February 1999

UILU-ENG-99-2202
DC-187

University of Illinois at Urbana-Champaign

Multi-User Flow Control as a Nash Game: Performance of Various Algorithms

Rajiv Tharmeswaran Maheswaran

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801

SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 12, 1999	

4. TITLE AND SUBTITLE
Multi-User Flow Control as a Nash Game: Performance of Various Algorithms

5. FUNDING NUMBERS

6. AUTHOR(S)
MAHESWARAN, Rajiv Tharmeswaran

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801-2307

8. PERFORMING ORGANIZATION REPORT NUMBER
UILU-ENG-99-2202
DC-187

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Bolling Air Force Base, DC 20332-6448

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

We investigate the convergence of various update schemes that yield Nash equilibrium solutions for transmission policies of multiple users attempting to minimize deviations from an assigned transmission rate and deviations from a desired queue length in a bottleneck node of a telecommunication network. The environment where each user has several traffic types available for transmission is also investigated, and update algorithms that converge to Nash solutions are presented. Then, the objective function is further generalized to include penalties for drastic changes in transmission rates, and a method for obtaining a Nash equilibrium solution is developed and simulated. Finally, the behavior of the queue is simulated and accrued cost is calculated for the case where the algorithms are applied online to analyze the performance of the update schemes.

14. SUBJECT TERMS
Multi-user rate-based flow control, linear-quadratic control, linear-quadratic differential games, Nash equilibria, high speed networks, convergence of decentralized algorithms

15. NUMBER IF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

JUNE 1998

(date)

WE HEREBY RECOMMEND THAT THE THESIS BY

RAJIV THARMESWARAN MAHESWARAN

ENTITLED MULTI-USER FLOW CONTROL AS A NASH GAME:

PERFORMANCE OF VARIOUS ALGORITHMS

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

Tamer Basar

Director of Thesis Research

N. Narayana Rao

Head of Department

Committee on Final Examination†

Chairperson

† Required for doctor's degree but not for master's.

O-517

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

GRADUATE COLLEGE DEPARTMENTAL FORMAT APPROVAL

THIS IS TO CERTIFY THAT THE FORMAT AND QUALITY OF PRESENTATION OF THE THESIS SUBMITTED BY RAJIV THARMESWARAN MAHESWARAN AS ONE OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE ARE ACCEPTABLE TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING.

JUNE 24, 1998
Date of Approval

James Hutchinson
Departmental Representative
by Nina Parsons

MULTI–USER FLOW CONTROL AS A NASH GAME: PERFORMANCE OF VARIOUS ALGORITHMS

BY

RAJIV THARMESWARAN MAHESWARAN

B.S., University of Wisconsin–Madison, 1995

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1998

Urbana, Illinois

ABSTRACT

We investigate the convergence of various update schemes that yield Nash equilibrium solutions for transmission policies of multiple users attempting to minimize deviations from an assigned transmission rate and deviations from a desired queue length in a bottleneck node of a telecommunication network. The environment where each user has several traffic types available for transmission is also investigated, and update algorithms that converge to Nash solutions are presented. Then, the objective function is further generalized to include penalties for drastic changes in transmission rates, and a method for obtaining a Nash equilibrium solution is developed and simulated. Finally, the behavior of the queue is simulated and accrued cost is calculated for the case where the algorithms are applied online to analyze the performance of the update schemes.

ACKNOWLEDGMENTS

I would like to express my appreciation and gratitude to my advisor, Dr. Tamer Başar, for his invaluable guidance, patience and insight; to my parents, Dr. Murugesapillai Maheswaran and Nirmala Maheswaran and sister, Ahila Maheswaran, for their endless support and encouragement; and to my grandparents, Manonmani Ganesiah, Veluppillai Ganesiah, Buvaneswary Murugesapillai, and Kandiah Murugesapillai, who always have and always will inspire me to reach beyond my grasp.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

We consider a bottleneck node in a telecommunication network where several users are assigned a certain portion of the available bandwidth. It is often required that the queue at the node be close to a certain desired length to assure that the node is neither underutilized nor too congested. The goal for each user is to minimize deviations from their assigned bandwidth and deviations from the desired queue length.

Flow control is often performed dynamically, requiring feedback information to adjust the transmission rates. In TCP/IP, the user can obtain information about the state of the network from the advertised window that it receives from the node and also from the round-trip delay time for acknowledgments [1]. In the Available Bit Rate (ABR) mode of ATM, users receive feedback information from Resource Management (RM) cells that contain information about the congestion level of the network through a congestion indicator (CI) and supportable transmission rates through an explicit rate (ER) field [2].

In many telecommunication networks (such as the Internet and best-effort modes of ATM), flow control is performed in a decentralized manner where users are responsible for their own transmission policy and requires only their own information to determine

their policy. Cansever has derived algorithms that converge to Pareto-optimal solutions where users minimize the ratio of throughput to delay denoted as the power of the virtual circuit [3].

Any attempt to find policies in a decentralized manner that lead to a stable operating point for the network node invites game theoretic analysis. The condition where no user is willing to deviate from its current control is analogous to a Nash equilibrium in the game theoretic setting. Modelling telecommunication network problems as a dynamic game has produced Nash equilibria solutions in many settings such as capacity allocation in routing [4], congestion control in product form networks [5], and flow control in Markovian queueing networks [6].

Because flow control is often performed in a decentralized manner depending only on local information, individual users may not have all the information to obtain a Nash equilibrium solution. Thus, iterative algorithms based on feedback information that converge to Nash equilibrium controls that can be applied online are desirable. Another factor to consider is that users may wish to transmit more than one type of traffic. For example, in the Internet, one source may alternate between sending voice, video and data in the same connection. All three types of traffic have different quality-of-service requirements. Douligeris and Mazumdar have used a game theoretic perspective to show the existence of Nash equilibria in a multiclass traffic environment and have provided algorithms that converge under certain symmetry conditions [7], [8].

In this thesis, we consider various decentralized update schemes, where users need only local information and feedback from the node, and compare their rates of convergence

to established Nash equilibrium solutions [9]. We consider the case where users attempt to minimize deviations from an assigned bandwidth and deviations from a desired queue length. Algorithms for both single and multitype traffic structures are investigated. We also investigate the case where users are penalized for drastic variations in transmission rates. Finally, we study the behavior of the system and optimality of the algorithms when the update schemes are implemented online.

The thesis is organized as follows. In Chapter 2, the basic model and background is presented. In Chapter 3, several update schemes are presented and simulated. The convergence rates of the various schemes are compared and analyzed. In Chapter 4, the multitype traffic environment is introduced. An algorithm to reach a Nash equilibrium is derived and convergence for various update schemes is investigated. In Chapter 5, the objective of each user is expanded to the more general setting where there is a penalty on drastic changes in the transmission rate. A method for obtaining a Nash equilibrium solution is developed and simulated in the two-user case. In Chapter 6, the performance of the update schemes when implemented online are investigated through analysis of the behavior of the queue and cost accrued. In Chapter 7, some conclusions are given and areas for future research are identified.

CHAPTER 2

BASIC MODEL AND BACKGROUND

The basic model adopted is described by several users seeking service or bandwidth from a node with a known total available bandwidth denoted by $s(t)$, where t is the continuous time variable. Let $a_m(t)$ be the portion of the total available bandwidth assigned to the m-th user, where $m \in \mathcal{M} = \{1, \ldots, M\}$; M = number of customers currently accessing the node. Hence, the bandwidth available to user m at time t is $a_m s(t)$ and $\sum_{m=1}^{M} a_m = 1$. Let $q(t)$ denote the length of the queue at the node and $r_m(t)$ denote the transmission rate of user m, both at time t. The dynamics of the queue are then described by the following:

$$\frac{dq}{dt} = \sum_{m=1}^{M} r_m(t) - a_m s(t) \tag{2.1}$$

It is desired to minimize both the variance of the queue length from a desired queue length and the variance of the transmission rate of the users from the assigned bandwidth. We introduce two new variables, $x(t) := q(t) - \bar{Q}$ where $\bar{Q}$ is some desired queue length, and $u_m(t) := r_m(t) - a_m s(t)$. Thus, the queue dynamics can be modified and expressed as

$$\frac{dx}{dt} = \sum_{m=1}^{M} u_m(t) \tag{2.2}$$

To accomplish the stated goals, a quadratic cost function was introduced for each user minimizing the new variables around zero:

$$J_m(u) = \int_0^\infty \left(|x(t)|^2 + \frac{1}{c_m} |u_m(t)|^2 \right) dt \tag{2.3}$$

where c_m for each user m defines their ability to vary about their assigned bandwidth relative to the other users and their need to reduce the total queue length.

In this framework, we seek a state-feedback policy M-tuple $\mu_m^* := (\mu_1^*, \ldots, \mu_M^*)$, that would lead to a Nash equilibrium [10], i.e.,

$$J_m(\mu^*) = \inf_{\mu_m \in \mathcal{U}_m} J_m([\mu_m | \mu_{-m}^*]) \tag{2.4}$$

where $u_m(t) = \mu_m(x(t))$, $\mu_m(\cdot) \in \mathcal{U}_m$ (the space of all state-feedback policies for user m) and where $[\mu_m | \mu_{-m}^*]$ is the policy M-tuple where user i uses the policy μ_i^* if $i \neq m$ and user m uses μ_m. Equilibrium is reached when, given all the other users' policies, no improvement can be made by changing one's current policy. The multipolicy where this is true for all users simultaneously is a Nash equilibrium solution.

