On the performance of primal/dual schemes for congestion control in networks with dynamic flows

Luo, Jun.; Ma, Kexin.; Ravi Mazumdar.

2008

Luo, J., Ma, K., & Ravi, M. (2008). INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. On the performance of primal/dual schemes for congestion control in networks with dynamic flows. 2008:(pp. 326-330)

https://hdl.handle.net/10356/90695

https://doi.org/10.1109/INFOCOM.2008.74

On the Performance of Primal/Dual Schemes for Congestion Control in Networks with Dynamic Flows

Kexin Ma[†] Ravi Mazumdar[*] Jun Luo[*]
[†]Qualcomm Incorporated CDMA Technologies
San Diego, CA, USA
Email: kma@qualcomm.com
[*]Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, ON, Canada
Email: mazum@eceweb.uwaterloo.ca, j7luo@engmail.uwaterloo.ca

Abstract— **Stability and fairness are two design objectives of congestion control mechanisms; they have traditionally been analyzed for long-lived flows (or *elephants*). It is only recently that short-lived flows (or *mice*) have received attention. Whereas stability has been established for the existing primal-dual based control mechanisms, the performance issue has been largely overlooked. In this paper, we study utility maximization problems for networks with dynamic flows. In particular, we consider the case where sessions of each class results in flows that arrive according to a Poisson process and have a length given by a general distribution. The goal is to maximize the long-term expected system utility that is a function of the number of flows and the rate (identical within a given class) allocated to each flow. Our results show that, as long as the average amount of work brought by the flows is strictly within the network stability region, the rate allocation and stability issues are decoupled. While stability can be guaranteed by, for example, a FIFO policy, utility maximization becomes an unconstrained optimization that results in a static rate allocation for flows. We also provide a queueing interpretation of this seemingly surprising result and show that not all utility functions make sense for dynamic flows. Finally, we use simulation results to show that indeed the open-loop algorithm maximizes the expected system utility.**

I. Introduction

Congestion control plays an important role in modern communication networks. It involves two complementary design objectives, namely *stability* and *fairness*. While the stability prevents the delay from going beyond a tolerable level, the fairness is measured through a utility function that represents the satisfaction of a flow on the assigned resource or transmission rate. Neglecting either objective can result in a trivial problem, because, for example, stability can be universally achieved by always assigning zero rate to a flow. A well designed congestion control scheme will maintain the network stability while optimizing a criterion based on the user utilities such as certain functions of the assigned rate. The seminal work of Kelly, Maulloo and Tan [3] provides a fundamental framework to solve such a global optimization problem in the context of wired networks with a fixed number of flows. The most important message conveyed by [3] is that the original global optimization problem can be solved by a distributed iterative algorithm.

Based on [3], congestion control has been studied extensively in the context of wired networks. Yaiche, Mazumdar and Rosenberg [13] studied the 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 contrast to [3], [13] uses the user throughputs to determine the performance characteristics. A primal-dual based framework was then introduced to solve for the user rates in a distributed manner. A formulation using general concave utility functions but with the same structural results can be found in [9]. More recently, Lin and Shroff [7] adopt the same techniques used in [13] and extend the results to the networks where multipath routing is allowed. In this case, the utility function is not strictly concave. Thus, a modified version of primal-dual algorithm based on a penalty function is proposed.

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 [10]. There was no strong proof that these mechanisms would meet the stability and fairness objectives when subjected to dynamic flows.

Recently, research has focused on networks with flows that arrive and depart dynamically [1], [6] with the aim of studying the stability issues related to the primal-dual framework. Bonald and L. Massoulié [1] 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 [6] 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. They considered logarithmic utilities as in [13], as well as a more general class of utility functions parameterized by a parameter α, introduced in [12] and referred to as fair bandwidth sharing in the literature. The advantage is that this allows one to study a very large class of behaviors corresponding to max-min fairness, proportional fairness, and other points on the Pareto surface. The analysis was based on Poisson arrivals and exponential holding times for the flows. These results were extended to the case of more general holding time distributions in [11] assuming time-scale separation and the effect of the utility functions was also presented suggesting that logarithmic utilities are more desirable. More recently [2] presents a fluid network approach for general file size distributions for the fair bandwidth sharing policy.

By now there have been a large number of extensions related to the stability issue for both wireline and wireless networks when using primal-dual type algorithms [8]. However, one key connection that has been missing is to relate the primal-dual algorithms to the original optimization problem. In other words, what **objective** does the primal-dual mechanism optimize when flows are dynamic? If fairness objective is not in the picture and stability is the only consequence, some simpler mechanisms might suffice. Addressing this question is indeed the focus of this paper.

