OFEX Controller to Improve Queueing and User Performance in Multi-bottleneck Networks

Jungang Liu and Oliver W.W. Yang

We have designed and investigated a new congestion control scheme, called optimal and fully explicit (OFEX) controller. Different from existing relatively explicit controllers, this new scheme can provide not only optimal bandwidth allocation but also a fully explicit congestion signal to sources. It uses the congestion signal from the most congested link instead of the cumulative signal from the flow path. In this way, it overcomes the drawback of relatively explicit controllers exhibiting bias toward multi-bottlenecked users and significantly improves their convergence speed and source throughput performance. Furthermore, our OFEX-controller design considers a dynamic model by proposing a remedial measure against the unpredictable bandwidth changes in contention-based multi-access networks. Compared with former works/controllers, this remedy also effectively reduces the instantaneous queue size in a router and thus significantly improves queuing delay and packet loss performance. We have evaluated the effectiveness of the OFEX controller in OPNET. The experimental comparison with the existing relatively explicit controllers verifies the superiority of our new scheme.

Keywords: Congestion control, optimality, fully explicit, fairness, multi-bottleneck networks.

I. Introduction

Since the seminal work laid by Kelly [1], network utility maximization (NUM) has emerged as a promising framework in network congestion control protocol design and received wide investigation. Many literatures, for example, [2]–[8], have been published since. There are also books or tutorials on the theory of NUM; for example, [9]–[10].

Based on the two categories of existing congestion control algorithms (that is, primal and dual) generalized in [7] and [11], our investigation showed that dual algorithms should be classified more carefully into "relatively explicit algorithms" and "fully explicit algorithms" because this would help to expose the shortcomings of the existing NUM-based dual algorithms, all of which are relatively explicit protocols to the best of our knowledge. This sub-classification provides the reasons of our research and therefore motivates us to propose remedies, which form the basis of our study.

The primal algorithms, as defined in [7] and [11], broadly correspond to implicit congestion control mechanisms where noisy feedback from the network is averaged at sources using increase/decrease rules, which are commonly found in transmission control protocol (TCP) and TCP + random early detection/marking approaches [12].

For relatively explicit algorithms, they broadly correspond to congestion control approaches where sources can adjust their sending rates according to certain algorithms (for example, end-user control equations or demand functions) in response to the congestion information fed back by links. We consider the congestion signals in these types of algorithms to be relatively explicit because the links do not directly feed the allowed sending rate or rate changes back to sources. Instead, it is the duty of the sources to derive the sending rate according to their

Manuscript received Jan 10, 2013; revised Sept. 14, 2013; accepted Nov. 8, 2013.

Parts of this work were supported by a Research Discovery Grant (#RGPIN42878) and an Accelerated Grant from NSERC (Natural Sciences and Engineering Research Council), Canada.

Jungang Liu (phone: +1 6135625800 Ext. 6198, jliu115@uottawa.ca) and Oliver W.W. Yang (yang@eecs.uottawa.ca) are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ontario, Canada.

http://dx.doi.org/10.4218/etrij.14.0113.0036

own algorithms. In the relatively explicit algorithms, the Lagrangian multiplier used in optimizations is often economically interpreted by revenue management as the total price that a source has to pay for a network [13].

The fully explicit algorithms broadly correspond to the explicit congestion control protocols where the links directly allocate their capacities to the passing flows (according to some averaging mechanism) by feeding back the allowed sending rate (hence fully explicit signal) back to sources.

A primal rate and a dual rate control protocol based on utility functions has been proposed [1], in which each user s, has a utility $U_s(x_s)$ (by assuming the sending rate of user s is x_s). The objective is to maximize the aggregate utility of all users based on a series of network capacity constraints. Each user implements an additive increase and multiplicative decrease of its data rate based on feedback information cumulated from links [14]; consequently, network-wise proportional fairness is achieved among the users. A method similar to [1] was used to solve the utility function–based problem using different marking implementations of the algorithm [2]. They proved the convergence of the algorithm in both synchronous and asynchronous conditions. Specifically, each link updates its congestion signal based on the received rates. Then, the relatively explicit congestion signals from all links along the path are summed up and fed back to the sources, which use demand functions to compute new sending rates.

Based on implicit congestion information, as in TCP, a pricing mechanism [15] modeled with NUM was introduced to not only guarantee fairness but to also maximize the overall utility of users. The algorithm can adjust the congestion prices (that is, the relatively explicit congestion signal) that users will use to regulate their window size so as to achieve a system optimum. Further, the equilibrium prices are such that the system optimum can achieve a weighted proportional fairness.

In contrast to solving the congestion control problem for single-path routing with NUM, the utility function method for multi-path routing networks has been applied [6] where a distributed algorithm was developed and analyzed to solve a NUM problem. Specifically, a "dual-update algorithm" computes the relatively explicit congestion signals attached to the dual variables, and a "primal update algorithm" computes the source sending rate for different paths given the congestion signals.

To understand the connection between existing network congestion control algorithms and optimal control theory, utility functionals that can be maximized by existing algorithms have been investigated [16], and results have shown that there exist meaningful utility functionals whose maximization can lead to the celebrated primal, dual, and primal-dual algorithms.

More dual traffic control algorithms can be found in [17]–[20]. However, our investigation reveals that they all share the following issues:

1) The algorithms can "bias" multi-bottlenecked flows in terms of their source sending rates (or throughput) and their convergence performance, due to the fact that their source is adjusting the sending rates according to cumulative congestion signals. The more bottlenecks a flow passes through, the more "bias" it encounters. Such an issue is unavoidable in these kind of duality models because of their network-wise proportional fairness mechanism [21]. As a matter of fact, such an issue has already been considered in TCP/IP networks, and the correcting measures have been discussed in [22] accordingly. In this paper, our research for explicit congestion control algorithms is based on the same consideration. That is, there is no compelling reason for long flows passing through multiple bottlenecks to suffer from low throughput [22].

2) Most of the algorithms take on a fixed link bandwidth assumption that can make contention-based networks ineffective (such as, the multi-access Ethernet or IEEE 802.11) because deterministic link capacity is usually unavailable in such networks. Furthermore, there is no easy way to predict the contentions/interferences that can happen anytime in the highly dynamic networks; therefore, estimations to link bandwidth are quite often inaccurate (due to conservative or over-estimation) or simply wrong; these have resulted in stability issues [23], [24]. There are some NUM-based works that have considered varying link capacity. For example, stability/sensitivity issues were studied in a network with varying link capacity (for example, in [5]), but no remedial measure against bandwidth variations was proposed. Others (for example, [25]-[27]) have only modeled the link capacity as a function of the battery power of a node in wireless ad hoc networks and as such, these models are not applicable to wired networks.

