design method, we shall shape the Gershgorin bands using an analytical approach that is based on the gain margin specification. Figure 2.3 depicts the block diagram of the control system we consider for our adaptive multi-loop PI rate-based router controller design. In this AQM control system, $P(s)$ is the AQM control plant (the MIMO IP router with controlled sources) discussed above. The transfer function matrix $C(s)$ is the multiloop PI rate-based router controller. Therefore, we can define the open-loop transfer function of MIMO AQM control system as

$$G(s) = P(s)C(s) = \begin{bmatrix} g_{11}(s) & \cdots & g_{1N}(s) \\ \vdots & & \\ g_{N1}(s) & \cdots & g_{NN}(s) \end{bmatrix} .$$

Since the PI rate-based router controller $c_{ii}(s) = K_{Pi} + \frac{K_{Ii}}{s}$ regulates the advertised source transmission rate $q'_{ij}(t)$ based on the queue length $x_i(t)$ of the output port i, each diagonal element $g_{ii}(s)$ of $G(s)$ is the target of our multi-loop controller design, while the other elements $g_{ji}(s)$ $(j \neq i)$ in the same column can be considered as the interaction on the diagonal element.

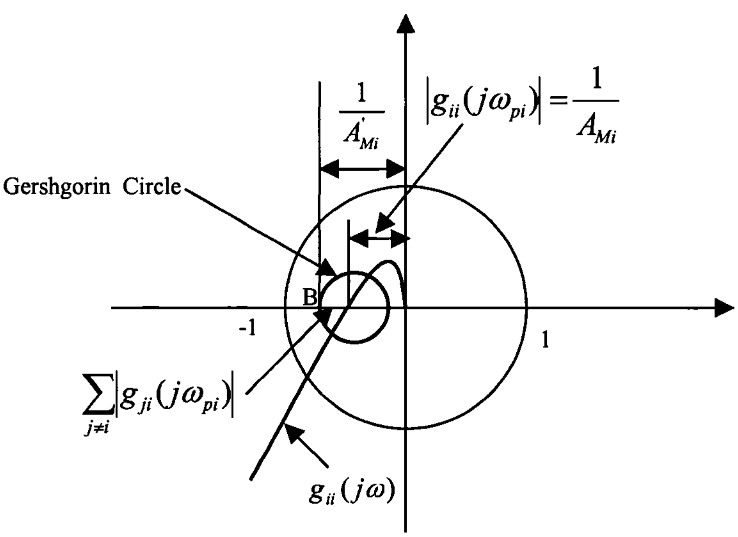

Figure 6.2: A Typical Nyquist Diagram with Gershgorin Circle at the Phase Crossover Frequency

Figure 6.2 is a typical Nyquist curve of a diagonal element $g_{ii}(j\omega)$ with one of Gershgorin circles shown at its phase crossover frequency ω_{pi}. The Gershgorin band (as mentioned informally before) consists of a band of many such circles with centers $g_{ii}(j\omega)$ and radii $\sum_{j,j \neq i} |g_{ji}(j\omega)|$ for $0 \le i, j \le N$. By focusing on point B in Figure 6.2 where a particular

Gershgorin circle intersects the negative axis at the phase crossover frequency ω_{pi} of $g_{ii}(s)$, we can define the gain margin of this Gershgorin circle centered on $g_{ii}(s)$ to be $A'_{Mi} = \frac{1}{|B|}$.

We want to shape the Gershgorin bands of circles by specifying the gain margin $A'_{Mi} \quad i = 1, \cdots, N$ of each circle so that it does not encircle the point (-1+j0) in order to achieve system stability according to the DNA Stability Theorem [Maci89]. Note that the gain margin A'_{Mi} in a SISO control system only defines one point and cannot guarantee that its Nyquist curve does not encircle the point (-1+j0). However, we can extend this SISO technique of shaping the Nyquist Curves to the Gershgorin bands by specifying the gain margin, which is a well-accepted technique in practice and in classical control. This is done as following:

By using a simple geometry in Figure 6.2 substituting from Equation (6.9) and considering $\sum_{j=1}^{N} \alpha_{ji} = 1$, the gain margin A_{Mi} of $g_{ii}(s)$ can be expressed in terms of A'_{Mi} as

$$
\begin{aligned}
A_{Mi} &= A'_{Mi}\left(1 + \frac{\sum_{j,j\neq i}|g_{ji}(j\omega_{pi})|}{|g_{ii}(j\omega_{pi})|}\right) \\
&= A'_{Mi}\left(1 + \frac{\sum_{j,j\neq i}|p_{ji}(j\omega_{pi})||c_{ii}(j\omega_{pi})|}{|p_{ii}(j\omega_{pi})||c_{ii}(j\omega_{pi})|}\right) \\
&= A'_{Mi}\left(1 + \frac{\sum_{j,j\neq i}|p_{ji}(j\omega_{pi})|}{|p_{ii}(j\omega_{pi})|}\right) = A'_{Mi}\left(1 + \sum_{j,j\neq i}\left|\frac{\alpha_{ji}M_i e^{-j\omega_{pi}\bar{\tau}_i}}{j\omega_{pi}}\right| \Big/ \left|\frac{\alpha_{ii}M_i e^{-j\omega_{pi}\bar{\tau}_i}}{j\omega_{pi}}\right|\right) \\
&= A'_{Mi}\left(1 + \frac{\sum_{j,j\neq i}\alpha_{ji}}{\alpha_{ii}}\right) = A'_{Mi}\left(\frac{\alpha_{ii} + \sum_{j,j\neq i}\alpha_{ji}}{\alpha_{ii}}\right) \\
&= A'_{Mi} / \alpha_{ii}
\end{aligned}
\tag{6.10}
$$

Equation (6.10) says that the gain margin A_{Mi} of $g_{ii}(s)$ can be readily obtained as long as the parameters A'_{Mi} and α_{ii} are known. Therefore designing a PI rate-based router controller for a given gain margin A_{Mi} becomes a classical control problem [Ogat02]. From Equations (6.3), (6.4), (6.8) and (6.9), we can obtain the open-loop transfer function of the dominant loop i as

$$g_{ii}(s) = p_{ii}(s)c_{ii}(s) = \frac{\alpha_{ii}M_i e^{-s\bar{\tau}_i}}{s}\left(K_{Pi} + \frac{K_{Ii}}{s}\right) \quad . \tag{6.11}$$

From the definition of the gain margin [Ogat02], we can obtain

$$-\frac{\pi}{2} + \arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \frac{\pi}{2} - \omega_{pi}\bar{\tau}_i = -\pi \quad , \tag{6.12}$$

$$\frac{\alpha_{ii}M_i\sqrt{K_{Pi}^2\omega_{pi}^2+K_{Ii}^2}}{\omega_{pi}^2}=\frac{1}{A_{Mi}} \quad . \tag{6.13}$$

In order to compute the two unknown parameters, the proportional gain K_{Pi} and the integral gain K_{Ii} of the PI rate-based router controller, based on the known gain margin A_{Mi}, we first determine the relationship between the two unknown parameters. From our many simulation results (some of which shown in Figure 6.8 and Figure 6.9 later on), the relationship $K_{Pi}=(2+\sqrt{3})K_{Ii}/\omega_{pi}$ is a good choice to achieve good AQM performance, where ω_{pi} is the phase crossover frequency of the dominant loop i. Thus we can rewrite Equations (6.12) and (6.13) as

$$-\frac{\pi}{2}+\arctan\left(\frac{(2+\sqrt{3})K_{Ii}\omega_{pi}/\omega_{pi}}{K_{Ii}}\right)-\frac{\pi}{2}-\omega_{pi}\bar{\tau}_i=-\pi \quad , \tag{6.14}$$

and
$$\frac{2\sqrt{2+\sqrt{3}}\alpha_{ii}M_iK_{Ii}}{\omega_{pi}^2}=\frac{1}{A_{Mi}} \quad . \tag{6.15}$$

Using the relationship $K_{Pi}=(2+\sqrt{3})K_{Ii}/\omega_{pi}$ and Equation (6.10), we can obtain K_{Pi} and K_{Ii} from Equations (6.14) and (6.15) as

$$K_{Pi}=\frac{5\sqrt{2+\sqrt{3}}\pi/12}{2A'_{Mi}M_i\bar{\tau}_i} \quad , \tag{6.16}$$

$$K_{Ii}=\frac{(5\pi/12)^2}{2\sqrt{2+\sqrt{3}}A'_{Mi}M_i\bar{\tau}_i^2} \quad . \tag{6.17}$$

When the network parameter M_i and τ_i for the dominant loop i changes, PI rate-based router controller parameters K_{Pi} and K_{Ii} for the dominant loop i do not change, and this will force the gain margin to change. We can simplified Equations (6.12) and (6.13) as

$$\arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right)=\omega_{pi}\bar{\tau}_i \quad , \tag{6.18}$$

$$A'_{Mi}=\frac{\omega_{pi}^2}{M_i\sqrt{K_{Pi}^2\omega_{pi}^2+K_{Ii}^2}} \quad . \tag{6.19}$$

We will apply Equations (6.18) and (6.19) to monitor the gain margin of the dominant loop i, but we need to first consider Equation (6.18) in the following 2 different cases:

Case (1): when $K_{Pi}\omega_{pi}/K_{Ii}>1$ (that is, $\omega_{pi}>K_{Ii}/K_{Pi}$)

Based on the approximation that $\arctan(x)\approx\pi/2-\pi/(4x)$ when $x>1$, we can obtain

$$\pi / 2 - \pi K_{Ii} /(4K_{Pi}\omega_{pi}) = \omega_{pi}\tau_i \quad ,$$

$$\omega_{pi}^2 - \frac{\pi}{2\tau_i}\omega_{pi} + \frac{\pi K_{Ii}}{4K_{Pi}\tau_i} = 0 \quad . \tag{6.20}$$

The largest positive real root of Equation (6.20) will be the default value of phase crossover frequency ω_{pi} that will be used to calculate the gain margin of the dominant loop i by Equation (6.19).

Case (2): when $0<K_{Pi}\omega_{pi}/K_{Ii}\leq 1$ (that is, $\omega_{gi}<\omega_{pi}\leq K_{Ii}/K_{Pi}$)

In this case, we can obtain the gain crossover frequency ω_{gi} of the dominant loop i from the smallest positive real root of the following Equation (6.21). From the definition of the phase margin [Ogat02], we can obtain

$$\frac{\alpha_{ii}M_i\sqrt{K_{Pi}^2\omega_{gi}^2 + K_{Ii}^2}}{\omega_{gi}^2} = 1$$

which can be simplified as

$$\omega_{gi}^4 - \alpha_{ii}^2 M_i^2 K_{Pi}^2 \omega_{gi}^2 - \alpha_{ii}^2 M_i^2 K_{Ii}^2 = 0 \quad . \tag{6.21}$$

Then we can draw the graphs of $\arctan\left[\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right]$ and $\omega_{pi}\tau_i$ in the same plane in the range of $\omega_{gi}<\omega_{pi}\leq K_{Ii}/K_{Pi}$. The intersection of the two graphs is the solution of Equation (6.18), which is the phase crossover frequency ω_{pi} to be used to calculate the gain margin of the dominant loop i of the AQM control system by Equation (6.19). If these two graphs have no intersection, it means that the gain margin is less than 1, and the adaptive PI rate-based router controller need to self-tune.

In summary, when the network parameter (i.e., the number M_i of active controlled sources and the average round-trip time τ_i for the dominant loop i) changed so significantly that the gain margin of the dominant loop i falls outside its specified interval, the PI rate-based router controller for the dominant loop i needs to self-tune to give a new A_{Mi}' value for the dominant loop i. As a whole, we first use Equation (6.20) to obtain ω_{pi}. If we find $K_{Pi}\omega_{pi}/K_{Ii}<1$ or Equation (6.20) has no positive real roots, then we can obtain ω_{pi} using the method proposed by Case (2).

<u>Another method to monitor the change of the network traffic</u>

Since the above method to calculate the gain margin of the MIMO control system is complicated, it may not be easy for real-time implementation. Therefore, we would also suggest another approach. From Equation (6.19), we can see that the gain margin A'_{Mi} is a function of M_i and ω_p, while ω_p is a function of τ_i according to Equation (6.18). Based on Lemma 6.1, the gain margin decreases when the number M_i of the active controlled sources increases. Based on Lemma 6.2, we can also verify that the gain margin decreases when the average round trip time τ_i increases. Therefore, based on our practical experience from many simulation results (some of which shown in performance evaluation later on), we may want to choose a parameter $\beta_i = M_i\tau_i$ to monitor the number M_i of the active controlled sources and the average round trip time τ_i instead of the gain margin A'_{Mi}. When the network parameter M_i and τ_i for the dominant loop i changed so greatly that caused the monitoring parameter $|\beta_{inew} - \beta_{iold}| > 0.3\min\{\beta_{inew}, \beta_{iold}\}$ (that is $\beta_{inew} > 1.3\beta_{iold}$ or $\beta_{inew} < 0.7\beta_{iold}$, the adaptive PI rate controller for the dominant loop i will self-tune based on Equations (16) and (17) and update $\beta_{iold} = \beta_{inew}$. We will adopt this method to monitor the change of the network environment in our performance evaluation later on.

Lemma 6.1:

If the current number M'_i of active controlled sources is less than the previous number M_i (that is, $M'_i < M_i$) and multiloop PI rate controller is designed based on the previous number M_i for the dominant loop i, ***then*** the AQM control system with the current number M'_i will have more gain than that with the previous number M_i (that is, $A'_{Miv} > A'_{Mi}$), which can then always guarantee the stability of the AQM control system.

Proof:

From Equations (6.12), (6.13), (6.14), (6.15), (6.16) and (6.17), we obtain the PI rate controller $c_{ii}(s) = K_{Pi} + \frac{K_{Ii}}{s}$ for the dominant loop i of the MIMO AQM control system based on the respective number M_i. For the dominant loop i of MIMO AQM control system with M_i controlled sources, the open-loop transfer function is $\left(K_{Pi} + \frac{K_{Ii}}{s}\right)\frac{\alpha_{ii}M_i}{s}e^{-s\tau_i}$. Then from the definition of the gain margin [Ogat02], we can obtain

$$-\frac{\pi}{2} + \arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \frac{\pi}{2} - \omega_{pi}\tau_i = -\pi \quad ,$$

which can be simplified as

$$\arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \omega_{pi}\tau_i = 0 \quad , \tag{6.22}$$

$$\frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}} \frac{\alpha_{ii}M_i}{\omega_{pi}} \left| e^{-j\omega_{pi}\tau_i} \right| = \frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}} \frac{\alpha_{ii}M_i}{\omega_{pi}} = \frac{1}{A_{Mi}} \quad . \tag{6.23}$$

Similarly, for the dominant loop i of MIMO AQM control system with $M_i^{'}$ controlled sources, we can obtain

$$\arctan\left(\frac{K_{Pi}\omega_{pi}^{'}}{K_{Ii}}\right) - \omega_{pi}^{'}\tau_i = 0 \quad , \tag{6.24}$$

$$\frac{\sqrt{K_{Pi}^2\omega_{pi}^{'2} + K_{Ii}^2}}{\omega_{pi}^{'}} \frac{\alpha_{ii}M_i^{'}}{\omega_{pi}^{'}} \left| e^{-j\omega_{pi}^{'}\tau_i} \right| = \frac{\sqrt{K_{Pi}^2\omega_{pi}^{'2} + K_{Ii}^2}}{\omega_{pi}^{'}} \frac{\alpha_{ii}M_i^{'}}{\omega_{pi}^{'}} = \frac{1}{A_{Miv}} \quad . \tag{6.25}$$

From Equations (6.22) and (6.24), we can determine that $\omega_{pi}^{'} = \omega_{pi}$. Therefore, based on Equations (6.23) and (6.25), we can obtain

$$A_{Miv} / A_{Mi} = M_i / M_i^{'} > 1 \quad ,$$

that is,

$$A_{Miv} > A_{Mi} \quad . \tag{6.26}$$

Then for the dominant loop i of the MIMO AQM control system with M_i controlled sources, we rewrite Equation (6.11) as

$$A_{Mi}^{'} = \alpha_{ii} A_{Mi} \quad . \tag{6.27}$$

Similarly, for the dominant loop i of MIMO AQM control system with $M_i^{'}$ controlled sources, we can also obtain

$$A_{Miv}^{'} = \alpha_{ii} A_{Miv} \quad . \tag{6.28}$$

Therefore by combining the Equations (6.26), (6.27) and (6.28), we can get

$$A_{Miv}^{'} = \alpha_{ii} A_{Miv} > \alpha_{ii} A_{Mi} = A_{Mi}^{'} \quad ,$$

that is, $A_{Miv}^{'} > A_{Mi}^{'}$.

Lemma 6.2:

If the current average round trip time $\bar{\tau}_i^{'}$ is shorter than the previous average round trip time $\bar{\tau}_i$ (that is, $\bar{\tau}_i^{'} \le \bar{\tau}_i$) and the multi-loop PI rate controller is designed based on the previous average round trip time τ_i, ***then*** the AQM control system with the current average round trip time $\tau_i^{'}$ will have more gain than that with the previous average round trip time τ_i (that is, $A'_{Miv} > A'_{Mi}$), which can always guarantee the stability of the AQM control system.

Proof:

For the dominant loop i of MIMO AQM control system with the average round trip time τ_i, the open-loop transfer function is $\left(K_{Pi} + \frac{K_{Ii}}{s}\right)\frac{\alpha_{ii}M_i}{s}e^{-s\tau_i}$. As discussed in Lemma 1, from the definition of the gain margin [Ogat02], we can obtain

$$\arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \omega_{pi}\tau_i = 0 \quad , \tag{6.22}$$

$$\frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}}\frac{\alpha_{ii}M_i}{\omega_{pi}}\left|e^{-j\omega_{pi}\tau_i}\right| = \frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}}\frac{\alpha_{ii}M_i}{\omega_{pi}} = \frac{1}{A_{Mi}} \quad . \tag{6.23}$$

From the definition of the phase margin [Ogat02], we can obtain

$$\frac{\sqrt{K_{Pi}^2\omega_{gi}^2 + K_{Ii}^2}}{\omega_{gi}}\frac{\alpha_{ii}M_i}{\omega_{gi}}\left|e^{-j\omega_{gi}\tau_i}\right| = \frac{\sqrt{K_{Pi}^2\omega_{gi}^2 + K_{Ii}^2}}{\omega_{gi}}\frac{\alpha_{ii}M_i}{\omega_{gi}} = 1 \quad , \tag{6.29}$$

where ω_{gi} and ω_{pi} are gain crossover frequency and phase crossover frequency of the dominant loop i of MIMO AQM control system respectively. Similarly, for the dominant loop i of MIMO AQM control system with the average round trip time τ'_i, we can obtain

$$\arctan\left(\frac{K_{Pi}\omega'_{pi}}{K_{Ii}}\right) - \omega'_{pi}\tau'_i = 0 \quad , \tag{6.30}$$

$$\frac{\sqrt{K_{Pi}^2\omega'^2_{pi} + K_{Ii}^2}}{\omega'_{pi}}\frac{\alpha_{ii}M_i}{\omega'_{pi}}\left|e^{-j\omega'_{pi}\tau'_i}\right| = \frac{\sqrt{K_{Pi}^2\omega'^2_{pi} + K_{Ii}^2}}{\omega'_{pi}}\frac{\alpha_{ii}M_i}{\omega'_{pi}} = \frac{1}{A_{Miv}} \quad , \tag{6.31}$$

and

$$\frac{\sqrt{K_{Pi}^2\omega'^2_{gi} + K_{Ii}^2}}{\omega'_{gi}}\frac{\alpha_{ii}M_i}{\omega'_{gi}}\left|e^{-j\omega'_{gi}\tau'_i}\right| = \frac{\sqrt{K_{Pi}^2\omega'^2_{gi} + K_{Ii}^2}}{\omega'_{gi}}\frac{\alpha_{ii}M_i}{\omega'_{gi}} = 1 \quad . \tag{6.32}$$

From Equations (6.29) and (6.32), we can then determine that $\omega'_{gi} = \omega_{gi}$.