In this paper, we study the utility maximization problem in networks with dynamic flows. We assume the flow length is random with finite second moment 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, an appropriate objective would be 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 as has been assumed in the prior work on stability, we can achieve the stability and fairness objectives independently. In other words, utility maximization becomes an open-loop procedure. Moreover, we investigate the system steady-state behavior in terms of delay. Finally, we study via simulations the open-loop approach as well as algorithms in [1], [6]. The results demonstrate that, while all these algorithms guarantee stability, the approach presented here maximizes the long-term expected utility.

The rest of the paper is structured as follows. In Section II, we present the system model and problem formulation. Section III and Section IV obtain the principal result showing that the stability and control issues can be decoupled. Section V provides an explanation on the results in Section III through queueing interpretation; we also discuss the possible file length distribution and issue of choosing the right utility functions in this section. Section VI considers the steady-state behavior in terms of delay. We report our simulation results in Section VII before finally conclude our paper in Section VIII.

II. 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 $L \times S$ incidence or routing matrix that represents the routes of the flows: $A_s^l = 1$ if the flows of class $s \in S$ uses 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 each flow brings with it a file for transfer whose size is exponentially distributed 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 $\Theta = \{\vec{\rho} | \sum_{s=1}^{S} A_s^l \rho_s \le R^l, \forall l\}$.

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] We assume that each flow 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.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:

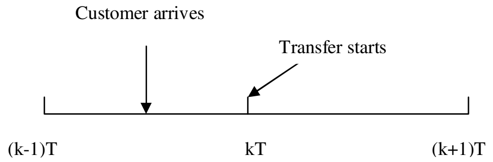

Fig. 1. Transmission Model

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

$$\text{subject to } \lim_{t\to\infty} \frac{1}{t} \int_{t=0}^{\infty} \sum_{s=1}^{S} A_s^l n_s^t(t) x_s(t) dt \le R_l, \quad \forall l \qquad (2)$$

(1) has an interpretation of maximizing expected long-term system utility. The constraint (2) addresses stability requirement.

[1] Our results can be readily extended to the case where the link capacity is time-varying and the routes are not pre-defined. We refer to our companion work [5] for details.

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

Suppose the queueing process at each source node is in equilibrium and ergodic (we will justify this assumption in Section V). Let $\nu(\vec{n}^t, \vec{x})$ denote the density (which we assume exists without 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 of product form (follows from a truncation of the $M/G/\infty$ model). Consequently, let $\nu(n_s^t|\vec{x})$ denote the conditional density of flow n_s^t given $\vec{x}$, then we have $\nu(\vec{n}^t, \vec{x}) = [\prod_{s=1}^{S} \nu(n_s^t|\vec{x})] p(\vec{x})$, where $p(\vec{x})$ denotes the joint density of $\vec{x}$.

Then (1) can be written as the following problem:

$$\max_{\vec{x}} \int_X \sum_{s=1}^{S} \left[\sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) \nu(n_s^t|\vec{x}) \right] p(\vec{x}) d\vec{x} \tag{3}$$

subject to

$$\int_X \sum_{s=1}^{S} A_s^l \left[\sum_{n_s^t=0}^{\infty} n_s^t x_s \nu(n_s^t|\vec{x}) \right] p(\vec{x}) d\vec{x} \leq R_l, \quad \forall l \tag{4}$$

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 density $p(\vec{x})$. Define

$$\begin{aligned} g(\vec{x}) &= \sum_{s=1}^{S} \sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) \nu(n_s^t|\vec{x}) \\ &= \sum_{s=1}^{S} \log(x_s) \sum_{n_s^t=0}^{\infty} n_s^t \nu(n_s^t|\vec{x}) \\ &= \sum_{s=1}^{S} \log(x_s) E[N_s^t|\vec{x}] \\ &= \sum_{s=1}^{S} \log(x_s) \rho_s / x_s \end{aligned} \tag{5}$$

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 (5) into (3), we have

$$\max_{\vec{x}} \int_X g(\vec{x}) p(\vec{x}) d\vec{x} \tag{6}$$

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

$$\frac{\partial g(\vec{x})}{\partial x_s} = \frac{\rho_s [1 - \log(x_s)]}{x_s} \tag{7}$$

$$\frac{\partial^2 g(\vec{x})}{\partial x_s^2} = \frac{\rho_s [2\log(x_s) - 3]}{x_s^3} \tag{8}$$

$$\frac{\partial^2 g(\vec{x})}{\partial x_s x_t} = 0, \; s \neq t \tag{9}$$

From (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 (8) and (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 (7).