3) The methods to match the incoming traffic rate to the link capacity in these algorithms may incur long queuing delays or even cause packet loss after an overload. For example, a sudden influx of data can cause the incoming traffic rate to become greater than the link capacity. If this overload persists, the queue will quickly build up and even overflow the buffer. The queue can never dissipate even if the incoming traffic matches (remains equal to but not less than) the link capacity afterwards. We have noticed that the traditional controllers, explicit control protocol (XCP) [28] and rate control protocol (RCP) [29], relax such an issue by considering persistent or instantaneous queue size in their models under a fixed link capacity. However, both of them are based on the classical linear time-invariant control models and not NUM-based models [30].

We also note that, so far, many works have only considered a

bandwidth of less than 150 Mbps, which is not representative of today's high-speed networks, for example, see [5], [6], [15], and [31]. There are works that consider a high link bandwidth, as in [32], but they are not fully explicit.

In view of the aforementioned shortcomings of the current NUM-based relatively explicit protocols, we are motivated to propose a new scheme to address these issues. The general objective of this work is to design a fully explicit congestion controller that is superior to existing relatively explicit controllers so as to allow for the design of a fair, stable, zero-loss, low-delay, and high-utilization network. Specifically, we would like to do the following: adapt the existing NUM approach to obtain an optimal and fully explicit rate-allocation mechanism that is capable of feeding congestion signals back from the most congested link, change the application domain of proportional fairness to fit the new scheme, model our scheme so that it is adaptable to time-varying network conditions, and simulate and evaluate our fully explicit congestion controller in high-bandwidth scenarios using the packet-based OPNET modeler [33].

To model the fully explicit controller as categorized above, we shall consider social welfare[1)] instead of end-user utility (as in the existing works). To do so, we formulate our problem based on the links of a network because they manage how much resource (that is, bandwidth) each end user should obtain, just as in the distribution of wellbeing by governments. However, the fundamental difference of our work is that our algorithm is based on the optimization for each link rather than the global optimization of all users and all links together as done in other existing works [4], [6] and [8]. This allows for a distributed algorithm where each link along a data path can desire a source sending rate, which means that we can design our new scheme with the congestion signal fed back from the most congested link (which results in network-wise and max-min fairness) instead of the cumulative congestion signal over the flow path. In this way, we avoid a centralized global optimization and reduce the traffic required to broadcast the link prices in the existing works. On the other hand, applying proportional fairness across the network in our new scheme will be impractical, and we will rethink its application in our scheme.

Theoretically, we will formulate and solve our problem with a convex optimization approach. Based on the duality technique, the scheme can be converted into a dual problem so that we can economically interpret the Lagrangian multiplier as "a unit bandwidth price" that a router uses to "sell" its link bandwidth. Then, we employ the resource management (RM) theory from economics [35] to interpret such a "price" under various traffic conditions, just like plane ticket prices fluctuate according to travel seasons. To rectify the unpractical and static assumption of link bandwidth in former works, we would like to include the time-varying feature of networks to allow our controller to take on more general applications. To this regard, we will incorporate the queue size $q(t)$ into our controller model as a remedial measure since queue size is always able to reflect traffic or bandwidth dynamics upon congestion. In the meantime, we hope such a remedial measure can also encompass an ability to overcome the big queuing delay and potential packet loss problems of existing algorithms.

We shall verify the effectiveness of our fully explicit traffic controller using the OPNET Modeler [33], which is capable of providing configurations and implementations close to real-world internet practices.

The contributions of our work lie in: identifying a new class of NUM-based algorithms, called fully explicit congestion control protocol, for the proper formulation of an optimized traffic congestion controller; extending the NUM approach to formulate a distributed OFEX congestion controller, which exercises link-wise proportional fairness to achieve network-wise max-min fairness, thus overcoming the drawback of existing relatively explicit congestion controllers "biasing" multi-bottlenecked users; and proposing the use of $q(t)$ as a remedial measure against time-varying link capacities, thus making the OFEX controller applicable in general networks. Furthermore, with the remedial measure, the queue size can be kept at a low level upon congestion, thus improving queuing delay performance and reducing packet-loss probability. To the best of our knowledge, no other NUM-based controllers consider queue size in their design.

II. Network Operation, Modeling, and Assumptions

1. Network Layout and Operation Principle

In a heterogeneous network topology, hosts are attached to access routers which cooperate with core routers to enable end-to-end communications. Using a graph model description, we represent such a network by $G = (S, L)$, where $S = \{s_1, s_2, s_3 \ldots\}$ represents the set of hosts and $L = \{l_1, l_2, l_3 \ldots\}$ represents the set of links in all routers of a network. For simplicity later on, we may omit the subscript of an element in the set S or L to use it to denote any element in the set. For example, we may use l to represent any link in the set L.

Congestion occurs when many flows traverse and overload a link, thus causing a bottleneck effect. Along an end-to-end data path, one or more links may be bottlenecks. Therefore, we need to implement some control schemes in routers to prevent

1) Social welfare originally means the provision of a minimal level of wellbeing and social support for all citizens from government or organizations [34].

internet congestion. Below is the general operation principle of our OFEX controller, in which sources adjust their sending rates according to the most congested link on their flow path.

To support the new algorithm to be given later, two new fields are needed in a packet header. One field is the flow weight ψ_s that each source $s \in S_l(t)$ uses to embed a "price" they are willing to pay for the network. The other field is Req. rate in the packet header, and this is used to record the permissible sending rate of a router. Each router along the data path will use the new algorithm to compute a permissible source transmission rate with a link-wise proportional fairness and compare it with the rate already recorded in the Req. rate field. If the former is smaller than the latter, the Req. rate field in the packet header will be updated; otherwise it remains unchanged. After the packet arrives at its destination, the value of the Req. rate field reflects the permissible rate from the most congested router along the path. The receiver then sends this value (fully explicit congestion signal) back to the source via the acknowledgement (ACK) packet, and the source then updates its sending rate accordingly. In this way, our scheme actually exercises network-wise max-min fairness. One can see that the local optimization in each link, to support the fully explicit signaling and the distributed nature of our OFEX controller, are what distinguishes our work from existing works—that is, the relatively explicit controllers, where they model the problem with an integrated optimization for all users and links and exercise the network-wise proportional fairness.

