Congestion Control in Networks with Dynamic Flows

by

Kexin Ma

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2007

©Kexin Ma, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

Congestion control in wireline networks has been studied extensively since the seminal work by Mazumdar *et al* in 1998. It is well known that this global optimization problem can be implemented in a distributed manner. Stability and fairness are two main design objectives of congestion control mechanisms. Most literatures make the assumption that the number of flows is fixed in the network and each flow has infinite backlog for transfer in developing congestion control schemes. However, this assumption may not hold in reality. Thus, there is a need to study congestion control algorithm in the presence of dynamic flows. It is only until recently that short-lived flows have been taken into account. In this thesis, we study utility maximization problems for networks with dynamic flows. In particular, we consider the case where each class of flows arrive according to a Poisson process and has a length given by a certain distribution. The goal is to maximize the long-term expected system utility, which is a function of the number of flows and the rate (identical within a given class) allocated to each flow. Our investigation shows that, as long as the average work brought in by the arrival processes is strictly within the network stability region, the fairness and stability issues are independent. While stability can be guaranteed by, for example, a FIFO policy, utility maximization becomes an unconstrained optimization. We also provide a queueing interpretation of this seemingly surprising result and show that not all utility functions make sense under dynamic flows. Finally, we use simulation results to show that our algorithm indeed maximizes the expected system utility.

Acknowledgements

First of all, I want to thank my supervisor Professor Ravi. R. Mazumdar, who has been making tremendous effort in supporting my research. He is one of the most talented and knowledgable professors in the area of networking and stochastic processes. In the past two years, he does not only guide me in performing academic research, but he also teaches me the way and attitude to be an excellent researcher. These are essential qualities that one must have to be a successful person, and I will benefit from these skills for my lifetime. When it comes to his teaching style, his class always allows one to fully explore his potential and think actively. Personally, Professor Mazumdar is a very hardworking and open-minded person. He tries to provide us the best learning environment and experience ever possible all the time. It is my honor to study under his supervision.

Secondly, I would like to thank my parents who have been supporting me and giving me advise in making important decisions in my life. To support my study at Canada, they always give me the best they have and I am touched from the bottom of my heart. Without their encouragement and love, I will not be who I am today.

I also want to thank my best friend Lina Ma, whom I have known for six years. We have had a lot of discussion about researches and more often the discussions turn into arguments eventually. However, these arguments are always embedded with inspiration which advances our researches. Furthermore, she is a very positive and happy person, who makes my research life a joyful one as well. I will never forget this beautiful memory in my life.

Finally, I want to express my gratitude to Natural Sciences and Engineering Research Council of Canada who has provided funding to me for my entire master degree studies. Without these funding, my dream will not come true and I will not be able to meet so many brilliant professors and colleagues.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Congestion control plays a very important role in modern communication networks. Particularly, for traffics which can tolerate variation of delays, the sources must be able to adjust their transmission rate adaptively according to the current condition in the network. Otherwise, the network performance will be degraded significantly and the end users will experience a high loss rate. Therefore, the requirement of such a control mechanism raises the notation of rate control or congestion control. In this thesis, only wired network is analyzed. However, the analysis can be carried over to wireless networks with the same techniques. The main difference is that the bandwidth of each link is a random variable, and an explicit channel model is required.

To develop an efficient and meaningful congestion control scheme, there are several issues to be considered [19]:

1. Efficient bandwidth allocation to users with different requirements
2. The crucial notation of fairness
3. The ability to implement the control scheme in a distributed manner with minimal communication overheads
4. The performance of the network is maximized if the above congestion control scheme is used

Item 1 ensures that no bandwidth is wasted or overloaded. This objective suggests that the solution should satisfy Pareto optimality. Since there are many sources competing for the limited bandwidth, item 2 imposes some rules to guarantee fairness, which is also a required objective of the optimization. Item 3 specifies the requirement in practical implementation. Essentially, the control algorithm must be scalable with the size of the network. The last item defines the main objective of the control algorithm. It also implies that there is a need to quantify the network performance. Stability and fairness are two nonseparable objectives of the optimization. Neglecting either objective usually renders the problem meaningless. A well designed congestion control scheme will maintain the network stability and fairness while maximizing the total system performance.

The seminal work of Yaiche, Mazumdar and Rosenberg [19] studied this optimization problem from a game theoretical point of view. This work focuses on developing an algorithm, which not only provides the rate settings of flows that are Pareto optimal from the point of view of the whole system, but are also consistent with the fairness axioms of game theory.

In their work, the network performance is measured by assigning each user a utility function of the transmission rate. The objective is to maximize the social benefit, which is the sum of user utilities. They have shown that this global maximization problem can be implemented in a distributed manner by applying the so-called gradient projection method in optimization theory. Moreover, it has been shown that the solution obtained has the property of *proportional fairness* termed by Kelly [5] if the utility functions are logarithmic functions of the allocated bandwidth. In fact, Mazumdar has shown that this solution corresponds to a Nash bargaining solution (NBS), but the definition of NBS does not require logarithmic utility functions.

Inspired by Mazumdar's framework, many researchers have developed utility based congestion control algorithms. For example, Lin and Shroff [10] adopt the same techniques used in [19] and extends the research to networks where multipath routing is allowed. We note that a common assumption made by aforementioned proposals is that the number of flows in the system is fixed and each flow has infinite backlog to transfer. Therefore, these control mechanisms aim at controlling the long-lived flows and hoping that the short-lived flows may "fly" through the network with little delay or loss [12]. There is no strong proof that these mechanisms will meet the stability and fairness objectives when facing dynamic flows.

Recently, researches began to study the networks with flows that arrive and depart dynamically [3, 9]. Bonald and L. Massoulié [3] assume "middle-lived" flows: whose length is not infinite but long enough to allow the control algorithm to converge to its optimal value (also known as time-scale separation assumption). They show that the optimal rate allocation does guarantee network stability if the utility function is chosen carefully. Lin and Shroff [9] remove the time-scale separation assumption and prove that the network stability can still be achieved given the fact that the traffic intensity is within the network stability region. However, Lin and Shroff do not show what are the fairness objective being achieved and the objective being maximized.

In this work, we study the utility maximization problem in networks with dynamic flows. We assume that the flow length is determined by a random variable and we do not require the time-scale separation assumption. The utility per flow is defined as a function of the transmission rate allocated to it and the total system utility is the sum over all flow utilities. Since flows arrive and depart dynamically, our objective is to maximize the long-term expected system utility, under the link capacity constraints. Our analysis shows that, as long as the traffic intensity is within the network stability region, we can achieve the stability and fairness objectives independently: while stability can be guaranteed by, for example, a FIFO policy, utility maximization becomes an unconstrained optimization. Moreover, we investigate the system steady-state behavior in terms of delay. Finally, we perform numerical simulations on our algorithms as well as algorithms in [3, 9]. The results demonstrate that, while all these algorithms guarantee stability, ours maximizes the long-term expected utility.

The rest of the thesis is structured as follows: Chapter 2 presents the fundamental framework of congestion control. Furthermore, recent works about networks with random arrivals and departures are also analyzed. In Chapter 3, a study of congestion control for networks with random user arrivals and departures is presented. In this work, the aim is to maximize the average system utility instead of focusing on stability region as the existing works do. However, the stability region associated with the new strategy is studied as well. The superiority of the new algorithm is demonstrated through simulation results. Finally, Chapter 4 concludes the thesis and projects on the future development of congestion control.

Chapter 2

Related Works

2.1 Basic Framework

This section presents the fundamental framework by Mazumdar in congestion control. More specifically, Pareto optimality and Nash bargaining solution are the core ideas employed in developing congestion control scheme.

The network setting is as follows: There are N users who have infinite backlog to transfer. Let $X \subset \Re^N$ represent the space of all possible bandwidth allocation strategies. Each user $i \in N$ has a performance function f_i defined on X and a required minimum performance, u_i^0. Suppose that there exists at least one vector in X for which the performance vector $\vec{f} = (f_1, f_2, ..., f_N)$ is superior or equal to the minimum performance vector $\vec{u}^0 = (u_1^0, u_2^0, ..., u_N^0)$

The selection of the strategy $x \in X$ is based on the efficiency and fairness criteria. Let $U \subset \Re^N$ denote the set of all achievable performance. Clearly, U is a nonempty convex closed and upper-bounded set. The efficiency criteria is called Pareto optimality and its definition is: The point $u \in U$ is said to be Pareto optimal if for each $v \in U$, $v \geq u$, then $v = u$. The interpretation of a Pareto optimum is that it is impossible to find another point which leads to strictly superior performance for any user without sacrificing the performance of other users. In case of a network with N users, the space of the Pareto optimal points form a $N - 1$ dimensional hyper surface. An efficient congestion control algorithm should operate at one of these points on the surface.

To further narrow down the selection process, the fairness criteria plays a very important role. As we know, there are many different types of fairness such as max-min fairness. However, these fairness are not proper to use in the context of congestion control. A much better definition of fairness is the axioms from game theory. Particularly, NBS is chosen as the system operating point because an NBS has the following properties:

1. NBS is Pareto optimal.

2. The solution is unchanged if the performance objectives are scaled in the form of $au + b$. This property is also called scale invariant.

3. The solution is not affected by enlarging the domain if agreement can be found on a restricted domain. This property is also called the irrelevant-alternatives axiom.

4. The solution does not depend on the specific labels, i.e., users with the same minimum requirement and objectives will realize the same performance. This property is also called symmetry property.