To maximize (6), $p(\vec{x})$ should put all its mass at $\vec{x^*}$ if $\vec{x^*}$ satisfies (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) and (4) in to a deterministic one in the following:[3]

$$\max_{\vec{x} \in X} \sum_{s=1}^{S} E[N_s^t|x_s] U_s(x_s) \tag{10}$$

$$\text{subject to } \sum_{s=1}^{S} A_s^l E[N_s^t|x_s] x_s \leq R_l, \quad \forall l \tag{11}$$

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:

$$\int_X p(\vec{x}) d\vec{x} = \int_{\mathcal{X}} p(\vec{x}) d\vec{x} = 1 \tag{12}$$

where $\mathcal{X}$ is the set of $\vec{x}$ that solve problem (10) and (11). The solution is straightforward: $p(\vec{x})$ can be any distribution as long as (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 (11) refers to long-term congestion avoidance. If instantaneous congestion avoidance is required, (11) will be replaced by

$$\sum_{s=1}^{S} A_s^l n_s^t x_s \leq R_l, \quad \forall l, t \tag{13}$$

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.

III. 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

[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.

disappears naturally in our problem. The Lagrangian function associated with problem DL is

$$\begin{aligned} L(\vec{q},\vec{x}) &= \sum_{s=1}^{S} E[N_s^t|x_s]\log(x_s) \\ &\quad -\sum_{l=1}^{L} q_l(\sum_{s=1}^{S} A_s^l E[N_s^t|x_s]x_s - R_l) \end{aligned} \qquad (14)$$

where $\vec{q} = (q_1, q_2, \ldots, q_L)$ are the Lagrangian multipliers, and they represent the level of congestion. Then, the dual of problem DL is defined as

$$\min_{\vec{q}\geq 0} F(\vec{q}) \qquad (15)$$

where

$$F(\vec{q}) = \max_{\vec{x}\in X} L(\vec{q},\vec{x}) \qquad (16)$$

Let D denote the dual problem. To solve problem D, we consider the problem in (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

$$\begin{aligned} x_s^*(\vec{q}) &= \arg\max_{0<x_s\leq M_s} \{E[N_s^t|x_s]\log(x_s) \\ &\quad -E[N_s^t|x_s]x_s \sum_{l=1}^{L} A_s^l q_l\} \end{aligned} \qquad (17)$$

Substitute $E[N_s^t|x_s] = \lambda_s/(\mu_s x_s) = \rho_s/x_s$ into (17). Then, the solution of (17) can be expressed as

$$\begin{aligned} x_s^*(\vec{q}) &= \arg\max_{0<x_s\leq M_s} \{\log(x_s)/x_s\} \\ &= \min\{\arg\max(\log(x_s)/x_s), M_s\} \end{aligned} \qquad (18)$$

Since the function $\log(x)/x$ strictly increases first and then strictly decreases, the solution given in (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, 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 literature 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 [14].

The role of the dual variable updates is to stabilize the network and thus prevent network congestion. However, for the 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 via a standard Lyapunov argument based on the conditional drifts that the network is stable, where the network stability criterion is given by

$$\begin{aligned} \limsup_{t\to\infty} \frac{1}{t}\int_0^t 1_{\{\sum_{s=1}^{S} n_s(t)+\sum_{l=1}^{L} q_l(t)>M\}} dt \to 0 \\ as\ M \to \infty \end{aligned} \qquad (19)$$

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

$$\begin{aligned} E[N_s|x_s] &= E[N_s^w|x_s] + E[N_s^t|x_s], \quad \forall s \\ &= \lambda_s T/2 + \rho_s/x_s \end{aligned} \qquad (20)$$

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 (18), $x_s > 0$ is satisfied, and ρ_s is finite by definition. As a result, the first term within $\limsup$ of (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 (19).