It has been shown in [9] that there exists a Nash equilibrium solution, which exhibits appealing linear dependence on x, as follows:

$$\mu_m^*(x) = -\beta_m^* x, \quad m = 1, \ldots, M \tag{2.5}$$

where $\beta_m^*, m \in \mathcal{M}$, are solved from the set of coupled M equations

$$\beta_m = f_m(\beta_{-m}) := -\beta_{-m} + \sqrt{\beta_{-m}^2 + c_m} \quad , \quad m \in \mathcal{M}, \tag{2.6}$$

where $\beta_{-m} := \sum_{i \neq m} \beta_i$. It was also shown that if the users employ the following decentralized algorithm,

$$\beta_m^{(n+1)} = \begin{cases} f_m(\beta_{-m}^{(n)}) & \text{if } m \in K_n \quad n = 0, 1, \ldots \\ \beta_{-m}^{(n)} & \text{otherwise} \end{cases} \tag{2.7}$$

where n stands for an iteration step, then we have local convergence to the optimal $\beta^* := (\beta_1^*, \ldots, \beta_M^*)$ that solves Equation (2.6) and hence to μ^*. The algorithm will also converge globally under certain initial conditions. In this algorithm, K_n is a set which determines which users update at the n-th iteration. The first investigation in this paper is how the rate of convergence of the algorithm above varies with different update scenarios (as defined by K_n).

CHAPTER 3

UPDATE SCHEMES AND CONVERGENCE

3.1 Description of Update Schemes

Given the update algorithm $\beta_m^{(n+1)} = f_m(\beta_{-m}^{(n)}), \quad n = 0, 1, \ldots$ it is not necessary nor desirable for every user to use it at every iteration. If users update too often they might be reacting to initial guesses that are far from the equilibrium values, thus prolonging the convergence. On the other hand, being idle for long periods of time naturally extends the convergence process as well. It is how to strike a balance between these scenarios that was the goal of the investigation. Three different update scenarios are defined for the algorithm described in Equation (2.7):

- *Parallel Update (PU)*: $K_n = \{1, \ldots, M\} \ \forall n$. Every user updates at every iteration.
- *Round Robin (RR)*: $K_n = \{(n + k) \mathrm{mod} M + 1)\}$ where k is an arbitrary integer. Users update sequentially and only one user updates at a time (i.e., each user updates every M iterations and is idle otherwise and only one user is active during an iteration).

- *Randomized Update (RU)*: $K_n \subset \{1, \ldots, M\}$ where $Prob[i \in K_n] = p, \quad i = 1, \ldots, M$. During a particular iteration, each user updates independently with probability p and is idle with probability $1 - p$.

To investigate the relative convergence of the update scenarios, each scheme was implemented using MATLAB. Convergence over both uniform and varying cost structures, $\{c_m\}$ (the latter was generated by a uniform distribution over a given segment of the positive real line) were investigated. Also, convergence over uniform and varying initial values for the algorithm, $\{\beta_m^{(0)}\}$, were investigated (as in the cost case, the latter was generated by a uniform distribution over a given segment of the positive real line). In the case of nonuniform initial values, the number of iterations to convergence was averaged over several trials with differing initial values generated uniformly over the same set.

The convergence rate was measured by the number of iterations necessary until the following stopping criterion is met:

$$\sum_{m=1}^{M} |\beta_m^{(n)} - \beta_m^{(n-N)}| \leq M\epsilon \tag{3.1}$$

where $\epsilon > 0$ is sufficiently small and N is defined by the following:

- $N = 1$ for Parallel Update (PU)
- $N = M$ for Round Robin (RR)
- $N = E[\tau]$ for Randomized Update (RU) where τ is the earliest iteration where $\beta_m^{(\tau)} \neq \beta_m^{(0)} \ \ \forall m \in M$ (i.e., the first time when all users have updated at least once).

The N's were chosen to give a common basis of comparison among the differing update schemes. With the above definition of N, the algorithm is run until the difference in the iterates between stages where it is known that every user has made at least one update is sufficiently small. We use $M\epsilon$ in the RHS of Equation (3.1) so that we are consistent when comparing simulations with different numbers of users. With such a criterion, we continue until the average absolute difference in the iterates is less than ϵ.

3.2 Comparison of Update Schemes

All three schemes were simulated with $\epsilon = 5 \times 10^{-5}$. The Randomized Update scheme was implemented with a probability of update of $p = 0.5$. The results of the simulations for the uniform cost, $c_m = 1 \ \forall m$, and uniform initial conditions, $\beta_m^{(0)} = 0 \ \forall m$, are shown in Figure 3.1. For Randomized Update, 10 trials were overlaid to show the effect of the variance of the update times on the number of iterations until meeting the stopping criterion. The results were similar when the users began with nonuniform initial conditions and nonuniform costs (both generated randomly).

It can be seen that the Randomized Update scheme significantly outperforms the Parallel update and Round Robin schemes, especially as the number of users increases. An intuitive explanation for this performance is that the Randomized Update finds a balance between overreacting to inaccurate information (Parallel Update) and being idle for too long between updates (Round Robin).

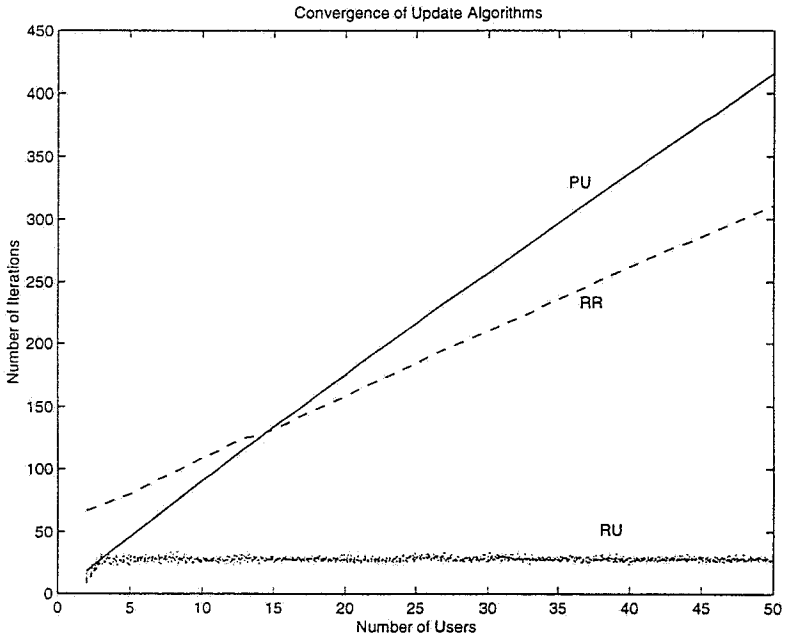

Figure 3.1 Convergence Rates for PU, RR, and RU for Uniform Costs and Zero Initial Values

3.3 Optimal Randomized Update

When implementing the Randomized Update scheme, an immediate question that follows is "which probability of update minimizes the number of iterations to convergence?" To investigate this, we can linearize the update algorithm around the final β values to yield the following one-step characterization:

$$\Delta\beta^{(n+1)} = B\Delta\beta^{(n)} + o(\Delta\beta^{(n)}) \tag{3.2}$$

$$\Delta\beta_k^{(n)} := \beta_k^{(n)} - \beta_k^* \tag{3.3}$$

where β_k^* is the optimal solution as shown in [9]. It is shown in [9] that the mk-th entry of B is given by

$$B_{mk} = \left.\frac{\partial f_m(\beta'_{-m})}{\partial \beta'_k}\right|_{\beta'=\beta^*} = b_m[1-\delta_{km}], \tag{3.4}$$

where

$$b_m := -1 + \frac{\beta^*_{-m}}{\sqrt{\beta^{*2}_{-m} + c_m}} = -1 + \frac{\beta^*_{-m}}{\bar{\beta}^*} \equiv -\frac{\beta^*_m}{\bar{\beta}^*} < 0 \tag{3.5}$$

and δ_{km} is the Dirac delta function. A sufficient condition for all eigenvalues of B to be in the interior of the unit disk is that $\sum_{k\neq m} |b_k| < 1$, which follows from Gersgorin's theorem [11]. Because $\sum_{k\neq m} |b_k| = \beta^*_{-m}/\bar{\beta} < 1$, we have local convergence of the Parallel update algorithm.

To model the Randomized Update, we introduce the Bernoulli random variable $Z_k^{(n)}$, such that $Prob[Z_k^{(n)} = 1] = p_k^{(n)}$ where $p_k^{(n)}$ is the probability that the k-th user updates on the n-th iteration. Thus, the Randomized Update linearized around the optimal solution is characterized by

$$\Delta\beta^{(n+1)} = \tilde{B}^{(n)}\Delta\beta^{(n)} + o(\Delta\beta^{(n)}) \tag{3.6}$$

where $\tilde{B}^{(n)}$ is obtained by multiplying the nondiagonal elements of the k-th row of B by $Z_k^{(n)}$ and replacing the diagonal element of the k-th row of B with $1 - Z_k^{(n)}$. With the simplifying assumption that $p_k^{(n)} = p_k$, we get

$$\Delta\beta_k^{(n+1)} = Z_k^{(n)} b_k \sum_{j\neq k} \Delta\beta_j^{(n)} + (1 - Z_k^{(n)})\Delta\beta_j^{(n)} + o(\Delta\beta) \tag{3.7}$$

$$|\Delta\beta_k^{(n+1)}| \leq Z_k^{(n)} |b_k| \sum_{j\neq k} |\Delta\beta_j^{(n)}| + (1 - Z_k^{(n)})|\Delta\beta_j^{(n)}| + o(\Delta\beta) \tag{3.8}$$

$$E|\Delta\beta_k^{(n+1)}| \leq p_k |b_k| \sum_{j\neq k} E|\Delta\beta_j^{(n)}| + (1 - p_k)E|\Delta\beta_j^{(n)}| + o(\Delta\beta) \tag{3.9}$$

where E denotes the expectation operator over the space generated by the Bernoulli random variables up to stage n. Let us choose p_k such that

$$p_k|b_k| \geq (1 - p_k) \quad i.e. \quad p_k \geq \frac{1}{1 + |b_k|}. \tag{3.10}$$

If the inequality (3.10) is satisfied, then inequality (3.9) becomes

$$E|\Delta\beta_k^{(n+1)}| \leq p_k |b_k| \sum_{j} E|\Delta\beta_j^{(n)}| + o(\Delta\beta) \tag{3.11}$$

$$\sum_k E|\Delta\beta_k^{(n+1)}| \leq \left(\sum_k p_k|b_k|\right) \cdot \left(\sum_j E|\Delta\beta_j^{(n)}|\right) + o(\Delta\beta) \tag{3.12}$$