The set of NBS can be found by solving the following optimization problem (P_J):

$$(P_J) \qquad Max \prod_{j \in J} (f_j(x) - u_j^0) \qquad x \in X_0$$

where J is the set of users who can achieve performance strictly superior to their minimum requirement, and $X_0 = \{x \in X : \vec{f}(x) > \vec{u}^0\}$. The uniqueness of NBS is given by the following theorem:

Theorem 2.1: Let $f_i(\cdot)$: $X \rightarrow \Re, i = 1, 2, ..., N$ be concave upper bounded functions defined on X which is a convex and compact subset of $\Re^N$. Then, there exists a unique NBS.

To convert the problem P_J into an additive structure, Mazumdar has proved the theorem below in [19].

Theorem 2.2: In addition to the assumption in Theorem 2.1, let f_j, $j \in J$ be injective on X_0.

Consider the two maximization problems P_J and P'_J:

$$(P_J) \qquad Max \prod_{j\in J}(f_j(x) - u_j^0) \qquad x \in X_0$$

$$(P'_J) \qquad Max \sum_{j\in J} ln(f_j(x) - u_j^0) \qquad x \in X_0$$

Then:

1. (P_J) has a unique solution
2. (P'_J) is a convex program and has a unique solution
3. (P_J) and (P'_J) are equivalent

It is interesting to note that the utility function used in P'_J is the same as the one used in Kelly's paper [5]. Kelly has used the term "proportional fairness" to define the fairness objective of using a log type utility function. In fact, Kelly's solution corresponds to a NBS. P'_J can be viewed as a problem of maximizing social welfare, which is the sum of individual utility. Intuitively, finding the optimal solution requires the cooperation of all users. However, it turns out that this optimization problem can be solved as a user-level problem. The details will be demonstrated in the next section.

To conclude this section, the optimization problem P'_J provides a new structure to analyze congestion control, and this seminal framework initiates a lot of researches in network utility maximization (NUM). The choice of log utility function is not a coincidence, but substantiated by proper efficiency and fairness reasoning. That is why many researchers adopt log function as their objectives in dealing with utility based optimization problems.

2.2 Congestion Control for Networks with Fixed Number of Users

2.2.1 Problem Formulation

This section discusses the results presented in [19]. We consider a network with L links and N static connections with infinite backlog to transfer. The capacity of each link is C_l. Let

$[A]$ be an $L \times N$ incidence matrix that represents the routes of the connections: $A_i^l = 1$ if the connection i goes through link l and $A_i^l = 0$ otherwise. Furthermore, each connection has a minimum bandwidth requirement MR_i and a peak rate PR_i. We assume that each link has enough capacity to provide strictly superior performance to the minimum requirement of users who utilize this link. The performance function $f_i(x)$ for user i is defined as x_i. Therefore, $u_i^0 = MR_i$ and $X_0 = \{x \in \Re^N | MR_i < x_i \leq PR_i \ \forall i \in N \ and \ Ax \leq \vec{C}\}$, where $\vec{C} = (C_1, C_2, ..., C_L)$ is the link capacity vector.

With respect to the framework described in the previous section, the NBS is an optimal and fair resource allocation of available network capacities to the N connections. The NBS is the solution of the following social or global optimization problem (S):

$$
\begin{array}{rccll}
 & \max\limits_{x_i} & \sum_{i=1}^{N} & ln(x_i - MR_i) & \\
Subject\ to & x_i & > & MR_i, & \forall i \\
 & x_i & \leq & PR_i & \forall i \\
 & \displaystyle\sum_{i=1}^{N} A_i^l x_i & \leq & C_l & \forall l
\end{array}
$$

The solution of problem S is obtained by using Lagrangian method and it is given by

$$
x_i = MR_i + min\{(PR_i - MR_i), \frac{1}{\sum_{l=1}^{L} A_i^l \mu_l}\}, \qquad i \in N
$$

where μ_l is the Lagrange multiplier for the capacity constraint, $\sum_{l=1}^{L} A_i^l x_i \leq C_l$ for all l. Indeed, μ_l has the interpretation that it represents the cost of using link l. Since solving the above problem requires the cooperation of all users or a centralized controller, it is not practical to implement it in a large network. To decentralize the problem, the following local optimization problem (U_i) catches the attention.

$$
\begin{array}{rcccl}
 & \max\limits_{x_i} & ln & (x_i - MR_i) - \alpha_i x_i & \\
Subject\ to & x_i & > & MR_i & \forall i \\
 & x_i & \leq & PR_i & \forall i
\end{array}
$$

where α_i is the cost of getting one unit of bandwidth for user i. Mazumdar has proved that if $\alpha_i = \sum_{l=1}^{L} A_i^l \mu_l$, the problem U_i yields the same solution as the problem S. The proof can be found in [19]. In other words, users do not have to cooperate in order to reach social optimum. This result suggests that it is possible to implement a distributed congestion control strategy which leads to an NBS if the cost of each link is available to end users.

2.2.2 Distributed Algorithm

The distributed algorithm is obtained by applying primal-dual algorithm to problem S. The associated Lagrangian equation is

$$L(x,\vec{\mu}) = \sum_{i=1}^{N} ln(x_i - MR_i) + \sum_{l=1}^{L} \mu_l (C_l - \sum_{i=1}^{N} A_i^l x_i)$$

where $\vec{\mu} = (\mu_1, \mu_2, ..., \mu_L)$ is the link cost vector. Then, the dual of problem S is defined as

$$\min_{\mu \geq 0} F(\vec{\mu}) \tag{2.1}$$

where

$$F(\vec{\mu}) = \max_{MR_i < x_i \leq PR_i} L(x,\vec{\mu}) \tag{2.2}$$

To solve (2.1), we consider the problem in (2.2) first. For a given $\vec{\mu}$, the problem is separable in i, $x_i(\vec{\mu})$ maximizes $L(x,\vec{\mu})$ if and only if

$$\begin{aligned} x_i(\vec{\mu}) &= \arg \max_{MR_i < x_i \leq PR_i} \{ln(x_i - MR_i) - x_i \sum_{l=1}^{L} A_i^l \mu_l\} && (2.3) \\ &= MR_i + min\{(PR_i - MR_i), \frac{1}{\sum_{l=1}^{L} A_i^l \mu_l}\}, \qquad i \in N && (2.4) \end{aligned}$$

Note that (2.3) is identical to the user problem U_i defined in the previous section. The solution of this linear programming problem can be found by many standard methods, and the complexity is low.

Now, we focus on solving the link cost vector $\vec{\mu}$, which is the solution of (2.1). The algorithm is based on gradient projection method with a constant step-size. The partial derivative of $L(x, \vec{\mu})$ is

$$\frac{\partial}{\partial \mu_l} L(x, \vec{\mu}) = C_l - \sum_{i=1}^{N} A_i^l x_i$$

Let $\gamma > 0$ denote the step-size. Then $\vec{\mu}$ can be solved by the following recursive equation

$$\mu_l(k+1) = [\mu_l(k) + \gamma(\sum_{i=1}^{N} A_i^l x_i - C_l)]^+, \forall l \tag{2.5}$$

where $[\cdot]^+$ denote the projection to $[0, +\infty]$, and x_i is given by (2.3). Let $N(i)$ denote the number of links crossed by user i and define

$$K = \sqrt{L}(\sum_{i=1}^{N}(PR_i - MR_i)^2 N(i))$$

If $\gamma \in (0, 2/K)$, then $x_i(\vec{\mu})$ will converge to the NBS.

(2.4) and (2.5) are often termed primal update and dual update respectively. In real-time online implementation, the time is slotted with length T. At the end of each time slot, the system executes these two equations.

It is important to note that by applying the primal-dual algorithm, the original global problem is decomposed into two local optimization procedures. Each link updates its cost according to the local traffic, and the each end user updates its bandwidth allocation according to the total cost on its path. The only communication required is the broadcast of link costs to end users. Most literatures in the area of congestion control are based on the same techniques developed by Mazumdar.

2.3 Congestion Control for Networks with Multipath Routing

2.3.1 Problem Formulation

In [10], the primal-dual control scheme has been extended to networks where multipath routing is allowed by Lin and Shroff. The number of users in the system is assumed to be constant. The optimization problem is formulated as

$$\max_{x_{ij}\geq 0, m_i \leq \sum_{j=1}^{J(i)} x_{ij} \leq M_i, i=1,...,I} \sum_{i=1}^{I} f_i\left(\sum_{j=1}^{J(i)} x_{ij}\right) \tag{2.6}$$

$$\text{subject to} \sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^{l} x_{ij} \leq R^{l},\ \ for\ all\ l = 1, ..., L. \tag{2.7}$$

Generally, the problem (2.6) amounts to allocating resources $R^1, ..., R^L$ from network components $l = 1, 2, ..., L$ to users $i = 1, 2, ..., I$ such that the total system "utility" is maximized. The "utility" function $f_i(\cdot)$ represents the performance, or level of "satisfaction," of user i when a certain amount of resource is allocated to it. In practice, this performance measure can be in terms of revenue, welfare, or admission probability. The utility function $f_i(\cdot)$ is assumed to be concave. Each user i can have $J(i)$ alternative paths (a path consists of a subset of the network components). Let x_{ij} denote the amount of resources allocated to user i on path j. Then the utility $f_i(\sum_{j=1}^{J(i)} x_{ij})$, subject to $m_i \leq \sum_{j=1}^{J(i)} x_{ij} \leq M_i$, is a function of the sum of the resources allocated to user i on all paths. Hence, the resources on alternative paths are considered equivalent and interchangeable for user i. The constants E_{ij}^l represent the routing structure of the network: each unit of resource allocated to user i on path j will consume E_{ij}^l units of resource on network component l. ($E_{ij}^l = 0$ for network components that are not on path j of user i.) The inequalities in (2.7) represent the resource constraints at the network components (hence R^l can be viewed as the capacity of network component l, and $\sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij}$ is the total amount of resources consumed at network component l summed over all users and all alternative paths). The following assumptions are made: $R^l > 0$, $E_{ij}^l \geq 0$, $m_i \geq 0$ and $M_i > 0$ (M_i could possible be $+\infty$).