By drawing the Nyquist diagram of the transfer function $g_{ii}(s) = \left(K_{Pi} + \frac{K_{Ii}}{s}\right)\frac{\alpha_{ii}M_i}{s}e^{-s\tau_i}$, we can find the following useful Relationship 6.1 analytically.

<u>Relationship 6.1</u>:

If there exists the frequencies $\omega_{gi} \le \omega_1 < \omega_2 \le \omega_{pi}$, ***then*** we will have

$-\pi = \arg(g_{ii}(j\omega_{pi})) \le \arg(g_{ii}(j\omega_2)) < \arg(g_{ii}(j\omega_1)) \le \arg(g_{ii}(j\omega_{gi})) = -\pi + \varphi_{Mi}$ and

$\frac{1}{A_{Mi}} = |g_{ii}(j\omega_{pi})| \le |g_{ii}(j\omega_2)| < |g_{ii}(j\omega_1)| \le |g_{ii}(j\omega_{gi})| = 1$, and vice versa. This will be shown in Figure 6.3 and Figure 6.4 later on.

Let ω_{gi} and ω_{pi} be the gain crossover frequency and the phase crossover frequency of the dominant loop i of MIMO AQM control system with the average round trip time $\bar{\tau}_i$. In order to find the relationship between the gain margins A_{Miv} and A_{Mi}, we use $\omega_1 = \omega_{pi} > \omega_{gi} = \omega_{gi}^{'}$ and $\omega_2 = \omega_{pi}^{'}$ as two different frequency points. Then considering Equations (6.27) and (6.28), we have

$$
\begin{aligned}
&\arg(g_{ii}(j\omega_1)) - \arg(g_{ii}(j\omega_2)) = \arg(g_{ii}(j\omega_{pi})) - \arg(g_{ii}(j\omega_{pi}^{'})) \\
&= \left[-\pi + \arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \omega_{pi}\bar{\tau}_i^{'}\right] - \left[-\pi + \arctan\left(\frac{K_{Pi}\omega_{pi}^{'}}{K_{Ii}}\right) - \omega_{pi}^{'}\bar{\tau}_i^{'}\right] \\
&= \arctan\left(\frac{K_{Pi}\omega_{pi}}{K_{Ii}}\right) - \omega_{pi}\bar{\tau}_i^{'} = \omega_{pi}\bar{\tau}_i - \omega_{pi}\bar{\tau}_i^{'} > 0
\end{aligned}
.
$$

Then based on the Relationship 6.1, we can show that $\omega_{gi}^{'} = \omega_{gi} < \omega_1 = \omega_{pi} < \omega_2 = \omega_{pi}^{'}$. Applying the Relationship 6.1 again, we can obtain $|g_{ii}(j\omega_1)| = |g_{ii}(j\omega_{pi})| > |g_{ii}(j\omega_2)| = |g_{ii}(j\omega_{pi}^{'})|$, which can be expressed as $|g_{ii}(j\omega_{pi})| = \frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}} \frac{\alpha_{ii}M_i}{\omega_{pi}} \left|e^{-j\omega_{pi}\bar{\tau}_i}\right| = \frac{\sqrt{K_{Pi}^2\omega_{pi}^2 + K_{Ii}^2}}{\omega_{pi}} \frac{\alpha_{ii}M_i}{\omega_{pi}} = \frac{1}{A_{Mi}} > |g_{ii}(j\omega_{pi}^{'})| = \frac{1}{A_{Miv}}$, that is, $A_{Miv} > A_{Mi}$. By considering Equations (6.27) and (6.28), we can get $A_{Miv}^{'} > A_{Mi}^{'}$.

6.2.2 Fairness among Rate-Based Router Controlled Sources

Most congestion control algorithms show that the fairness can be achieved when the system reaches its steady state as $t \to \infty$ [KeMa98]. We define fairness here to be the condition that all the controlled sources routing through the same input port have the same steady transmission rate. From the multi-loop PI rate control algorithm in Equations (6.2) and (6.5), we can see that every controlled source node $j|i$ always try to transmit the packets into the network at the maximum allowed rate as advertised by the router. When the queue length of the output ports in the AQM router reaches the steady state, the fairness among the controlled source nodes is considered to be achieved, as shown in our performance evaluation of Figure 6.5, Figure 6.6, Figure 6.13 and Figure 6.14 later on.

6.2.3 Implementation of Rate-Based Router Control Protocol for MIMO IP router

We use the window-based calculation to convert the advertised source transmission rate into the advertised window size for the controlled source nodes in the IP networks, which has been well accepted in TCP/IP networks e.g., [AwOu02, KaHa02, LoAn05]. The advertised window size $win'_{ij}(t)$ for the controlled source node $j|i$ is computed as

$$win'_{ij}(t) = \tau_{ij} \cdot q'_{ij}(t) \quad . \tag{6.33}$$

Then from the Equation (6.5), the window size $win_{ij}(t)$ for the controlled source node $j|i$ is given as

$$win_{ij}(t) = \tau_{ij} \cdot q_{ij}(t) = \tau_{ij} \cdot q'_{ij}(t - \tau_{bij}) = win'_{ij}(t - \tau_{bij}) \quad , \tag{6.34}$$

where the window size will be restricted between 1 and the maximum window size $win_{ij}^{\max}$ allowed for the controlled source node $j|i$, i.e., $1 \le win_{ij}(t) \le win_{ji}^{\max}$.

From the description in Section 6.2.1, one can see the feature of our control algorithm is that we only need to know one important information from the controlled sources: the round trip time τ_{ij} between the source $j|i$, and its destination from which the average round trip time $\bar{\tau}_i$ is decided. This is one major difference from most heuristic AQM algorithms for TCP traffic. To implement the PI rate-based control algorithm in the IP networks, we need to define two fields in the IP packet headers:

(1) the round trip time τ_{ij}: this can be estimated by the source using the method proposed by [Stev94];

(2) the advertised window size $win'_{ij}(t)$: the original value is set to be the maximum window size $win_{ij}^{\max}$.

Normally an IP packet sent by the source $j|i$ will be routed through a series of AQM routers before it reaches its destinations. In order to prevent all the routers from congestion, we will compare the current advertised window size with the previous advertised window size in the IP packet and then record the smaller value in the header of IP packet.

After the IP packet reaches its destination successfully, the advertised window size will be extracted from the header of the IP packet and then integrated into its ACK packet. When the source $j|i$ receives the ACK packet finally, it extracts the advertised window size $win'_{ij}(t)$ from the ACK packet, and use it to set the current window size $win_{ij}(t)$.

In the real-time implementation, we use specific address queue to store the source addresses of the incoming IP packets from the controlled sources: (1) the source address of each IP packet under process will enter as a "packet" into the address queue per sampling time. This can be ignored if the same source address is found; (2) the address queue will drop "packet" at the head every average round trip time. This can evict the inactive controlled source address "packet" from the address queue. Therefore, the address queue size can approximate the number M_i of the controlled source nodes. Furthermore, the advertised window size $win'_{ij}(t)$ will be an integer and larger than 1. To simplify the calculation, we compute the average round trip time $\bar{\tau}_i$ by the averaging filter method in RED algorithm [FlJa93], i.e.,

$$\bar{\tau}_i(t) = (1 - w_{\tau i})\bar{\tau}_i(t-1) + w_{\tau i}\tau_{ij} \qquad j = 1, \cdots, M_i \quad . \tag{6.35}$$

where the averaging weight $w_{\tau i}$ is suggested to be $w_{\tau i} = 1/M_i$, based on the following Relationship 6.2.

Relationship 6.2:

If (M_i-1) controlled source nodes transmit the IP packets to their destinations through the input port i of the router, then the average round trip time $\bar{\tau}'_i = \sum_{j=1}^{M_i - 1}\tau_{ij} / (M_i - 1)$. The controlled source M_i starts to transmit the IP packets routed through the input port i of the router, so the new average round trip time will be

$$\bar{\tau}_i = \sum_{j=1}^{M_i}\tau_{ij} / M_i = \left((M_i - 1)\bar{\tau}'_i + \tau_{iM_i}\right) / M_i = (1 - 1/M_i)\bar{\tau}'_i + \tau_{iM_i} / M_i \,.$$

6.3 Adaptive Multiloop PI Rate-Based Router Controller Algorithm

We can now present the details of adaptive multiloop PI rate-based router control algorithm for the MIMO router as follows:

Router behavior:

Step 1: Estimate the number M_i of the controlled sources $j|i$ based on the address queue size, extract the round trip time τ_{ij} from the header of the IP packet that flows into the input port i of the router from the source node $j|i$, and then calculate the average round trip time $\bar{\tau}_i$ based on Equation (6.35);

Step 2: Calculate the initial value $\beta_{i0} = M_i / \tau_i$ for monitoring the traffic changes.

Step 3: From Equations (6.16) and (6.17), we can obtain the proportional gain K_{Pi} and integral gain K_{Ii} of PI rate-based router controller for the dominant loop i.

Step 4: Apply the adaptive multiloop PI rate-based router controller to calculate the advertised transmission rate q'_{ij_new} based on the instantaneous queue length $x_i(t)$ of the output port i of MIMO IP router, according to Equation (6.2).

Step 5: Apply Equation (6.33) to calculate the advertised window size win'_{ij_new}.

Step 6: Compare the current advertised window size win'_{ij_new} with the advertised window size field win'_{ij_old} of the IP header.

(6a) Set $win'_{ij\_old} = win'_{ij\_new}$ if we find that $win'_{ij\_new} < win'_{ij\_old}$;

(6b) Otherwise, the advertised window size field win'_{ij_old} remains unchanged.

Real-time monitoring for each dominant loop i ($i=1,2,\ldots N$):

Step 7: Calculate the current monitoring parameter $\beta_i = M_i\tau_i$.

(7a) If we find that $|\beta_i - \beta_{i0}| > 0.3\min\{\beta_{i0}, \beta_i\}$, then update $\beta_{i0} = \beta_i$. Go to Step 3.

(7b) If we find that $|\beta_i - \beta_{i0}| \le 0.3\min\{\beta_{i0}, \beta_i\}$, PI rate-based router controller would not self-tune for the dominant loop i, go to Step 4.

Destination behavior:

Extract the advertised window size win'_{ij} from the header of the IP packet, and then integrate them into the ACK packets when IP packet arrives in the destination.

Source behavior:

Extract the advertised window size win'_{ij} from the ACK packet and then regulate the current source window size after the controlled source $j|i$ receives ACK packet.

6.3.1 An Example

To illustrate the adaptive multiloop PI rate-based router controller design, we will consider a router with 2 input ports and 2 output ports. The forward time delays τ_{fij}, the feedback time delays τ_{bij} and the round trip time τ_{ij} for the controlled source node $j|i$ (j=1,2,3,4) that transmits the packets into the input port i (i=1,2) are listed in Table 6.1. The traffic distribution ratio of the first input port is $\alpha_{11}=0.8$ and $\alpha_{21}=0.2$, while the traffic distribution ratio of second input port is $\alpha_{22}=0.8$ and $\alpha_{12}=0.2$. For both dominant loops of the MIMO AQM control system, the gain margin $A'_{Mi}(i=1,2)$ is specified as 3.

Table 6.1: The Maximum Transmission Rates and Time Delays for the Controlled Source Nodes that Transmit the Packets into the Input Ports 1 and 2 of Router 1

Rate & / Node $j\|i$	Forward Delay τ_{fij} (ms)	Feedback Delay τ_{bij} (ms)	Round-Trip Time τ_{ij} (ms)
Source Node *1*|*1*	200	400	600
Source Node *2*|*1*	100	300	400
Source Node *3*|*1*	200	400	600
Source Node *4*|*1*	300	500	800
Source Node *1*|*2*	150	350	500
Source Node *2*|*2*	100	300	400
Source Node *3*|*2*	100	350	450
Source Node *4*|*2*	200	450	650

We assume that two controlled source nodes *1*|*1* and 2|*1* starts to transmit the packets to the input port *1*, while another two controlled source nodes *1*|*2* and 2|*2* starts to transmit the packets to the input port *2* at the beginning of the experiment. Therefore we can obtain the average round-trip times for two dominant control loops as $\tau_1=(\tau_{11}+\tau_{12})/2=500\text{msec}$ and $\tau_2=(\tau_{21}+\tau_{22})/2=450\text{msec}$ respectively. Therefore we can obtain the transfer function matrix of the MIMO router model as

$$P(s)=\begin{bmatrix}\dfrac{0.8*2e^{-0.5s}}{s} & \dfrac{0.2*3e^{-0.45s}}{s}\\ \dfrac{0.2*2e^{-0.5s}}{s} & \dfrac{0.8*3e^{-0.45s}}{s}\end{bmatrix}=\begin{bmatrix}\dfrac{1.6e^{-0.5s}}{s} & \dfrac{0.6e^{-0.45s}}{s}\\ \dfrac{0.4e^{-0.5s}}{s} & \dfrac{2.4e^{-0.45s}}{s}\end{bmatrix}. \tag{6.36}$$

From Equations (6.16) and (6.17), the diagonal multiloop PI rate-based router controller is given as

$$C(s)=\begin{bmatrix}0.4215+\dfrac{0.2957}{s} & 0\\ 0 & 0.3122+\dfrac{0.2433}{s}\end{bmatrix}. \tag{6.37}$$

The Nyquist diagrams of $g_{ii}(j\omega)$ with Gershgorin circle are shown in Figure 6.3 and Figure 6.4 respectively. It is shown that the gain margins $A'_{Mi}(i=1,2)$ of both domain loops are 3 as desired.

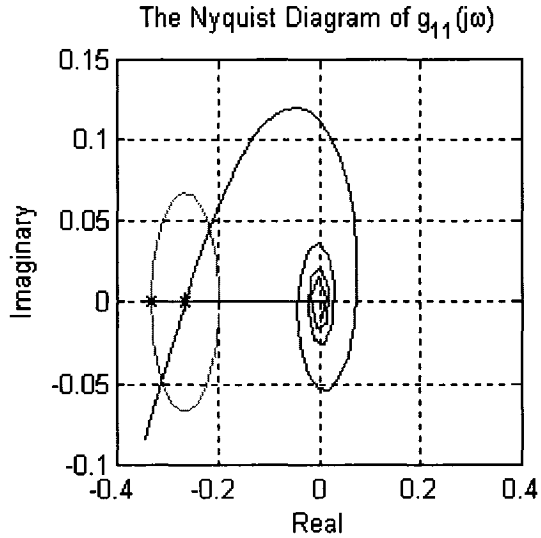

Figure 6.3: Nyquist Diagram of $g_{11}(j\omega)$ with Gershgorin Circle at the Phase Crossover Frequency

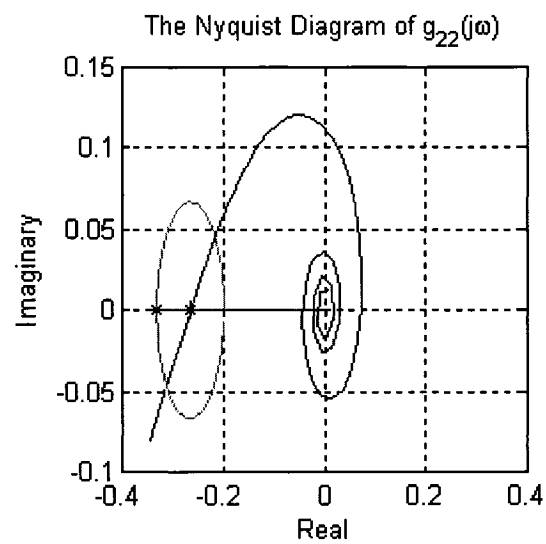

Figure 6.4: Nyquist Diagram of $g_{22}(j\omega)$ with Gershgorin Circle at the Phase Crossover Frequency

6.4 Performance Evaluation of Adaptive Multiloop PI Rate-Based Router Controller

We have verified our adaptive multiloop PI rate-based router controller using Matlab simulations based on the network topology shown in Figure 2.7. There are 2 Routers in the network. We make performance evaluation in the case of the varying uncontrolled traffic (that is, M_i changes) and the constant uncontrolled traffic (that is, M_i does not change) respectively and observe the transient behavior of the most congested AQM router. During all the following simulations, all the IP packets have the same size of *1024* bytes. The link bandwidth (i.e., service rate) of every output port is T3 (*44,736,00*bps, i.e., *5,461*packets/sec). Every MIMO router has two input ports and two output ports. For both dominant loops of the MIMO IP router, the gain margin $A'_{Mi}(i=1,2)$ is specified as 3. The buffer size of every output port is *2000* packets based on based on the bandwidth-delay product [HoMi01a]. The target buffer occupancy of the every output port is specified as *200* packets. The maximum window size allowed for all the controlled sources is specified as *3000* packets. The sampling time for PI rate controller is chosen as 1ms.

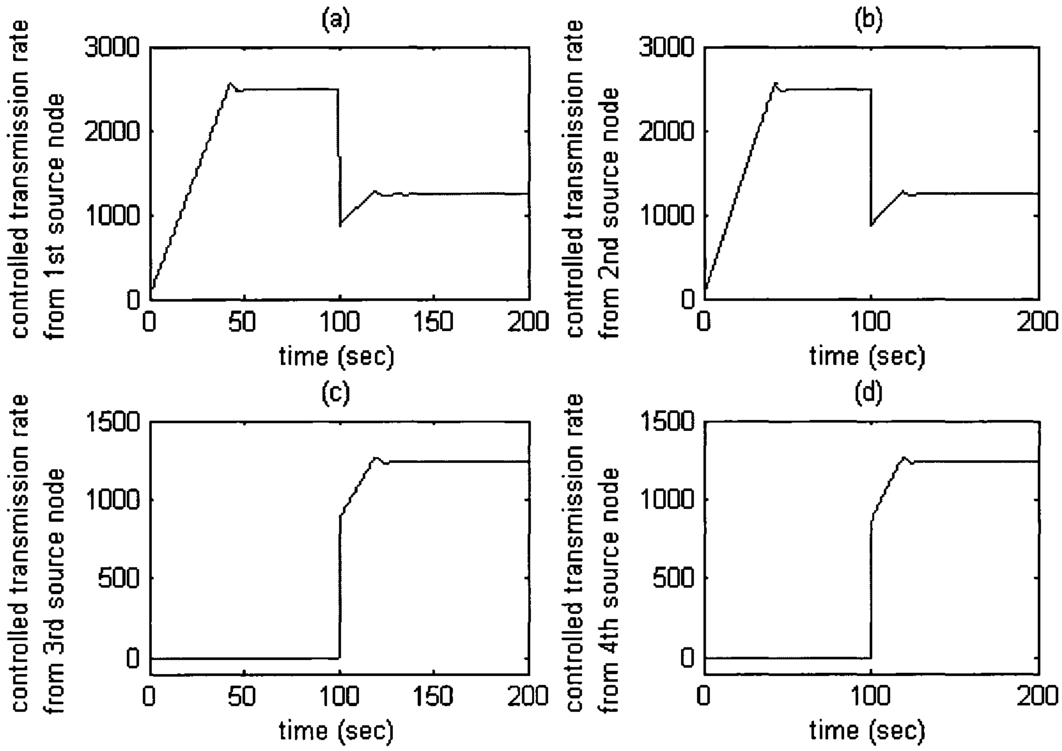

Figure 6.5: Instantaneous Transmission Rates (packets/sec) of 4 controlled Sources for 1st Input Port

6.4.1 Case with Varying Controlled Traffic

Under the case of varying uncontrolled traffic, we assume that the MIMO Router 1 is always more congested than the MIMO Router 2 any time. In this case study, there are 8 controlled source nodes that transmit the packets into the network, whose parameters are shown in Table 6.1. The uncontrolled guaranteed traffic flows into the input port *i* (*i=1,2*) at the same constant rate of *500* packets/sec. In the beginning of the experiment, four controlled source nodes $j|i$

($j=1,2$ with $i=1,2$) starts to transmit the packets into the input port i ($i=1,2$) respectively, as described in the example of Section 6.3.1. After time $t=100$s, another two controlled source nodes $j|i$ ($j=3,4$ with $i=1,2$) starts to send the packets to the AQM router, that is, every input port i of the router receives the packets from the 4 controlled source nodes respectively.