2. Link Optimization Model

Of the several outgoing links possible at a router, we focus on all traffic queuing on one outgoing link. A link $l, \forall l \in L$, in a router has a bandwidth capacity c_l with a set of passing flows originated from the set of sources $S_l(t)$, $S_l(t) \subset S$. Each source $s \in S_l(t)$ carries a weight ψ_s to link l so that it can obtain a proportion of the bandwidth with which to transmit data at a rate of x_s^l (that is, Req. rate). We define "revenue" as the total income the resource owner may collect. The bandwidth sharing algorithm in link l can thus be formulated as an optimization problem, called **PP** below

PP:

$$\max \quad \sum_{s \in S_l(t)} \psi_s U(x_s^l), \tag{1.1}$$

$$\text{subject to} \sum_{s \in S_l(t)} x_s^l \le c_l - \gamma q(t), \quad \forall l \in L, \tag{1.2}$$

$$\text{where} \quad x_s^l > 0, \quad s \in S_l(t). \tag{1.3}$$

Equation (1.1) is the objective function of link l to obtain the maximum "revenue" from forwarding traffic for all passing flows. The function $U(x_s^l)$ is a social welfare function[2] and is increasing strictly concave and twice continuously differentiable over x_s^l. The weight ψ_s can be regarded as a "payment" of source s. The bandwidth is allocated by link l according to the "payment" (from lowest to highest) from the sources; just like the ranking of social states. The parameter c_l is the nominal bandwidth (or say, face value of the bandwidth) of an outgoing link, which is predefined in a router. The actual link bandwidth, however, can be more than or less than c_l [36].

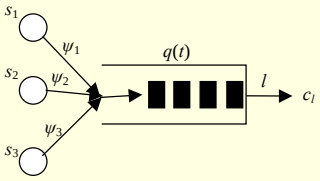


Fig. 1. Example router model.

The variable $q(t)$ is the instantaneous queue size, which can be accurately measured and monitored. The constant γ ($\gamma > 0$) is a threshold parameter for $q(t)$. Equation (1.2) is a constraint stipulating that the total incoming data rate $\sum_{s \in S_l(t)} x_s^l$ to link l, $\forall l \in L$, cannot exceed the link bandwidth c_l when the instantaneous queue is emptied.

Finally, equation and constraint (1.3) stipulates that source data rates are always positive, which means that once a flow is admitted to use a link l it is always allotted a portion of bandwidth. The total number of flows in link l is represented by $|S_l(t)|$. We define $\boldsymbol{\psi} = [\psi_1, \psi_2, \cdots, \psi_{|S_l(t)|}]$ as the weight vector and define $\mathbf{x_l} = [x_1^l, x_2^l, \cdots, x_s^l, \cdots, x_{|S_l(t)|}^l]$ as the data-rate vector of passing flows in link l, $\forall l \in L$. We let X_l be the domain of x_l, that is, the region where x satisfies (1.2) and (1.3).

With such modeling and formulation we hope the readers can better understand the contributions of our work mentioned in section I.

Figure 1 depicts an example to illustrate our example router model in which link l has three incoming flows and the total link capacity is c_l. The controller formulated in (1.1)–(1.3) can assign optimal sending rates to each flow according to the weight ψ_i, $i = 1, 2, 3$ (see details in section III).

III. OFEX Controller Design

We would like to choose $U(x_s^l) = \log(x_s^l)$ as the social welfare function for link l because it will give a convex

2) In economics, a social welfare function is a real-valued function that ranks conceivable social states (alternative complete descriptions of the society) from lowest to highest [37]. We borrow this terminology to draw the analogy between the ranking of social states and the way the sending rate of end users in the OFEX controller is determined.

objective function and is intimately associated with the concept of proportional fairness [38]. However, our modeling approach mandates that the proportional fairness in the OFEX controller has to be link-wise instead of network-wise as in the relatively explicit protocols. This is crucial for the OFEX controller to feed back the congestion signal from the most congested link as a fully explicit signal.

PP is a primal problem [39]. We are going to solve it with its duality so that we can obtain and interpret the Lagrange multiplier as a unit bandwidth price. Then, we apply one of the Karuch-Kuhn-Tucher (KKT) optimality conditions[3] to obtain the optimal data rate for all passing flows.

To convert **PP** to a Lagrange dual problem, we need to apply Lagrangian relaxation by associating a Lagrange multiplier $\lambda_l \geq 0$, $\forall l \in L$, to constraint (1.2). Then, we use it to augment the objective function. The Lagrangian of **PP** is

$$\eta(\lambda_l, x) = \sum_{s \in S_l(t)} \psi_s \log(x_s^l) - \lambda_l \Big(\sum_{s \in S_l(t)} x_s^l - c_l + \gamma q(t) \Big). \quad (2)$$

Note that the total "revenue" of link l now includes two parts. In addition to the revenue $\sum_{s \in S_l(t)} \psi_s \log(x_s^l)$ inherited from the objective function, it now has a congestion cost $\lambda_l \Big(\sum_{s \in S_l(t)} x_s^l - c_l + \gamma q(t) \Big)$. If we assume λ_l is the congestion cost per unit bandwidth, then when $\sum_{s \in S_l(t)} x_s^l > c_l$—that is, congestion is occurring because the aggregate incoming rate is greater than the link capacity—link l has to pay an additional cost to deal with the congestion, for example, using its queue buffer (that is, $q(t) > 0$) to temporarily accommodate for redundant data. Therefore, to avoid congestion, maintaining the congestion cost $\lambda_l \Big(\sum_{s \in S_l(t)} x_s^l - c_l \Big)$ to be less than or equal to zero is ideal. We shall visit this issue later on in "**Comment 2**".

Rearranging (2), we have

$$\begin{aligned} \eta(\lambda_l, x_l) &= \sum_{s \in S_l(t)} \psi_s \log(x_s^l) - \lambda_l \sum_{s \in S_l(t)} x_s^l + \lambda_l (c_l - \gamma q(t)) \\ &= \sum_{s \in S_l(t)} (\psi_s \log(x_s^l) - \lambda_l x_s^l) + \lambda_l (c_l - \gamma q(t)). \end{aligned} \quad (3)$$

Hence, the Lagrangian dual problem is

DP:

$$\min \ \Omega(\lambda_l), \quad (4.1)$$

$$\lambda_l \geq 0, \ \forall l \in L, \quad (4.2)$$

where $\Omega(\lambda_l)$ is the dual function given by

$$\Omega(\lambda_l) = \max_{x_s^l} \Big[\sum_{s \in S_l(t)} (\psi_s \log(x_s^l) - \lambda_l x_s^l) \Big] + \lambda_l (c_l - \gamma q(t)). \quad (5)$$

3) KKT optimality conditions [P244, 38] here refer to the series of constraints attached to the primal and dual problems of a convex optimization problem.