Thus, for convergence, we need $\sum_k p_k|b_k| < 1$. Combining the previous with the requirement on p_k in (3.10), a necessary condition for convergence is $\sum_k \frac{|b_k|}{1+|b_k|} < 1$, which is always satisfied because $\sum_k |b_k| = 1$. Thus, we can always select a set of p_k's for which we have convergence (i.e., $p_k = \frac{1}{1+|b_k|}$). However, by analyzing the symmetric case ($p = \frac{1}{1+\frac{1}{M}}$) we get the convergence rate to be

$$p \cdot \frac{1}{M} \cdot (M-1) + (1-p) = 1 - \frac{p}{M} = 1 - \frac{1}{1+M} = \frac{M}{M+1} \tag{3.13}$$

which is worse than the convergence rate for Parallel Update ($\frac{M-1}{M}$) [9]. So this analysis has not answered the original question posed as to the optimal update probability. To answer this question, we analyze the two-step linearized update around the optimal solution

$$\Delta\beta_k^{(n+1)} = \sum_k \hat{B}_{mk} \Delta\beta_k^{(n-1)} + o(\Delta\beta) \tag{3.14}$$

where $\Delta\beta$ is defined as in Equation (3.3) and

$$\hat{B} = \tilde{B}^{(n)}\tilde{B}^{(n-1)} = \begin{bmatrix} (1-Z_1^{(n)}) & Z_1^{(n)}b_1 & Z_1^{(n)}b_1 & \cdots & Z_1^{(n)}b_1 \\ Z_2^{(n)}b_2 & (1-Z_2^{(n)}) & Z_2^{(n)}b_2 & \cdots & Z_2^{(n)}b_2 \\ Z_3^{(n)}b_3 & Z_1^{(n)}b_1 & (1-Z_3^{(n)}) & \cdots & Z_1^{(n)}b_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_M^{(n)}b_M & Z_M^{(n)}b_M & Z_M^{(n)}b_M & \cdots & (1-Z_M^{(n)}) \end{bmatrix}$$

$$\cdot \begin{bmatrix} (1-Z_1^{(n-1)}) & Z_1^{(n-1)}b_1 & Z_1^{(n-1)}b_1 & \cdots & Z_1^{(n-1)}b_1 \\ Z_2^{(n-1)}b_2 & (1-Z_2^{(n-1)}) & Z_2^{(n-1)}b_2 & \cdots & Z_2^{(n-1)}b_2 \\ Z_3^{(n-1)}b_3 & Z_1^{(n-1)}b_1 & (1-Z_3^{(n-1)}) & \cdots & Z_1^{(n-1)}b_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_M^{(n-1)}b_M & Z_M^{(n-1)}b_M & Z_M^{(n-1)}b_M & \cdots & (1-Z_M^{(n-1)}) \end{bmatrix} \tag{3.15}$$

where

$$\hat{B}_{mm} = (1 - Z_m^{(n)})(1 - Z_m^{(n-1)}) + \sum_{j \neq m} Z_m^{(n)} Z_j^{(n-1)} b_m b_j \tag{3.16}$$

$$\hat{B}_{mk} = (1 - Z_m^{(n)}) Z_k^{(n-1)} b_m + Z_m^{(n)} (1 - Z_k^{(n-1)}) b_m + \sum_{j \neq m,k} Z_m^{(n)} Z_j^{(n-1)} b_m b_j. \tag{3.17}$$

By replacing $\Delta\beta_k^{(n-1)}$ with the maximum of all such values over all users and taking expectation over the space generated by the Bernoulli random variables in Equation (3.14), we have

$$E|\Delta\beta_m^{(n+1)}| \leq \sum_k E\{|\hat{B}_{mk}| \cdot \max_j |\Delta\beta_j^{(n-1)}|\} + o(\Delta\beta) \tag{3.18}$$

$$\max_m E|\Delta\beta_m^{(n+1)}| \leq \max_m \left(\left(\sum_k E|\hat{B}_{mk}| \right) \left(\max_j E|\Delta\beta_j^{(n-1)}| \right) \right) + o(\Delta\beta) \tag{3.19}$$

$$\| \Delta\beta^{(n+1)} \| \leq \left(\max_m \sum_k E|\hat{B}_{mk}| \right) \left(\| \Delta\beta^{(n-1)} \| \right) \tag{3.20}$$

From our structure of the Randomized Update model, we can assume that $\{Z_k^{(n-1)}\}$ is independent of $\{Z_k^{(n)}\}$. Because $Z_m^{(n)}, (1 - Z_m^{(n)}), b_m b_k$ are all nonnegative, from Equation (3.16), it follows that $\hat{B}_{mm} \geq 0$ (a.s.). Hence,

$$E[|\hat{B}_{mm}|] = E[\hat{B}_{mm}] = (1 - p_m^{(n)})(1 - p_m^{(n-1)}) + \sum_{j \neq m} p_m^{(n)} p_j^{(n-1)} b_m b_j \tag{3.21}$$

For $k \neq m$,

$$
\begin{aligned}
E[|\hat{B}_{mk}|] &= E\{|\hat{B}_{mk}| \big| Z_m^{(n)} = 1, Z_k^{(n-1)} = 1\} \cdot \text{Prob}(Z_m^{(n)} = 1, Z_k^{(n-1)} = 1) \\
&\quad + E\{|\hat{B}_{mk}| \big| Z_m^{(n)} = 1, Z_k^{(n-1)} = 0\} \cdot \text{Prob}(Z_m^{(n)} = 1, Z_k^{(n-1)} = 0) \\
&\quad + E\{|\hat{B}_{mk}| \big| Z_m^{(n)} = 0\} \cdot \text{Prob}(Z_m^{(n)} = 0) \qquad (3.22) \\
&= \Big| \sum_{j \neq m,k} p_j^{(n-1)} b_m b_j \Big| \; p_m^{(n)} p_k^{(n-1)} \\
&\quad + \Big| b_m + \sum_{j \neq m,k} p_j^{(n-1)} b_m b_j \Big| \; p_m^{(n)} (1 - p_k^{(n-1)}) \\
&\quad + \Big| b_m p_k^{(n-1)} \Big| \; (1 - p_m^{(n)}) \qquad (3.23)
\end{aligned}
$$

where the second equality is obtained by substituting from Equation (3.17) and taking expectations. The next issue is dealing with the absolute values in Equation (3.23). In the first term, we can drop the absolute value because $b_m b_j,\ p_j^{(n-1)} > 0$. In the second term we can replace $|\cdot|$ with $-(\cdot)$ since $(1 + \sum_{j \neq m,k} p_j^{(n-1)} b_j) > 0$ and $b_m < 0$. In the final term, we can again replace $|\cdot|$ with $-(\cdot)$ because $p_k^{(n-1)} > 0$ and $b_m < 0$. This yields

$$
\begin{aligned}
E[|\hat{B}_{mk}|] &= \sum_{j \neq m,k} p_j^{(n-1)} b_m b_j p_m^{(n)} p_k^{(n-1)} \\
&\quad - (b_m + \sum_{j \neq m,k} p_j^{(n-1)} b_m b_j) \; p_m^{(n)} (1 - p_k^{(n-1)}) \\
&\quad - b_m p_k^{(n-1)} (1 - p_m^{(n)}) \qquad (3.24) \\
&= (\sum_{j \neq m,k} p_j^{(n-1)} b_m b_j p_m^{(n)}) (2 p_k^{(n-1)} - 1) \\
&\quad - b_m (p_m^{(n)} (1 - p_k^{(n-1)}) + p_k^{(n-1)} (1 - p_m^{(n)}) \qquad (3.25)
\end{aligned}
$$

By substituting Equation (3.21) and Equation (3.25) into Equation (3.20), the condition for contraction is $\alpha < 1$ where

$$\begin{aligned}\alpha \;:=\; \max_{m,n}\Bigg\{ & (1-p_m^{(n)})(1-p_m^{(n-1)}) + \sum_{j\neq m} p_m^{(n)} p_j^{(n-1)} b_m b_j \\ & + \sum_{k\neq m}\left((\sum_{j\neq m,k} p_j^{(n-1)} b_m b_j p_m^{(n)})(2p_k^{(n-1)}-1)\right) \\ & - b_m \sum_{k\neq m}\left(p_m^{(n)}(1-p_k^{(n-1)}) + p_k^{(n-1)}(1-p_m^{(n)})\right)\Bigg\} \end{aligned} \tag{3.26}$$

Under the case where $p_j^{(n)} = p \;\; \forall j, n$, we have

$$\alpha = \max_{m,n}\left\{(1-p)^2 + p^2 b_m \sum_{j\neq m} b_j + p^2(2p-1)b_m(M-2)\sum_{j\neq m} b_j - 2(M-1)p(1-p)b_m\right\}. \tag{3.27}$$

Furthermore, for the symmetric case where $b_m = \frac{-1}{M} = b_j \;\; \forall j$, we have

$$\alpha = (1-p)[1-p+2(\frac{M-1}{M})p] + p^2(\frac{M-1}{M^2})[1+(2p-1)(M-2)] \tag{3.28}$$

$$= (1-p)[1+(\frac{M-2}{M})p] + p^2(\frac{M-1}{M^2})[1+(2p-1)(M-2)] \tag{3.29}$$

If $M \gg 1$

$$\alpha \rightarrow (1-p)(1+p) + p^2(2p-1) \tag{3.30}$$

$$= 1 - p^2 + 2p^3 - p^2 \tag{3.31}$$

$$= 2p^3 - 2p^2 + 1 \tag{3.32}$$

$$\min_p \alpha(p) \Rightarrow 6p^2 - 4p = 0 \Rightarrow p = \frac{2}{3} \tag{3.33}$$

$$\alpha(\frac{2}{3}) = \frac{19}{27} \tag{3.34}$$

Thus, under uniform costs and constant update probability we have obtained the probability of update under which the deviation from the optimal solution diminishes the fastest.