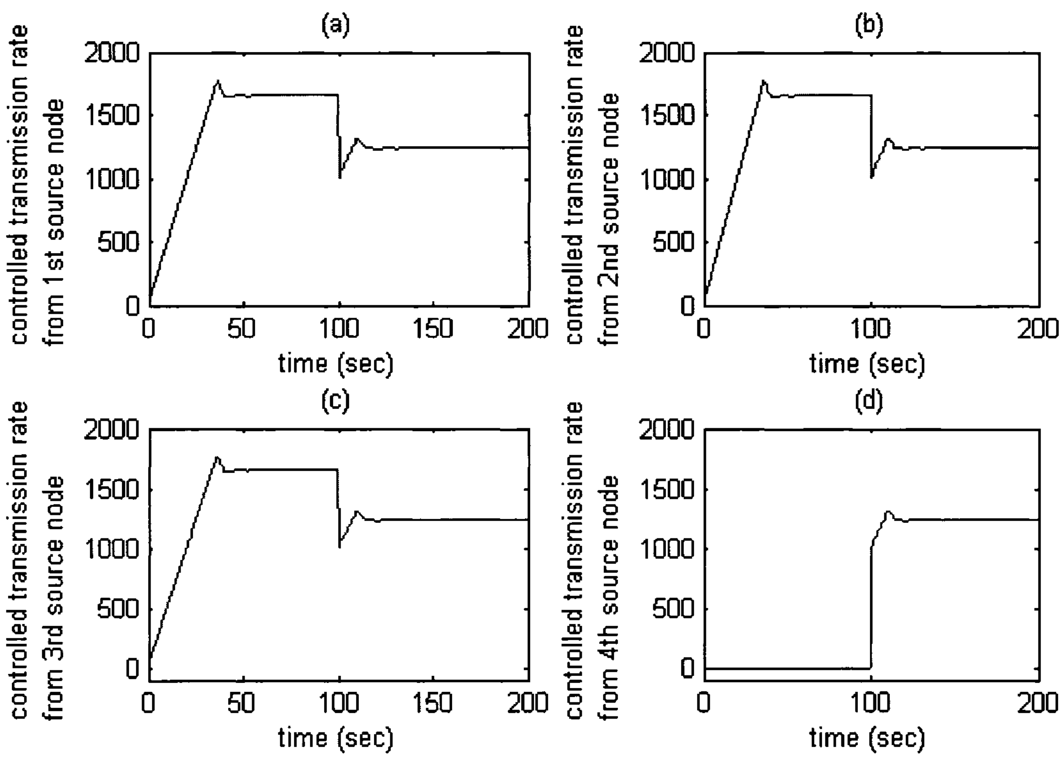

Figure 6.6: Instantaneous Transmission Rates (packets/sec) of 4 controlled Sources for 2nd Input Port

The instantaneous transmission rates of the 8 controlled source nodes for the AQM router are shown in Figure 6.5 and Figure 6.6. The total transmission rates for the input ports and the instantaneous queue lengths of the output ports in the router are shown in Figure 6.7. At time $t=0$s, two controlled source nodes $j|i$ ($j=1,2$ with $i=1,2$) started to send the packets to the AQM router. Source $1|1$ increases its transmission rate and approaches a steady value of around 2480 packets/sec. At time $t=100$s, more controlled source nodes are added in the network, the average round trip times for both dominant loops become longer, both monitoring parameters fall outside their specified intervals, so adaptive PI rate-based router controllers located in both output ports self-tune. Source $1|1$ decreases its transmission rate and approaches to a new steady state. Source $3|1$ increases its transmission rate and reaches a steady state. With the adaptive PI rate-based router controllers applied, the queue lengths in both output ports have small fluctuations. In summary, the steady-state value of the queue length in both output ports reaches the specified target buffer occupancy (*200* packets) in the two different periods and avoids the buffer overflow, which prevents the congestion in the network.

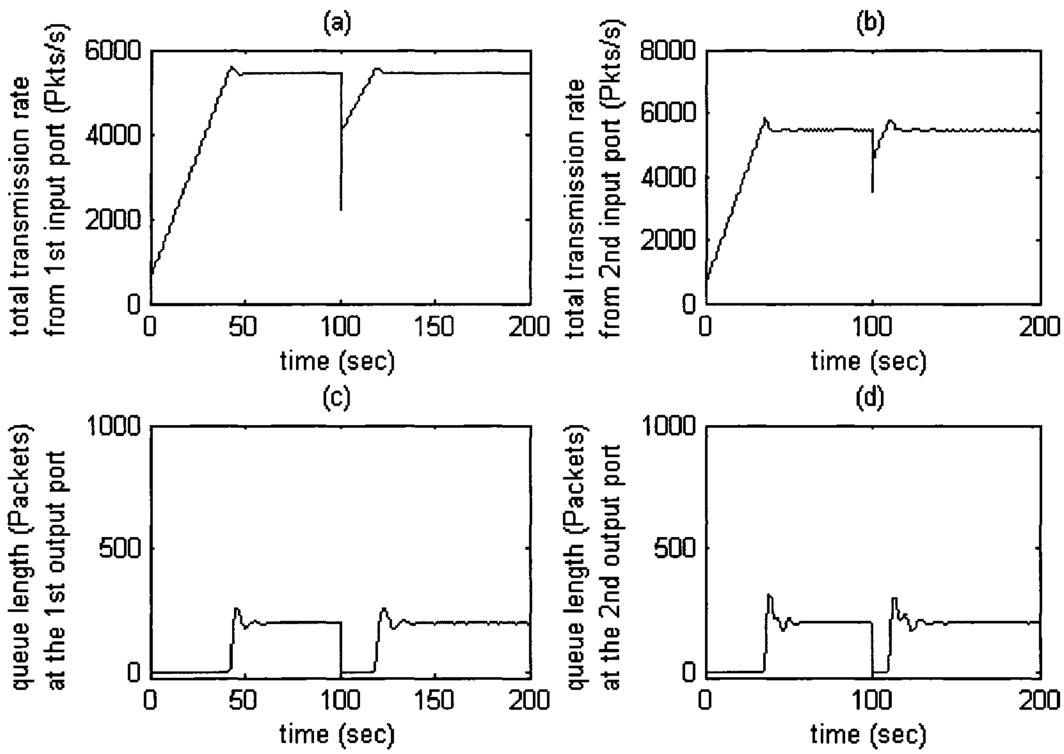

Figure 6.7: Transient Behavior of 2x2 AQM Router
(a) Total transmission rate for 1st input port
(b) Total transmission rate for 2nd input port
(c) Instantaneous queue length of the first output port
(d) Instantaneous queue length of the second output port

6.4.1.1 Impact of Gain Margin on AQM Performance of MIMO IP Router

We will study the impact of the different gain margin on the AQM performance of most congested MIMO router (i.e., Router 1 in Figure 2.7). We specify different gain margin to design the PI rate-based router controller for the time $t=0$s to *100*s, and then observe the transient behavior (e.g., queue length) of the MIMO router.

Figure 6.8 and Figure 6.9 show the queue length for the two output ports upon the different gain margins. As evident from Figure 6.8 and Figure 6.9, when the gain margin increases, settling time increases, i.e., queue length takes more time to reach the steady state. If the gain margin is too large (e.g., 6), the settling time to reach the target buffer occupancy (200 packets) is extremely long (e.g., the queue length with gain margin of 6 still can not reach steady state at

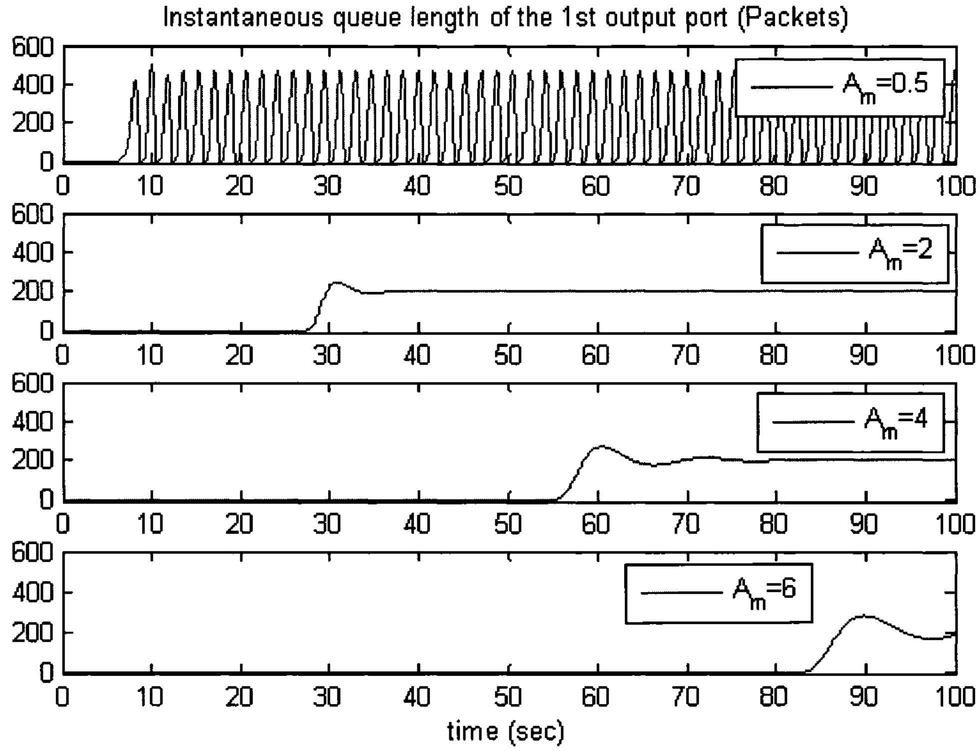

Figure 6.8: Instantaneous Queue Length of the 1[st] Output Port upon Different Gain Margin

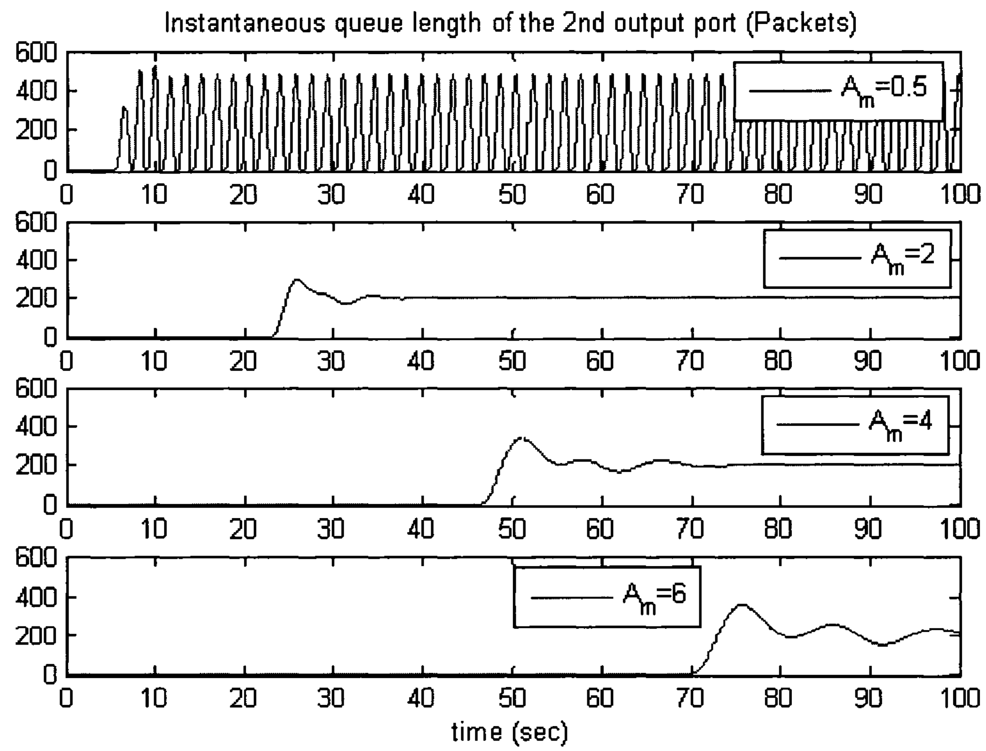

Figure 6.9: Instantaneous Queue Length of the 2[nd] Output Port upon Different Gain Margin

the time $t=100$s). When the gain margin less than 1 (e.g., 0.5), the queue length becomes unstable and cannot approach the target buffer occupancy (*200* packets), as shown in Figure 6.8 and Figure 6.9, which also corresponds to the Nyquist stability theorem [Ogat02].

Table 6.2: The Maximum Transmission Rates and Time Delays for the Controlled Source Nodes that Transmit the Packets into the Input Ports 1 and 2 of Router 2

Rate & / Node $j\|i$	Forward Delay τ_{fij} (ms)	Feedback Delay τ_{bij} (ms)	Round-Trip Time τ_{ij} (ms)
Source Node *1*|*1*	300	300	600
Source Node *2*|*1*	200	200	400
Source Node *3*|*1*	300	300	600
Source Node *4*|*1*	400	400	800
Source Node *1*|*2*	250	250	500
Source Node *2*|*2*	200	200	400
Source Node *3*|*2*	300	300	600
Source Node *4*|*2*	250	250	500

6.4.2 Case with Constant Controlled Traffic

In this case study, we have multiple 2x2 routers where two routers are most congested routers in different period. We assume that Router 2 has the same structure (i.e., traffic distribution ratio) as Router 1, as shown in the example of Section 6.3.1. The uncontrolled transmission rates for the input ports of both routers fluctuate between 10% and 40% of the service rate of the respective output links, as shown in Figure 6.10 and Figure 6.11. The 8 controlled source nodes $j|i$ ($j=1,2,3,4$ with $i=1,2$) start to transmit the packets in the beginning of the experiment. Based on Table 6.1 and Table 6.2, the average round trip times for 2 dominant loops can also obtained as $\tau_1 = 600\text{msec}$ and $\tau_2 = 500\text{msec}$ respectively. So we can get the same transfer function matrix for MIMO Router 1 and Router 2, i.e.,

$$P(s) = \begin{bmatrix} \dfrac{0.8*4e^{-0.6s}}{s} & \dfrac{0.2*4e^{-0.5s}}{s} \\ \dfrac{0.2*4e^{-0.6s}}{s} & \dfrac{0.8*4e^{-0.5s}}{s} \end{bmatrix} = \begin{bmatrix} \dfrac{3.2e^{-0.6s}}{s} & \dfrac{0.8e^{-0.5s}}{s} \\ \dfrac{0.8e^{-0.6s}}{s} & \dfrac{3.2e^{-0.5s}}{s} \end{bmatrix}. \quad (6.38)$$

From Equations (6.16) and (6.17), the diagonal multiloop PI rate-based router controller was given as

$$C(s) = \begin{bmatrix} 0.1756 + \frac{0.1027}{s} & 0 \\ 0 & 0.2107 + \frac{0.1478}{s} \end{bmatrix} \quad (6.39)$$

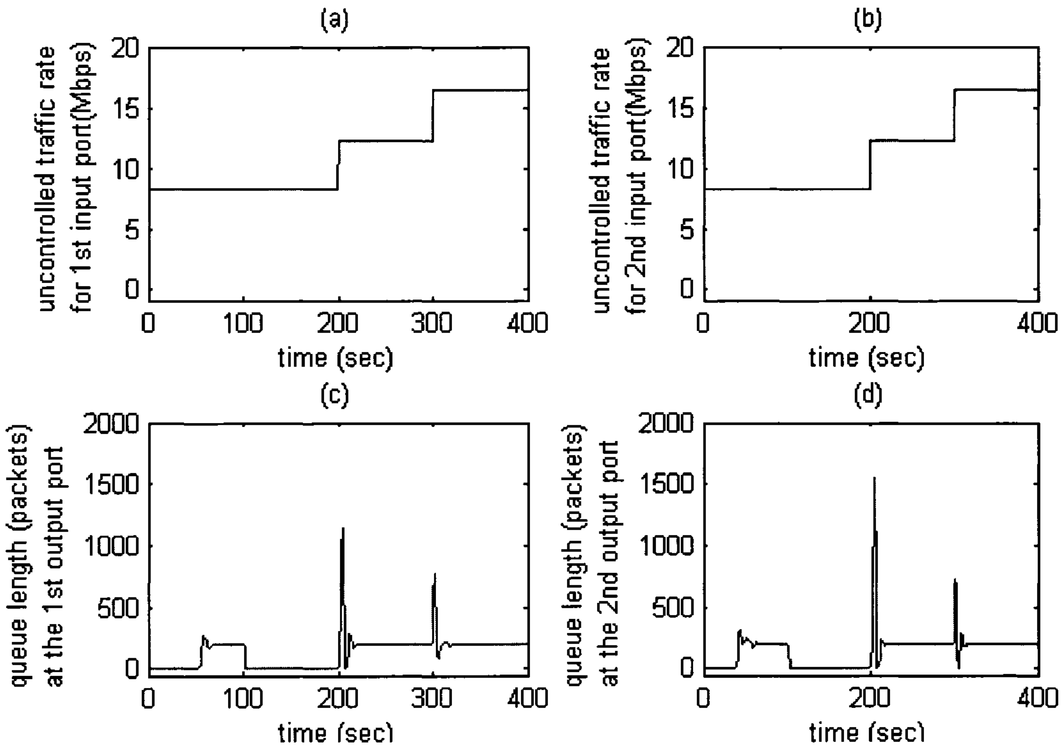

Figure 6.10: Transient Behavior of Router 1

(a) Uncontrolled transmission rate for 1st input port
(b) Uncontrolled transmission rate for 2nd input port
(c) Queue length of 1st input port
(d) Queue length of 2nd input port

Figure 6.10(a), Figure 6.10(b), Figure 6.11(a) and Figure 6.11(b) show the instantaneous uncontrolled transmission rates for the guaranteed service traffic flowing into the input ports of Router 1 and Router 2 respectively. It can be seen that the guarantee service traffic flows into different input ports of the MIMO router with different transmission rate at different time for different duration. The instantaneous queue length of the output ports in the MIMO Router 1 and Router 2 are shown in Figure 6.10(c), Figure 6.10(d), Figure 6.11(c) and Figure 6.11(d). It can be seen that the queue length of the output ports can adapt to the fluctuation of the uncontrolled traffic flowing into the input ports very well. Furthermore, there is no queue in the less congested router when the AQM control system reaches its steady state, while the queue length in the most congested router reaches the target buffer occupancy.