IV. 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

$$\begin{aligned} L(\vec{q},\vec{x}) &= \sum_{s=1}^{S} E[N_s^t|x_s]\log(x_s) \\ &\quad -\sum_{l=1}^{L} q_l(\sum_{s=1}^{S} A_s^l n_s^t x_s - R_l) \end{aligned} \qquad (21)$$

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

$$\min_{\vec{q}\geq 0} F(\vec{q}) \qquad (22)$$

where

$$F(\vec{q}) = \max_{\vec{x}\in X} L(\vec{q},\vec{x}) \qquad (23)$$

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

$$\begin{aligned} x_s^*(\vec{q}) &= \arg\max_{0<x_s\leq M_s} \{E[N_s^t|x_s]\log(x_s) - n_s^t x_s \sum_{l=1}^{L} A_s^l q_l\} \\ &= \arg\max_{0<x_s\leq M_s} \{\frac{\rho_s \log x_s}{x_s} - n_s^t x_s \sum_{l=1}^{L} A_s^l q_l\} \end{aligned} \qquad (24)$$

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

$$\frac{\partial}{\partial q_l} L(\vec{q}, \vec{x}) = R_l - \sum_{s=1}^{S} A_s^l n_s^t x_s \tag{25}$$

Thus, q_l is updated through

$$q_l(k+1) = [q_l(k) + \gamma(\sum_{s=1}^{S} A_s^l n_s^t x_s - R_l)]^+, \quad \forall l \tag{26}$$

where γ is the step-size. To ensure the convergence of this algorithm, we set $\gamma = 1/k$. Since n_s^t is random, x_s^* and q_l will converge converge as processes. As the result of the projection operation $[\cdot]^+$, $E[q_l] > 0$ and the second term in (24) will be non-negative $\forall t$. Therefore, x_s^* will be always be dominated by the solution given in (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 (24) and (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.

V. Queueing Interpretation and Discussion

We now provide intuitive explanation for the results described in Section III. First of all, we will justify the fact that the queuing process at each source node is ergodic as mentioned in Section II. In Fig. 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 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.

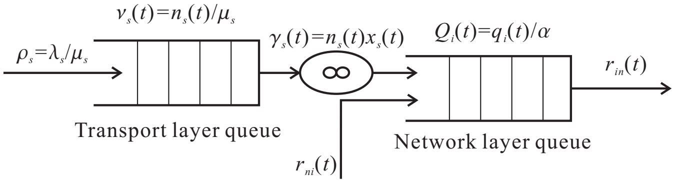

Fig. 2. Relationship Between Transport Layer and Network Layer Queues

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 [4]. Indeed, any work conserving policy will ensure stability. This fact can be proved by looking at the expected one-step drift of the queues:

$$E[q_l(k+1) - q_l(k)] = \sum_{s=1}^{S} A_s^l x_s E[N_s^t] - R_l \tag{27}$$

where x_s is given by (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 (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 (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

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

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

In addition, if we take the general α utility $U_s(x_s) = w_s \frac{x_s^{1-\alpha}}{1-\alpha}$ introduced by Mo and Walrand [12], then the optimal solution is given by

$$x_s^* = \begin{cases} M_s, \text{if } \alpha > 1 \\ 0, \text{if } 0 < \alpha < 1 \end{cases} \tag{28}$$

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 $\log(\cdot)$ utility that we have taken) is not suitable in the case of dynamic flows as $x_s = 0$ is not a feasible solution.

VI. Delay Analysis

In this section, we present the transport layer delay analysis of our algorithm. The queueing model is shown in Fig. 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 our paper. Here we 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

$$D_s^t = W + F_s/x_s \tag{29}$$

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.

$$\begin{aligned} f_{D_s^t}(d) &= \int f_w(d-\tau) f_{F_s/x_s}(\tau) d\tau \\ \Rightarrow \quad f_{D_s^t}(d) &= \begin{cases} \frac{1}{T}(1 - e^{-\mu_s x_s d}) & \text{if } 0 \le d \le T \\ \frac{1}{T} e^{-\mu_s x_s d}(e^{\mu_s x_s T} - 1) & \text{if } d \ge T \end{cases} \end{aligned}$$

VII. Numerical Results

In this section, we will compare our proposed algorithm with current works 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.

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{S} \int_0^t n_s^t(t) U_s(x_s(t)) dt \tag{30}$$

$$\lim_{t\to\infty} \frac{1}{t} \int_0^t n_s^t(t) x_s(t) dt, \forall s \tag{31}$$

(30) is the average system utility and (31) 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.