3.4 Performance of Randomized Update

To verify the validity of the result (3.33), the Randomized Update scheme was simulated over the probability range $p \in [0.3, 0.9]$ using uniform costs and zero initial values with $\epsilon = 5 \times 10^{-7}$. Because the derivation of the best update probability depended on local analysis, an adjusted iteration counter was also introduced. This counter began counting iterations only after $\sum_{m=1}^{M} |\beta_m^{(n)} - \beta_m^{(n-N)}| \leq M\bar{\epsilon}$ where $\bar{\epsilon} = 0.5$. This had the effect of only considering the rate of convergence once the update variables were within a sufficiently small neighborhood of the optimal solution, thus mirroring our assumptions in the derivation more closely. The results of the simulation are shown in Figure 3.2.

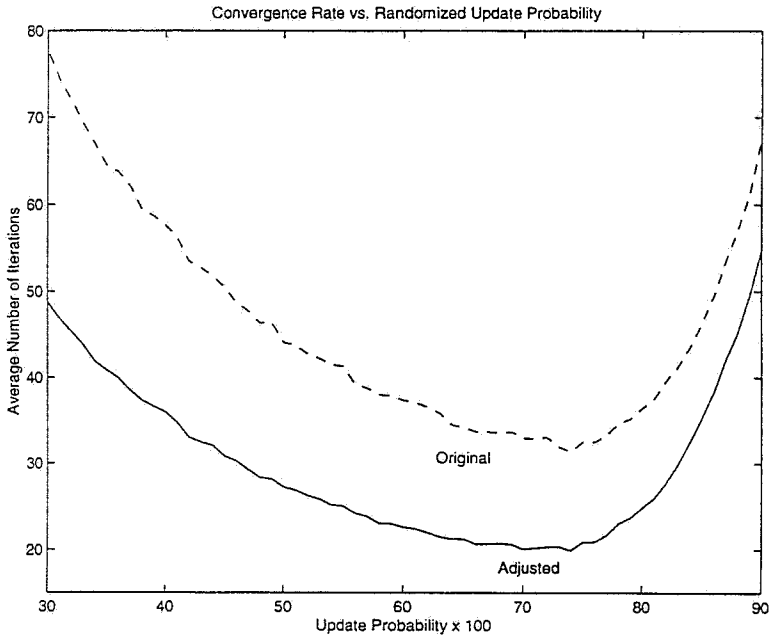

Figure 3.2 Original and Adjusted Convergence Rate of Randomized Update Scheme versus Update Probability

From the simulations, the optimal update probability seems to lie at approximately 0.74. However, it is noted that the rate of convergence does not vary significantly in the interval $[0.65, 0.75]$, verifying that the theoretical optimal update probability yields a rate of convergence very comparable to the optimal rate derived from the simulations. The effects of the bounds made in the derivation and other sources of the discrepancy is still an issue to be investigated.

CHAPTER 4

MULTITYPE TRAFFIC

4.1 Multitype Traffic Model

In the previous model, transmission policies were calculated based on the assumption that only one type of traffic was available to each user m, and the cost for that traffic type was denoted by c_m. We now consider the case where each user has several types of traffic available. At each time t, the user m transmits traffic type $i_m \in \mathcal{S}_m \subset \Omega$ where Ω is the set of all available traffic types and $|\mathcal{S}_m| = s_m$. Thus at each time t, there exists a traffic constellation $i = i_1 i_2 \cdots i_M \in \mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \mathcal{S}_M$ which is the M-tuple of traffic types being transmitted by the users. The behavior of the traffic is modeled by a continuous time Markov jump process taking values in a finite state space $\mathcal{S}$ $(s := |\mathcal{S}| = \Pi_{i=1}^{M} s_i)$. An element $\theta \in \mathcal{S}$ represents a possible traffic constellation of the M users. The jump process is characterized by the transition rate matrix $\Lambda = \{\lambda_{ij}\}$, $i, j \in \mathcal{S}$, where the λ_{ij}'s are real numbers such that $\lambda_{ij} \geq 0 \quad \forall i \neq j$ and $\lambda_{ii} = -\sum_{j \neq i} \lambda_{ij} \quad \forall i \in \mathcal{S}$.

In our earlier cost structure described in Equation (2.3), the value of c_m reflects the balance that the m-th user wishes to achieve between minimizing the deviation from the desired queue length and minimizing the deviation from assigned bandwidth. Similarly,

under the multitype structure, each user can associate the performance goals under a given constellation i, with the cost $c_m(i)$. Thus, if the m-th user wished to put more emphasis on minimizing deviation in queue length in constellation i when compared to constellation j, the user would choose $c_m(i) > c_m(j)$ to allow higher variations in the transmission rate to facilitate quicker convergence to the desired queue length. Under the multitype structure, each user minimizes the cost

$$J_m(\mu) := E^{\mu}\left[\int_0^{\infty}\left(|x(t)|^2 + \frac{1}{c_m(\theta(t))}|u_m(t)|^2\right)dt\Big|x(0) = x_0, \theta(0) = i_0\right] \tag{4.1}$$

where $i_0 \in \mathcal{S}$, $\theta(t)$ is the value of the Markov jump process at time t, and E^{μ} denotes the expectation operator under multipolicy μ.

It was shown in [9] that when each user minimizes the cost defined above, there exists a Nash equilibrium given by $\mu_m(x,i) = -\beta_m(i)x, \quad m = 1,\ldots,M$, where $\beta_m(i) = c_m(i)P_m(i)$, and $\{P_m(i), i \in \mathcal{S}, m \in \mathcal{M}\}$ is defined through the following coupled equations, where $\beta_{-m} := \sum_{j\neq m}\beta_j$:

$$-2\beta_{-m}(i)P_m(i) - P_m(i)^2 c_m(i) + 1 + \sum_{j\in\mathcal{S}}\lambda_{ij}P_m(j) = 0, \qquad i \in \mathcal{S},\ m = 1,\ldots,M. \tag{4.2}$$

4.2 Derivation of Algorithm

Substituting for $P_m(i)$ in Equation (4.2) and multiplying by $c_m(i)$, for $m = 1, 2, \ldots, M$, we have

$$-2\beta_{-m}(i)\beta_m(i) - \beta_m(i)^2 + c_m(i) + \sum_{j\in\mathcal{S}} \lambda_{ij}\frac{\beta_m(j)}{c_m j}c_m(i) = 0. \tag{4.3}$$

Removing the $\beta_m(i)$ term from the summation and grouping it with the first term, we have

$$-\beta_m(i)^2 + \beta_m(i)\left(-2\beta_{-m}(i) + \lambda_{ii}\right) + c_m(i) + \sum_{j\in\mathcal{S}, j\neq i} \lambda_{ij}\frac{\beta_m(j)}{c_m j}c_m(i) = 0 \tag{4.4}$$

$$\beta_m(i)^2 + \beta_m(i)\left(2\beta_{-m}(i) - \lambda_{ii}\right) - c_m(i)\left(1 + \sum_{j\in\mathcal{S}, j\neq i} \lambda_{ij}\frac{\beta_m(j)}{c_m j}\right) = 0 \tag{4.5}$$

By solving the quadratic equation (4.5), we obtain a solution for $\beta_m(i)$ as a function of $\beta_{-m}(i)$, $c_m(i)$ $\forall i \in \mathcal{S}$, and λ_{ij} $\forall j \in \mathcal{S}$, where $\beta_{-m}(i) := \sum_{l\neq m}\beta_l(i)$ is the only term that does not solely depend on information pertaining to user m. Thus, if the server broadcast $\bar{\beta}_m^{(n)}(i) := \sum_{l=1}^{M}\beta_l(i)$ $\forall i \in \mathcal{S}$ at every iteration n , user m could use the solution of the quadratic equation (4.5) to obtain iterates using the following decentralized algorithm:

$$\beta_m^{(n+1)}(i) = -(\beta_{-m}^{(n)} - \frac{\lambda_{ii}}{2}) + \sqrt{(\beta_{-m}^{(n)} - \frac{\lambda_{ii}}{2})^2 + c_m(i)\left[1 + \sum_{j\neq i}\lambda_{ij}\frac{\beta_m^{(n)}(j)}{c_m(j)}\right]} \tag{4.6}$$

Under this algorithm, each user has a value $\beta_m(i)$ for every possible traffic constellation, thus each can make up to s updates whenever feedback is received. This algorithm requires the server not only to broadcast $\bar{\beta}_m^{(n)}(i)$ for all traffic constellations but also the current constellation under which the system is operating, allowing the user to know the value of $\beta_m(i)$ to use online.

Simulations have shown both that the algorithm converges for a robust set of initial conditions and also that the maximum eigenvalue of the first-order derivative matrix obtained from local analysis remains less than one throughout the iterations but an analytical proof of convergence is lacking. Under various simulations, the maximum eigenvalue has reached values in excess of 0.99 and even Gersgorin's theorem [11] does not offer a tight enough bound to show convergence. In the following, we investigate how the convergence of the multitype traffic algorithm varies for update scenarios similar to those implemented in the original algorithm (Parallel Update, Round Robin, and Randomized).

4.3 Multitype Traffic Update Schemes

As with the algorithm presented earlier through Equations (2.6) and (2.7), it is of interest to investigate the effects of the differing update schemes on the algorithm described by Equation (4.6). Because the update variables $\{\beta_m^{(n)}(i)\}$ incorporate both the user and the traffic constellation, the algorithm solves for sM variables, where M is the number of users and s is the total number of traffic constellations. We say that $\beta_m^{(n)}(i)$

updates on the n-th iteration if $m \in K_n$ and $i \in L_n$ and is idle otherwise. The update schemes are defined by K_n and L_n as follows:

- *Parallel Update (PU)*: $K_n = \{1, \ldots, M\}, \quad L_n = \{1, \ldots, s\} \quad \forall n.$
- *Round Robin by User (RR-1)*: $K_n = \{(n+k) \bmod M + 1)\}, \quad L_n = \{1, \ldots, s\}$, where k is an arbitrary integer.
- *Round Robin by Traffic (RR-2)*: $K_n = \{1, \ldots, M\}, \quad L_n = \{(n+k) \bmod s + 1)\}$, where k is an arbitrary integer.
- *Randomized Update (RU)*: $K_n \subset \{1, \ldots, M\}, \quad L_n \subset \{1, \ldots, s\}$, where $Prob[m \in K_n, i \in L_n] = p, \qquad m = 1, \ldots, M \quad i = 1, \ldots, s.$

We also replace the previous stopping criterion described in Equation (3.1) with

$$\sum_{m=1}^{M} |\beta_m^{(n)} - \beta_m^{(n-N)}| \leq Ms\,\epsilon \tag{4.7}$$

where N is defined by the following

- $N = 1$ for Parallel Update (PU)
- $N = M$ for Round Robin by User (RR-1)
- $N = s$ for Round Robin by Traffic (RR-2)
- $N = E[\tau]$ for Randomized Update (RU) where τ is the earliest iteration where $\beta_m^{(\tau)}(i) \neq \beta_m^{(0)}(i) \quad \forall m \in M, \quad i \in \mathcal{S}$ (i.e., the first time when all update variables have updated at least once).