Problem (2.6) is referred as the *multipath utility maximization problem*. Essentially, once the network can support multipath routing, the resource allocation problem changes from a *single-path* utility maximization to a *multi-path* utility maximization problem. The multi-path nature of the problem leads to several difficulties in constructing solutions suitable for online implementation. One of the main difficulties is that, once some users have multiple alternative paths, the objective function of problem (2.6) is no longer strictly concave, and hence the dual of the problem may not be differentiable at every point. Note that this lack of strict concavity is mainly due to the linearity $\sum_{j=1}^{J(i)} x_{ij}$. (The objective function in (2.6) is still not strictly concave even if the utility function f_i are strictly concave.) On the other hand, the requirement that the solutions must be implementable online also imposes a number of important restrictions on the design space. These restrictions are outlined below:

- The solution has to be distributed because these communication networks can be very large and centralized solutions are not scalable.
- In order to lower the communication overhead, the solution has to limit the amount of information exchanged between the users and different network components. For example, a solution that can adjust resource allocation based on online measurements is preferable to one that requires explicit signaling mechanisms to communicate information.
- It is also important that the solution does not require the network components to store and maintain per-user information (or per-flow information). Since the number of users sharing a network can be large, solutions that require maintaining per-user information will be costly and will not scalable to large networks.
- In the case where solution uses online measurements to adjust resource allocation, the solution should also be resilient to measurement noise due to estimation errors.

In [10], Lin and Shroff developed a distributed solution to multi-path utility maximization problem with the following major technical contributions:

- A rigorous analysis of the convergence of the distributed algorithm is provided. The analysis is done without requiring the *two-level convergence structure* that is typical in standard techniques in the convex programming literature for dealing with the lack of strict concavity of the problem. Note that algorithms based on these standard techniques are required to have an *outer level* of iterations where each outer iteration consists of an *inner level* of iterations. For the convergence of this class of algorithm to hold, the inner level of iterations must converge before each outer iteration can proceed. Such a two-level convergence structure may be acceptable for offline computation, but not suitable for online implementation because in practice it is difficult for the network to decide in a distributive fashion when the inner level of iterations can stop. A main contribution of this work is to establish the convergence of the distributed algorithm without requiring such a two-level convergence structure.

- By providing convergence, an easy-to-verify bounds on the algorithm parameters (i.e., step-size) to ensure convergence is provided. Note that when distributed algorithms based on solution are implemented online, a practically important question is how to choose the parameters of the algorithm to ensure efficient network control. Roughly speaking, the step-sizes used in the algorithm should be small enough to ensure stability and convergence, but not too small such that the convergence becomes unnecessarily slow. The main part of this work addresses the question of parameter selection by providing a rigorous analysis of the convergence of the distributed algorithm.

- The convergence of the algorithm in the presence of measurement noise is studied, and guidelines on how to choose the step-sizes to reduce the disturbance in the resource allocation due to noise are provided.

- The impact of the inherent nature of the multi-path problem on instability and oscillation is studied.

2.3.2 Distributed Algorithm

As mentioned earlier, the objective function is not strict concave. This nature creates difficulty, and the standard primal-dual technique can not be applied directly. However, primal-dual method is still preferred because of its elegant decomposition. To overcome the difficulty of lack of strict concavity, the idea from Proximal Optimization Algorithms is adopted.

The idea is to add a quadratic term to the objective function, which transforms the original problem to a strictly concave problem. Let $\vec{x}_i = [x_{ij}, j = 1, ..., J(i)]$ and

$$C_i = \left\{ \vec{x}_i | x_{ij} \geq 0 \; for \; all \; j \; and \; \sum_{j=1}^{J(i)} x_{ij} \in [m_i, M_i] \right\}, \; i = 1, ..., I. \tag{2.8}$$

Let $\vec{x} = [\vec{x}_1, ..., \vec{x}_I]^T$ and let C denote the Cartesian product of C_i, i.e., $C = \otimes_{i=1}^{I} C_i$. Let y_{ij} denote the auxiliary variable for each x_{ij}. Let $\vec{y}_i = [y_{ij}, j = 1, ..., J(i)]$ and $\vec{y} = [\vec{y}_1, ..., \vec{y}_I]^T$. The optimization problem can be rewritten as

$$\max_{\vec{x}\in C, \vec{y}\in C} \sum_{i=1}^{I} f_i \left(\sum_{j=1}^{J(i)} x_{ij} \right) - \sum_{i=1}^{I} \sum_{j=1}^{J(i)} \frac{c_i}{2} (x_{ij} - y_{ij})^2 \tag{2.9}$$

$$\text{subject to} \sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^{l} x_{ij} \leq R^l, \ \ for \ all \ l \tag{2.10}$$

where c_i is a positive number. It has been proved that the solution of (2.9) is equivalent to that of (2.6). In fact, if $\vec{x}^*$ is the solution of (2.6), then $\vec{x} = \vec{x}^*, \vec{y} = \vec{x}^*$ is the solution of (2.9).

The proximal optimization problem proceeds as follows:

`Algorithm P:` At the tth iteration

P1 Fix $\vec{y} = \vec{y}(t)$ and maximize the augmented objective function with respect to $\vec{x}$.

$$\max_{\vec{x}\in C} \sum_{i=1}^{I} f_i \left(\sum_{j=1}^{J(i)} x_{ij} \right) - \sum_{i=1}^{I} \sum_{j=1}^{J(i)} \frac{c_i}{2} (x_{ij} - y_{ij})^2 \tag{2.11}$$

$$\text{subject to} \sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^{l} x_{ij} \leq R^l, \ \ for \ all \ l \tag{2.12}$$

Remark: The above maximization problem is taken over $\vec{x}$. The additional quadratic term in (2.11) converts the problem to a strictly concave structure. Thus, the solution of (2.11) is always unique. Let $\vec{x}(t)$ be the solution of this problem.

P2 Set $\vec{y}(t) = \vec{x}(t)$

As $t \to \infty$, the iterations will converge to $\vec{x}^*$. Step P1 involves solving a global optimization problem. Since it is strictly concave, its solution is given by the primal-dual algorithm below:

$$\vec{x}_i^*(t) = \max_{\vec{x}_i \in C_i} \left\{ f_i\left(\sum_{j=1}^{J(i)} x_{ij}\right) - \sum_{j=1}^{J(i)} x_{ij}q_{ij} - \sum_{j=1}^{J(i)} \frac{c_i}{2}(x_{ij}-y_{ij})^2 \right\}, \forall i \quad (2.13)$$

$$q^l(t+1) = \left[q^l(t) + \alpha^l\left(\sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij}(t) - R^l\right)\right]^+, \forall l \quad (2.14)$$

where q^l is the Lagrangian multiplier associated with constraint (2.12).

Remark: Algorithm `P` involves a two-level convergence structure. Each outer iteration P1 consists of an inner level of iterations (2.13) and (2.14). For the convergence of algorithm `P` to hold, inner level of iterations must converge before each outer iteration P2 can proceed. Such a two-level structure is not suitable for online implementation as the network components can not determine when the inner level iteration should stop in a distributive manner.

To solve the above problem, the following modified algorithm has been proposed:

`Algorithm A`: Fix $K \geq 1$. At the tth iteration:

A1 Fix $\vec{y} = \vec{y}(t)$ and use equation (2.13) on the dual variable $\vec{q}$ for K times. To be precise, let $\vec{q}(t,0) = \vec{q}(t)$. Repeat for each $k = 0, 1, ..., K-1$:

$$q^l(t,k+1) = \left[q^l(t,k) + \alpha^l\left(\sum_{i=1}^{I}\sum_{j=1}^{J(i)} E_{ij}^l x_{ij}(t,k) - R^l\right)\right]^+, \forall l \quad (2.15)$$

A2 Let $\vec{q}(t+1) = \vec{q}(t,K)$. Let $\vec{z}(t)$ be the primal variable that solves (2.13). Set

$$y_{ij}(t+1) = y_{ij}(t) + \beta_i(z_{ij}(t) - y_{ij}(t)), \forall i, j \quad (2.16)$$

where $0 < \beta_i \leq 1$. The convergence of algorithm `A` is given by the following proposition.

Proposition 2.1: Fix $1 \leq K \leq \infty$. As long as the step-size α^l is small enough, algorithm A will converge to a stationary point $(\vec{y}^*, \vec{q}^*)$ of the algorithm, and $\vec{x}^* = \vec{y}^*$ will solve the original problem (2.6). The sufficient condition for convergence is

$$max_l \alpha^l < \begin{cases} \frac{2}{\mathcal{S}\mathcal{L}} \min_i c_i, & \text{if } K = \infty; \\ \frac{1}{2\mathcal{S}\mathcal{L}} \min_i c_i, & \text{if } K = 1; \\ \frac{4}{5K(K+1)\mathcal{S}\mathcal{L}} \min_i c_i, & \text{if } K > 1. \end{cases} \tag{2.17}$$

where $\mathcal{L} = \max\{\sum_{l=1}^{L} E_{ij}^l, i = 1, ..., I, j = 1, ..., J(i)\}$, and $\mathcal{S} = \max\{\sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l, l = 1, ..., L\}$.