Lemma 1 and Comment 1 below will justify the legitimacy of our transformation from problem **PP** to **DP**.

Lemma 1: Strong duality holds between the primal problem **PP** in (1.1)–(1.3) and the Lagrangian dual problem **DP** in (4.1)–(4.2), that is, they have zero duality gap.

Proof: For a convex optimization problem, if Slater's condition [39] holds, the primal and the dual problem would have a zero duality gap. To see so, we need to check constraints (1.2) and (1.3) to see whether they meet Slater's condition. Since the constraint (1.3) is of the form $y = ax + b$, it is an affine function; therefore, it automatically meets Slater's condition (more precisely, (1.3) is the refined Slater condition). On the other hand, the constraint (1.2) is not affine. However, if there exists x_s^l so that (1.2) holds under $\sum_{s \in S_l(t)} x_s < c_l - \gamma q(t)$, then Slater's condition can also be met. By checking (1.2), one sees that there indeed exists a strictly feasible point, for example, $x_s^l = 0.5(c_l - \gamma q(t))/|S_l(t)|, \ s \in S_l(t), \ \text{where } c_l > 0$. Therefore, Slater's condition holds and the proof follows. ■

Comment 1 [39]: The establishment of lemma 1 enables us to solve the problem **PP** (1.1)–(1.3) by solving the problem **DP** (4.1)–(4.2), which is brought about by Lagrangian relaxation.

Theorem 1: The problem **DP** in (4.1)–(4.2) can be solved by the gradient descent method guided by

$$\lambda_l^{(k+1)} = [\lambda_l^{(k)} + \delta \cdot \Big(\sum_{s \in S_l(t)} x_s^l - c_l + \gamma q(t) \Big)]^+, \quad (6)$$

where $[a]^+ = \max(0, a)$, and δ is a constant step size with $\delta > 0$.

***Proof*:** Standard gradient descent method [39] allows us to write

$$\lambda_l^{(k+1)} = \lambda_l^{(k)} - \delta \cdot \Omega'(\lambda_l). \quad (7)$$

From (6), we can obtain

$$\Omega'(\lambda_l) = \frac{d\Omega(\lambda_l)}{d\lambda_l} = -\sum_{s \in S_l(t)} x_s^l + (c_l - \gamma q(t)). \quad (8)$$

Therefore

$$\lambda_l^{(k+1)} = \lambda_l^{(k)} + \delta \cdot \Big(\sum_{s \in S_l(t)} x_s^l - c_l + \gamma q(t) \Big). \quad (9)$$

Consider (4.2), and use $[a]^+$ to force λ_l in (9) to take on only non-negative values, and (6) follows. ■

Comment 2: Equation (6) allows us to interpret how the link bandwidth can be managed from an economical perspective using RM.

When congestion is taking place, that is, when $\sum_{s \in S_l(t)} x_s > c_l$ and $q(t) > 0$, λ_l increases in each iteration because $\sum_{s \in S_l(t)} x_s^l - c_l + \gamma q(t)$ is positive. Otherwise (that is, when

http://dx.doi.org/10.4218/etrij.14.0113.0036

$\sum_{s\in S_l(t)} x_s < c_l$ and $q(t) = 0$), λ_l decreases. Economically, one may interpret λ_l as "a unit link bandwidth price." Depending on the traffic loads, a link l, $\forall l \in L$, would adjust its λ_l; just as the way plane fares are adjusted by each airline company according to high or low travelling seasons. The parameter λ_l stays unchanged when $\sum_{s\in S_l(t)} x_s^l = c_l$ (Note: a very small $q(t)$ is allowed in this case, as seen in our simulation later), which means λ_l has converged to its optimum λ_l^*. In such a case, we say that the traffic demand (that is, the aggregate incoming traffic) and the bandwidth c_l of link l, $\forall l \in L$, have struck a balance.

Upon a sudden shrinkage of c_l when the link is already fully utilized, $q(t)$ must build up due to the decreased link output capability. Equation (6) stipulates that λ_l must increase to reflect the shortage of link bandwidth so that the sources can adjust their sending rate accordingly (see theorem 2 below). Note that the constant c_l in the controller remains the same all the time. The physical bandwidth shrinkage can be caused by interference, a contention, or the joining-in of some uncontrolled traffics, such as user datagram protocol (UDP) flows. In contrast, recall that the relatively explicit controllers were designed without considering $q(t)$ in their models; thus, they have no ability to detect the bandwidth dynamics which subsequently degrade their application.

Theorem 2: The optimal data rate that each passing flow is allocated by link l, $\forall l \in L$, is simply

$$x_s^{l*} = \frac{\psi_s}{\lambda_l^*}. \tag{10}$$

***Proof*:** For optimization, we set the gradient $\partial \eta(\lambda_l, x_l) / \partial x_s^l = 0$ from (4)

$$\frac{\partial \eta(\lambda_l, x_l)}{\partial x_s^l} = \frac{\psi_s}{x_s^{l*}} - \lambda_l^* = 0 \rightarrow x_s^{l*} = \frac{\psi_s}{\lambda_l^*}. \tag{11}$$

Thus, the unique optimal data rate x_s^{l*} in (11) follows when λ_l converges to λ_l^* with (6). ■

Due to space limit, the convergence and time-delayed stability analyses are not shown here.

IV. Performance Evaluation

We shall evaluate the performance and capability of our OFEX controller through a series of experiments.

1. Simulated Networks

We use OPNET modeler [33] to verify the effectiveness of our new control scheme. With reference to the simulated network topologies in Fig. 2, we shall first study the single-bottleneck network, which is a special case of multi-bottleneck networks and which has been the subject of many analyses, for example, [28], [29], and [40]–[42]. Therefore, we shall use it as a reference and for comparison with other relatively explicit controllers later.

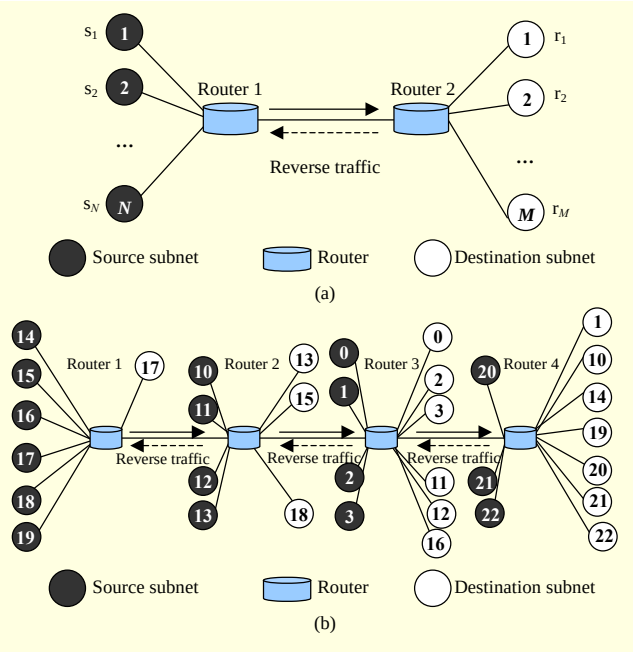


Fig. 2. Network simulation topology: (a) single-bottleneck network and (b) multi-bottleneck network.