4.4 Comparison of Update Schemes

The algorithm presented in Equation (4.6) was implemented using MATLAB under the four mentioned update scenarios under uniform costs and zero initial conditions with $\epsilon = 5 \times 10^{-5}$. The results of the simulations for the uniform cost, $c_m(i) = 1 \ \ \forall m, i$, and uniform initial conditions, $\beta_m(i)^{(0)} = 0 \ \ \forall m$, are shown in Figure 4.1. The number of traffic constellations, s, was set to three. The number of iterations to convergence was averaged over 50 trials where the matrix Λ was generated randomly for each trial. The Randomized Update scheme was implemented with a probability of update of $p = 0.5$. The results were similar when the users began with nonuniform initial conditions and nonuniform costs (both generated randomly).

It can be seen, as before, that the Randomized Update significantly outperforms the other update schemes. The optimal update probability for the algorithm presented in Equation (4.6) is still open for investigation.

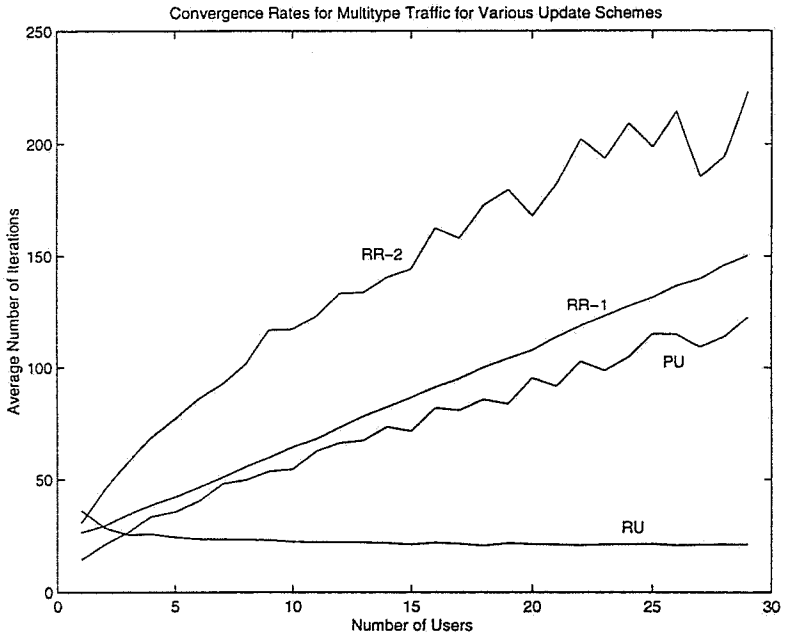

Figure 4.1 Convergence Rates for Multitype Traffic for PU, RU, RR-1 and RR-2 for Uniform Costs and Zero Initial Values

CHAPTER 5

JITTER COST

We now consider a more general class of performance measures by placing a cost on jitter (i.e., penalizing excessive variations in the transmission rate). The cost function for user m is accordingly modified as follows:

$$J_m(u) = \int_0^\infty \left(|x(t)|^2 + \frac{1}{c_m}|u_m(t)|^2 + \frac{1}{d_m}|\frac{d}{dt}r_m(t)|^2 \right) dt, \tag{5.1}$$

where d_m determines how quickly user m may vary the rate of transmission relative to his or her needs to reduce variance from a desired queue length and reduce variance from an assigned bandwidth.

An issue of importance in the jitter case is what variables user m's transmission rate at time t will be allowed to depend on. The control variable for which we will solve here is the rate of change of the transmission rate of user m, which we denote by v_m, i.e.,

$$v_m(t) := \frac{d}{dt}r_m(t) \tag{5.2}$$

The easiest, as far as analytic derivation goes, is to let $v_m(t)$ depend not only on $x(t)$, but also on $u(t) := (u_1(t), \ldots, u_M(t))$. Hence the new state in this case is

$$\underline{x} := [x, u_1, \ldots, u_M]^T, \tag{5.3}$$

which evolves according to

$$\frac{d}{dt}\underline{x} = A\underline{x} + \sum_{i=1}^{M} B_i v_i + c \tag{5.4}$$

where

$$A = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad B_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad c = \begin{bmatrix} 0 \\ -a_1\dot{s}(t) \\ \vdots \\ -a_M\dot{s}(t) \end{bmatrix} \tag{5.5}$$

with the nonzero term in B_i being the $(i+1)$-th entry and $\dot{s}(t) = \frac{d}{dt}s(t)$.

In conformity with this, we can also rewrite the cost function for the m-th user into standard LQR form in the following manner:

$$J_m = \int_0^\infty \left(\underline{x}^T Q_m \underline{x} + \frac{1}{d_m}(v_m)^2 \right) \; dt \tag{5.6}$$

$$Q_m = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & 0 & 0 & 0 & \vdots \\ \vdots & \vdots & 0 & \frac{1}{c_m} & 0 & \vdots \\ \vdots & \vdots & 0 & 0 & 0 & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \tag{5.7}$$

where $\frac{1}{c_m}$ is the $(m+1)$-th diagonal element of Q_m.

A Nash equilibrium solution for this problem (which turns out to be strongly time consistent with respect to $\underline{x}$) [10] is:

$$v_m = -d_m B_m^T (P_m \underline{x} + \zeta_m) \quad m \in \mathcal{M} \tag{5.8}$$

where $\{\zeta_m\}$ are obtained from the following coupled linear equations:

$$\zeta_m + \tilde{F}^T \zeta_m + d_m P_m B_m B_m^T \zeta_m + P_m (c - \sum_{i \in \mathcal{M}} d_i B_i B_i^T \zeta_i) = 0 \qquad m = 1, \ldots, M \tag{5.9}$$

and P_m's solve the coupled Riccati equations:

$$P_m \tilde{F} + \tilde{F} P_m + d_m P_m B_m B_m^T P_m + Q_m = 0 \tag{5.10}$$

where

$$\tilde{F} := A - \sum_{i=1}^{M} d_i B_i B_i^T P_i, \tag{5.11}$$

all this being true provided that such P_i's exist that make $\tilde{F}$ stable.

The question that remains is "how do we go about determining such P_m's?" Towards this end, let us assume that we begin with some initial guess, $\{P_1^{(0)}, \ldots, P_M^{(0)}\}$, that makes $\tilde{F}$ stable and then use an algorithm of the form

$$P_m^{(n+1)} = f_m(P_1^{(n)}, \ldots, P_{m-1}^{(n)}, P_{m+1}^{(n)}, \ldots, P_M^{(n)}), \qquad m = 1, \ldots, M \tag{5.12}$$

where the function f_m is defined implicitly via the corresponding relation:

$$P_m^{(n+1)} \tilde{F}^{(n)} + \tilde{F}^{(n)^T} P_m^{(n+1)} + d_m P_m^{(n+1)} B_m B_m^T P_m^{(n+1)} + Q_m = 0 \tag{5.13}$$

$$\tilde{F}^{(n)} := A - \sum_{i=1}^{M} d_i B_i B_i^T P_i^{(n)} \tag{5.14}$$

Thus, at every iteration it is necessary to solve M independent Riccati equations given the values of $\{P_m\}$ from the previous iteration. Simulations of this algorithm show that it does indeed converge to a solution for the two-user case. It can further be shown that the optimal cost of a jitter problem obtained from using this algorithm approaches the optimal cost of the no-jitter solution discussed earlier as $d_m \to \infty$ for the two-user case. Simulations results that verify this can be seen in Appendix C.

The information structure that we have picked, which allowed v_m to depend not only on x but also on u, may not be desirable for the problem at hand because online implementation of the policies that emerge from this may not be possible, with each user required to have access to the instantaneous transmission rates of the other users. A more realistic information structure is the one where v_m depends at time t only on the present and possibly past values of x. Derivation of a Nash equilibrium solution in this case is a more challenging task as the standard sufficiency results based on dynamic programming [10] do not apply here.

CHAPTER 6

ONLINE PERFORMANCE

6.1 Online Queue Dynamics and Accrued Cost

We have analyzed in the previous chapters the rate of convergence of various update scenarios and seen that they converge to a stable equilibrium solution. We now investigate the behavior of the queue length and cost accrued to each user when the updates are done online. If the users employ controls of the structure $u(t) = -\beta(t)x(t)$, the queue evolves according to the following dynamics:

$$\frac{dx}{dt} = \sum_{i=1}^{M} u_i = -\sum_{i=1}^{M} \beta_i(t)x = -x\sum_{i=1}^{M} \beta_i(t) =: -x\bar{\beta}(t), \tag{6.1}$$

where the last equality is due to the definition of $\bar{\beta}$. Let us assume that the users begin using the controls $u_m = -\beta_m^{(0)}x,\ m = 1, 2, \ldots, M$ at time t_0, where $\beta_m^{(0)}$ is some positive real number for all users. Let the users follow a decentralized update scheme where the n-th update occurs at $t_n = \sum_{j=0}^{n-1} \delta_j$ for $n \geq 1$. This assumes that $\{\delta_n\}$ is the sequence of delay times required for the server to get information back to the users and all users receive this information at the same time. The users' controls can be described as follows:

$$u_i(t) = -\beta_i^{(n)}(t)x(t) \qquad t_n \leq t \leq t_{n+1} = t_n + \delta_n \tag{6.2}$$

where $\beta_i^{(n)}$ is the positive real number which is the n-th iterate of the i-th user for the update scheme being used. Substituting into Equation (6.1), we have

$$\frac{dx}{dt} = -x\bar{\beta}^{(n)} \qquad t_n \leq t \leq t_{n+1} \tag{6.3}$$

where $\bar{\beta}^{(n)} := \sum_{i=1}^{M} \beta_i^{(n)}$ is a positive real number. Solving for $x(t)$, we have

$$x(t) = x(t_n)\ e^{-\bar{\beta}^{(n)}(t-t_n)}, \qquad t_n \leq t \leq t_{n+1} \tag{6.4}$$

Let $\delta_n = \delta \ \ \forall n$. Then, given the initial deviation from the desired queue length x_0 at t_0, x(t) evolves as follows:

$$x(t) = x_0\ e^{-\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}}\ e^{-\bar{\beta}^{(n)}(t-n\delta)} \qquad t_0 + n\delta \leq t \leq t_0 + (n+1)\delta, \tag{6.5}$$

where $\sum_{l=0}^{-1} \bar{\beta}^{(l)} = 0$ and n is any nonnegative integer.