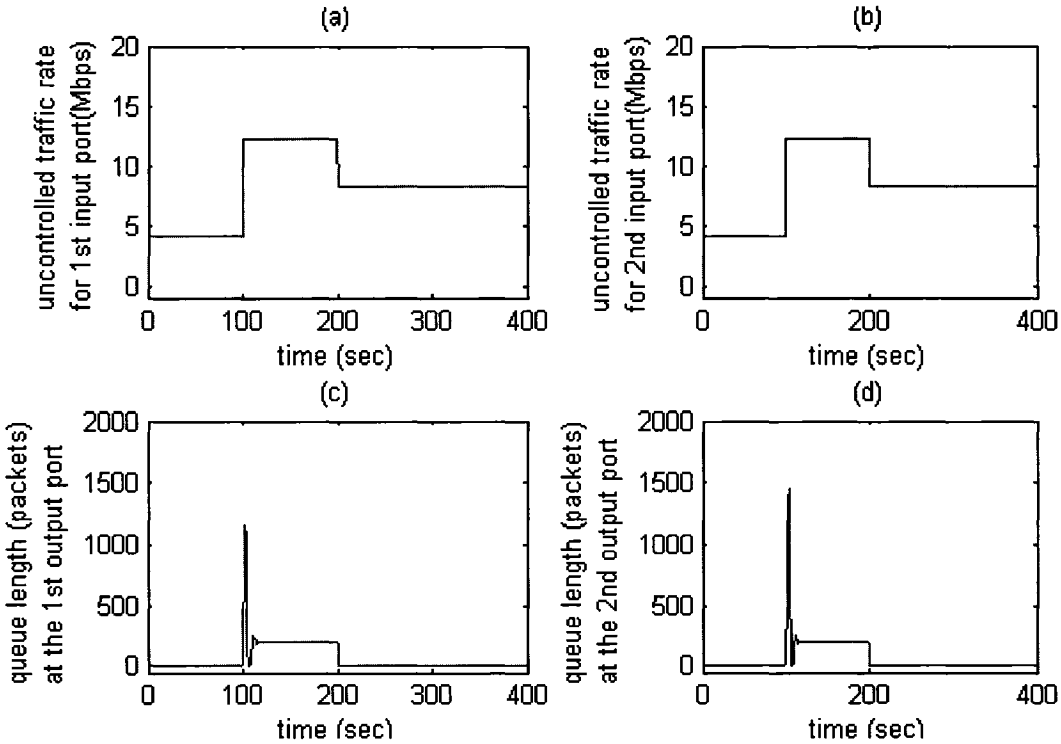

Figure 6.11: Transient Behavior of Router 2

(a) Uncontrolled transmission rate for 1st input port
(b) Uncontrolled transmission rate for 2nd input port
(c) Queue length of 1st input port
(d) Queue length of 2nd input port

The instantaneous advertised transmission rates for the controlled source nodes $j|i$ ($j=1,2,3,4$ with $i=1,2$) are shown in Figure 6.12. The instantaneous transmission rates of the 8 controlled source nodes $j|i$ ($j=1,2,3,4$ with $i=1,2$) for the best-effort service traffic flowing into the input ports i are shown in Figure 6.13, Figure 6.14 respectively. It is shown that the controlled source nodes always try to send the packets at the maximum allowed rate to catch the rest bandwidth left over from the guarantee service traffic. Moreover, it can be seen that the advertised transmission rate from the less congested router will reach the controlled source through IP packets and ACK packets. This prevents all the routers from congestion.

Between time t=*0s* and t=10*0s*, both uncontrolled transmission rates for the input port i (i=1,2) of Router 1 are *1000* packets/sec, and the queue length of the output port i (i=1,2) reaches the steady state value (*200* packets) at the time t=*76*s and *70*s respectively. Both uncontrolled transmission rates for the input port i (i=1,2) of Router 2 were 50*0*packets/sec, and the queue of the output port i ($i=1,2$) are empty. This corresponds to the following facts: the advertised source

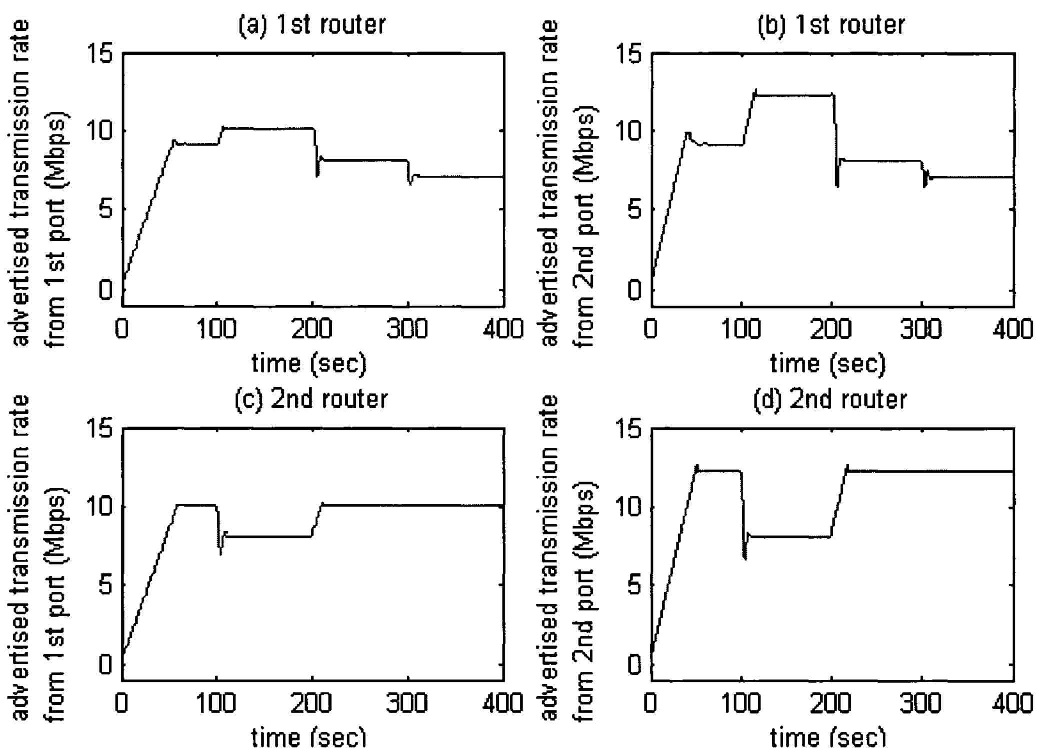

Figure 6.12: Instantaneous Advertised Controlled Transmission Rate
(a) from 1st port of 1st router
(b) from 2nd port of 1st router
(c) from 1st port of 2nd router
(d) from 2nd port of 2nd router

transmission rate (i.e., advertised window size) by the Router 1 is less than that from Router 2, which will be informed to the sources finally. This means that the Router 1 is more congested than Router 2, and our rate-based control protocol prevents both routers from congestion. At time *t=100*s, both the uncontrolled transmission rates for the input port *i* (*i*=1,2) of the Router 2 increase to *1500* packets/sec, while both uncontrolled transmission rates for the input port *i* (*i*=1,2) of the Router 1 remains unchanged, which hints that the Router 2 becomes more congested than Router 1. Consequently, the advertised source transmission rate by the Router 2 is less than that from Router 1, which will be informed to the sources finally. All the transmission rates of the source controlled nodes reduce, e.g., the source *1|1* decreases from *9.14*Mbps to *8.12*Mbps as expected. The queue length of the output port *i* (*i*=1,2) in Router 2 reaches the steady state value (*200* packets) at the time t=*130*s and *130*s respectively, while both queue of the output port *i* (*i*=1,2) in Router 1 became empty at the time *t=102s*. The similar case can be seen between time $t=200$s to 400s, as shown in Figure 6.10, Figure 6.11, Figure 6.12, Figure 6.13 and Figure 6.14.

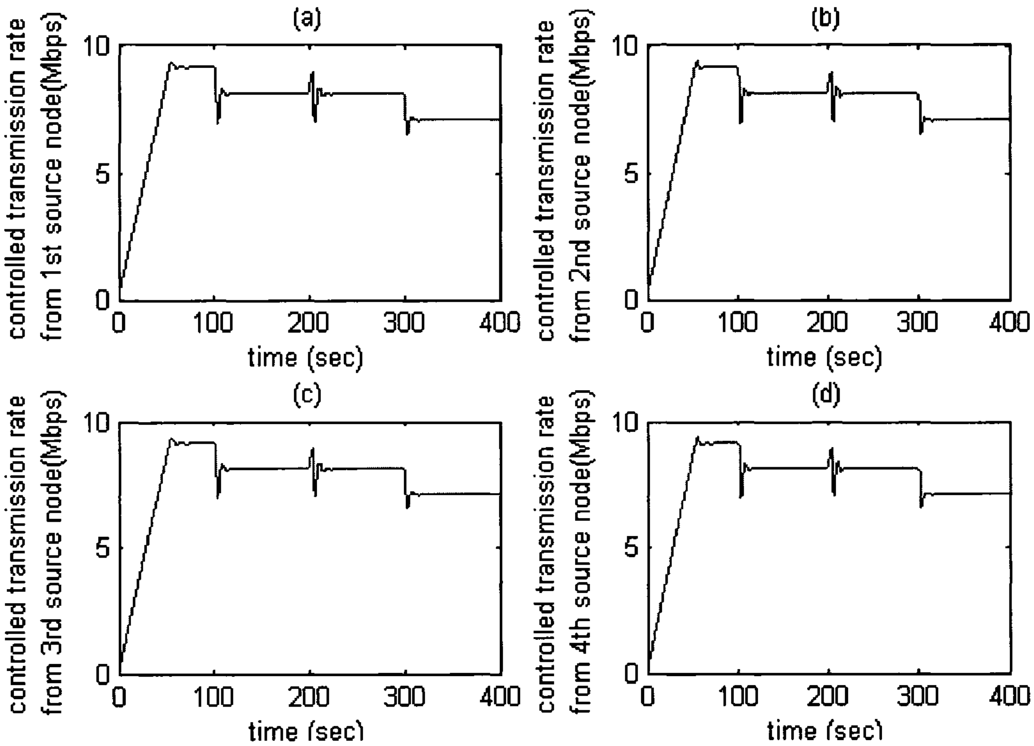

Figure 6.13: Instantaneous Transmission Rates of Controlled Source Nodes for 1st Input Port

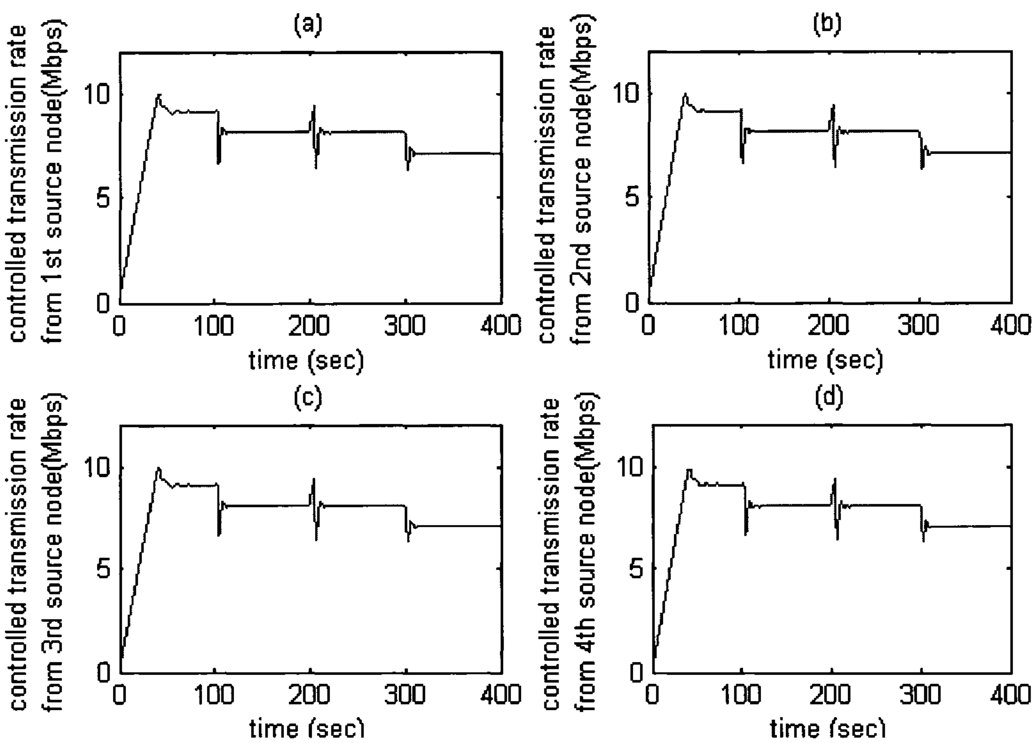

Figure 6.14: Instantaneous Transmission Rates of Controlled Source Nodes for 2nd Input Port

In summary, it can be shown that the rate-based MIMO AQM control system exhibits good transient behavior when there exists great fluctuation in the uncontrolled traffic flowing into the MIMO router.

6.5 Concluding Remarks

We have designed an adaptive multiloop PI rate-based router controller for the AQM of an MIMO router supporting best-effort service traffic in the Internet. We have employed the Direct Nyquist Array method in the control design. This design method allows the user to specify the stability robustness of the AQM control system through gain margin specification. Every controlled source node always transmits the packets at its maximum allowed transmission rate to support the best-effort service traffic, thus maximizing the bandwidth utilization of the Internet. Our adaptive multiloop PI rate-based router controller can clamp the steady queue length in every output port around the target buffer occupancy. Our simulations demonstrate that the AQM MIMO control system can adapt to the fluctuation of the uncontrolled traffic so well that the network exhibits good transient behavior. Our multiloop PI rate-based router controller is suitable for the rate control of streaming media transmission such as Voice over IP in the Internet [HaFl03].

Chapter 7 Adaptive Utility-Based Source Controller for Internet Traffic

In this final chapter, we will design an adaptive utility-based source controller for end-to-end congestion in the Internet. Multiple controlled sources transmit the packets into their destinations simultaneously and share the limited network bandwidth through a series of IP routers in the Internet. Each AQM router runs the RED (Random Early Detection) algorithm that uses ECN (Explicit Congestion Notification) packet marking strategy to provide the congestion information. An adaptive utility-based controller is placed in every source node to regulate the source transmission rate based on the feedback congestion information (in ACK packets) from the IP routers. The pole placement technique in classical control theory is used to allow the user to achieve a good transient network performance. We select a heuristic parameter and assign an interval to monitor the network congestion status. The adaptive utility-based source controller self-tunes only when the change of network parameters (the round-trip time and total packet marking rate) drifts the monitoring parameter outside its specified interval.

7.1 The Utility-Based Source Controller Design Based on Pole Placement

We will design our utility-based source controller for the Internet traffic based on the pole placement technique. Based on the network utility model described in Section 2.3, the closed-loop transfer function of the network control system in Equation (2.2) can be obtain as

$$G(s) = \frac{\mathrm{K_r w_r}}{s + \mathrm{K_r}\alpha_r e^{-s\tau_r}} \quad . \tag{7.1}$$

In order to simplify the controller design later on, we can linearize our nonlinear control system in Equation (7.1) by making a first-order Pade approximation [Bequ02] of $e^{-s\tau} \approx (1 - s\tau/2)/(1 + s\tau/2)$ when $|s\tau| = |\omega_n \tau| \le 1.2$, as shown in Figure 7.1.

Thus we can rewrite Equation (7.1) as

$$G(s) = \frac{\mathrm{K_r w_r}(1 + s\tau_r/2)}{s(1 + s\tau_r/2) + \mathrm{K_r}\alpha_r(1 - s\tau_r/2)} = \frac{\mathrm{K_r w_r}(s + 2/\tau_r)}{s(s + 2/\tau_r) + \mathrm{K_r}\alpha_r(-s + 2/\tau_r)} \quad . \tag{7.2}$$

So we can obtain the characteristic equation of the closed-loop network control system as

$$s^2+(\frac{2}{\tau_r}-K_r\alpha_r)s+\frac{2K_r\alpha_r}{\tau_r}=0 \quad . \tag{7.3}$$

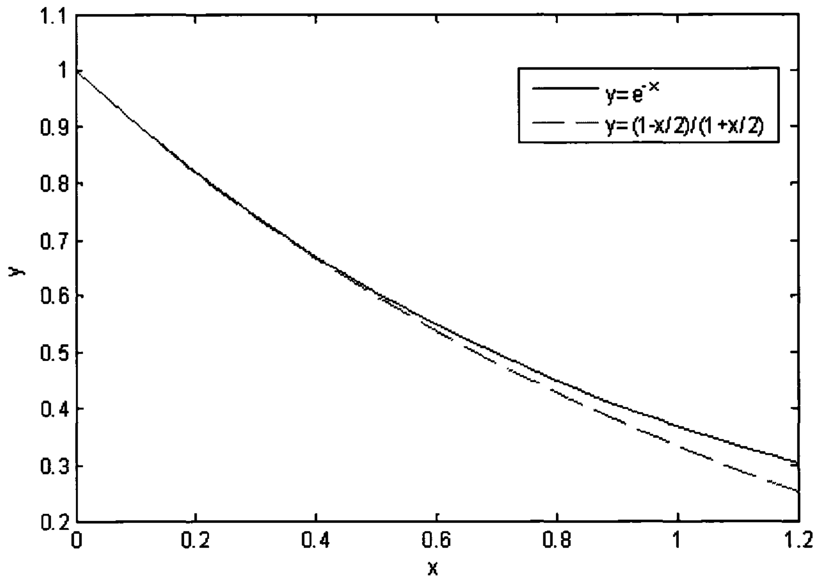

Figure 7.1: Pade Approximation of e^{-x} as (1-x/2)/(1+2/x)

Let s_1 and s_2 be the roots of the Equation (7.3), which are also the poles of the second-order closed-loop control system. To guarantee the stability of the closed-loop control system, s_1 and s_2 will be placed in the left-half s plane [Ogat02]. For $0<\zeta\leq 1$, we can obtain a pair of complex conjugate closed-loop poles ($s_1, s_2=-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$), that is, $|s_1|=|s_2|=\omega_n$. Therefore, from Equation (7.3), we can obtain

$$\frac{2}{\tau_r}-K_r\alpha_r=2\zeta\omega_n \quad , \tag{7.4}$$

$$\frac{2K_r\alpha_r}{\tau_r}=\omega_n^2 \quad . \tag{7.5}$$

From Equation (7.5), we can determine the natural frequency as

$$\omega_n=\sqrt{\frac{2K_r\alpha_r}{\tau_r}} \quad . \tag{7.6}$$

Considering the condition $\omega_n\tau_r\leq 1.2$ required by the Pade approximation, we can obtain $\omega_n\tau_r=\sqrt{2K_r\alpha_r\tau_r}\leq 1.2$, i.e.,

$$K_r\leq\frac{18}{25\alpha_r\tau_r} \quad . \tag{7.7}$$

Then substituting Equation (7.6) back into Equation (7.4), we can obtain

$$\frac{2}{\tau_r} - K_r\alpha_r = 2\zeta\sqrt{\frac{2K_r\alpha_r}{\tau_r}} \quad , \tag{7.8}$$

which can be simplified as

$$K_r^2 + \frac{4}{\alpha_r\tau_r}\left(1+2\zeta^2\right)K_r + \left(\frac{2}{\alpha_r\tau_r}\right)^2 = 0 \quad . \tag{7.9}$$

Solving Equation (7.9) based on the condition from Equation (7.7), we can obtain

$$K_r = \frac{2}{\alpha_r\tau_r}\left[1+2\zeta^2 - 2\zeta\sqrt{\zeta^2+1}\right] \quad . \tag{7.10}$$

We obtain the proportional utility gain K_r based on the linearized network control system in Equation (7.2), but we implement our utility-based source control algorithm in the original non-linear network control system in Equation (2.2). Such control system design method has been widely adopted in the industrial control system e.g., [AsHa92, AsHa95, HoHo03], which has been well accepted in network traffic control e.g., [HoMi01a, HoYa04c].