In reality, the total load $\sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij}(t,k)$ is estimated through online measurements with nonnegligible noise. To consider this measurement noise into the model, algorithm `A` is replaced by

`Algorithm AN:`

A1-N:

$$q^l(t, k+1) = \left[q^l(t,k) + \eta_t \alpha^l \left(\sum_{i=1}^{I} \sum_{j=1}^{J(i)} E_{ij}^l x_{ij}(t,k) - R^l + N^l(t,k) \right) \right]^+, \forall l \tag{2.18}$$

A2-N:

$$y_{ij}(t+1) = y_{ij}(t) + \eta_t \beta_i (z_{ij}(t) - y_{ij}(t)), \forall i, j \tag{2.19}$$

where η_t is a positive sequence which goes to zero as $t \to \infty$, and $N^l(t,k)$ is the measurement noise on link l. The convergence of algorithm `AN` is given by the following proposition.

Proposition 2.2: If

$$\sum_{t=1}^{\infty} \eta_t = \infty, \sum_{t=1}^{\infty} \eta_t^2 < \infty \tag{2.20}$$

$$E[N^l(t) | \vec{x}(s), \vec{y}(s), \vec{q}(s), s \le t] = 0, \forall l \tag{2.21}$$

$$\sum_{t=1}^{\infty} \eta_t^2 E\|N^l(t)\|^2 < \infty, \forall l \tag{2.22}$$

then the algorithm `AN` will converge almost surely to a stationary point $(\vec{y}^*, \vec{q}^*)$ of algorithm `A`.

2.4 Congestion Control for Networks with Dynamic Users

2.4.1 System Model

This class of work studies a network with random dynamic arrivals and departures of users. The motivation is that the number of users in the system changes constantly in today's Internet. Therefore, the class of algorithms described in the previous section may never converge to the optimal solution. However, this body of works still adopt the primal-dual technique and can be viewed as an extension of the static analysis. Interestingly, although all works in this class use utility based problem, they focus on exploring stability region rather than maximizing social welfare.

The network setting is identical to the static case except that N is not a fixed number anymore. There are S classes of users, and users of class s arrive to the network according to a Poisson process with parameter λ_s. Each user brings a file to transfer whose length is exponentially distributed with mean $1/\mu_s$. Note that μ_s has no relationship with the link cost anymore, and the link cost is denoted by q_l in this case. The load brought by each class of users is $\rho_s = \lambda_s/\mu_s$. Let $\vec{\rho} = (\rho_1, \rho_2, ..., \rho_S)$ be the load vector. Let $n_s(t)$ denote the number of users in class s and $x_s(t)$ denote the bandwidth allocation to class s at time t. The assumption being made is that users in the same class receive equal amount of bandwidth. Similar to the static case, let A be the routing matrix. Instead of having A_i^l, we have A_s^l which means that all users in class s follow the same routing path. Another minor change is that the minimum bandwidth required is 0 and peak rate is M_s for all users in class s.

Now, let's present three important definitions first before we discuss the results of researches done so far. The first definition is stability. A system is stable if

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t 1_{\{\sum_{s=1}^{S} n_s(t)+\sum_{l=1}^{L} q_l(t)>M\}} dt \to 0, \ as\ M \to \infty$$

The second definition is the largest stability region, which is given by

$$\Theta = \{\vec{\rho} | \sum_{s=1}^{S} A_s^l \rho_s \le C_l, \ \forall l\}$$

The last definition is called the time-scale separation assumption. The assumption states that the data rates $x_s(t)$ at each time instant t are adjusted instantaneously to the optimal rate allocation computed by the static global optimization problem S with $N = \sum_{s=1}^{S} n_s(t)$. Furthermore, a controller satisfying this assumption is called a perfect congestion controller.

The utility function discussed in this body of work is not restricted to log function anymore. A more general form of the utility function is

$$U_s(x_s) = w_s \frac{x_s^{1-\alpha}}{1-\alpha},\ \alpha > 0 \ and \ \alpha \neq 1 \tag{2.23}$$

This utility function is also referred as weighted α-bandwidth sharing utility function.

2.4.2 Recent Results

Most results are based on the time-scale separation assumption. In [3], the ergodicity of the stochastic process $\vec{n}(t) = [n_1(t), ..., n_S(t)]$ is considered. Conditions on the traffic intensities ρ_s for which, starting from any initial state, the number of flows on each route remains finite with probability 1 are derived. Clearly, the following conditions are necessary:

$$\sum_{s=1}^{S} A_s^l \rho_s \leq C_s, \forall l \tag{2.24}$$

The evolution of $\vec{n}(t)$ is governed by a Markov process, i.e. the arrival process is Poisson and the size of the file for transfer is exponentially distributed. Its transition rates are given by

$$\begin{aligned} n_s(t) \quad &\rightarrow \quad n_s(t)+1, \qquad \text{with rate } \lambda_s, \\ n_s(t) \quad &\rightarrow \quad n_s(t)-1, \qquad \text{with rate } \mu_s x_s(t) n_s(t), \end{aligned}$$

The following proposition gives the condition for stability with α utility function, starting from any initial state.

Proposition 2.3: The Markov process $\vec{n}(t)$ is ergodic if and only if traffic condition (2.24) is satisfied.

The proof consists of the the study of fluid system defined by

$$N_s(t) = \lim_{w\to\infty} \frac{n_s(wt)}{w}$$
$$\sum_{s=1}^{S} n_s(0) = w$$

If the limit exists, $\sum_{s=1}^{S} N_s(0) = 1$. Given an initial distribution of the fluid $N_s(0)$, it follows from the strong law of large number that the evolution of the fluid system $N_s(t)$ is uniquely defined by the differential equations:

$$\frac{d}{dt}N_s = \lambda_s - \mu_s\Lambda_s(t), \text{ for all } s,t \text{ such that } N_s(t) > 0 \tag{2.25}$$

where $\Lambda_s(t) = n_s(t)x_s(t)$ is the total bandwidth allocated to class s at time t. Let vector $\Lambda = (\Lambda_s)$ be the solution of the optimization problem:

$$\max_{\Lambda} \sum_{s=1}^{S} w_s N_s^{\alpha} \frac{\Lambda_s^{1-\alpha}}{1-\alpha}$$
$$\text{subject to } \sum_{s=1}^{S} A_s^l \Lambda_s \le C_l, \ \forall l$$

Consider the following function defined on the set of $|\mathcal{S}|$ dimensional positive vectors:

$$F(u) = \sum_{s=1}^{S} w_s \mu_s^{-1} \rho_s^{-\alpha} \frac{u_s^{\alpha+1}}{\alpha+1}$$

From (2.25),

$$\frac{d}{dt}F(N) = \sum_{s=1}^{S} w_s \rho_s^{-\alpha} N_s^{\alpha}(\rho_s - \Lambda_s) \tag{2.26}$$

Consider now the function

$$G(u) = \sum_{s=1}^{S} w_s N_s^{\alpha} \frac{u_s^{1-\alpha}}{1-\alpha}$$

The vector Λ attains the maximum of this function over the domain specified by the capacity constraint. Thus, for any vector u in this convex domain, the gradient of G satisfies $G'(u)(u-\Lambda) \leq 0$. By concavity of G, we conclude that:

$$G'(u)(u-\Lambda) \leq 0$$

Under the capacity constraint, there exists $\epsilon > 0$ such that the vector $u = (\rho_s(1+\epsilon))$ satisfies the capacity constraint. Applying the previous inequality,

$$\sum_{s=1}^{S} w_s \rho_s^{-\alpha} N_s^{\alpha} (\rho_s(1+\epsilon) - \Lambda_s) \leq 0$$

Equivalently, in view of (2.26), this reads

$$\frac{d}{dt} F(N) \leq -\epsilon \sum_{s=1}^{S} w_s \rho_s^{\alpha+1} N_s^{\alpha}$$

Using straightforward bounds, there exists a positive constant β such that

$$\frac{d}{dt} F(N) \leq -\beta F(N)^{\frac{\alpha}{\alpha+1}}$$

This implies that if $F(N(T)) = 0$ for some $T > 0$, $F(N(t)) = 0$ for all $t \geq T$. In addition, integrating this equation yields for all $t \geq 0$ such that $F(N(t)) > 0$,

$$F(N(t)) \leq \left(F(N(0))^{\frac{1}{\alpha+1}} - \frac{\beta}{\alpha+1} t \right)^{\alpha+1}$$

Recalling that $\sum_{s=1}^{S} N_s(0) = 1$, this implies that $F(N(t))$ and thus $N(t)$ are identically equal to zero for all $t \geq T$, with

$$T = \frac{\alpha+1}{\beta} \left(\frac{1}{\alpha+1} \sum_{s=1}^{S} \mu_s^{-1} \rho_s^{-\alpha} \right)^{\frac{1}{\alpha+1}}$$

Recently, Lin and Shroff have proved that the time-scale separation assumption is not necessary to achieve the largest stability region. In [9], the time is divided into slots of length T, and the link costs are updated at the end of each slot. Having defined the structure, they proposed the following distributed algorithm:

$$x_s(t) = x_s(kT) = min\{(\frac{w_s}{\sum_{l=1}^{L} A_s^l q_l(kT)})^{1/\beta}, M_s\},\ for\ kT \leq t < (k+1)T \quad (2.27)$$

$$q_l((k+1)T) = [q_l(kT) + \gamma_l(\sum_{s=1}^{S} A_s^l \int_{kT}^{(k+1)T} n_s(t)x_s(kT)dt - TC_l)]^+ \quad (2.28)$$

The following proposition has been proved in [9].

Proposition 2.4: Assume that utility functions are of the form in (2.23) for some $\beta > 1$, and that the data rates are controlled by (2.27) and (2.28). Let $\overline{S} = \max_l \sum_{s=1}^{S} A_s^l$ denote the maximum number of classes using any link, and let $\overline{L} = \max_s \sum_{l=1}^{L} A_s^l$ denote the maximum number of links crossed by any class. If

$$\max_l \gamma_l \leq \frac{1}{T\overline{S}\overline{L}} \frac{2^\beta - 1}{16} \min_s \frac{w_s}{\rho_s M_s^\beta}$$

then for any offered load $\vec{\rho}$ that resides strictly inside Θ, the system is stable.