By substituting for the control in the cost for the m-th user described in Equation (2.3), we have

$$\begin{aligned} J_m &= \int_0^\infty x(t)^2 + \frac{1}{c_m}\left(-\beta_m(t)x(t)\right)^2 dt \\ &= \int_0^\infty \left(1 + \frac{\beta_m(t)^2}{c_m}\right) x(t)^2 dt. \end{aligned} \tag{6.6}$$

But because $\beta_m(t)$ takes on the value of the iterates and is constant during intervals of length δ, we have

$$J_m = \sum_{n=0}^{\infty} \int_{n\delta}^{(n+1)\delta} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) x(t)^2 dt. \tag{6.7}$$

Substituting the expression in Equation (6.5) for $x(t)$, we have

$$\begin{aligned}
J_m &= \sum_{n=0}^{\infty} \int_{n\delta}^{(n+1)\delta} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) \left[x_0^2 \, e^{-2\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}} \, e^{-2\bar{\beta}^{(n)}(t-n\delta)}\right] dt \\
&= x_0^2 \sum_{n=0}^{\infty} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) \left(e^{-2\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}}\right) \left(e^{2\bar{\beta}^{(n)} n\delta}\right) \int_{n\delta}^{(n+1)\delta} e^{-2\bar{\beta}^{(n)} t} dt \\
&= x_0^2 \sum_{n=0}^{\infty} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) \left(e^{2\bar{\beta}^{(n)} n\delta - 2\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}}\right) \frac{e^{-2\bar{\beta}^{(n)} t}}{-2\bar{\beta}^{(n)}} \Bigg|_{n\delta}^{(n+1)\delta} \\
&= x_0^2 \sum_{n=0}^{\infty} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) \left(e^{2\bar{\beta}^{(n)} n\delta - 2\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}}\right) \frac{e^{-2\bar{\beta}^{(n)} n\delta} - e^{-2\bar{\beta}^{(n)}(n+1)\delta}}{2\bar{\beta}^{(n)}} \\
&= x_0^2 \sum_{n=0}^{\infty} \left(1 + \frac{\beta_m^{(n)^2}}{c_m}\right) \left(e^{-2\delta \sum_{l=0}^{n-1} \bar{\beta}^{(l)}}\right) \left(\frac{1 - e^{-2\bar{\beta}^{(n)}\delta}}{2\bar{\beta}^{(n)}}\right)
\end{aligned} \tag{6.8}$$

where $\sum_{l=0}^{-1} \bar{\beta}^{(l)} = 0$ and the last factor within the summation in Equation (6.8) becomes δ if $\bar{\beta}^{(n)} = 0$. By expressing $x(t)$ and J_m as functions of the iterates, we can generate the queue dynamics of the system and the accrued cost for each user when the updates are done online.

6.2 Simulation of Queue Dynamics for Online Updates

The behavior of the queue was simulated in MATLAB under a variety of update algorithms, cost structures and initial conditions. A sample of the results obtained are shown in Appendix B. The queue dynamics were simulated over the time interval $t \in [0, 3]$, with the time between updates taken as $\delta = 0.25$ seconds.

An immediate property that is verified is the monotonicity of the convergence of the queue to the desired queue length. Because we begin with nonnegative values for $\{\beta_m^{(0)}\}$, the update algorithm described in Equation (2.6) yields positive values for all $\{\beta_m^{(n)}\}_{n>1}$ because $c_m > 0 \quad \forall m$. Thus, from Equations (6.3)-(6.5), it can be seen that $|x(t)|$ is nondecreasing until the first iterate and then decreases monotonically toward zero afterwards. This is shown in Figure B.1. This result is intuitive as well, because the structure of our control ($u_m = r_m - a_m s = -\beta_m x$) implies that for $\beta_m > 0$, we transmit lower than our assigned rate ($a_m s$) when the queue is greater than the desired length ($x > 0$) thus reducing the queue length ($\frac{dx}{dt} \propto -x < 0$) and similarly, we transmit higher than our assigned rate when the queue is lower than the desired length ($x < 0$) thus increasing the queue length ($\frac{dx}{dt} \propto -x > 0$). It is also noted that, in our structure, there is never an overshoot, i.e., if queue is initially greater than the desired value it will at no point dip below the desired value even as the users change their controls. This is because our control assumes instantaneous or near instantaneous feedback of queue status. If, however, queue length information is sent back periodically where the time between samples is long, one may see oscillations around the target queue length.

A second result of the queue analysis, as shown in Figures B.2 - B.4, is that knowing the equilibrium value at the beginning does not necessarily lead to faster convergence to the desired queue length. Equation (6.5) shows that $\{\bar{\beta}^{(n)}\}$ is the determining factor in the rate of convergence and this is reflected in the simulations as update schemes with $\beta_m^{(0)} = 0.1$ had slower convergence to zero than update schemes with $\beta_m^{(0)} = 1$. Again, this follows intuitively as larger $\beta_m^{(0)}$s imply that the users are more dramatically reducing their transmission rates and thus accept less bandwidth to allow the queue to diminish more rapidly. The simulations show that PU is better suited to deal with low $\beta_m^{(0)}$ values than RR especially for a large number of users. This is because PU can change the low $\beta_m^{(0)}$ values after one iteration while RR must wait M iterations until all the users reach one update, thus making $\{\bar{\beta}^{(n)}\}$ remain at a lower value for a longer period of time. The performance of RU is generally comparable to PU as long periods of inactivity though possible are generally unlikely. The RU simulations show that the initial guess, $\{\beta_m^{(0)}\}$, is the most significant factor in determining the convergence of the queue length to the desired value as can be seen in Figure B.4 where four trials of each initial condition are shown.

Finally, queue simulations also verify the role of c_m in the basic cost that each user optimizes as described in Equation (2.3). Intuitively, a larger value of c_m would reflect a larger tolerance in variation in u_m and because we have $\beta_m > 0$, larger c_m would allow user m to reduce the transmission rate, r_m, without a high penalty, thus facilitating greater reduction in the queue. So, we expect the queue to diminish faster for cost structures with higher c_m's. Simulations verify this, as can be seen in Figure B.5.

6.3 Calculation of Accrued Cost for Online Updates

Using Equation (6.8), the accrued cost under various $\{\beta_m^{(n)}\}$ sequences were calculated. Analysis was restricted to the two-user case where $c_1 = c_2 = 1$, as this was sufficient to demonstrate the relevant properties of the accrued cost. The cost was also calculated over the interval $[0, 10]$, with $\delta = 1$ and $x_0 = 1$. A realization that becomes readily apparent is that the numerical value of the accrued cost alone is not a reasonable judge of optimality. For example, if both users begin with the equilibrium values that solve Equation (2.6), $\beta_m^{(0)} = \beta_m^* \approx 0.5774$, the accrued cost is 0.5774, but if both users begin with $\beta_m^{(0)} = 1$ and use PU, the accrued cost is 0.5032. This does not imply, however, that employing PU with $\beta_m^{(0)} = 1$ which converges to $\beta_m^* \approx 0.5774$ is a superior control to using $\beta_m^{(n)} = 0.5774 \ \ \forall n$.

User m's control is based on minimizing the cost J_m based on the assumption that the other user(s) is using a certain given strategy or, equivalently, a given $\{\beta_{-m}^{(n)}\}_{n\geq 0}$. When this is accomplished simultaneously for all users, we have a Nash equilibrium. Thus, the accrued cost for user m using a certain sequence $\{\beta_m^{(n)}\}_{n\geq 0}$ given $\{\beta_{-m}^{(n)}\}_{n\geq 0}$ can only be compared to the accrued cost for other sequences $\{\hat{\beta}_m^{(n)}\}_{n\geq 0}$ given the same information $\{\beta_{-m}^{(n)}\}_{n\geq 0}$. The comparison in the example above is invalid because in the case where $\beta_m^{(0)} = 0.5774$, the information is the sequence $\{\beta_{-m}^{(n)}\} = \{0.5774, 0.5774, 0.5774, 0.5774, \ldots\}$, whereas in the PU case the information is the sequence $\{\beta_{-m}^{(n)}\} = \{1.0000, 0.4142, 0.6682, 0.5345, 0.5994, \ldots\}$, and they are not the same.

Let us look at the illustrative case where the first user applies the control $\beta_1^{(n)} = 0.1 \quad \forall n \geq 0$ and the second user applies the control $\beta_2^{(0)} = 1, \quad \beta_2^{(n)} = f_m(\beta_1^{(n-1)}) = 0.9050 \quad \forall n > 0$, where f_m is defined as in Equation (2.6). Under this case, $J_1 = 0.4841$ and $J_2 = 0.9067$. Even though $J_2 > J_1$ for $c_1 = c_2 = 1$, this does not imply that the first user's control is a superior control or that the second user's application of the algorithm is not optimal. As mentioned earlier, one cannot look at the value of the accrued cost alone to judge a policy. Given $\beta_1^{(n)} = 0.1 \quad \forall n \geq 0$, if the second user varies slightly from the control used earlier (e.g., $\beta_2^{(0)} = 1, \quad \beta_2^{(n)} = 0.9250 \quad \forall n > 0$ or $\beta_2^{(0)} = 1, \quad \beta_2^{(n)} = 0.8850 \quad \forall n > 0$), J_2 increases (to 0.9068 for both cases given). Thus, the second user's control is optimal given the control sequence of the first user for $n > 0$. The same cannot be said for the first user. If the first user had instead used the sequence $\beta_1^{(0)} = 0.1, \quad \beta_1^{(n)} = f_m(0.9050) = 0.4437 \quad \forall n > 0$, J_1 would have been reduced to 0.4502.