In order to keep the proportional gain K_r of utility-based controller constant, we have to adjust the damping ratio ζ whenever the network parameters α_r or τ_r change. Under this scenario, Equation (7.8) becomes

$$\zeta = \frac{1}{2}\left(\sqrt{\frac{2}{K_r\alpha_r\tau_r}} - \sqrt{\frac{K_r\alpha_r\tau_r}{2}}\right) \quad . \tag{7.11}$$

Equation (7.11) can now be used to monitor the damping ratio ζ of the utility-based control system. In general, we would like to specify an interval $[\zeta_1, \zeta_2]$, where ζ_1 and ζ_2 are the upper and lower bounds of ζ respectively. When the damping ratio ζ of the utility-based control system falls outside this specified interval due to the dramatic change of the network parameters (α_r and τ_r), the utility-based controller need to self-tune to give a new K_r value. The choice of ζ_1 and ζ_2 will be discussed later.

Another real-time monitoring method

The starting point is to rewrite Equation (7.11) as

$$K_r\alpha_r\tau_r = 4\zeta^2 - 4\zeta\sqrt{\zeta^2+1} + 2 \quad . \tag{7.12}$$

Note that we can draw a plot for the relationship between $K_r\alpha_r\tau_r$ and ζ based on Equation (7.12), as shown in Figure 7.2. For various values of ζ (e.g., ζ=*0.6, 0.8* and *1*), we have corresponding values of $K_r\alpha_r\tau_r$ (e.g., $K_r\alpha_r\tau_r$=*0.64, 0.46* and *0.34*). It means that we can also use the interval of

$K_r\alpha_r\tau_r$ (i.e., $0.34 \le K_r\alpha_r\tau_r \le 0.64$) to monitor the change of the network parameters instead of ζ. When the value $K_r\alpha_r\tau_r$ of the utility-based control system falls outside its specified interval (i.e., $0.34 \le K_r\alpha_r\tau_r \le 0.64$) upon the dramatic change of the network parameters (α_r and τ_r), the utility-based controller need to self-tune to give new value for proportional gain K_r.

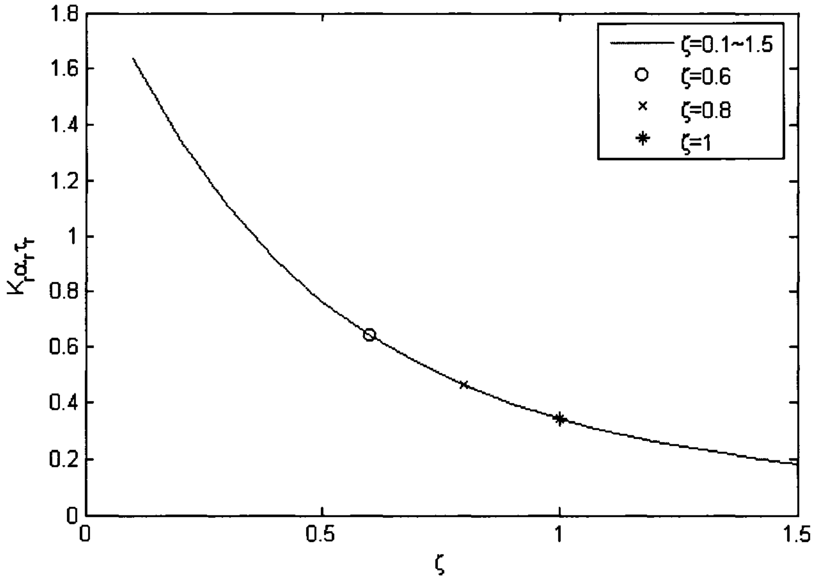

Figure 7.2: The Relationship between $K_r\alpha_r\tau_r$ and ζ

7.2 Fairness among Utility-Based Controlled Sources

Most congestion control algorithms show that the fairness can be achieved when the system reaches its steady state as $t \to \infty$ [KeMa98]. We define fairness here to be the condition that all utility-based controlled sources have the steady transmission rates proportional to their source prices. From the utility-based control algorithm expressed in Equation (2.1 and 2.2), we can see that every controlled source node r always tries to transmit the data into the network according to the price (i.e., source price and route price) assigned to it. When the transmission rates of the source nodes converge to their equilibrium points, the fairness among the controlled source nodes is achieved, as shown in our performance evaluation of Figure 7.3 later on.

7.3 Implementation of Utility-Based Source Control Algorithm

We have adopted the RED algorithm for AQM in every router and used the ECN packet-marking strategy to mark our packets based on current congestion information (i.e., link price). Then we convey the total packet-marking rate $\alpha_r(t)$ (i.e., route price) to the source node r through the ACK packets when the packets reach their destinations.

Considering the stochastic characteristics of the network, we compute the average value $\overline{\alpha}_r(t)$ of total packet-marking rate $\alpha_r(t)$ by the averaging filter method in the RED algorithm [Jaco88, FlJa93], i.e.,

$$\overline{\alpha}_r(t) = (1 - w_\alpha)\overline{\alpha}_r(t-1) + w_\alpha \alpha_r(t) \quad . \tag{7.13}$$

We need $\overline{\alpha}_r(t)$ to accomplish the following:

(1) to calculate the proportional gain K_r (through Equation (7.10), which will then be used in the utility-based source control algorithm in Equation (7.10) to obtain the source transmission rate (based on instantaneous feedback network congestion information i.e., $\alpha_r(t)$);

(2) to compute the monitoring parameter $K_r\alpha_r\tau_r$ for the network control system in order to adjust to the change of the network environment;

(3) to calculate the damping ratio ζ by Equation (7.11) to observe the change of ζ in our experiment.

To be compatible for the current dominant TCP/IP protocol in the Internet, we convert the source transmission rate calculated by Equation (2.2) into the advertised window size for the controlled source nodes. Similar methods have been well accepted in TCP/IP networks e.g., [HoYa05c, KaHa02, LoAn05]. Thus the advertised window size $win_r(t)$ for the controlled source node r is computed as

$$win_r(t) = \tau_r \cdot q_r(t) \quad . \tag{7.14}$$

where the window size must be an integer and be restricted between 1 and the maximum window size $win_r^{\max}$ allowed for the controlled source node r, i.e., $1 \le win_r(t) \le win_r^{\max}$.

We now introduce the behaviors of a controlled source, an AQM router and a destination in our utility-based source controller.

<u>Behavior of Source r</u>

Step 1: Specify an interval of ζ to be $\zeta_1 \le \zeta \le \zeta_2$ with a nominal value, and the corresponding interval of $K_r\alpha_r\tau_r$ to be $K_r\alpha_r\tau_{r\_min} \le K_r\alpha_r\tau_r \le K_r\alpha_r\tau_{r\_max}$.

Step 2: Estimate the round trip time τ_r by using the method proposed by [HaFl03]. Then record and keep its transmission rate for a round-trip time, according to utility-based source control algorithm in Equation (2.2).

Step 3: Extract the instantaneous total marking rate $\alpha_r(t)$ from the ACK packet when it receives an ACK packet.

Step 4: Apply the utility-based source control algorithm in Equation (2.2) to calculate the current transmission rate $q_r(t)$, based on instantaneous marking rate $\alpha_r(t)$ so as to prevent the network from congestion.

Step 5: Apply Equation (7.14) to calculate the source window size $win_r(t)$.

Real-time Monitoring of Source *r*

Step 5: Apply Equation (7.10) to compute the proportional gain K_r of the utility-based source controller based on the current information in the source *r*, i.e., the average marking rate $\overline{\alpha}_r(t)$ and the current round trip time τ_r.

Step 6: Utilize Equation (7.13) to compute the average value $\overline{\alpha}_r(t)$ of the total marking rate.

Step 7: Compute the parameter $K_r\alpha_r\tau_r$ of the control system based on the current information in the source *r*, i.e., the average marking rate $\overline{\alpha}_r(t)$ and the round trip time τ_r.

Step 8: (8a) If $K_r\alpha_r\tau_r$ falls outside the specified interval, then go to Step 5.

(8b) If $K_r\alpha_r\tau_r$ is still within the interval, the utility-based controller would not self-tune. Then go to Step 6.

AQM Router Behavior

Step 1: Apply RED to calculate the packet marking probability (i.e., link price) based on the current congestion situation [FlJa93].

Step 2: Use the ECN marking strategy to mark the packets [Floy94].

Destination Behavior

Step 1: Check whether the received packets are marked or not, when the packets arrive their destination,

Step 2: Calculate the total packet-marking rate $\alpha_r(t)$ (i.e., route price) in a fixed period. (The total packet-marking rate $\alpha_r(t)$ is defined as the ratio of the total number of marked packets and the total number of received packets in the period [ZhYa04]).

Step 3: Integrate the total packet-marking rate $\alpha_r(t)$ into the ACK packet and send it back to the source.

7.3.1 Initial Condition of Source r

In order to achieve a desirable transient response of a second-order system, we need to choose the ζ_1 and ζ_2 values for the interval of ζ. Note that in general, too small a ζ value yields excessive overshoot in the transient response, and a system with a large ζ value would respond sluggishly [Ogat02], as shown in Figure 7.4 of Section 7.4. From Equations (7.7) and (7.10) on the condition $\omega_n\tau_r \leq 1.2$, we can also obtain $1+2\zeta^2-2\zeta\sqrt{\zeta^2+1}\leq 9/25$, i.e., $\zeta \geq 8/15 \approx 0.53$. Then from our many performance evaluation (e.g., Figure 7.4) and many others not shown here, we have determined that $0.6\leq\zeta\leq 1$ would give a satisfactory transient response. Therefore, we can specify the nominal value of ζ to be 0.8 (the mean of 0.6 and 1), so that our adaptive utility-based source controller can clamp ζ around the range of $0.6\leq\zeta\leq 1$ upon changes in network environment.

It is obvious that the calculation of the parameter $K_r\alpha_r\tau_r$ is much simpler than the calculation of ζ by Equation (7.11). So in our experiments, the interval of $K_r\alpha_r\tau_r$ (i.e., $0.34\leq K_r\alpha_r\tau_r\leq 0.64$) is used to monitor the change of the network parameters instead of ζ, as we discussed above. When the value $K_r\alpha_r\tau_r$ of the utility-based control system falls outside its specified interval upon the dramatic change of the network parameters (α_r and τ_r), the utility-based source controller needs to self-tune to give a new value for the proportional gain K_r.

Parameter w_α

To determine w_α of Equation (7.13), we have conducted many experiments dealing with the stochastic characteristic of the data stream in the Internet, and the consideration of measurement accuracy, from which we recommend the averaging weight to be $w_\alpha=0.01$.

Parameter $\overline{\alpha}_{r\_initial}$

The initial value $\overline{\alpha}_{r\_initial}$ of $\overline{\alpha}_r(t)$ is set to be the mean of $\alpha_r(t)$ during the first round-trip time period, i.e., $\overline{\alpha}_{r\_initial}=\frac{1}{\tau_r}\sum_{t=0}^{\tau_r}\overline{\alpha}_r(t)=\frac{1}{\tau_r/\delta}\sum_{n=1}^{\tau_r/\delta}\overline{\alpha}_r(n)$, where δ is sampling time.

7.3.2 An Example

Considering a utility-based controlled source r, we have $w_r=20$, $\overline{\alpha}_r=0.225$ and $\tau_r=0.6$s. If we specify the interval of ζ to be $0.6\leq\zeta\leq 1$ and the nominal $\zeta=0.8$, then we can obtain K_r=3.422 based on Equation (7.10). Next, we apply the utility-based source control algorithm in Equation (2.2) to regulate the transmission rate of source r to prevent the Internet from congestion. At the mean time, we use the parameter $K_r\alpha_r\tau_r$ to monitor the change of the network parameters online.

7.4 Performance Evaluation of Utility-Based Source Controller

Since our adaptive utility-based source controller can regulate the source transmission rate over time and can provide an optimal transmission rate [KeMa98], we are particularly interested in analyzing in Section 7.4.1 the transient behavior of every source transmission rate and fairness among all the sources. Furthermore, we also like to see in Section 7.4.2 how our controller will adapt the source transmission rate to the dramatic change of network parameters and guarantee the global stability of the network. We assume that all the source window sizes (calculated by utility-based control algorithm) are smaller than their maximum window sizes in our experiment.

7.4.1 The Transient Response of Different Sources

We perform Matlab simulations to see how well our controller achieves a smooth transmission rate. Two different sets of experiments are examined: (1) Verify the fairness among the sources with different prices upon the same network environment; (2) Study the impact of different damping ratio ζ on the transient network performance.

Without loss of generality, we simulate a network with N=3 AQM routers located between the source and its destination. Although many sources transmit the packets to their destination through these 3 AQM routers, we only present the comparison results on the behaviors of 3 sources with different weight w_r and the packets sent by these 3 sources routed through the same 3 AQM routers, as shown in Figure 2.7. The weights w_r of the source node r ($r=1,2,3$) are *10*, *20* and *30* respectively. The round-trip times τ_r of the source node r ($r=1,2,3$) are *1s*, *0.6*s and *0.4s* respectively. The initial values of all the source transmission rates are set to zero, i.e., $q_1(0)=q_2(0)=q_3(0)=0$. We assume that the RED and the ECN marking probabilities p_j for the packets in AQM router j ($j=1,2,3$) are uniformly distributed within the intervals [0, 0.1], [0, 0.15] and [0,0.2] respectively. Thus, the average of total marking rate can be approximated as $\overline{\alpha}_r = 0.1/2+0.15/2+0.2/2 = 0.225$. Such control system parameter approximation is well accepted in control system design e.g., [AsHa92, AsHa95, HoHo03]. When the time $t\to\infty$, the $\alpha_r(t)$ will be also converge to

$$\alpha_r^* = \lim_{T\to\infty}\frac{1}{T}\int_0^T (p_1(t)+p_2(t)+p_3(t))dt = 0.225.$$

First, in order to make comparison among the transient behaviors of the 3 different sources (that send the packets routed through the same 3 AQM routers), we require the damping ratio of the 3 sources to be the same $\zeta=0.8$. Then the network parameter values of Source 1 become

$w_1=10$, $\overline{\alpha}_1 = 0.225$ and $\tau_1=1$s. Under the case of ζ=0.8, we can obtain K_r=2.053 based on Equation (7.10). Then we apply utility control algorithm in Equation (2.2) to regulate the transmission rates of Sources *1* to *3*.

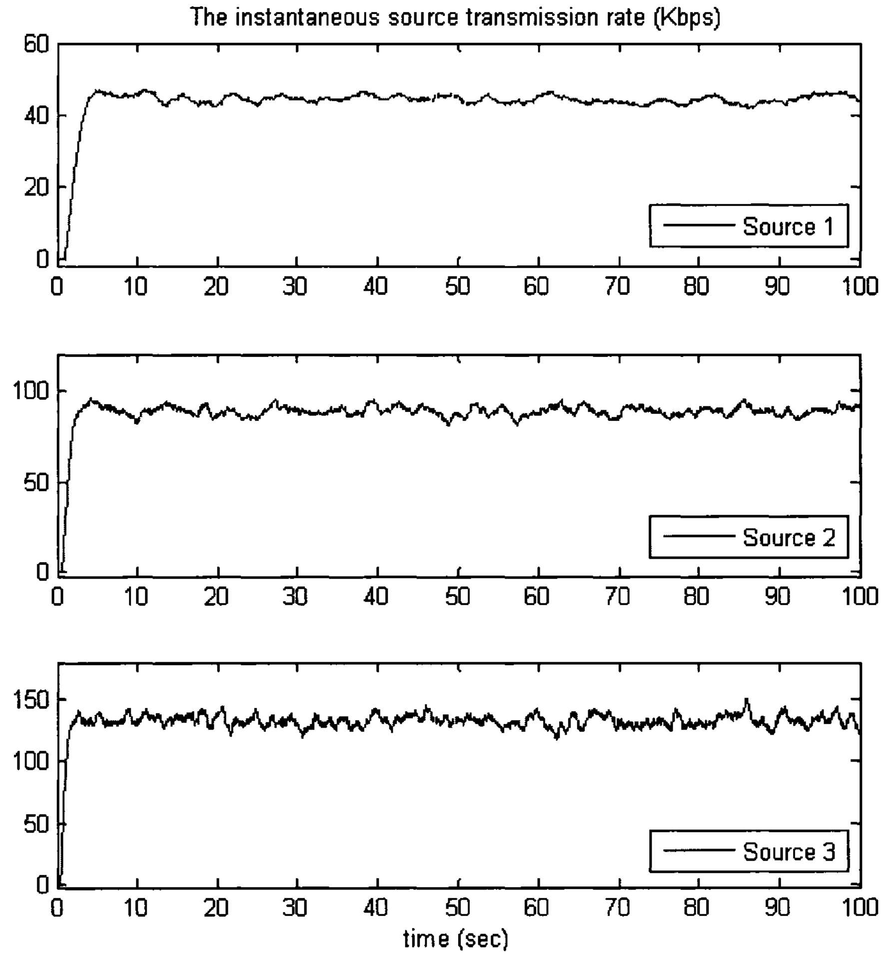

Figure 7.3: Instantaneous Transmission Rates of Different Sources *r* (*r=1,2,3*)

Figure 7.3 shows the instantaneous transmission rates of different sources. We can see that all 3 sources exhibit good transient behavior under the proper damping ratio $\zeta=0.8$. It is shown that the transmission rates of the source *r* (*r=1,2,3*) converge to their equilibrium points $q_1^* = \frac{w_1}{\alpha_1^*} = \frac{10}{0.225} = 44.44Kbps$, $q_2^* = \frac{w_2}{\alpha_2^*} = \frac{20}{0.225} = 88.89Kbps$ and $q_3^* = \frac{w_3}{\alpha_3^*} = \frac{30}{0.225} = 133.33Kbps$ respectively, as anticipated. The fairness among these 3 sources is achieved when their transmission rates approach their respective equilibrium points, as we can observe from Figure 7.3.