A. Single-Bottleneck Network

Figure 2(a) shows a single-bottleneck network with multiple controlled source subnets (or say, groups) s_i (i = 1, 2, … , N) and destination subnets r_j (j = 1, 2, … , M). The source or destination nodes with the same number are the subnet pairs, which form the source-receiver data flows in the network. The receivers in the destination subnets produce the reverse traffic by piggybacking ACK information to the sources. To investigate the controller behavior in the most congested router we choose Router 1 as the only bottleneck in Fig. 2(a), whereas Router 2 is configured to have sufficiently high service rate so that congestion will never happen there.

We shall use $M = N = 11$ for subnets that can run various internet applications such as long-lived ftp, short-lived http, or unresponsive UDP-like flows (also called uncontrolled ftp [41]). At any instance, some sources are active and send traffic to their corresponding destinations. Since the link bandwidth we would like to simulate has a magnitude of Gigabits per second to produce congestion, we use 20 flows in each subnet to generate enough traffic.

Table 1. Source characteristics in single-bottleneck network.

Subnet ID	Source ID	Flow No.	RTPD (ms)
1	1 – 20	ftp 1 – 20	80
2	21 – 40	ftp 21 – 40	120
3	41 – 60	ftp 41 – 60	160
4	61 – 80	ftp 61 – 80	200
5	81 – 100	ftp 81 – 100	240
6	101 – 120	http 1 – 20	80
7	121 – 140	http 21 – 40	120
8	141 – 160	http 41 – 60	160
9	161 – 180	http 61 – 80	200
10	181 – 200	http 81 – 100	240
11	201	UDP 1	160

The flow configuration is summarized in Table 1. There are five ftp subnets and five http subnets. There is also one UDP flow that will require some dedicated but constant bandwidth when running. The round-trip propagation delay (RTPD) includes the forward-path propagation delay and the feedback propagation delay but does not include the queuing delay, which depends on the instantaneous queue size in the experiments. As seen, the RTPD takes on different values in different groups.

The buffer capacity B in Router 1 and Router 2 is approximately equal to the bandwidth-delay product (BDP). So, B may be different in the different experiments below. All ftp packets have the same size of 1,024 bytes [41], while the http packet size is uniformly distributed in [800, 1,300] bytes [43].

To show the robustness of the OFEX controller, our experiments mainly focus on the testing of long-lived ftp sources, unless otherwise stated. The sporadic short-lived http-flows just act as the disturbance of the ftp traffic, and their transfer size follows real web-traffic scenarios by using a Pareto distribution [44] with an average of 30 packets [28]. The arrivals of http flows follow a think time [44], which is uniformly distributed in [0.1, 30] seconds.

B. Multi-bottleneck Network

Figure 2(b) illustrates a multi-bottleneck network with four routers. As with Fig. 2(a), the numbers marked on the nodes designate the source or destination subnets attached to each router. For example, the ftp flows in source subnet 16 will go through Routers 1, 2, and 3 before they reach their corresponding receivers in destination subnet 16. Note that we do not number the subnets consecutively (that is, all the way from 1 to 22) in Fig. 2(b) to allow for future network extension.

Table 2. Source characteristics in multi-bottleneck network.

Subnet ID	Source ID	Flow No.	RTPD (ms)
0	1 – 20	ftp 1 – 20	52
1	21 – 40	ftp 21 – 40	80
2	41 – 60	ftp 41 – 60	96
3	61 – 80	UDP 1 – 20	75
10	1 – 20	ftp 1 – 20	180
11	21 – 40	ftp 21 – 40	160
12	41 – 60	http 1 – 20	150
13	61 – 80	UDP 1 – 20	200
14	1 – 20	http 1 – 20	100
15	21 – 40	ftp 1 – 20	50
16	41 – 60	ftp 21 – 40	170
17	61 – 80	ftp 41 – 60	80
18	81 – 100	ftp 61 – 80	110
19	101 – 120	ftp 81 – 100	290
20	1 – 20	ftp 1 – 20	180
21	21 – 40	ftp 21 – 40	166
22	41 – 60	UDP 1 – 20	95

The link bandwidth from Router 1 to Router 4 is 1.52 Gbps, 3.02 Gbps, 2.14 Gbps, and 1.86 Gbps, respectively. As seen, we did not set the link bandwidth from Router 1 to Router 4 in a monotonically increasing or decreasing manner, thus making it much less artificial. With such a bandwidth configuration, some routers may become a bottleneck. For example, Router 3 may encounter congestion because upstream Router 2 has a bigger link bandwidth, allowing more traffic to pass through; thus, making Router 3 congested.

Table 2 tabulates the source characteristics in the multi-bottleneck network. Other settings, such as packet size, have the same values as discussed for the single-bottleneck network.

Since the behavior and performance of the sources within each subnet are quite similar, we shall show the results of one source from each group in the following experiments for reasons of brevity.

2. Robustness against Bandwidth Dynamics

As discussed before, the existing NUM-based relatively explicit controllers have no way to tackle the link bandwidth dynamics (for example, as inside a contention-based multi-access network). The OFEX controller introduces the instantaneous queue size $q(t)$ in the system model as a remedial

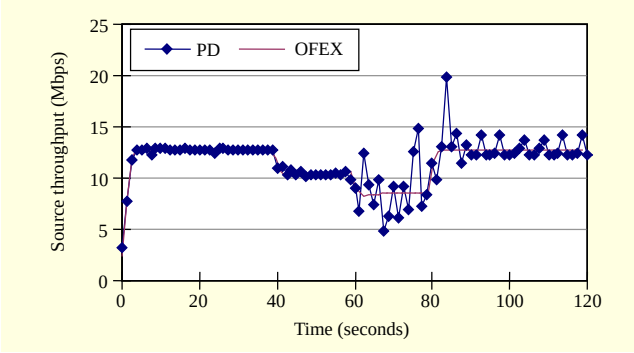


Fig. 3. Source dynamics.