In the earlier case, we analyzed the controls assuming we had access to the entire control sequence of the other user, but in the online implementation, users only have access to controls of the past with one iteration delay. If the first user applied the control $\beta_1^{(0)} = 0.1, \quad \beta_1^{(n)} = f_m(\beta_2^{(n-1)})$ while the second user applied the sequence $\{\beta_2^{(n)}\} = \{1.0000, 0.9050, 0.9050, 0.9050, \ldots\}$, we would have $\{\beta_1^{(n)}\} = \{0.1000, 0.4142, 0.4437, 0.4437, \ldots\}$. This is the same control sequence for the first user as when the entire sequence of the second user is known except that $\beta_1^{(1)} = 0.4142$ instead of 0.4437. This makes $J_1 = 0.5478$, which is greater than the cost under the "blind" strategy of $\beta_1^{(n)} = 0.1 \quad \forall n \geq 0$. Again, this does not mean that the first user would be better off using the "blind" strategy. Intuitively, what is happening is that the first user is transmitting at a high rate and putting

the responsibility of managing the queue on the other user. However, if the second user follows the same "blind" strategy as the first user, both users' costs will dramatically increase to 2.4832. Thus, only the algorithm defined in Equation (2.7) assures users that their objectives are being optimized regardless of the controls applied by the other users. Even though they may yield lower costs for a subset of possible control sequences of the other users, all other strategies risk dramatically undesirable performance if the other users stray from that subset.

CHAPTER 7

CONCLUSION

7.1 Summary

We have investigated the performance of various iterative algorithms that attempt to find a Nash equilibrium solution for transmission rates in an M-user bottleneck node environment. Under the objective of minimizing deviation from a desired queue length and minimizing deviation from an assigned transmission rate, it was shown that a randomized update scheme achieves the quickest convergence to an equilibrium solution, especially when the number of users is large. The Randomized Update algorithm was analyzed to obtain a theoretical optimal rate of update and the validity of the resulting optimal probability was verified through simulations. A more general problem where each user could transmit one of several types of traffic was considered. An update algorithm to achieve a Nash equilibrium was derived and shown to converge through simulations. The convergence rate of various update schemes using the multitype update algorithm was investigated, and it was shown that the Randomized Update again converged fastest to an equilibrium solution especially with a large number of users. We then considered a more general objective where users were also penalized for drastic changes in transmission

rate. A two-step iterative algorithm to reach a Nash equilibrium solution was developed. Simulations showed that the algorithm converged and also that the objective cost under the jitter structure approached the cost under the original structure in the limit as the cost on jitter was diminished. Finally, the behavior of the queue and significance of the accrued cost were investigated in the case where the users applied the algorithms online. Simulation of the queue verified the expected properties of monotonicity, and also revealed the significance of the initial transmission rates and cost structure to the dynamics of the system. Calculation of the accrued cost for various control sequences helped illustrate the game-theoretic nature of the problem and the optimality of the Nash solution.

7.2 Analysis and Proposition for Future Research

One conclusion that can be extracted from the simulations of the various update schemes is the superiority of randomization. When used with contractive mappings that form the algorithms, randomized update schemes seem to find the balance between updating too often on inaccurate information and waiting too long between updates. Under the original cost structure, the slight discrepancy in the theoretical and simulated optimal probability is still an issue to be investigated. For the randomized update scheme in the multitype environment, the optimal update probability is also an issue to be investigated. An analytical proof for the convergence of the multitype algorithm is still open and because bounds on the spectral radius of the mapping have not yielded this result, it would seem that explicit computation of the eigenvalues of the matrix of the

linearized evolution of iterates is necessary. The model for the multitype traffic is very cumbersome and update algorithms require large amounts of information to be passed between the node and the users. Also, the required feedback grows rapidly as users are added to the system. Alternative models and algorithms that yield solutions that do not require such large overheads would be desirable. Finally, with respect to the jitter cost structure, the current solution requires feedback of transmission rates of all users which may be both undesirable and unimplementable. Information structures that facilitate a more decentralized control while still yielding a consistent solution should be investigated. When simulating algorithms, additional complexity such as varying delay in information feedback among users and discrete and delayed feedback of the queue length can be included to more accurately model real systems. The main result remains the effectiveness of randomized algorithms. In fact, simulations show that the number of iterations to convergence for randomized algorithms does not increase as the number of users increase, revealing a property that merits further investigation. Randomization has been successfully applied to two specific update algorithms in this paper. The general question that this leaves us with is "Can the optimality of convergence of randomized update schemes be shown in a general dynamic game setting, and under what conditions does this optimality hold?"

APPENDIX A

CALCULATION OF E[τ]

We show the derivation of the expression used to calculate E[τ], the expected number of iterations necessary for all users to have updated at least once. This is used in the stopping criterion in various Randomized Update schemes. Let σ_n = the event that all M users have updated at least once in exactly n iterations. We assume each user m updates independently of all other users with equal probability of update, p_m, at every iteration.

$$E[\tau] = \sum_{n=1}^{\infty} n \cdot \mathcal{P}(\sigma_n) \tag{A.1}$$

where $\mathcal{P}(\cdot)$ is a probability measure over the space of the sequence of Bernoulli random variables $\{Z_m^{(k)}\}_{k\geq 0}$, and $m = 1, 2, \ldots, M$ where $\mathcal{P}(Z_m^{(k)} = 1) = p_m$. Let ω_n = the event that all M users have updated at least once in n or fewer iterations, i.e., $\omega_n = \cup_{i=1}^{n} \sigma_i$. Because $\{\sigma_n\}$ are disjoint events and the users update independently we have

$$\mathcal{P}(\sigma_n) = \mathcal{P}(\omega_n) - \mathcal{P}(\omega_{n-1}) \tag{A.2}$$

$$\begin{aligned} \mathcal{P}(\omega_n) &= \Pi_{i=1}^{M} \mathcal{P}(\gamma_i^n) \\ &= \Pi_{i=1}^{M} \left(1 - \mathcal{P}(\bar{\gamma}_i^n)\right) \end{aligned} \tag{A.3}$$

where γ_i^n = the event that the i-th user has updated at least once in n or fewer iterations and $\bar{\gamma}_i^n$ is the complement of γ_i^n, i.e., $\bar{\gamma}_i^n$ = the event that the i-th user has not updated at least once in n or fewer iterations $\Leftrightarrow Z_i^{(k)} = 0, \quad k = 1, \ldots, n$. Because in our update schemes we have $p_m = p \ \ \forall m$, we have

$$\begin{aligned} \mathcal{P}(\bar{\gamma}_i^n) &= (1 - p_i)^n \\ \Rightarrow \quad \mathcal{P}(\omega_n) &= \Pi_{i=1}^{M} \left(1 - (1 - p_i)^n\right) \\ &= \Pi_{i=1}^{M} \left(1 - (1 - p)^n\right) \\ &= \left(1 - (1 - p)^n\right)^M \\ \Rightarrow \quad \mathcal{P}(\sigma_n) &= \left(1 - (1 - p)^n\right)^M - \left(1 - (1 - p)^{n-1}\right)^M \\ \Rightarrow \quad E[\tau] &= \sum_{n=1}^{\infty} n \cdot \left(\left(1 - (1 - p)^n\right)^M - \left(1 - (1 - p)^{n-1}\right)^M\right) \end{aligned} \tag{A.4}$$

APPENDIX B

QUEUE DYNAMICS FOR ONLINE UPDATES

The following figures show the results of MATLAB simulations of the queue dynamics under various update algorithms, cost structures and initial conditions referred to in Section 6.2. In Figure B.1, four trials of RU ($p = \frac{2}{3}$) with each initial condition are displayed. In Figure B.4, four trials of each initial condition are displayed. Update times are marked with circles in Figures B.1-B.4.