Several remarks should be emphasized. First, the contribution of the above work is that no time-scale separation is required to achieve stability region Θ. Secondly, careful thought should indicate that (2.27) is identical to (2.4) with $MR_i = 0$. Similarly, (2.28) is identical to (2.5) except that the correction term is replaced by an integration operation. The reason is that $n_s(t)$ is a random number. Thus, the total traffic acting on a link must be measured by performing integration as opposed to simple addition. It seems that the proposed algorithm is only a modification of the solution in static case. Lastly, while (2.4) and (2.5) converge to a fixed equilibrium point, (2.27) and (2.28) converge to a stationary stochastic process, which is due to the randomness of the arrival and departure processes. Since the algorithm does not require time-scale separation assumption, the allocation process $x_s(t)$ may not reach the optimal solution computed by problem S with $N = \sum_{s=1}^{S} n_s(t)$ at all time.

A natural question to ask at this point is if we can achieve the largest stability region and maximize the system utility simultaneously. In other words, we need to take Lin and Shroff's result one step further. Although stability is an important issue, the problem is meaningless if utility maximization is overlooked. In the next chapter, a systematic study of utility maximization problem in the context of dynamic users is presented. We will show that the answer to the question is affirmative.

Chapter 3

Utility Maximization for Networks with Dynamic Users

3.1 System Model and Problem Formulation

In this section, we describe our system model and define the associated optimization problem. We consider a network with L links and S classes of flows. We denote the sets of links and classes by $\mathcal{L}$ and $\mathcal{S}$, respectively. The capacity of each link $l \in \mathcal{L}$ is R_l. $[A]$ is an $L \times S$ matrix that represents the routes of the flows: $A_s^l = 1$ if the flows of class $s \in S$ go through link l and $A_s^l = 0$ otherwise.[1] The arrival process of the flows of any class s is Poisson with rate λ_s and the durations are of an arbitrary length distribution with mean μ_s^{-1}. Thus, the traffic intensity brought by flows of class s is $\rho_s = \lambda_s/\mu_s$. We further assume that $\vec{\rho} = [\rho_s]$ is within the stability region defined by $\Theta = \{\vec{\rho} | \sum_{s=1}^{S} A_s^l \rho_s \leq R^l, \forall l\}$.

For each class s, let $x_s(t)$ denote the rate allocated for each flow at time t, and let $U_s(x_s(t)) = \log x_s(t)$ be the utility received by the flow of class s when the allocated transmission rate is $x_s(t)$. The utility function represents the level of satisfaction of a flow, and different utility functions will achieve different fairness objectives. Here, $\log(\cdot)$ function will ensure proportional fairness[2] defined in Chapter 2. We assume that each flow

[1] Our results can be readily extended to the case where the link capacity is time-varying and the routes are not pre-defined.

[2] As we will show later, we are not taking the log utility function by chance. It seems to be the only

of class s has a maximum transmission rate, M_s.

Let $n_s(t), s = 1, 2, \ldots, S$ denote the number of flows of class s that are present in the system, and $\vec{x}(t) = [x_1(t), x_2(t), \ldots, x_S(t)]$ denote the rate vector at time t. In our model, time is slotted and the length of each slot is T seconds. Flows arriving within a slot will start transmission at the beginning of the next slot as shown in Fig. 3.1. Therefore, $n_s(t)$ can be decomposed into two parts, $n_s(t) = n_s^w(t) + n_s^t(t)$, where $n_s^w(t)$ represent the flows waiting for transmission and $n_s^t(t)$ represent the flows transmitting data. Therefore, the global optimization problem can be formulated as:

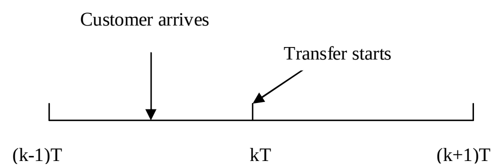

Figure 3.1: Transmission Model

$$\max_{\vec{x}(t) \in X(t)} \lim_{t \to \infty} \frac{1}{t} \int_{t=0}^{\infty} \sum_{s=1}^{S} n_s^t(t) U_s(x_s(t)) dt \tag{3.1}$$

$$\text{subject to} \lim_{t \to \infty} \frac{1}{t} \int_{t=0}^{\infty} \sum_{s=1}^{S} A_s^l n_s^t(t) x_s(t) dt \leq R_l, \quad \forall l \tag{3.2}$$

(3.1) has an interpretation of maximizing expected long-term system utility. The constraint (3.2) addresses stability requirement.

Suppose the queueing process at each source node is ergodic (we will justify this assumption in Section 3.4). Let $\nu(\vec{n}^t, \vec{x})$ denote the density (which we assume exists without the loss of generality) of the joint distribution of $\vec{n}^t$ and $\vec{x}$ in equilibrium. Given $\vec{x}$, the stationary distribution of $\vec{n}$ can be shown to be conditionally independent (follows from a truncation of the $M/G/\infty$ model). Let $\nu(n_s^t|\vec{x})$ denote the conditional density of flow x_s given $\vec{x}$, then $\nu(\vec{n}^t, \vec{x}) = [\prod_{s=1}^{S} \nu(n_s^t|\vec{x})]p(\vec{x})$, where $p(\vec{x})$ denotes the joint density of $\vec{x}$.

meaningful utility under dynamic flows.

Then (3.1) can be written as the following problem:

$$\max_{\vec{x}} \int_X \sum_{s=1}^{S} \left[\sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) \nu(n_s^t|\vec{x}) \right] p(\vec{x}) d\vec{x} \tag{3.3}$$

subject to

$$\int_X \sum_{s=1}^{S} A_s^l \left[\sum_{n_s^t=0}^{\infty} n_s^t x_s \nu(n_s^t|\vec{x}) \right] p(\vec{x}) d\vec{x} \leq R_l, \quad \forall l \tag{3.4}$$

where $X = \{\vec{x}|x_s \in (0, M_s], s = 1, 2, \ldots, S\}$. Therefore, the maximization problem is essentially about finding an optimal joint distribution $p(\vec{x})$. Define

$$\begin{aligned} g(\vec{x}) &= \sum_{s=1}^{S} \sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) \nu(n_s^t|\vec{x}) \\ &= \sum_{s=1}^{S} \log(x_s) \sum_{n_s^t=0}^{\infty} n_s^t \nu(n_s^t|\vec{x}) \\ &= \sum_{s=1}^{S} \log(x_s) E[N_s^t|\vec{x}] \\ &= \sum_{s=1}^{S} \log(x_s) \rho_s / x_s \end{aligned} \tag{3.5}$$

To evaluate $E[N_s^t|\vec{x}]$, we have applied Little's law in the above derivation. It is easy to see that the expected service time excluding the waiting time is $1/(\mu_s x_s)$ in equilibrium. Thus, by Little's law, $E[N_s^t|x_s] = \lambda_s/(\mu_s x_s) = \rho_s/x_s$. Substitute (3.5) into (3.3), we have

$$\max_{\vec{x}} \int_X g(\vec{x}) p(\vec{x}) d\vec{x} \tag{3.6}$$

Now, we are going to investigate the properties of $g(\vec{x})$ to obtain the structure of $p(\vec{x})$. The first order and second order partial derivative of $g(\vec{x})$ are given by

$$\frac{\partial g(\vec{x})}{\partial x_s} = \frac{\rho_s(1 - \log(x_s))}{x_s} \tag{3.7}$$

$$\frac{\partial^2 g(\vec{x})}{\partial x_s^2} = \frac{\rho_s(2\log(x_s) - 3)}{x_s^3} \tag{3.8}$$

$$\frac{\partial^2 g(\vec{x})}{\partial x_s x_t} = 0, \ s \neq t \tag{3.9}$$

According to (3.7), it is easy to see that $g(\vec{x})$ has a unique global maxima at $x_s^* = e$ for all s. In addition, from (3.8) and (3.9), we can conclude that $g(\vec{x})$ is strictly concave if $0 < x_s < e^{3/2}$ and strictly convex if $x_s > e^{3/2}$ for all s. Within the convex region, the minima occurs at $x_s = \infty$, which can be inferred from (3.7).

To maximize (3.6), $p(\vec{x})$ should put all its mass at $\vec{x^*}$ if $\vec{x^*}$ satisfies (3.4) and $0 < x_s^* \leq M_s$ for all s. This is because $g(\vec{x^*}) > g(\vec{x})$ for all $\vec{x} \neq \vec{x^*}$. Otherwise, $p(\vec{x})$ should put all its mass at one of its boundary points of the solution space. This is because that $g(\vec{x})$ strictly increases until it reaches the global maxima, and then strictly decreases on each of its dimension. In either case, $p(\vec{x})$ is a Dirac delta function.

As a consequence, we can transfer the stochastic optimization problem (3.3) and (3.4) into a deterministic one in the following:[3]

$$\max_{\vec{x} \in X} \sum_{s=1}^{S} E[N_s^t | x_s] U_s(x_s) \tag{3.10}$$

$$\text{subject to } \sum_{s=1}^{S} A_s^l E[N_s^t | x_s] x_s \leq R_l, \quad \forall l \tag{3.11}$$

where N_s^t is a random variable representing the number of class s flows in transmission when rate x_s is assigned. Let DL denote the above optimization problem.