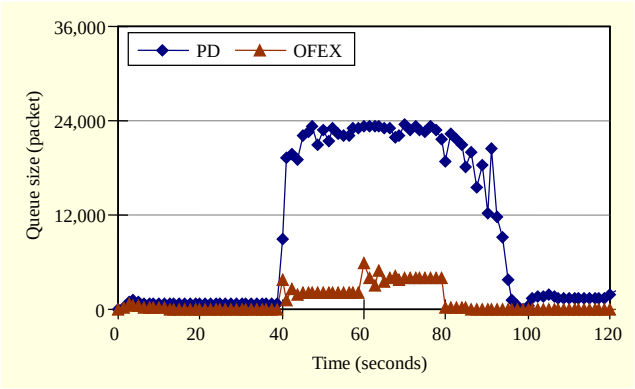


Fig. 4. Instantaneous queue size $q(t)$.

measure to detect such occurrences. The purpose of this experiment is to show that the OFEX controller has a better performance than the relatively explicit controllers in the presence of dynamic link bandwidth.

We set up a 1 Gbps-link in Router 1 of Fig. 2(a). To produce the bandwidth dynamics, the link capacity is reduced to 880 Mbps at t = 40 s and 675 Mbps at t = 60 s. This is one common scenario in contention-based networks, for example, when some UDP traffic swarms in. Finally, the bandwidth will recover to 1 Gbps at t = 80 s.

The parameters of the OFEX controller are $\delta = 0.05$, $\lambda_l^{(0)} = 20$, $\gamma = 1$, and B = 24,000 packets. The weights ψ_s in the flows from ftp groups 1 to 5 are 12,207.03, 4,882.81, 9,765.62, 7,324.22, and 2,441.4, respectively. The relatively explicit controller uses the same parameters and configuration as with the OFEX controller except for the parameter γ because it does not have such a parameter.

Figure 3 demonstrates the source-sending-rate dynamics of the two controllers upon changes of link bandwidth. For the source behavior of the relatively explicit controller, after the transient in the first three seconds, its source sending rate stabilizes at 13.1 Mbps. When the bandwidth decreases at t = 40 s, its sending rate decreases and stabilizes at 10.2 Mbps. However, the second bandwidth reduction at t = 60 s makes the relatively explicit controller render unstable behavior in that its sending rate starts to oscillate. When the link bandwidth goes back to the original value of 1 Gbps at t = 80 s, its sending rate recovers but still has some small fluctuations.

Note that the controller can stabilize in the first two occasions but not thereafter. The reason could be due to the queue buildup in the relatively explicit controller. As seen in Fig. 4 below, the relatively explicit controller has no queue-control measure, and its queue size builds up quickly after the first link bandwidth reduction at t = 40 s. After the second bandwidth reduction at t = 60 s, its queue size has become unstable; so has the queuing delay. As known, the queuing delay is part of the round-trip time (RTT), and an unstable RTT could make source sending rates oscillate.

In contrast, the OFEX controller as shown in Fig. 3 is stable at all stages irrespective of bandwidth dynamics. That is, when the bandwidth shrinks at t = 40 s and 60 s, or when the bandwidth recovers at t = 80 s, the source sending rate hardly fluctuates. This verifies that the introduction of $q(t)$ in the OFEX controller model greatly improves the system's stability performance against link bandwidth dynamics.

Figure 4 shows the $q(t)$ for the two controllers. The relatively explicit controller has a queue length of 1,300 packets after the router is started. When the bandwidth undergoes the first reduction at t = 40 s, its queue size sharply increases to the full capacity of the buffer, (that is, 24,000 packets) and there it remains with some oscillations. The packet loss becomes unavoidable thereafter. The buffer is kept overflowed through the second bandwidth reduction from t = 60 s. Only after the bandwidth goes back to the original value at t = 80 s, does the queue size of the relatively explicit controller begin to decrease and eventually settle at 2,300 packets.

In contrast, the queue size of the OFEX controller depicted in Fig. 4 shows better performance than that of the relatively explicit controller. As seen, the OFEX controller renders a near-zero queue size (around 30 packets) after the router is started in the first eight seconds. Upon the first bandwidth shrink at t = 40 s, the queue size of the OFEX controller rises to a level of 2,000 packets. The second bandwidth shrink at t = 60 s, elevates the queue size to around 4,000 packets, which is only one-sixth of the buffer size, so there is no packet loss at all. When the bandwidth recovers to its original capacity, the queue size of the OFEX controller goes back to its smooth, near-zero level accordingly.

In summary, along with other experiments whose results are not shown here due to space limitation, the OFEX controller shows much better performance in its queue size due to the introduction of the instantaneous queue size $q(t)$ in its model, and significantly improves the queuing-delay performance and system stability in the contention-based networks.

3. Performance Improvement of Multi-bottleneck Users

The purpose of this experiment is to verify the superiority of the OFEX controller over the relatively explicit controller in improving the throughput and convergence speed of the multi-bottlenecked users. The comparison of the $q(t)$ of the two controllers is omitted here due to space limitations.

The comparison of the two controllers is conducted with the multi-bottleneck network settings described in section IV, 1*B*, where $\delta = 0.05$ and $\lambda_l^{(0)} = 20$ in all the routers. We also use the same source weight vector ψ_s in the two controllers. For example, sources in subnet 0 pay a "price" of 7,324.20 in both controllers, and sources in subnet 20 pay a "price" of 2,441.40 in both controllers. For brevity, we skip enumerating the "prices" paid by other sources. In addition, upon light traffic the routers request a minimum payment of 0.001, which is set close to zero. Two of the multi-bottlenecked flows will be used to compare the performances of the two controllers. They are called Flow A and Flow B and represent flows that pass through three bottlenecks and four bottlenecks, respectively.

Figure 5 demonstrates the instantaneous sending rate of Flow A from subnet 10. The flows in subnet 10 pass through three routers; that is, Routers 2, 3, and 4. As seen, the convergence time of the OFEX controller is nine seconds and its steady state sending rate is 24.13 Mbps. In comparison, the convergence time and the steady state sending rate of the relatively explicit controller are fourteen seconds and 21.87 Mbps, respectively. Hence, the OFEX controller shows a faster convergence speed of five seconds and higher throughput of 2.26 Mbps, which means an improvement of 35.71% and 10.33%, respectively over the relatively explicit controller.

Figure 6 shows the instantaneous source sending rate of Flow B from subnet 19, where the flows have 4 bottlenecks; that is, from Routers 1 to 4. As with Fig. 5, the same trend can be observed. The sources with the OFEX controller have higher sending rates and shorter convergence times than those with the relatively explicit controller. In particular, as shown in Fig. 6, the OFEX controller converges 1.7 s faster (that is, an improvement of 35.42%) than the relatively explicit controller in that the former spends 3.1 s and the latter spends 4.8 s converging to the steady state.