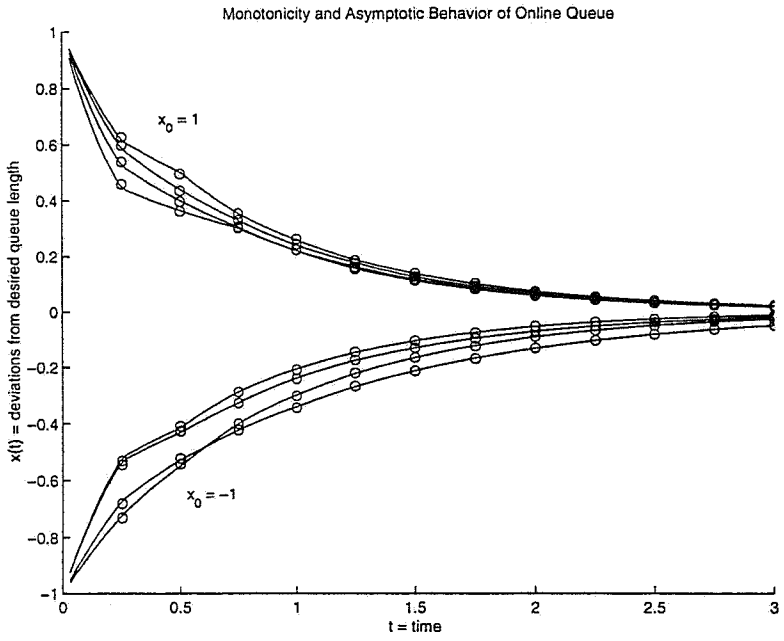

Figure B.1 Monotonicity and Asymptotic Behavior of Online Queue Dynamics

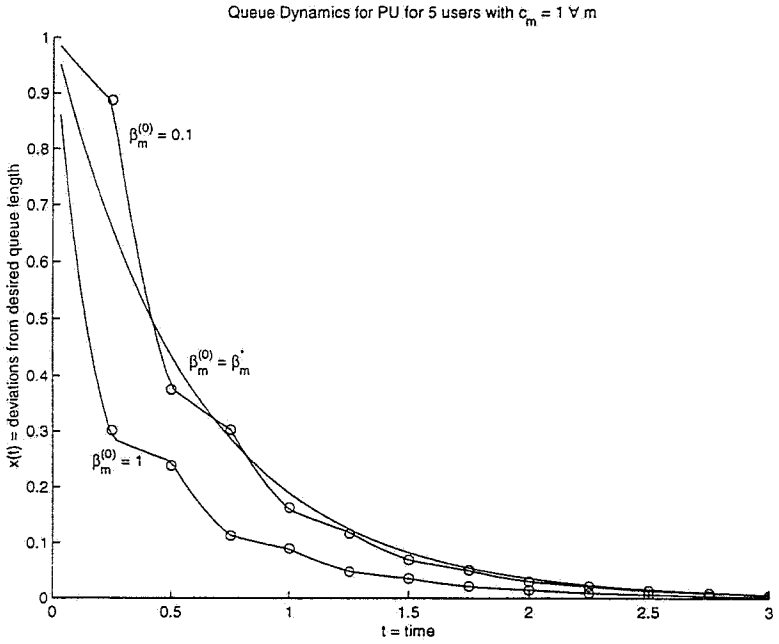

Figure B.2 Queue Dynamics for PU for 5 users with $c_m = 1 \ \ \forall \ m$

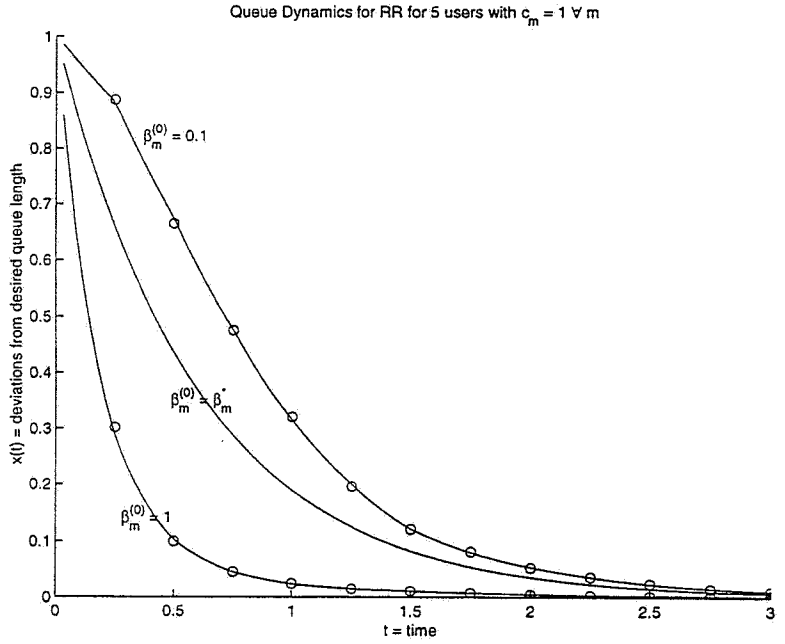

Figure B.3 Queue Dynamics for RR for 5 users with $c_m = 1 \ \ \forall \ m$

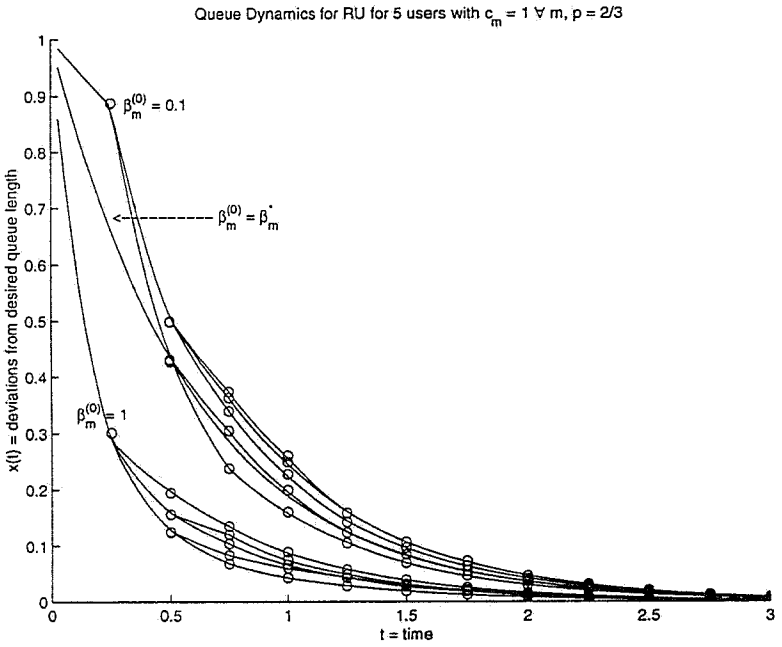

Figure B.4 Queue Dynamics for RU for 5 users with $c_m = 1 \ \forall \ m, \quad p = \frac{2}{3}$

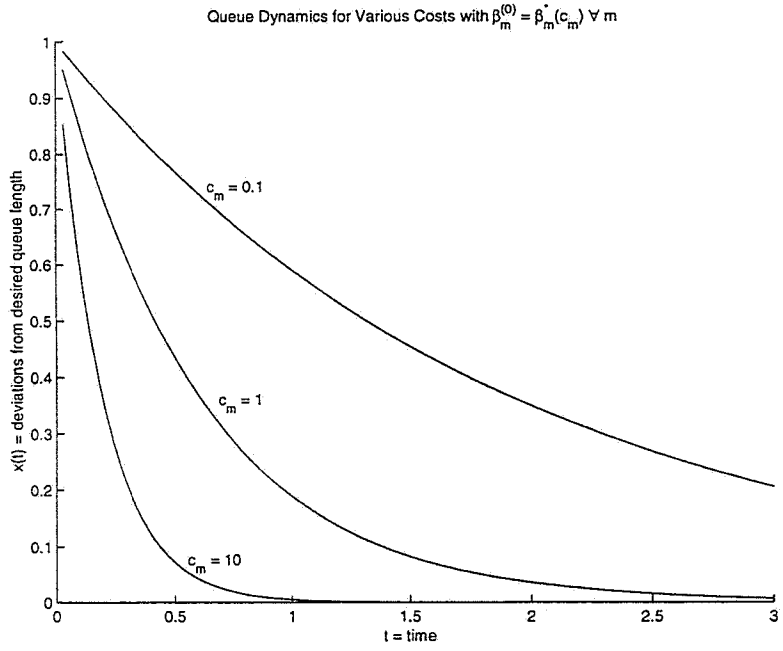

Figure B.5 Queue Dynamics for Various Costs with $\beta_m^{(0)} = \beta_m^*(c_m) \ \forall \ m$

APPENDIX C

JITTER COSTS IN THE LIMIT

The following table shows that the limit of the jitter solution cost (Equation (5.1)) as the weight on jitter ($\frac{1}{d_m}$) tends to zero is the cost in the original problem (Equation (2.3)).

Table C.1 Jitter Costs for Various 2-User Cost Structures as $d_m \to \infty$

d_1	d_2	c_1	c_2	J_1	J_2
1	1	1	1	1.12421	1.12421
10	10	1	1	0.79518	0.79518
10^2	10^2	1	1	0.65487	0.65487
10^3	10^3	1	1	0.60306	0.60306
10^4	10^4	1	1	0.58562	0.58562
10^5	10^5	1	1	0.57998	0.57998
∞	∞	1	1	0.57735	0.57735
10^3	10^3	100	100	0.17964	0.17964
10^4	10^4	100	100	0.11242	0.11242
10^5	10^5	100	100	0.07952	0.07952
10^6	10^6	100	100	0.06549	0.06549
10^7	10^7	100	100	0.06031	0.06031
10^8	10^8	100	100	0.05856	0.05856
∞	∞	100	100	0.05774	0.05774
1	1	1	2	1.07256	1.07681
10	10	1	2	0.69633	0.74489
10^2	10^2	1	2	0.51155	0.60826
10^3	10^3	1	2	0.43450	0.56026
10^4	10^4	1	2	0.40682	0.54457
10^5	10^5	1	2	0.39764	0.53959
10^6	10^6	1	2	0.39469	0.53802
∞	∞	1	2	0.39332	0.53729

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks.* Upper Saddle River: Prentice-Hall, 1992.

[2] F. Bonomi and K. W. Fendick, "The rate-based flow control framework for the available bit rate ATM service," *IEEE Network*, vol. 9, pp. 25-39, March/April 1995.

[3] D. Cansever, "Decentralized algorithms for flow control in networks," in *Proceedings of the 25th Conference on Decision and Control*, 1986, pp. 2107-2112.

[4] Y. A. Korilis and A. A. Lazar, "Architecting noncooperative networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1241-1251, September 1995.

[5] Y. A. Korilis and A. A. Lazar, "On the existence of equilibria in noncooperative optimal flow control," *Journal of the Association for Computing Machinery*, vol. 42, pp. 584-613, May 1995.

[6] M. T. Hsiao and A. A. Lazar, "Optimal decentralized flow control of Markovian queueing networks with multiple controllers," *Performance Evaluation*, vol. 13, pp. 181-204, November 1991.

[7] C. Douligeris and R. Mazumdar, "User optimal flow control in an integrated environment," in *Proceedings of the Indo-U.S. Workshop on Signals and Systems*, 1988, pp. 273-285.

[8] C. Douligeris and R. Mazumdar, "A game theoretic perspective to flow control in telecommunication networks," *Journal of the Franklin Institute*, vol. 329, pp. 383-402, 1992.

[9] E. Altman and T. Başar, "Multi-user rate-based flow control," in *Proceedings of the 36th Conference on Decision and Control*, 1997, pp. 2916-2921.

[10] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory.* London: Academic Press, 1995.

[11] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis.* New York: Cambridge University Press, 1991.