We can further deduce the optimal joint distribution $p(\vec{x})$ by setting the following equality:

$$\int_X p(\vec{x}) d\vec{x} = \int_{\mathcal{X}} p(\vec{x}) d\vec{x} = 1 \tag{3.12}$$

where $\mathcal{X}$ is the set of $\vec{x}$ that solve problem (3.10) and (3.11). The solution is straightforward: $p(\vec{x})$ can be any distribution as long as (3.12) is met. If $g(\vec{x})$ has an unique maxima on X, $\mathcal{X} = \{\vec{x}^*\}$ is a singleton and thus $p(\vec{x}) = \delta(\vec{x} - \vec{x}^*)$ where δ is the Dirac delta function.

Note that the constraint given in (3.11) refers to long-term congestion avoidance. If instantaneous congestion avoidance is required, (3.11) will be replaced by

$$\sum_{s=1}^{S} A_s^l n_s^t x_s \leq R_l, \quad \forall l \tag{3.13}$$

[3]Although the density functions $\nu(n_s^t | \vec{x})$ are still involved, they are completely specified by $\vec{x}$ and the Poisson assumption on the arrivals.

Let DS denote this modified problem. In later sections, we will show the trade-off between these two problem formulations. Intuitively, problem DS has a more stringent constraint, which implies that its performance might be worse than that of the problem DL.

3.2 Distributed Algorithm and Stability Analysis

This section presents the derivation of the distributed control algorithm and its stability analysis for problem DL. The standard primal-dual technique is employed to find the solution. Note that the objective function does not have the convex property and there will be a duality gap. We will ignore this issue now since we will show that the duality gap actually disappears naturally in our problem. The Lagrangian function associated with problem DL is

$$L(\vec{q},\vec{x}) = \sum_{s=1}^{S} E[N_s^t|x_s]\log(x_s) - \sum_{l=1}^{L} q_l(\sum_{s=1}^{S} A_s^l E[N_s^t|x_s]x_s - R_l) \tag{3.14}$$

where $\vec{q} = (q_1, q_2, \ldots, q_L)$ are the Lagrangian multipliers, and they represent the level of congestion. q_l has the same interpretation as the dual variable μ_l introduced in Chapter 2. Then, the dual of problem DL is defined as

$$\min_{\vec{q}\geq 0} F(\vec{q}) \tag{3.15}$$

where

$$F(\vec{q}) = \max_{\vec{x}\in X} L(\vec{q},\vec{x}) \tag{3.16}$$

Let D denote the dual problem. To solve problem D, we consider the problem in (3.16) first. For a given $\vec{q}$, the problem is separable in s, $\vec{x}(\vec{q})$ maximizes $L(\vec{q},\vec{x})$ if and only if $\vec{x}(\vec{q}) = (x_1(\vec{q}), x_2(\vec{q}), \ldots, x_S(\vec{q}))$, where

$$x_s^*(\vec{q}) = \arg\max_{0<x_s\leq M_s}\{E[N_s^t|x_s]\log(x_s) - E[N_s^t|x_s]x_s\sum_{l=1}^{L} A_s^l q_l\} \tag{3.17}$$

Substitute $E[N_s^t|x_s] = \lambda_s/(\mu_s x_s) = \rho_s/x_s$ into (3.17). Then, the solution of (3.17) can be expressed as

$$\begin{aligned} x_s^*(\vec{q}) &= \arg\max_{0<x_s\leq M_s} \{\log(x_s)/x_s\} \\ &= \min\{\arg\max(\log(x_s)/x_s), M_s\} \end{aligned} \tag{3.18}$$

Since the function $\log(x)/x$ strictly increases first and then strictly decreases, the solution given in (3.18) is a global optimal solution.

An interesting observation is that the solution of x_s is **independent** of the dual variable $\vec{q}$. In other words, we are showing that the utility maximization is fully decoupled from the stability issue. This implies that the algorithm does not require the feedback from the network in finding the optimal transmission rate, and the duality gap does not affect the optimization at all. This result actually simplifies the implementation of the control algorithm significantly. In classical literatures about distributed utility maximization algorithm, the noise and delay associated with the feedback of dual variable updates usually create non-trivial difficulties. Although some recent works have demonstrated that the algorithm will converge to the optimal solution, they usually require assumptions such as the noise must be unbiased and the variance of the noise must be bounded. The details can be found in [20].

The role of the dual variable updates is to stabilize the network and thus prevent network congestion. However, for long-term average, this step is naturally achieved. If the traffic intensity ρ_s is strictly within the stability region Θ, then we can show that the network is stable, where the network stability criterion is given by

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t 1_{\{\sum_{s=1}^S n_s(t)+\sum_{l=1}^L q_l(t)>M\}}dt \to 0 \ as\ M \to \infty \tag{3.19}$$

where n_s denotes the number of flows in class s. In other words, the number of flows at each source node and the queues at each link must be finite. By Little's law, we have

$$\begin{aligned} E[N_s|x_s] &= E[N_s^w|x_s] + E[N_s^t|x_s], \quad \forall s \\ &= \lambda_s T/2 + \rho_s/x_s \end{aligned} \tag{3.20}$$

where the term $T/2$ comes from the fact that given a Poisson arrival occurs within interval $[0, T]$, the expected arrival time is $T/2$. To have a bounded number of flows, x_s must be strictly greater than zero and the mean of the file length must be finite. According to (3.18), $x_s > 0$ is satisfied, and ρ_s is finite by definition. As a result, the first term within lim sup of (3.19) converges to zero. Since the system is not lossy, the load injected into the network is ρ_s by each class in equilibrium. Thus, the load imposed on each link is $\sum_{s=1}^{S} A_s^l \rho_s$. If $\vec{\rho} \in \Theta$ is satisfied, queues at each link will be bounded for all work conserving scheduling policies, and this fact provides the convergence of the second term in (3.19).

3.3 Distributed Algorithm for Instantaneous Congestion Control

Let us now consider the problem where the allocations are such that the instantaneous capacity constraints are not allowed to be violated. i.e. we study the solution for the problem DS, which provides instantaneous congestion avoidance. Again, primal-dual method is applied. The Lagrangian function associated with problem DS is

$$L(\vec{q}, \vec{x}) \;=\; \sum_{s=1}^{S} E[N_s^t | x_s] \log(x_s) - \sum_{l=1}^{L} q_l \Big(\sum_{s=1}^{S} A_s^l n_s^t x_s - R_l\Big) \tag{3.21}$$

where $\vec{q} = (q_1, q_2, \ldots, q_L)$ are the Lagrangian multipliers for link capacity constraints. Then, the dual of problem DS is defined as

$$\min_{\vec{q} \geq 0} F(\vec{q}) \tag{3.22}$$

where

$$F(\vec{q}) = \max_{\vec{x} \in X} L(\vec{q}, \vec{x}) \tag{3.23}$$

For a given $\vec{q}$, the problem is separable in s, $\vec{x}(\vec{q})$ maximizes $L(\vec{q}, \vec{x})$ if and only if $\vec{x}(\vec{q}) = (x_1(\vec{q}), x_2(\vec{q}), \ldots, x_S(\vec{q}))$, where

$$\begin{aligned} x_s^*(\vec{q}) &= \arg \max_{0 < x_s \leq M_s} \{E[N_s^t | x_s] \log(x_s) - n_s^t x_s \sum_{l=1}^{L} A_s^l q_l\} \\ &= \arg \max_{0 < x_s \leq M_s} \{\frac{\rho_s \log x_s}{x_s} - n_s^t x_s \sum_{l=1}^{L} A_s^l q_l\} \end{aligned} \tag{3.24}$$

The dual problem is solved by using gradient projection method. The partial derivative of $L(\vec{q}, \vec{x})$ is

$$\frac{\partial}{\partial q_l} L(\vec{q}, \vec{x}) = R_l - \sum_{s=1}^{S} A_s^l n_s^t x_s \tag{3.25}$$

Thus, q_l is updated through

$$q_l(k+1) = [q_l(k) + \gamma(\sum_{s=1}^{S} A_s^l n_s^t x_s - R_l)]^+, \qquad \forall l \tag{3.26}$$

where γ is the step-size. To ensure the convergence of this algorithm, we set $\gamma = 1/k$. Since n_s^t is a random variable, x_s^* and q_l will converge to two stochastic processes. As the result of the projection operation $[\cdot]^+$, $E[q_l] > 0$ and the second term in (3.24) will be a non-negative number all the time. Therefore, x_s^* will be always smaller than or equal to the solution given in (3.18), and $E[x_s^*] < e$. Since $g(\vec{x})$ is a strictly increasing function until it reaches its global maxima, $x_s = e$, the performance of the solution given by (3.24) and (3.26) will be worse than that of the long-term congestion avoidance algorithm. In addition, a smaller transmission rate will induce a longer delay.

Thus, the trade-offs between instantaneous and long-term congestion avoidance are utility and delay. If stability is the only requirement, the long-term congestion control solution has much more advantages in terms of implementation and complexity. For this reason, all the discussion from this point on will be focusing on the algorithm of problem DL unless explicit explanation is made.