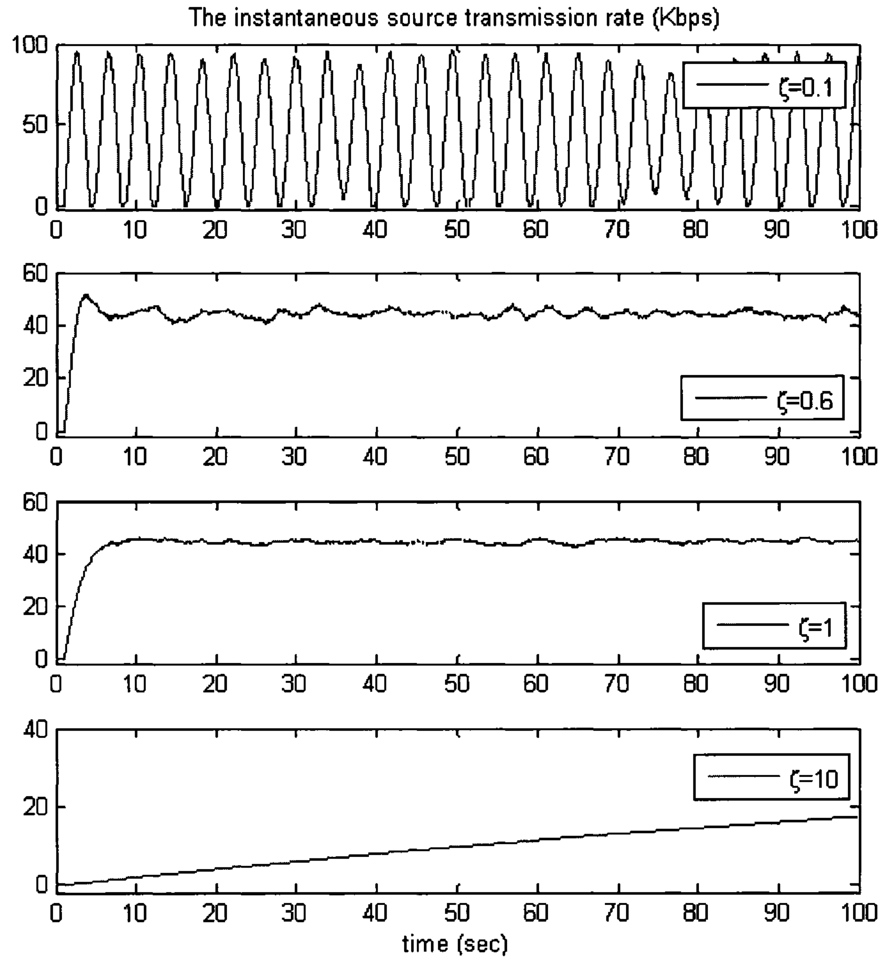

Figure 7.4: Instantaneous Transmission Rates of Different Sources r ($r=1,2,3$)

7.4.1.1 Impact of ζ on Transient Response of Source Transmission Rate

We have performed simulations for Source *1* to demonstrate the impact of ζ on the transient response of source transmission rate. Figure 7.4 shows the instantaneous transmission rates of Source 1 for different ζ values. We can see that the maximum percent overshoot decreases and settling time becomes longer as the damping ratio ζ increases within the range of $0.6 \le \zeta \le 1$. Both source transmission rates converge to an equilibrium point $q_1^* = \frac{w_1}{\alpha_1^*} = \frac{10}{0.225} = 44.44 Kbps$ in the case of $\zeta=0.6$ and $\zeta=1$, as we expected. When $\zeta=0.1$, the source transmission rate exhibits large oscillations, which suggests an unstable behavior of the network. On the other hand, when $\zeta>1$, the overdamped system has a slow response, especially for the case of $\zeta=10$, where the transmission rate still does not converge to its equilibrium point until $t=100$s.

7.4.2 Adaptive Utility-Based Source Controller versus Fixed Utility-Based Source Controller

We want to demonstrate the importance of adaptive control in the network traffic control by comparing the adaptive utility-based source controller with the fixed utility-based source controller. Here the proportional gain of a fixed utility-based source controller is fixed no matter how large the change of the network parameters, while that of an adaptive utility-based source controller should be re-computed if the monitoring parameter $K_r\alpha_r\tau_r$ drifts outside its specified interval.

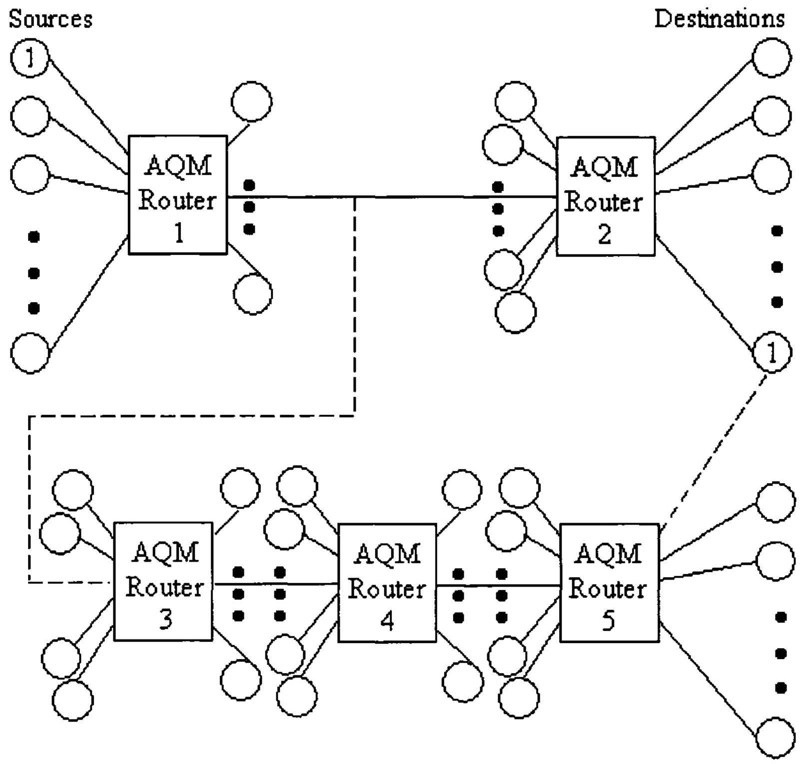

Figure 7.5: Utility-Based Controlled Sources with 5 AQM Routers

We consider a network with 5 AQM routers in Figure 7.5 that we have implemented the fixed utility-based source control algorithm and the adaptive utility-based source control algorithm for Source *1* respectively. In this network, from the time $t=0$s to *100*s, the packets are transmitted by Source *1* and routed through 2 AQM routers (Router *1* and Router *2*) before they reach the destination. At time $t=100$s, we turn off Router *2* (e.g., system crashed), so that the packets have to by-pass Router 2 and have found a new route through Router *3*, *4* and *5* (i.e., the new path is now Routers *1*, *3*, *4* and *5*) before they arrive at the destination. The monitored interval of damping ratio ζ is specified as $0.6 \leq \zeta \leq 1$ and the nominal value is specified as $\zeta=0.8$. So the corresponding interval of the monitoring parameter $K_r\alpha_r\tau_r$ is selected to be $0.34 \leq K_r\alpha_r\tau_r \leq 0.64$.

It is assumed that the RED and the ECN marking probabilities p_j for the packets in AQM router j ($j=1,2,3,4,5$) are uniformly distributed on [0, 0.1], [0, 0.2], [0, 0.15], [0, 0.2] and [0,0.25] respectively. The round-trip time τ_1 between Source 1 and its destination is 1s if the packets are routed through Routers 1 and 2, while the round trip time τ_1 is 1.5s if the packets are routed through Routers 1, 3, 4 and 5. At the beginning of the simulation, the initial value of average total packet marking rate $\overline{\alpha}_1(t)$ is calculated as the mean of instantaneous total packet marking rate $\alpha_1(t)$ during the first round-trip time period, as discussed in Section 7.3. Figure 7.6 shows the instantaneous total packet marking rate $\alpha_1(t)$, and the average marking rate $\overline{\alpha}_1(t)$ (see Equation (7.13) before and after Router 2 shuts down.

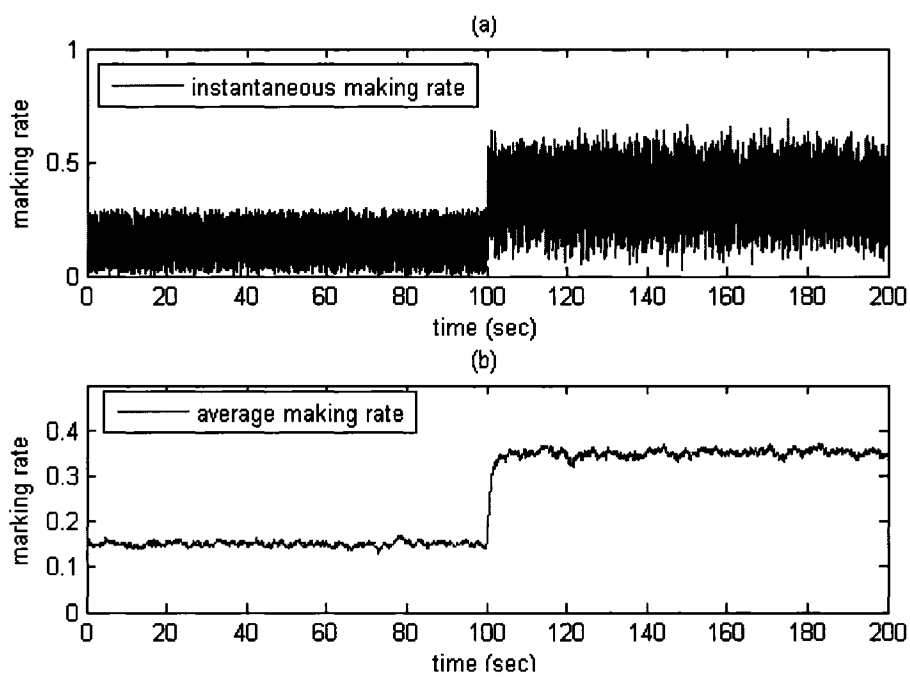

Figure 7.6: The total packet marking rate of Source 1
(a) instantaneous total packet marking rate $\alpha_1(t)$
(b) average total packet marking rate $\overline{\alpha}_1(t)$

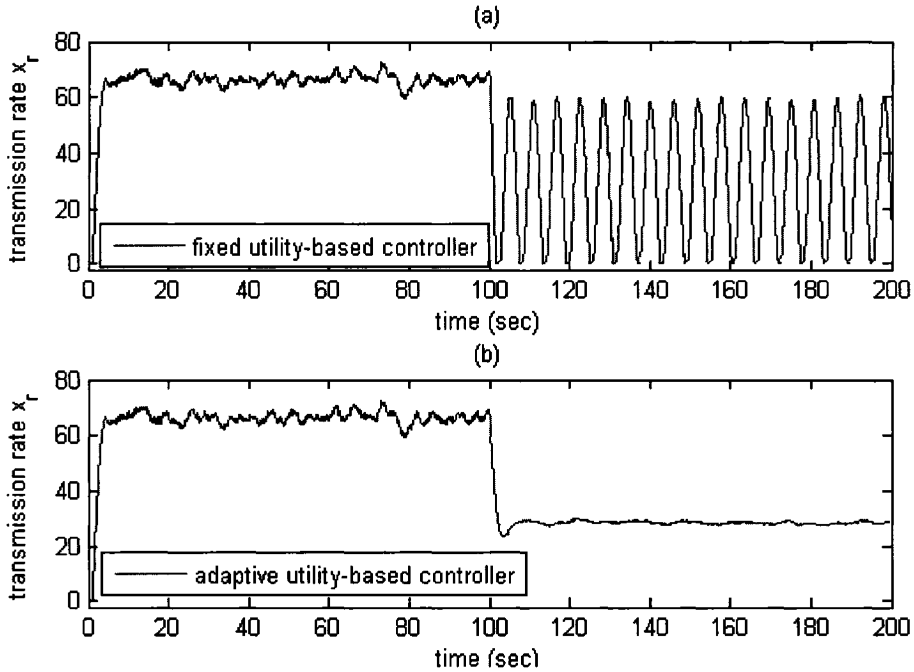

Figure 7.7: Instantaneous Transmission Rate of Source 1
(a) regulated by fixed utility-based control algorithm
(b) regulated by adaptive utility-based control algorithm

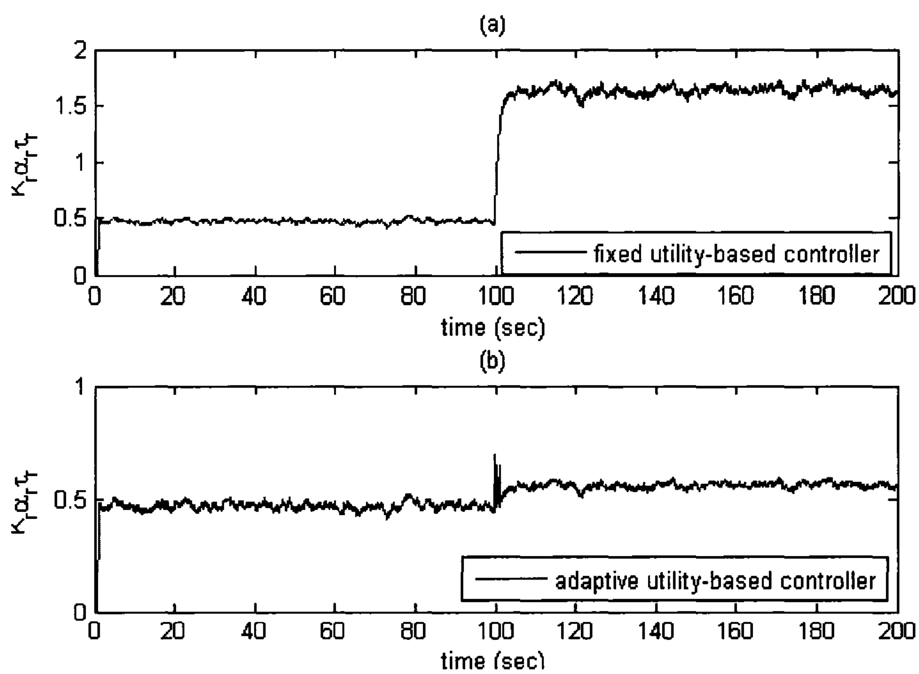

Figure 7.8: Instantaneous Monitoring Parameter $K_r\alpha_r\tau_r$
(a) for fixed utility-based control algorithm
(b) for adaptive utility-based control algorithm

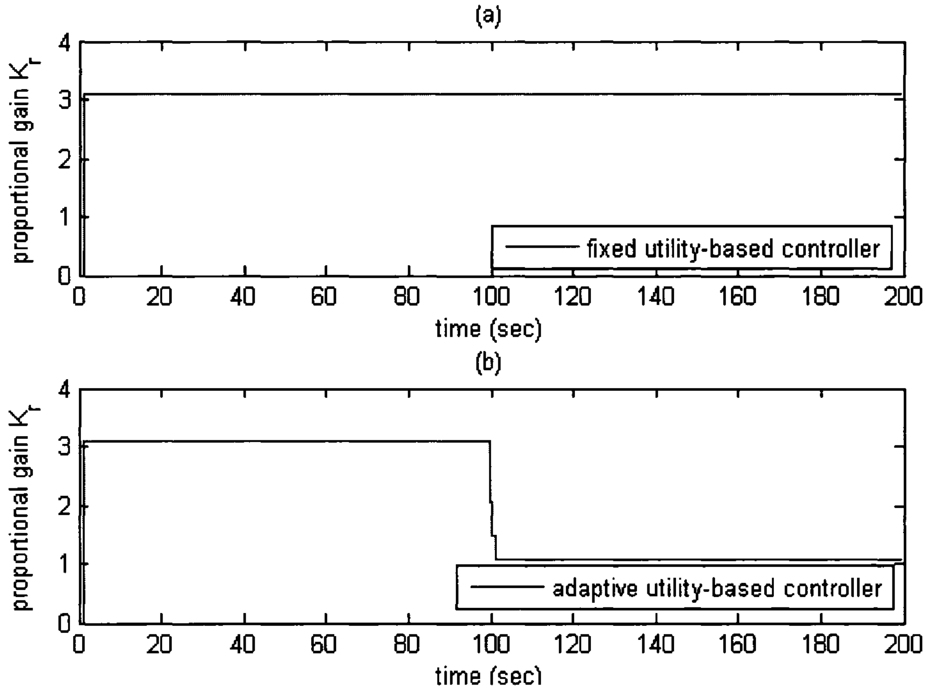

Figure 7.9: Instantaneous Proportional Gain K_r
(a) for fixed utility-based control algorithm
(b) for adaptive utility-based control algorithm

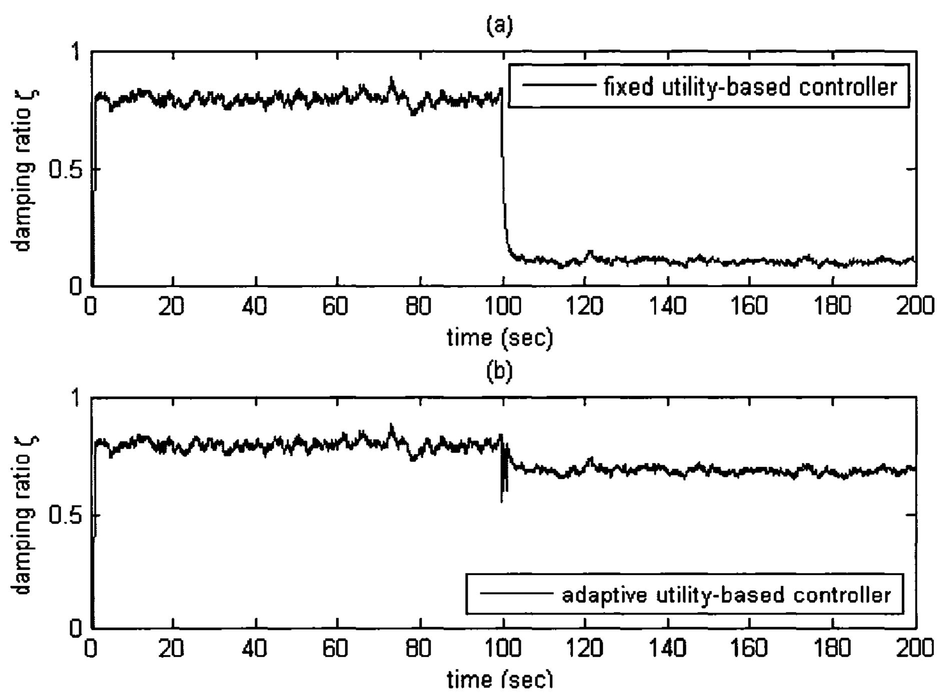

Figure 7.10: Instantaneous Damping Ratio Calculated by Equation (7.11)
(a) for fixed utility-based control algorithm
(b) for adaptive utility-based control algorithm

Figures 7.7 to 7.9 show respectively the instantaneous source transmission rate, the instantaneous monitoring parameter $K_r\alpha_r\tau_r$ and the instantaneous proportional gain K_r performance. Both the fixed utility-based source control algorithm and the adaptive utility-based source control algorithm are compared. In our experiment, we also calculate the damping ratio ζ online to observe the change of ζ, as shown in Figure 7.10.

From the time $t=0$s to 100s, the total packet marking rate is uniformly distributed on [0, 0.3], and the average value of the total packet marking is clamped around 0.15, as shown in Figure 7.6. The monitoring parameter $K_r\alpha_r\tau_r$ has only a little fluctuation around 0.46, and falls within the interval of $0.34 \le K_r\alpha_r\tau_r \le 0.64$, so the adaptive utility-based controller remains unchanged. Both proportional gains of the fixed utility-based source controller and the adaptive utility-based source controller remain constant, as shown in Figure 7.9. Both source transmission rates converge to the equilibrium point $q_r^* = \frac{w_r}{\alpha_r^*} = \frac{10}{0.15} = 66.67 Kbps$ at time $t=5$s, as we anticipated. At time $t=100$s, Router 2 shuts down, the packets have to be routed through Routers 3, 4 and 5. The instantaneous total packet-marking rate α_r for Source *1* has undergone a great change and is now uniformly distributed in [0, 0.7], as shown in Figure 7.6. The round-trip time becomes longer. The monitoring parameter $K_r\alpha_r\tau_r$ goes outside the interval $0.34 \le K_r\alpha_r\tau_r \le 0.64$, as shown in Figure 7.8. So the adaptive utility-based controller needs to self-tune. The proportional gain K_r changes from 3.1 to 1.06 between time $t=100$s and $t=101$s, as shown in Figure 7.9(b). After $t=101$s, the monitoring parameter $K_r\alpha_r\tau_r$ has little oscillation around 0.56, which still falls within the interval of $0.34 \le K_r\alpha_r\tau_r \le 0.64$. So our adaptive utility-based controller remains unchanged. The source transmission rate converges to a new equilibrium point $q_r^* = \frac{w_r}{\alpha_r^*} = \frac{10}{0.35} = 28.57 Kbps$ at time $t=110$s, as shown in Figure 7.7(b). Since the fixed utility-based controller is fixed, the monitoring parameter $K_r\alpha_r\tau_r$ increases to around 1.62 (corresponding to a damping ratio $\zeta \approx 0.1$), and causes the large oscillation of the source transmission rate shown in Figure 7.7(a).