Upon comparing the Flow B throughput between the two controllers in Fig. 6, the OFEX controller shows much better performance than the three-bottlenecked flows observed in Fig. 5. Specifically, Fig. 6 shows that the sending rate of the flows with the OFEX controller is 3.17 Mbps, and the sending rate with the relatively explicit controller is about 2.01 Mbps. The difference between them is 1.16 Mbps, which means the OFEX controller achieves a throughput improvement of 57.71%.

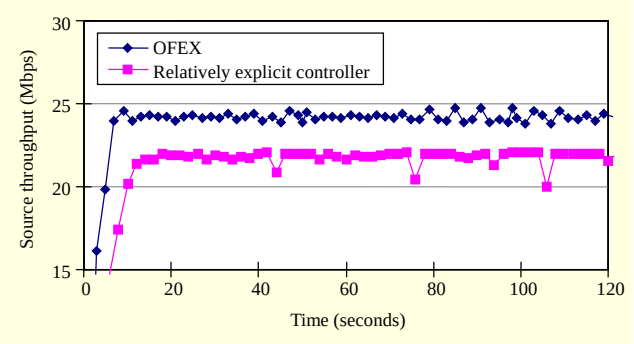


Fig. 5. Source sending rate in subnet 10.

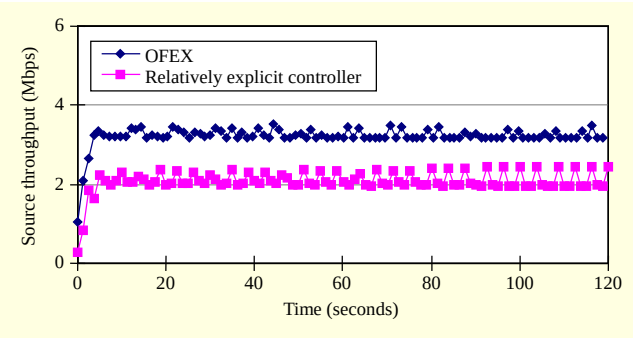


Fig. 6. Source sending rate in subnet 19.

Summarizing the comparisons in Figs. 5 and 6, the OFEX controller conspicuously improves the source throughput and convergence speed of the multi-bottlenecked users. In particular, the more bottlenecks a flow passes, the better throughput performance the OFEX controller shows; for example, Flow B illustrated above.

V. Conclusion

We have formulated and investigated an optimal and fully-explicit congestion controller (called the OFEX controller) that has a natural economic interpretation. In contrast to the existing relatively explicit NUM–based dual algorithms, the OFEX controller not only provides optimal solutions but also fully-explicit congestion signals to sources. The new scheme overcomes the drawback of the relatively explicit controllers in biasing against multi-bottlenecked users. Furthermore, the time-varying feature of the OFEX controller significantly improves the queuing performance and system stability against link bandwidth dynamics. The OPNET simulations have verified the effectiveness and superiority of the OFEX controller.

References

[1] F.P. Kelly, A. Maulloo, and D. Tan, "Rate Control for

Communication Networks: Shadow Prices, Proportional Fairness and Stability," *J. Operational Res. Soc*., vol. 49, no. 3, Mar. 1998, pp. 237–252.

[2] S.H. Low and D.E. Lapsley, "Optimization Flow Control I: Basic Algorithm and Convergence," *IEEE/ACM Trans. Netw*., vol. 7, no. 6, Dec. 1999, pp. 861–874.

[3] Y. Qiu and P. Marbach, "Bandwidth Allocation in Ad Hoc Networks: a Price-Based Approach," *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar. 30 – Apr. 3, 2003, pp. 1–6.

[4] T. Harks, "Utility Proportional Fair Bandwidth Allocation: An Optimization Oriented Approach," *Proc. QoS Multiservice IP Netw*., vol. 3375, Feb. 2–4, 2005, pp. 61–74.

[5] G. Zhang, Y. Wu, and Y. Liu, "Stability and Sensitivity for Congestion Control in Wireless Networks with Time Varying Link Capacities," *Proc. IEEE Int. Conf. Netw. Protocols*, Boston, MA, USA, Nov. 6–9, 2005, pp. 401–412.

[6] X. Lin and N. Shroff, "Utility Maximization for Communication Networks with Multipath Routing," *IEEE Trans. Autom. Contr*., vol. 51, no. 5, May 2006, pp. 766–781.

[7] F.P. Kelly and G. Raina, "Explicit Congestion Control: Charging, Fairness, and Admission Management," *Next-Generation Internet Architectures and Protocols,* Cambridge University Press, 2010.

[8] D. Pradas and M. V-Castro, "NUM-Based Fair Rate-Delay Balancing for Layered Video Multicasting over Adaptive Satellite Networks," *IEEE J. Sel. Areas Commun*., vol. 29, no. 5, May 2011, pp. 969–978.

[9] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhauser, 2004.

[10] D. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE J. Sel. Areas Commun*., vol. 24, no. 8, Aug. 2006, pp. 1439–1451.

[11] F.P. Kelly, "Fairness and Stability of End-to-End Congestion Control," *European J. Contr*., vol. 9, no. 2, 2003, pp. 159–176.

[12] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Netw*., vol. 1, no. 4, Aug. 1993, pp. 397–413.

[13] M. Bouhtou, M. Diallo, and L. Wynter, "Capacitated Network Revenue Management through Shadow Pricing," *Proc. Netw. Group Commun*., Sept. 16–19, 2003, pp. 342–351.

[14] J. Mo and J. Walrand, "Fair End-to-End Window-Based Congestion Control," *IEEE/ACM Trans. Netw*., vol. 8, no. 5, Oct. 2000, pp. 556–567.

[15] R. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE /ACM Trans. Netw*., vol. 10, no. 2, Apr. 2002, pp. 272–286.

[16] J. Lavaei, J.C. Doyle, and S. Low, "Utility Functionals Associated with Available Congestion Control Algorithms," *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 14–19, 2010, pp. 1–9.

[17] S. Athuraliya and S. Low, "Optimization Flow Control with Newton-like Algorithm," *Telecommun. Syst*., vol. 15, no. 3–4, Dec. 2000, pp. 345–358.

[18] K. Kar, S. Sarkar, and L. Tassiulas, "A Simple Rate Control Algorithm for Maximizing Total User Utility," *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, vol. 1, Apr. 22–26, 2001, pp. 133–141.