3.4 Queueing Interpretation and Discussion

We now provide intuitive explanation for the results described in Section 3.2. First of all, we will justify the fact that the queuing process at each source node is ergodic as mentioned in Section 3.1. In Fig. 3.2, we illustrate a simplified version of the queueing systems under investigation. For the ease of exposition, we concatenate the two queues that hold both n_s^w (waiting queue) and n_s^t (transmission queue) into one transport layer queue. Since the first queue is a pure delay block, $n_s^w(t)$ is stationary. Moreover, the second

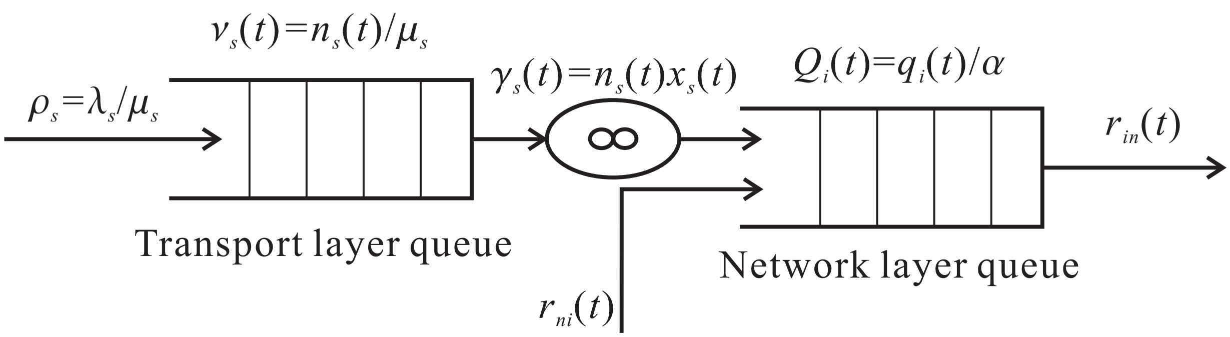

Figure 3.2: Relationship Between Transport Layer and Network Layer Queues

queue is of $G/M/\infty$ type because the service rate is scaled with $n_s^t(t)$. As a result, this queue is “self-stabilizing” and stationary, i.e., it is always stable no matter what intensity ρ_s is. Consequently, $n_s(t) = n_s^w(t) + n_s^t(t)$ is also a stationary process. Note that the arrival process of the second queue is in the form of periodic bursts with varying number of customers.

Secondly, the network achieves the largest stability region without the time-scale separation assumption. Given the fact that $\vec{\rho} \in \Theta$, stabilizing the network layer queue is straightforward: a normal FIFO policy would work [6]. Indeed, any work conserving policy will ensure stability. This fact can be proved by looking at the expected one-step drift of the queues:

$$E[q_l(k+1) - q_l(k)] = \sum_{s=1}^{S} A_s^l x_s E[N_s^t] - R_l \tag{3.27}$$

where x_s is given by (3.18). In fact, $E[N_s^t]$ should be written as $E[N_s^t|x_s]$ because N_s^t is a function of the control strategy. From the previous discussion, $E[N_s^t|x_s] = \rho_s/x_s$. Substitute this expression into the right hand side of (3.27) to obtain $\sum_{s=1}^{S} A_s^l \rho_s - R_l \leq 0$, which indicates that the expected one-step drift is non-positive.

Our results hold for a large class of file length distribution including even heavy tail distributions with finite mean; the performance of the algorithm is insensitive to this distribution. However, the expected number of flows at each source node and their sojourn time are linearly related to the mean of the distribution.

The utility function can be expressed in a more general form as well, and the convexity of $g(\vec{x})$ can be more complicated. Thus, there may be multiple optimal solutions which maximizes (3.6). However, from the system utility maximization's point of view, the mass distribution over these points will not affect the total utility. This fact implies that the structure of $\nu(\vec{x})$ can still be a delta function.

If the utility function is a linear function of the transmission rate ($U(x) = \gamma x$), the system utility is constant and independent of transmission rate. A simple proof is

$$\begin{aligned} & \max_{\vec{x}\in X} \sum_{s=1}^{S} E[N_s^t|x_s]U_s(x_s) \\ = & \max_{\vec{x}\in X} \sum_{s=1}^{S} \frac{\rho_s}{x_s}\gamma x_s \\ = & S\gamma \end{aligned}$$

However, from individual user's point of view, individual flow's utility is maximized if $x_s = M_s$.

In addition, if we take the general α utility (2.23) introduced by Mo and Walrand [14], then the optimal solution is given by

$$x_s^* = \begin{cases} M_s, \text{if } \alpha > 1 \\ \\ 0, \text{if } 0 < \alpha < 1 \end{cases} \tag{3.28}$$

for $\alpha \neq 1$ and all s. The solution can be obtained by checking the first order partial derivative of $g(\vec{x})$

$$\begin{aligned} g(\vec{x}) &= \sum_{s=1}^{S} U_s(x_s)E[N_s^t|\vec{x}] \\ &= \sum_{s=1}^{S} w_s \frac{x_s^{1-\alpha}}{1-\alpha}\frac{\rho_s}{x_s} \\ &= \frac{w_s\rho_s}{1-\alpha}x_s^{-\alpha} \\ \frac{\partial}{\partial x_s}g(\vec{x}) &= \frac{\alpha w_s\rho_s}{\alpha-1}x_s^{-\alpha-1} \end{aligned}$$

Note that $g(\vec{x})$ is either a strictly increasing or strictly decreasing function depending on α. Consequently, the optimal solution is one of the two boundary points. This solution shows that the general α utility (apart from the log utility that we have taken) is not suitable in the case of dynamic flows as $x_s = 0$ is not a feasible solution.

3.5 Delay Analysis

In this section, we present the transport layer delay analysis of our algorithm. The queueing model is shown in Fig. 3.2. Let D_s^{ee} denote the end-to-end delay of class s flows. Then we have $D_s^{ee} = D_s^t + D_s^q$, where D_s^t and D_s^q are the queueing delay at transport layer and within the network (along the path towards destination), respectively. The evaluation of network queueing delay is out of the scope of this thesis. In this thesis, we only focus on D_s^t.

The transport layer delay for class s flows D_s^t consists of the waiting time and the transmission time. Let W be the random variable denoting the waiting time and F_s be the random variable denoting the file length of class s. Since the arrival process is Poisson, given that an arrival occurs, W has an uniform distribution in the interval $[0, T]$. The transport layer delay can be written as

$$D_s^t = W + F_s/x_s \tag{3.29}$$

where the second term F_s/x_s is an exponential distribution with rate $x_s\mu_s$. Since the arrival time is independent of the file length, the distribution of D_s^t is given by the convolution of the distribution of W with the distribution of F_s/x_s.

$$f_{D_s^t}(d) = \int f_w(d-\tau) f_{F_s/x_s}(\tau) d\tau \tag{3.30}$$

$$f_{D_s^t}(d) = \begin{cases} \frac{1}{T}(1 - e^{-\mu_s x_s d}), \text{if } 0 \le d \le T \\ \\ \frac{1}{T} e^{-\mu_s x_s d}(e^{\mu_s x_s T} - 1), \text{if } d \ge T \end{cases} \tag{3.31}$$

3.6 Numerical Results

In this section, we will compare our proposed algorithm with the works introduced in Chapter 2 for networks with random arrivals and departures to demonstrate the superiority of our scheme. To facilitate our discussion, let A denote our proposed algorithm. Let B and C denote the algorithms with time-scale separation assumption and the one proposed by Lin and Shroff without time-scale separation assumption respectively. We will consider both one-hop and multi-hop network configurations in simulation. The following two objectives will be compared.

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{S} \int_0^t n_s^t(t) U_s(x_s(t)) dt \tag{3.32}$$

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t n_s^t(t) x_s(t) dt, \forall s \tag{3.33}$$

(3.32) is the average system utility and (3.33) is the average throughput for each class. Before we demonstrate the simulation results, we will present a brief description about the operations of algorithms B and C. All algorithms run in discrete time and time is slotted with length T.

3.6.1 Operation of Algorithm B

With time-scale separation assumption, algorithm B solves the following optimization problem at the beginning of each time slot.

$$\max_{x\in X} \sum_{s=1}^{S} n_s \log(x_s) \tag{3.34}$$

$$\text{subject to } \sum_{s=1}^{S} A_s^l n_s x_s \le R_l, \forall l \tag{3.35}$$

where x_s denote the individual flow transmission rate of class s, and n_s is the number of class s flows in the system. The Lagrangian is given by

$$L(x, \vec{\mu}) = \sum_{s=1}^{S} n_s \log(x_s) - \sum_{l=1}^{L} \mu_l (\sum_{s=1}^{S} A_s^l n_s x_s - R_l) \tag{3.36}$$

where $\vec{\mu} = \{\mu_1, \mu_2, ..., \mu_L\}$. For a given $\vec{\mu}$, the solution of x_s is

$$x_s = \min\{1/(\sum_{l=1}^{L} A_s^l \mu_l), M_s\} \tag{3.37}$$

Suppose that M_s is a very large number, then $x_s = 1/(\sum_{l=1}^{L} A_s^l \mu_l)$. Substitute this expression into the complementary slackness equation, we have

$$\mu_l(\sum_{s=1}^{S} A_s^l n_s/(\sum_{l=1}^{L} A_s^l \mu_l) - R_l) = 0, \forall l \tag{3.38}$$

The explicit expression of μ_l depends on the routing structure. Suppose the network is a one-hop network and only class s flows cross link l, then the explicit expression for μ_l is

$$\begin{aligned} &\mu_l(\frac{n_s}{\mu_l} - R_l) = 0 \\ &\rightarrow \mu_l = \frac{n_s}{R_l} \end{aligned} \tag{3.39}$$

where n_s denote the number of class s flows crossing link l. Substitute (3.39) into the primal solution, we get

$$x_s = \frac{R_l}{n_s} \tag{3.40}$$

Equation (3.40) provides us the optimal solution to which the primal-dual algorithm will converge in equilibrium.

Suppose that the number of flows is dynamic and the primal-dual algorithm employed converges on a much faster scale than the dynamic of n_s. In the extreme case, we assume that the algorithm converges instantly, and the transmission rate is updated with equation (3.40) at the beginning of each time slot for one-hop network topology. This rate update mechanism adopts the time-scale separation assumption and describes the operation of algorithm B.