It is interesting to observe from these experiments and others not shown here that our result with a non-zero round-trip delay always converges to an equilibrium point. With reference to Kelly's model without delays, we would suspect that our source control algorithm can also make optimal use of the network resources.

7.5 Concluding Remarks

We have extended Kelly's network utility model without delays to a general utility model for the Internet with communication delays. We have developed an adaptive utility-based source controller for the Internet traffic based on this new model, thus providing a desirable pricing mechanism for the network providers. The pole placement method from the classical control theory was employed in the controller design, and a simple equation for obtaining the utility-based source controller has been derived. By selecting a proper interval of damping ratio ζ, we can achieve a satisfactory transient response of the network control system. The utility-based source controller is located in the every source and it can calculate the source transmission rate based on its own source price and the instantaneous feedback route congestion information (i.e., route price) through ACK packets. Every source transmission rate can always converges to an equilibrium state. We have demonstrated by simulations that with our utility-based source controller applied, we can achieve desirable network performance in source transmission rate. Furthermore, relatively smooth transmission rate makes our adaptive utility-based source control algorithm quite suitable for rate control of multimedia transmission such as audio and video stream transmission in the Internet [HaFl03].

Chapter 8 Conclusions

In this thesis, we have proposed the designs of various controllers for Internet in order to prevent the end-to-end congestion in the Internet. Our designs are summarized as follows.

First, we have proposed adaptive P and PI window-based router controllers for the AQM of TCP traffic in the Internet. Based on the queue size in the router, P and PI window-based router controllers regulate the TCP source window size indirectly through adjusting packet drop probability in the router.

Next, a network model for a typical SISO IP router is proposed for the design of an adaptive PI rate-based router controller for best-effort streaming media traffic in the Internet. This controller is located in the IP router and can calculate the advertised source transmission rate based on the instantaneous queue length so that the source node can regulate transmission rate to prevent network congestion. Frequency domain technique is applied in the controller design to achieve good AQM performance by specifying a proper phase margin.

Third, we have proposed a network model for a MIMO IP router and then design an adaptive multi-loop PI rate-based router controller for best-effort streaming media traffic in the Internet. We have employed the DNA (Direct Nyquist Array) method in the controller design. The adaptive PI rate-based router controller is located in every output port of the MIMO router and can calculate the respective advertised source transmission rates based on the instantaneous queue length of the buffer. Every controlled source node always sends the IP packets to the network at the maximum allowed transmission rate, thus providing the best-effort service traffic and making the maximum use of network resources.

Forth, we have extended Kelly's network utility model without delays to a general utility model for the Internet with communication delays. We have developed an adaptive utility-based source controller for Internet traffic based on pole placement method. The utility-based source controller is located in the every source and it can calculate the source transmission rate based on its own source price and the instantaneous feedback route congestion information (i.e., route price) through ACK packets. Relatively smooth source transmission rate can always converge to an equilibrium state.

Our adaptive controllers have the capability to self-tune only when the dramatic changes of network traffic cause the monitoring parameter to drift outside the specified interval. Otherwise, the controllers remain unchanged. Our control system can adapt very well to great traffic load changes.

In summary, we have looked at 3 different types controllers for Internet traffic: window-based router controller (Chapters 3 and 4), rate-based router controller (Chapters 5 and 6) and utility-based source controller (Chapter 7). Unlike many heuristics, all these are supported by control theory; thus our controllers can always guarantee the stability of the network control system.

8.1 Future Work

There are still many interesting problems left for continuous development on our new proposed congestion controllers. For future work, we can extend our study of the Internet traffic control to the following items:

(1) Implement the flows using the rate-based router control algorithm to compare the flows using the AIMD source control over window-based router control in the same network environment;

(2) Combine the control theory with the theory of differential games to allocate the limited bandwidth in wireless networks;

(3) Extend our control system design method for the power control in optical communication networks.

References

[AsGo95] K.J. Astrom, G.C. Goodwin, and P.R. Kumar (editors), *Adaptive Control, Filtering, and Signal Processing*, Vol. 74, The IMA Volumes in Mathematics and its Applications, Springer-Verlag, 1995.

[AsHa92] K.J. Astrom, C.C. Hang, P. Persson, W.K. Ho, "Towards intelligent PID control", *Automatica*, 28(1), 1992, pp. 1-9.

[AsHa93] K. J. Astrom, T. Hagglund, C. C. Hang, and W. K. Ho, "Automatic Tuning and Adaptation for PID Controllers-A Survey," *IFAC J. Control Eng. Practice*, Vol.1, no.4, pp.699-714, 1993.

[AsHa95] K.J. Astrom, T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, second edition, North Carolina, 1995.

[AsWi95] K.J. Astrom and B. Wittenmark, *Adaptive Control,* 2nd edition. Addison-Wesley Publication Co., 1995.

[AtLi01] S. Athuraliya, V. H. Li, S. H. Low and Q. Yin, "REM: Active Queue Management", *IEEE Network*, 15(3), May/June 2001, pp 48-53.

[AwOu01] J. Aweya, M. Ouellette, and D. Y. Montuno, "A Control Theoretic Approach to Active Queue Management", *Computer Networks Journal*, Vol. 36, Issue 2–3, July 2001, pp. 203–35.

[AwOu02] James Aweya, Michel Ouellette and Delfin Y. Montuno, "Using TCP rate control for queue input-output rate matching", *International Journal of Communication Systems*, 15(10), December 2002, pp. 907-920.

[Bequ02] B. Bequette, *Process Control: Modeling, Design And Simulation*, Prentice Hall, 2002.

[BlCi02] F. Blanchini, R.L. Cigno and R. Tempo, "Robust Rate Control for Integrated Services Packet Networks", *IEEE/ACM Transactions on Networking, Vol. 10, No. 5*, October 2002, pp. 644-652.

[BlLo02] F. Blanchini, R. Lo Cigno, R. Tempo, "Robust Rate Control for Integrated Services Packet Networks", *IEEE/ACM Transactions on Networking*, **10**(5), 2002, pp. 644-652.

[BrCl98] B. Braden, D. Clark and et al. "Recommendations on Queue Management and Congestion Avoidance in the Internet," *IETF RFC2309*, April 1998.

[Bris86] E.H. Bristol, "The EXACT Pattern Recognition Adaptive Controllers, A User-oriented Commercial Success", Adaptive and learning systems, New York, Plenum Press, 1986, pp. 149-163.

[CaGe04] D. Cavendish; M. Gerla, S. Mascolo, "A control theoretical approach to congestion control in packet networks", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 5, October 2004, pp. 893-906.

[CrBe97] M. Crovella and A. Bestavros, "Self Similarity in World Wide Web Traffic: Evidence and Possible Causes", *IEEE/ACM Transactions on Networking,* vol. **5**(6), 1997, pp.835-846.

[ChYa04] Qiang Chen and Oliver W. W. Yang, "On Designing Self-Tuning Controllers for AQM Routers Supporting TCP Flows Based on Pole Placement, " *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 10, Dec. 2004, pp.1965-1974.

[Desh89] Deshpande P.B. *Multivariable Process Control*, Instrument Society of America, Research Triangle Park, NC, 1989.

[DeSr03] S. Deb and R. Srikant, "Global stability of congestion controllers for the Internet", *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp. 1055-1060, 2003.

[DeYi03] X. Deng, S. Yi, G. Kesidis, and C.R. Das, "A Control Theoretical Approach for Designing Adaptive AQM Schemes", Proceedings of IEEE Globecom, San Francisco, CA, U.S.A, December 2003.

[ElDe03] A. El-Hoiydi and J.D. Decotignie, "Simulation of Low Power MAC Protocols for Wireless Sensor Networks", Proceedings of 5th IFAC International Conference on Fieldbus Systems and their Applications FET, Aveiro, Portugal, July 2003.

[FeSh02] W. Feng, K. Shin, D. Kandlur and D. Saha "The Blue: Active Queue Management Algorithms", *IEEE/ACM Transactions on Networking*, Vol. 10, No. 4, August 2002, pp. 513-528.

[FlFa99] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, August 1999, pp. 458-472.

[FlGu01] S. Floyd and R. Gummadi and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED," UC Berkeley *ICIR Technical Report*, Available at http://www.icir.org/floyd/papers/adaptiveRed.pdf, August 2001.

[FlHa00] S. Floyd, M. Hanley, J. Padhye and J. Widmer "Equation-Based Congestion Control for Unicast Applications", *Proceedings of ACM SIGCOMM 2000*, pp. 43-56.

[FlJa93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397-413.

[Floy94] S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review*, Vol.24, No.5, 1994, pp.10-23.

[FuWa98] H.K. Fung, Q.G. Wang and T.H. Lee, "PI Tuning in Terms of Gain and Phase Margins", *Automatica*, Vol. 34, 9, pp. 1145-1149, September 1998.

[GeLo02] M. Gerla, R. Locigno, S. Mascolo and W. Weng, "Generalized Window Advertising for TCP Congestion Control", *European Transactions on Telecommunications*, No. 6, pp. 549-562, Nov/Dec. 2002.

[GeWe00] M.Gerla, W.Weng, R.Lo Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet", *Proceedings of IEEE Globecom 2000*, San Francisco, CA, USA, November 2000.

[HaAs91] C.C. Hang, K.J. Astrom and W.K. Ho, "Refinements of the Ziegler-Nichols tuning formula", *IEE Proceedings on Control Theory and Applications, Volume 138, Issue 2*, March 1991, pp.111-118.

[HaFl03] M. Hanley, S. Floyd, J. Padhye and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", *RFC 3448*, Proposed Standard, January 2003.

[HaLe93] C.C. Hang, T.H. Lee and W.K. Ho, *Adaptive Control,* Instrument Society of America, Research Triangle Park, NC, 1993.

[HaLi04] D.C. Hanselman, B.L. Littlefield, *Mastering MATLAB 7*, Prentice Hall, U.S.A., 2004

[Hash89] E. Hashem, "Analysis of random drop for gateway congestion control", *Report LCS TR-465*, Laboratory for Computer Science, MIT, Cambridge, MA, 1989.

[HiOh03] H. Hiroyuki, H. Ohsaki and M. Murata, "Steady State and Transient Behavior Analyses of TCP Connections considering Interactions between TCP connections and Network", Proceedings of the Symposium on Applications and the Internet, 2003.

[HoCh03] Yang Hong, Qiang Chen and Oliver W.W. Yang, "Design of TCP Traffic Controllers for AQM Routers Based on Gain Margin Specification", *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003)*, Montreal, Canada, July 2003, pp 452-457.

[HoHo03] W.K. Ho, Y. Hong, A. Hansson, H. Hjalmarsson and J.W. Deng, "Relay Auto-Tuning of PID Controllers using Iterative Feedback Tuning" *Automatica, Volume 39, Issue 1*, January 2003, pp.149-157.

[HoLe01] W.K. Ho, T.H. Lee, H.P. Han and Y. Hong, "Self-Tuning IMC-PID Control with Interval Gain and Phase Margin Assignment", *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 3, May 2001, pp. 535-541.

[HoMi01a] C.V. Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", *Proceedings of IEEE/INFOCOM*, April 2001, pp.1726-1734.

[HoMi01b] C.V. Hollot, Vishal Misra, Don Towsley and Wei-Bo Gong, "A Control Theoretic Analysis of RED," *Proceedings of IEEE/INFOCOM*, April 2001, pp.1510-1519.

[HoMi02] C.V. Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong, "Analysis and Design of Controllers for AQM Routers Supporting TCP Flows", *IEEE Transactions on Automatic Control*, Vol. 47, No. 6, September 2002, pp. 945-959.

[HoYa03] Yang Hong and Oliver W.W. Yang, "An End-to-End PI Congestion Controller for Best-Effort Traffic: Single Source Case", 39(5), *IEE Electronics Letters, March 2003,* pp 480-482.

[HoYa04a] Yang Hong, Oliver W.W. Yang, "Self-Tuning PI Controller for AQM Routers With Interval Gain Margin Assignment", *Proceedings of 38th annual conference on Information Sciences and Systems (CISS 2004)*, Princeton University, U.S.A, March 2004, pp. 455-460.

[HoYa04b] Yang Hong, Oliver W.W. Yang, "Design of Self-Tuning Proportional TCP Flow Controller for AQM Routers Based on Gain Margin Specification", *Proceedings of 22nd Biennial Symposium on Communications (QBSC 2004)*, Kingston, Canada, June 2004, pp. 280-282.

[HoYa04c] Yang Hong, Oliver W.W. Yang and ChangCheng Huang, "Self-Tuning PI TCP Flow Controller for AQM Routers With Interval Gain and Phase Margin Assignment", *Proceedings of IEEE Globecom*, Dallas, U.S.A, November 2004, pp. 1324-1328.

[HoYa05a] Yang Hong, Oliver W.W. Yang, "Design of Utility-Based Congestion Controller for Internet Traffic Based on Pole Placement", *Proceedings of 39th annual conference on Information Sciences and Systems (CISS 2005)*, Johns Hopkins University, U.S.A, March 2005.

[HoYa05b] Yang Hong, Oliver W.W. Yang, "Self-Tuning PI Rate Controller for AQM Router Supporting Best-Effort Traffic", *Proceedings of IEEE International*

Conference on Communication (ICC 2005), Seoul, Korea, May 2005, pp.1539-1543.

[HoYa05c] Yang Hong, Oliver W.W. Yang, "Design of an Adaptive PI Rate Controller for Streaming Media Traffic Based on Gain and Phase Margins", To appear in *IEE Proceedings on Communications*.

[HuXu03] Cheng Huang, Lihao Xu, "SRC: Stable Rate Control for Streaming Media", in *Proceedings of IEEE Globecom*, San Francisco, U.S.A, December 2003, pp.4016–4021.

[Jaco88] V. Jacobson, "Congestion Avoidance and Control", *Proceedings of ACM SIGCOMM*, August 1988, pp. 314-329.

[JoTa01] R. Johari and D.K.H. Tan, "End-to-end congestion control for the Internet: Delays and stability", *IEEE/ACM Transactions on Networking*, Vol. 9, No. 6, December 2001, pp. 818-832.

[JiWe04] C. Jin, D.X. Wei and S.H. Low, "FAST TCP: motivation, architecture, algorithms, performance", *Proceedings of IEEE INFOCOM*, March 2004, pp.2490-2501.

[KaHa02] D. Katabi, M. Handley and C. Rohrs, "Congestion Control for High Bandwidth Delay Product Networks", *Proceedings of ACM SIGCOMM*, August 2002.

[KaKa00] S. Karandikar, S. Kalyanaraman, P. Bagal, and B. Packer, "TCP Rate Control", *ACM Computer Communications Review*, Vol. 30, January 2000, pp. 45–58.

[KeMa98] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, Vol. 49, 1998, pp. 237-252.

[KuSr03] S. Kunniyur and R. Srikant, "End-to-end congestion control: utility functions, random losses and ECN marks", *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, Oct. 2003, pp. 689-702.

[KuSr04] Srisankar Kunniyur and R. Srikant "An adaptive virtual queue (AVQ) algorithm for active queue management", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, April 2004, pp. 286-299.

[LaAn02] R.J. La and V. Anantharam, "Utility-based rate control in the Internet for elastic traffic", *IEEE/ACM Transactions on Networking*, Vol. 10, No. 2, April 2002, pp. 272-286.

[LoAn05] S.H. Low, L.L.H. Andrew, B.P. Wydrowski, "Understanding XCP: equilibrium and fairness", *Proceedings of IEEE INFOCOM*, March 2005, pp. 1025-1036.

[LoLa99] S. H. Low and D. E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence", *IEEE/ACM Transactions on Networking*, Vol. 7, No.6, December 1999, pp. 861–875.

[LoPa02] S. Low, F. Paganini, J. Wang, S. Adlakha, J.C. Doyle, "Dynamics of TCP/RED and a Scalable Control", *Proceedings of IEEE INFOCOM*, June 2002, pp. 239 – 248.

[LiBa03] S. Liu, T. Basar, and R. Srikant, "Controlling the Internet: A survey and some new results," *Proceedings of IEEE Conference on Decision and Control*, Hawaii, *2003*.

[LiMo97] Dong Li and Robert Morris, "Dynamics of Random Early Detection", Proceedings of ACM SIGCOMM, September 1997.

[LoAn05] S.H. Low, L.L.H. Andrew, B.P. Wydrowski, "Understanding XCP: equilibrium and fairness", *Proceedings of IEEE INFOCOM*, March 2005.

[LoPa02] S. Low, F. Paganini and et al. "Dynamics of TCP/RED and a scalable control", *Proceedings of IEEE INFOCOM*, June 2002, pp. 239 – 248.

[MaBo99] M. May, J. Bolot, C. Diot and B. Lyles, "Reasons Not to Deploy RED", *Proceedings of 7th International workshop on Quality of Service (IWQoS)*, June 1999, pp. 260-262.

[Maci89] Maciejowski J.M., *Multivariable Feedback Design*, Addison Wesley, 1989

[MaSe97] M. Mathis, J. Semke, J. Mahdavi and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoid Algorithm", *ACM Computer Communication Review*, Vol. 27, No. 3, July 1997.

[Mass02] L. Massoulie. "Stability of distributed congestion control with heterogeneous feedback delays," *IEEE Transactions on Automatic Control*, Vol. 47, No. 6, 2002, pp. 895-902.

[MiGo00] 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," *Proceedings of ACM/SIGCOMM*, 2000.

[MoLa99] Jeonghoon Mo, Richard J. La, V. Anantharam, and J. Walrand, "Analysis and Comparison of TCP Reno and TCP Vegas", *Proceedings of IEEE INFOCOM*, March 1999, New York, pp. 1556-1563.

[Ogat02] Katsuhiko Ogata, *Modern Control Engineering*, 4th Edition, Prentice Hall, 2002.

[OPNE00] OPNET Modeler Manuals, OPNET Version 7.0, OPNET Technologies, Inc., 2000.

[OtLa99] Ott, T.J., Lakshman, T.V. and Wong, L.H., "SRED: stabilized RED," in *Proceedings of IEEE/INFOCOM,* 1999, pp. 1346 –1355.

[PaPr99] Rong Pan, Balaji Prabhakar and Konstantinos Psoynis, "CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation", *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000, pp. 942-951.

[PaZh04] F. Paganini, Z. Wang, J. Doyle and S. Low, "Congestion control for high performance, stability and fairness in general networks", *IEEE/ACM Transactions on Networking*, Vol. 13, No. 1, pp. 43-56.

[RaAb03] P. Ranjan, E. H. Abed and R. J. La, "Communication delay and instability in rate-controlled networks", *Proceedings of 42nd Annual IEEE Conference on Decision and Control*, Hawaii, U.S.A, December 2003.

[ReHa99] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Real-time Streams in the Internet," *Proceedings of IEEE INFOCOM*, March 1999, pp.1337 – 1345.

[Rose70] Rosenbrock H. *State-Space and Multivariable Theory*, Wiley, New York, 1970.

[Stal03] William Stallings, *Data and Computer Communications*, 7th Edition, Prentice Hall, 2003

[Stev94] W.R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, Boston, 1994.

[Stev97] W. Richard Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms", *IETF RFC2001*, September 1997.