[19] K. Ma, R. Mazumdar, and J. Luo, "On the Performance of Primal/Dual Schemes for Congestion Control in Networks with Dynamic Flows," *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 15–17, 2008, pp. 326–330.

[20] R. Li et al., "A Unified Approach to Optimizing Performance in Network Serving Heterogeneous Flows," *IEEE/ACM Trans. Netw*., vol. PP, no. 99, 2010, pp. 1–14.

[21] S. Low et al., "Dynamics of TCP/AQM and a Scalable Control," *Proc. INFOCOM*, New York, NY, USA, vol. 1, June 23–27, 2002, pp. 239–248.

[22] S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-Way Traffic," *ACM Comput. Commmun*., *Rev*., vol. 21, no. 5, Oct. 1991, pp. 30–47.

[23] Y. Zhang and T.R. Henderson, "An Implementation and Eperimental Study of the Explicit Control Protocol (XCP)," *Proc. INFOCOM*, Maiami, FL, USA, vol. 2, Mar. 13–17, 2005, pp. 1037–1048.

[24] F. Abrantes and M. Ricardo, "XCP for Shared-Access Multi-rate Media," *ACM SIGCOMM Comput. Commun*., *Rev*., vol. 36, no. 11, July 2006, pp. 27–38.

[25] M. Chiang, "Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control," *IEEE J. Sel. Areas Commun*., vol. 23, no. 1, Jan. 2005, pp. 104–116.

[26] J. Papandriopoulos, S. Dey, and J. Evans, "Optimal and Distributed Protocols for Cross-Layer Design of Physical and Transport Layers in MANETs," *IEEE/ACM Trans. Netw*., vol. 16, no. 6, Dec. 2008, pp. 1392–1405.

[27] M. Belleschi et al., "Fast Power Control for Cross-Layer Optimal Resource Allocation in DS-CDMA Wireless Networks," *Proc. IEEE Int. Conf. Commun*., Dresden, Germany, June 14–18, 2009, pp. 1–6.

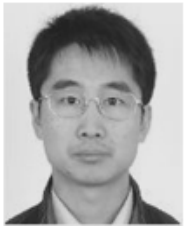
[28] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," *Proc. SIGCOMM*, Pittsburgh, PA, USA, Aug. 19–23, 2002, pp. 89–102.

[29] N. Dukkipati, N. McKeown, and A.G. Fraser, "RCP-AC Congestion Control to Make Flows Complete Quickly in Any Environment," *Proc. INFOCOM*, Barcelona, Spain, Apr. 23–29, 2006, pp. 1–5.

[30] J. Liu and O. Yang, "Convergence, Stability and Robustness Analysis of the OFEX Controller for High-Speed Networks," under preparation for journal submission. Accessed Nov. 2012. www.site.uottawa.ca/~jliu 115

[31] M. Andrews, L. Qian, and A. Stolyar, "Optimal Utility Based Multi-user Throughput Allocation Subject to Throughput

Constraints," *Proc. IEEE INFOCOM*, Maiami, FL, USA, vol. 4, Mar. 13–17, 2005, pp. 2415–2424.
[32] J. Chou and B. Lin, "Optimal Multi-path Routing and Bandwidth Allocation under Utility Max-Min Fairness," *Proc. Int. Workshop Quality Service*, Charleston, SC, USA, July 13–15, 2009, pp. 1–9.
[33] OPNET Modeler Manuals, *Opnet Technologies Inc*, 2012.
[34] Accessed Sept. 2012. http://en.wikipedia.org/wiki/Welfare
[35] I. Yeoman and U. McMahon-Beattie, *Revenue Management: a Practical Pricing Perspective*, Palgrave Macmillan, 2011.
[36] M. Welzl, "Traceable Congestion Control," *Proc. Int. Conf. Quality Future Internet Services Internet Charging QoS Technol*., Oct. 2002, pp. 273–282.
[37] Accessed Sept. 2012. http://en.wikipedia.org/wiki/Social_welfare_function
[38] F.P. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. Telecommun*., vol. 8, no. 1, 1997, pp. 33–37.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed., Cambridge, UK: Cambridge University Press, 2004.
[40] X. Guan et al., "Adaptive Fuzzy Sliding Mode Active Queue Management Algorithms," *Telecommun. Syst*., vol. 35, no. 1–2, June 2007.
[41] Y. Hong and O.W.W. Yang, "Design of Adaptive PI Rate Controller for Best-Effort Traffic in the Internet Based on Phase Margin," *IEEE Trans. Parallel Distrib. Syst*., vol. 18, no. 4, Apr. 2007, pp. 550–561.
[42] Y. Jing, Z. Chen, and G.M. Dimirovski, "Robust Fuzzy Observer-Based Control for TCP/AQM Network Systems with State Delay," *Proc. American Contr. Conf*., Baltimore, MD, USA, June 30 – July 2, 2010, pp. 1350–1355.
[43] Y. Zhang, D. Leonard, and D. Loguinov, "JetMax: Scalable Max-Min Congestion Control for High-Speed Heterogeneous Networks," *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 23–29, 2006, pp. 1–13.
[44] M.E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Trans. Netw*., vol. 5, no. 6, Dec. 1997, pp. 835–846.



Jungang Liu received his BS degree in automatic control and his MS degree in control theory and control engineering from Wuhan University of Technology, Wuhan, Hubei Province, China, in 1999 and 2002, respectively. He is currently pursuing his PhD in electrical and computer engineering at the University of Ottawa, Ontario, Canada. He was the recipient of the PhD Admission Scholarship of the University of Ottawa, Ontario, Canada and the 2011-2012 Ontario Graduate Scholarship, Ontario, Canada. He was also the recipient of the Teaching Assistant Excellence Award, University of Ottawa, Ontario, Canada, in 2013. His main research interests include internet traffic control; modeling and performance evaluation; automation; and control.

Oliver W.W. Yang received his PhD degree in electrical engineering from the University of Waterloo, Ontario, Canada. He is a professor in the school of electrical engineering and computer science, at the University of Ottawa, Ontario, Canada. He has worked for Northern Telecom Canada Ltd. as a consultant. He has served on the editorial board of IEEE Communication Magazine, and IEEE Communication Surveys & Tutorials, as well as serving as an associate director of the OCIECE (Ottawa-Carleton Institute of Electrical and Computer Engineering). His research interests are in the modeling, analysis, and performance evaluation of computer communication networks, their protocols, services, and interconnection architectures. The CCNR Lab, under his leadership, has been working on various projects in switch architecture, traffic control, traffic characterization, and other traffic engineering issues in both wireless and photonic networks, the results of which can be found in more than 400 technical papers. Dr. Yang is a senior member of the IEEE.