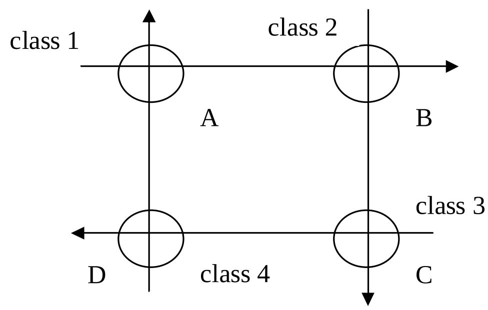

Figure 3.3: One-Hop Network Topology

3.6.2 Operation of Algorithm C

This algorithm has been introduced in Chapter 2. Here, we present the algorithm again for convenience

$$x_s(k) = \min\{\frac{1}{\sum_{l=1}^{L} q_l(k)H_s^l}, M_s\} \quad (3.41)$$

$$q_l(k+1) = [q_l(k) + \gamma_l(\sum_{s=1}^{S} H_s^l x_s(k) \int_{kT}^{(k+1)T} n_s^t(t)dt - TR_l)]^+ \quad (3.42)$$

where γ_l is the step-size. The main result claimed in [9] is that $x_s(t)$ and $n_s(t)$ will converge to stationary processes and the network can achieve the largest stability region Θ, provided the step-size is small enough.

3.6.3 Performance Comparison in One-hop Network

The network topology is shown in Fig. 3.3. This network has four links: AB, BC, CD and DA. Each link has a capacity of 10 units/second. There are four classes of flows whose file lengthes are exponentially distributed with a mean of 1 unit/flow. The arrival rates are 8, 8.5, 9 and 9.5 flows/second for class 1 to class 4. Thus, the loads brought by each class are 8, 8.5, 9 and 9.5 units/second. Each time slot is 10ms seconds long. The simulation results are shown in Table 3.1 and 3.2. Note that Table 3.1 also includes the relative performance comparison between algorithm A and C with algorithm C's performance as the baseline. Since algorithm B's performance is very low, its relative performance with respect to that of algorithm C is not included.

Table 3.1: Time Average Utility Comparison for One-hop Network

	A (% improvement)	B	C (baseline)
class 1	2.983 (+58.67%)	1.29	1.88
class 2	3.14 (+53.92%)	0.0055	2.04
class 3	3.32 (+46.26%)	-4.4764	2.27
class 4	3.47 (+29.96%)	-19.7793	2.67

Table 3.2: Time Average Throughput Comparison for One-hop Network

	A	B	C
class 1	8.108	8.02	8.06
class 2	8.54	8.46	8.53
class 3	9.01	9.03	9
class 4	9.43	9.54	9.5

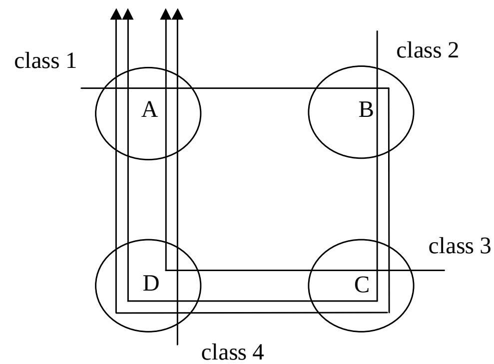

Figure 3.4: Multi-Hop Network Topology

Table 3.3: Time Average Utility Comparison for Multi-hop Network

	A (% improvement)	C (baseline)
class 1	0.3572 (+29.84%)	0.2751
class 2	0.7372 (+31.93%)	0.5588
class 3	1.1042 (+31.72%)	0.8383
class 4	1.2922 (+33.84%)	0.9655

Note that each class's throughput should be the same for all algorithms theoretically. The discrepancy appears in Table 3.2 is due to simulation.

3.6.4 Performance Comparison in Multi-hop Network

To further emphasize the advantages of our algorithm, we also investigate its performance in a multi-hop network shown in Fig. 3.4. The network parameters are identical to the previous example except the routing and arrival rates. In this example, the arrival rates are 1, 2, 3 and 3.5 flows/second for class 1 to class 4. Therefore, the load on link AB, BC, CD and DA are 1, 3, 6 and 9.5 units/second. Since the performance of algorithm B is not comparable with that of algorithm A and C, only A and C's simulation results are shown in Table 3.3 and 3.4.

Table 3.4: Time Average Throughput Comparison for Multi-hop Network

	A	C
class 1	0.9712	0.99
class 2	2	1.99
class 3	3	3
class 4	3.51	3.49

3.6.5 Simulation Results Discussion

According to the simulation results, algorithm A performs much better than the other two algorithms and maintains the throughput at the same time. This result can be explained from two different prospectives.

First, we analyze the algorithm from the stability's point of view. In classical literatures, utility maximization problem usually considers networks with fixed number flows. In addition, each flow is assumed to have infinite backlog to transfer. Therefore, the dual variable must be employed to regulate the flows to enure stability. However, when flow's arrival and departure are random, stability is not an issue if $\vec{\rho} \in \Theta$ is met and the transmission rate is strictly greater than zero. For this reason, the dual variable is not required to regulate the flows, and each flow will receive more utility. In some sense, it is a trade-off between stability and utility. If we know the system is operating within the stable region, we should not penalize the flows to ensure stability anymore.

Secondly, from the prospective of solution space, we can also verify the advantage of open-loop control. The constraints associated with these algorithms specify different solution space. For algorithm A, the solution is selected from a space which ensures long-term stability. For algorithm B and C, the solutions are chosen from spaces which ensure instantaneous and short-term congestion avoidance. If we rank these spaces according to their sizes, $A \supseteq C \supseteq B$. As the result, the performance of our algorithm should be at least as good as that of B and C. This analysis is consistent with the simulation results.

Chapter 4

Conclusion

The main contribution of this work is that a systematic study of utility maximization problem in networks with random user arrivals and departures is presented. We have found that the network utility maximization is independent of the network stability issue. If the network is operating within the capacity region and the network layer adopts a work conserving scheduling policy, the queue at each link remains finite. The time-scale separation assumption has no impact in determining the stability region. One way of interpreting these results is that primal-dual based congestion control schemes should be used for long-lived flows to prevent short-term congestion while short-lived flows need not to be controlled provided they do not bring excessive work.

For future works, we would like to take delay into consideration in addition to utility maximization. As we know, log utility function preserves the properties of an NBS, which will guarantee social welfare. However, it does not characterize individual's delay profile. Specially, in the context of dynamic arrivals and departures, delay is a very critical performance measurement. Whereas in the context of static networks, only fairness and utility are critical factors. Thus, utility function is not a complete reflection of the user's level of satisfaction in dynamic case. A new framework should be developed for dynamic connections with emphasis on delay performance. Perhaps, this can be done by using another type of utility function.

Bibliography

[1] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massachusetts, 1995.

[2] T. Bonald. Throughput Performance in Networks with Linear Capacity Contraints. In *Proc. of CISS*, 2006.

[3] T. Bonald and L. Massoulié. Impact of Fairness on Internet Performance. In *Proc. of the ACM SigMetric*, 1999.

[4] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource Allocation and Cross-Layer Control in Wireless Networks. *Foundations and Trends in Networking*, 1(1):1–144, 2006.

[5] F. P. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[6] P. R. Kumar. Re-entrant Lines. *Queueing Systems: Theory and Applications*, 13(1):87–110, 1993.

[7] J. W. Lee, R. Mazumdar, and N. B. Shroff. Opportunistic Power Scheduling for Dynamic Multi-Server Wireless Systems. *IEEE Transactions on Wireless Communications*, 5(6):1506–1515, 2006.

[8] X. Lin and N. B. Shroff. On the Stability Region of Congestion Control. In *42rd Annual Allerton Conference on Communication, Control, and Computing*, 2004.

[9] X. Lin and N. B. Shroff. The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks. *IEEE/ACM Transactions on Networking*, 14(2):302–315, 2006.

[10] X. Lin and N. B. Shroff. Utility Maximization for Communication Networks With Multipath Routing. *IEEE Transactions on Automatic Control*, 51(5):766–781, 2006.

[11] L. Ljung. Analysis of Recursive Stochastic Algorithms. *IEEE Transactions on Automatic Control*, AC-22(4):551–575, 1977.

[12] S. H. Low and R. Srikant. A Mathematical Framework for Designing a Low-Loss, Low-Delay Internet. *Networks and Spatial Economics*, 4(1):75–101, 2004.

[13] L. Massoulie. *Structural Properties of Proportional Fairness: Stability and Insensitivity.* Submitted to the Annals of Applied Probability, 2005.

[14] J. Mo and J. Walrand. Fairness End-to-End Window-based Cogestion Control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

[15] R. T. Rockafellar and R. Wets. *Stochastic Convex Programming: Basic Duality*, volume 62. Pacific Journal of Mathematics, 1976.

[16] G. Sharma, N. B. Shroff, and R. Mazumdar. Randomized Algorithms for Cross-Layer Network Control. In *Proc. of MILCOM*, 2006.

[17] A. Stolyar. *Mazimizaing Queueing Networks Utility Subject to Stability: Greedy Primal-Dual Algorithm*, volume 50. Queueing System, 2005.

[18] M. Vojnovic and J-Y. Boudec. On the Long-Run Behavior of Equation-Based Rate Control. In *Proc. of ACM Sigcom*, 2002.

[19] H. Yaiche, R. Mazumdar, and C. Rosenberg. A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks. *IEEE/ACM Transactions on Networking*, 8(5):667–678, 2000.

[20] J. Zhang, D. Zheng, and M. Chiang. The Impact of Stochastic Noisy Feedback on Distributed Network Utility Maximization. In *Proc. of the 26th IEEE INFOCOM*, 2007.