[WaTa04] J. Wang, A. Tang and S. H. Low, "Local stability of FAST TCP", *Proceedings of 43rd Annual IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec 2004.

[Whit71] P. Whittle, Optimization Under Constraints, Wiley, Chichester, 1971.

[WuRe01] Wei Wu, Yong Ren and Xiuming Shan, "A Self-Configuring PI Controller for Active Queue Management", *Proceedings of the 7th Asia-Pacific Communication Conference (APCC'2001)*, Tokyo Japan, Sep. 2001.

[Vinn02] G. Vinnicombe, "On the stability of networks operating TCP-like congestion control", *Proceedings of the IFAC World Congress*, Barcelona, Spain, 2002.

[YaMa00] H. Yaiche, R. R. Mazumdar and C. Rosenberg, "A game-theoretic framework for bandwidth allocation and pricing in broadband network", *IEEE/ACM Transactions on Networking*, October 2000, pp. 667–678.

[Zhan89] Lixia Zhang, "A New Architecture for Packet Switching Network Protocols", *MIT/LCS/TR-455*, Laboratory for Computer Science, Massachusetts Institute of Technology, August 1989.

[ZhCo03] Hongyi Zhang, Jun Cong and Oliver W.W. Yang, "Rate Control Over RED With Data Loss and Varying Delays", *Proceedings of IEEE/Globecom*, San Francisco, U.S.A, December 2003, pp.3035 –3040.

[ZhHo03] Honggang Zhang, C.V. Hollot, Don Towsley and Vishal Misra, "A Self-Tuning Structure for Adaptation in TCP/AQM Networks", *Proceedings of IEEE Globecom 2003*, pp. 1346 –1355.

[ZhYa97] H. Zhang, O. Yang and H. Mouftah, "Design of robust congestion controllers for ATM networks", *Proceedings of IEEE INFOCOM*, 1997.

[ZhYa02] C. Zhu, O. Yang, J. Aweya, M. Ouellette, and D. Montuno, "A comparison of active queue management algorithms using the OPNET modeler", IEEE Communications Magazine, vol. 40, no. 6, June 2002, pp. 158-167.

[ZhYa04] Hongyi Zhang, Oliver W.W. Yang, "Global Stability of Utility-Based Congestion Control with Delays", *Proceedings of IEEE Globecom*, Dallas, TX, U.S.A, November 2004, pp. 1431-4135.

Appendix A The TCP Window-Based Source Control

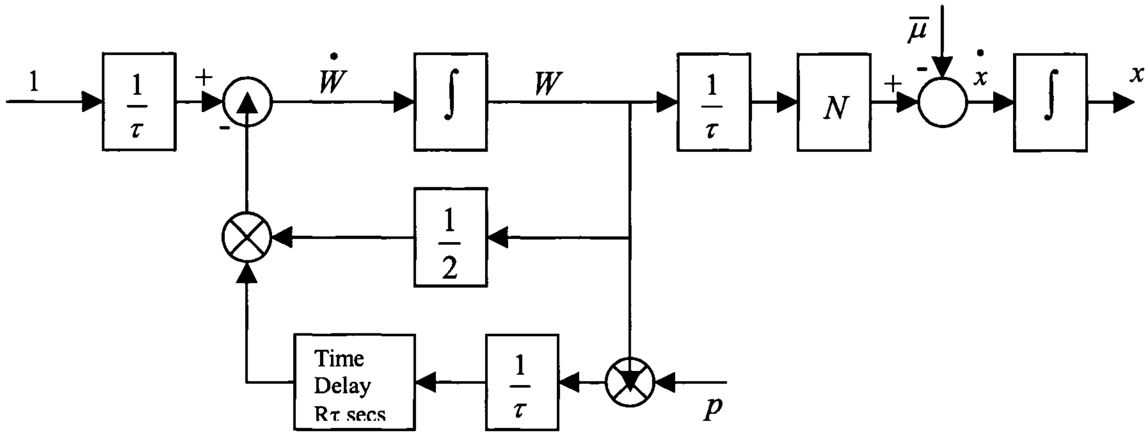

Figure A.1: TCP Window-Based Source Control

The non-linear dynamic model for TCP/AIMD/AQM is developed by [MiGo00] and is shown in Figure A.1. It is described by the following coupled, nonlinear differential equations:

$$\dot{W}(t) = \frac{1}{\tau(t)} - \frac{W(t)W(t-\tau(t))}{2\tau(t-\tau(t))} p(t-\tau(t)) \quad ,$$

$$\dot{x} = \frac{W(t)}{\tau(t)} N(t) - \mu \quad ,$$

where $\dot{x}$ denotes the time-derivative of x and μ is link capacity in packets/sec. It shows how the average queue length $x(t)$ reacts to the changes on the average window size $W(t)$, the average round-trip time $\tau(t)$, the traffic load $N(t)$, the link capacity μ and the nonlinear mechanism of the AQM algorithm (e.g., RED [FlJa93]). The RED algorithm will also be briefly introduced in Appendix B and further detailed information on the RED can be found in [FlJa93].

Appendix B The RED Controller

The compensator studied in [HoMi01b] is the well-known RED controller [FlJa93] consisting of a low-pass filter (LPF) and nonlinear gain element as shown in Figure A.1. The form of the LPF was derived in [HoMi01b] while nonlinear gain element is a mechanism that marks packets with a dropping probability p as a function of average queue size x_{avg}. As indicated in Figure A.1, p is varying between two queue thresholds min_{th} and max_{th}, with a slope of $L_{RED}=p_{max}/(max_{th}-min_{th})$. Combining the two elements, the transfer function model for RED [FlJa93] is $C_{RED}=L_{RED}/(S/K+1)$ where $K=\log_e(1-\alpha)/\delta$ and α is the queue averaging parameter while δ is the sampling frequency.

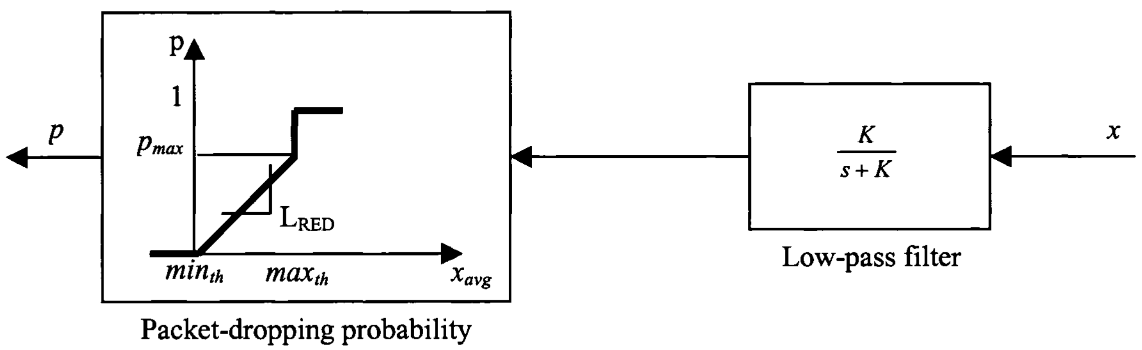

Figure A.2: RED as a Cascade of Low-Pass Filter and Nonlinear Gain Element

Based on the linearized model, design rules for obtaining a stable linear feedback control system with RED controller have been given in [HoMi01b]. However, such RED controller has a shortcoming of producing quite a long response time in the control system.

Appendix C The DNA Stability Theorem

Let the Gershgorin bands centered on the diagonal elements $g_{ii}(s)$ of $G(s)$ exclude the point (-1+j0). Let the *ith* Gershgorin band encircles the point (*-1*+j*0*) L_i times anticlockwise. Then, the closed-loop system is stable if and only if

$$\sum_{i=1}^{n} L_i = p_0$$

where p_0 is the number of unstable poles of $G(s)$ [Maci89].

Appendix D The Utility-Based Source Control Model

Following Kelly's work [KeMa98], we associate a route r with a source, and then assign a utility function $U_r(q_r)$ to each source r with respect to its transmission rate $q_r(t)$, so that we can formulate the resource allocation problem as a utility maximization problem:

$$\begin{aligned} &\max_{q\geq 0} \quad \sum_{R_r\in R} U_r(q_r) \\ &\text{subject to} \quad y \leq \mu \end{aligned} \quad , \tag{D.1}$$

where q is the vector of source transmission rates, μ is the vector of link capacities, and $y=Aq$. The constraint $y\leq\mu$ means that the aggregate incoming traffic rate $y_j = \sum_{R_r\in R} A_{jr} q_r$ at the router j cannot exceed the link capacity μ_j. Then we can obtain the Lagrangian [Whit71] for the problem in Equation (D.1) as

$$L(q,p) = \sum_{R_r\in R} U_r(q_r) + \sum_{j\in J} p_j(\mu_j - y_j) \quad . \tag{D.2}$$

where p_j is the Lagrange multiplier (also called the shadow price) associated with router j. We can interpret p_j as the price charged by Router j, i.e. per unit flow through Router j, on the total flow y_j through Router j. In network terminology, p_j represents the network congestion information from Router j (i.e. link price), which is transmitted by ACK packets back to inform the sources.

Since Router j does not know the utility functions of its sources, the optimal solution to the problem expressed by Equation (D.2) is generally not tractable. So we assign a logarithmic utility function $U_r(q_r)=w_r log(q_r)$ to each source r, where w_r is the price (per unit time) source r is willing to pay, or a weight assigned to the source (according to the price paid by the source). Therefore, in order to solve the problem expressed through Equation (D.1), we consider the following optimization problem:

$$L(q,p) = \max_{q\geq 0} \sum_{R_r\in R} w_r \log(q_r) - \sum_{j\in J} \int_0^{\sum_{s:j\in R_s} q_s} p_j(y)dy \quad . \tag{D.3}$$

where $\int_0^{\sum_{s:j\in R_s} x_s} p_j(y)dy$ is the total link price charged for all the incoming flows $\sum_{s:j\in R_s} q_s$, and $p_j(y)$ is the unit link price charged per flow. The optimal solution [KeMa98] to the problem expressed through (D.3) can be shown to be $q_r^* = w_r / \alpha_r^*$ with $\alpha_r^* = \sum_{j \in R_r} p_j$. Furthermore, a primal algorithm has been proposed in {KeMa98] to regulate the transmission rate of the source r in order to converge to $q_r^* = w_r / \alpha_r^*$ in the presence of no communication delays, i.e.

$$\dot{q}_r(t) = K_r \left(w_r - q_r(t) \sum_{j \in R_r} p_j \left(\sum_{s: j \in R_s} A_{js} q_s(t) \right) \right) \quad . \tag{D.4}$$

where K_r is the proportional gain, and $\sum_{j \in R_r} p_j \left(\sum_{s: j \in R_s} A_{js} q_s(t) \right)$ represents the total marking rate of the packets routed through a series of routers located between the source r and its destination [KeMa98]. It means that every source r adjusts its rate based on feedback information provided by the routers in the network in such as way that at an equilibrium the price w_r it is willing to pay equals to its aggregate cost [LaAn02]. It has been shown that with the utility-based control algorithm in Equation (D.4) applied, the network system without communication delays is globally stable if the marking function p_j is continuous, non-decreasing, non-negative and not identically zero [KeMa98].

Normally the communication delays in the Internet are not negligible, so we need to extend the primal algorithm in Equation (D.4) by including various delay components.

Appendix E The Ensemble Average of the Instantaneous Queue Lengths

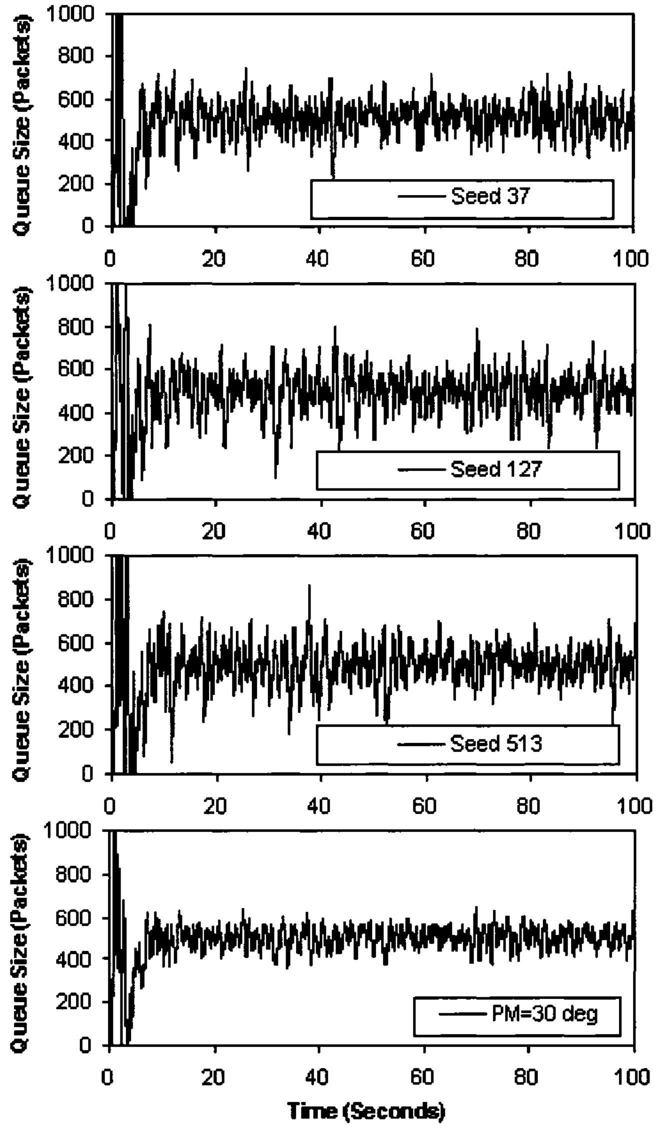

Figure E.1: Instantaneous Queue Size of the Fixed P Window-Based Router Controller with Phase Margin = 30°

We have conducted an interesting investigation in our simulations. One argument we received in the review of our work is that our instantaneous queue size only captures one instance of the time evolution of the control system and therefore may not be representative of the whole system. So here we repeat the simulation with different seeds (i.e., 37, 127 and 513) and determine their ensemble (statistical) average. For example, the top three figures of Fig.E.1 are the instantaneous queue size for a phase margin of 30 degrees when using different seeds, and the bottom figure is their ensemble average. We can see there is not much difference among the instantaneous queue sizes in the general behavior as well as their ensemble average.

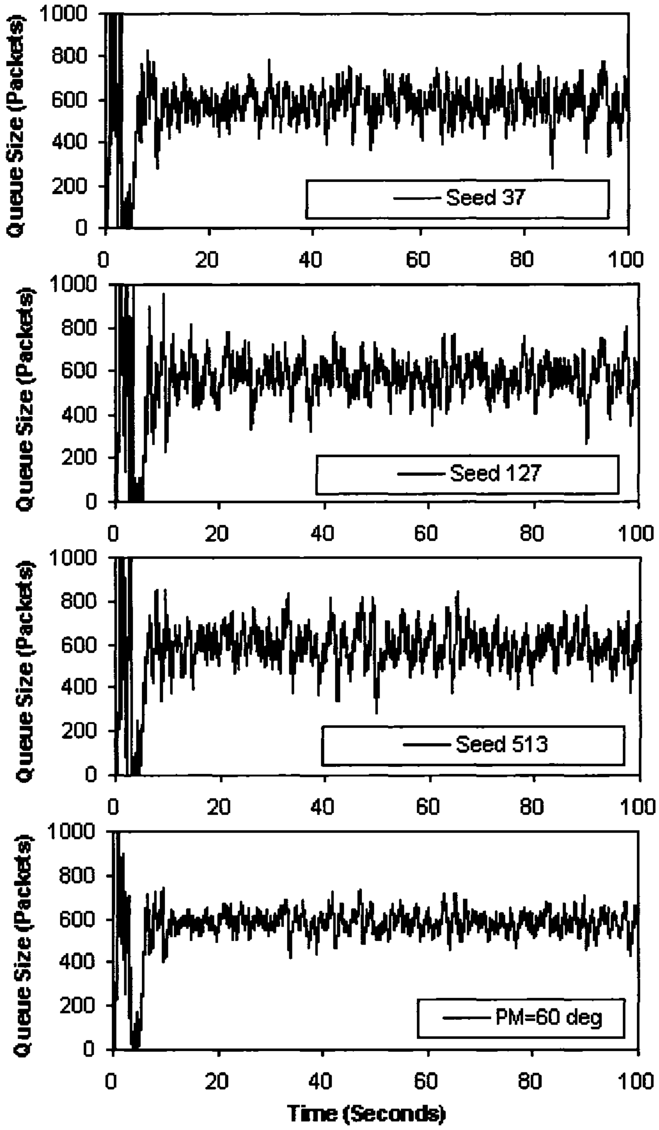

Figure E.2: Instantaneous Queue Size of the Fixed P Window-Based Router Controller with Phase Margin = *60°*

Similar observations are made in Figures E.2 and E.3 when the phase margins are respectively 60 degrees and 120 degrees. In fact, we have tested and found similar observations in other scenarios as well. Therefore, for the rest of the thesis, we have used the instantaneous queue length based on one seed, and assume that the ensemble average is most likely the same.

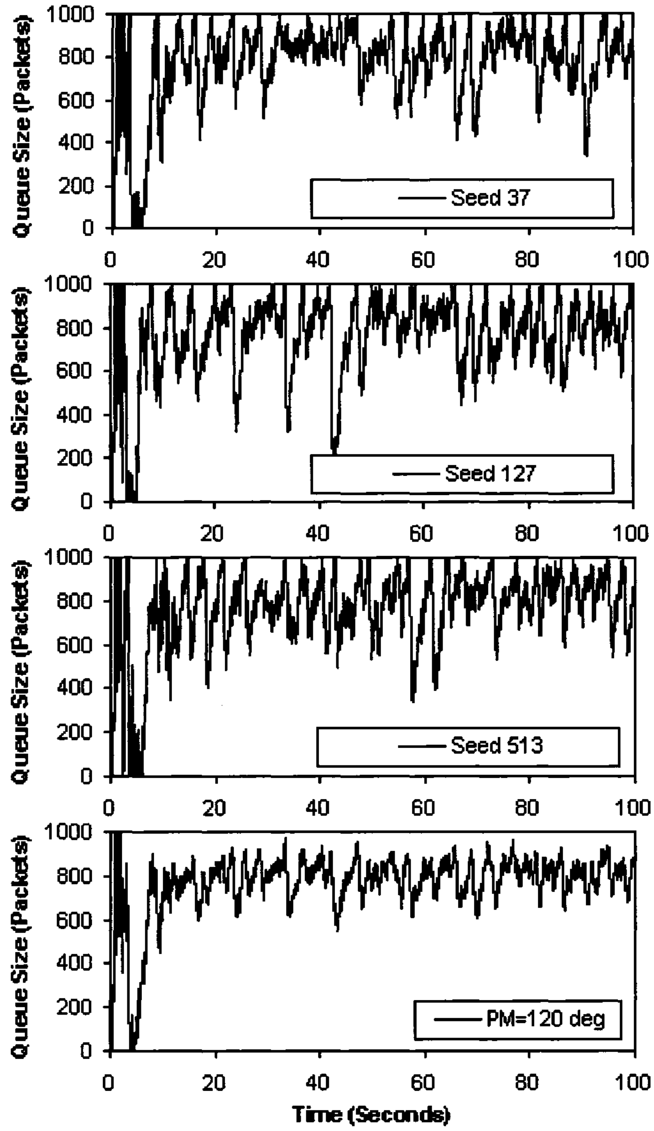

Figure E.3: Instantaneous Queue Size of the Fixed P Window-Based Router Controller with Phase Margin = *120°*