A. Operation of Algorithm B

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

$$\max_{x\in X} \sum_{s=1}^{S} n_s \log(x_s) \tag{32}$$

$$\text{subject to } \sum_{s=1}^{S} A_s^l n_s x_s \le R_l, \forall l \tag{33}$$

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

$$L(x, \vec{\mu}) = \sum_{s=1}^{S} n_s \log(x_s) - \sum_{l=1}^{L} \mu_l (\sum_{s=1}^{S} A_s^l n_s x_s - R_l) \tag{34}$$

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

$$x_s = \min\{1/(\sum_{l=1}^{L} A_s^l \mu_l), M_s\} \tag{35}$$

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

$$\mu_l (\sum_{s=1}^{S} A_s^l n_s / (\sum_{l=1}^{L} A_s^l \mu_l) - R_l) = 0, \forall l \tag{36}$$

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

$$\begin{aligned} &\mu_l(\frac{n_s}{\mu_l} - R_l) = 0 \\ &\rightarrow \mu_l = \frac{n_s}{R_l} \end{aligned} \tag{37}$$

where n_s denote the number of class s flows crossing link l. Substitute (37) into the primal solution, we get

$$x_s = \frac{R_l}{n_s} \tag{38}$$

Equation (38) 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 (38) 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.

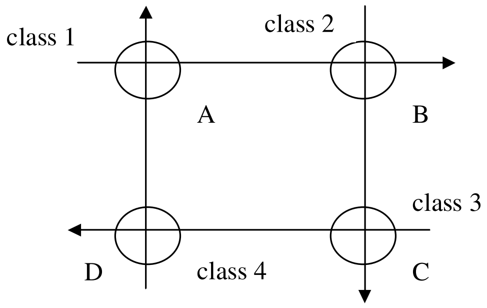

Fig. 3. One-Hop Network Topology

TABLE I
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

B. Operation of Algorithm C

The detailed derivation of this algorithm can be found in [6]. Here, we will only present the algorithm

$$x_s(q) = \min\{\frac{1}{\sum_{l=1}^{L} q_l H_s^l}, M_s\} \quad (39)$$

$$q_l(k+1) = [q_l(k) + \gamma_l(\sum_{s=1}^{S} H_s^l x_s(k) \int_{kT}^{(k+1)T} n_s^t(t)dt \\ -TR_l)]^+ \quad (40)$$

where γ_l is the step-size. The main result claimed in [6] 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.

C. Performance Comparison in One-hop Network

The network topology is shown in Fig. 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 I and II. Note that Table I 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.

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

TABLE II
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

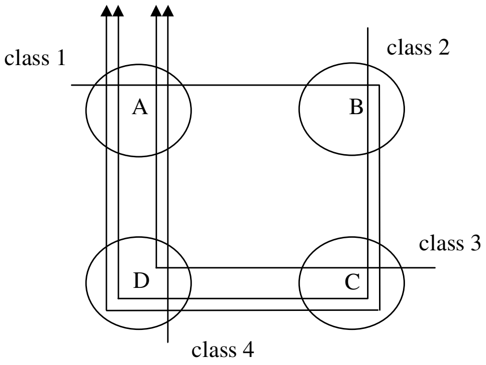

Fig. 4. Multi-Hop Network Topology

D. 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. 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 III and IV.

E. Simulation Results Discussion

According to the simulation results, algorithm A performs much better than the other two algorithms and maintains the

TABLE III
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

TABLE IV
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

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.

VIII. Concluding remarks

In this work, we have shown that, if we seek to maximize the long-term system utility subject to long-term stability in networks where random dynamic arrivals and departures are present, the control and stability issues are decoupled when the arriving sessions lie within the stability region. Our analysis shows that such an optimal control scheme is independent of the queues in the network, and the stability of the network is independent of the utility maximization problem. 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 sessions need not be controlled provided they do not bring excessive work. Such an approach will maximize network utility.

References

[1] T. Bonald and L. Massoulié. Impact of fairness on internet performance. In *Proc. of the ACM SigMetric*, 1999.

[2] C. Gromoll and R. J. Williams. Fluid limit of a network with fair bandwidth sharing and general document size distribution. Preprint, University of Virginia, 2006.

[3] 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.

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

[5] L. Le and R. Mazumdar. Performance and design of cross-layer algorithms in multihop wireless networks. Preprint, University of Waterloo http://www.ece.uwaterloo.ca/∼mazum/multihop_062007.pdf, 2007.

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

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

[8] X-J. Lin, N.B. Shroff, and R.S. Srikant. On connection-level stability of congestion-controlled communication networks. Preprint, Purdue University, 2005.

[9] S. H. Low and B.E. Lapsley. Optimization flow control: Basic algorithm and convergence. *IEEE/ACM Trans. on Networking*, 7(6):861–874, 1999.

[10] 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.

[11] L. Massoulie. Structural properties of proportional fairness: stability and insensitivity. Annals of Applied Probability, 2007, to appear.

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

[13] 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.

[14] 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.