Multi-path Aggregate Flow Control for Real-time Traffic Engineering

Jung-hoon Yun
Division of Electrical Engineering
School of Electrical Engineering &
Computer Sicence KAIST,
Daejon 305-701, Republic of Korea
Email: jhyun@netsys.kaist.ac.kr

Anseok Lee
Division of Electrical Engineering
School of Electrical Engineering &
Computer Sicence KAIST,
Daejon 305-701, Republic of Korea
Email: anseok@netsys.kaist.ac.kr

Song Chong
Division of Electrical Engineering
School of Electrical Engineering &
Computer Sicence KAIST,
Daejon 305-701, Republic of Korea
Email: song@ee.kaist.ac.k

***Abstract*—We present an online distributed traffic engineering method for ISP networks with multi-path routing. The method is based on edge-to-edge aggregate flow control that balances load and makes the network congestion-free in real time, responding to actual traffic demands whether they are underload or overloaded. Moreover, it allows ISPs to apply various bandwidth-sharing policies to edge-to-edge flows, as desired. Our simulations confirm that the proposed method works as designed for TCP sources that have their own end-to-end congestion control mechanism and enhance the performance and the efficiency of the the network.**

***Index Terms*—Traffic engineering, aggregate flow control, path diversity.**

I. Introduction

Traffic engineering (TE) plays a critical role in determining the performance and reliability of a network. The challenging issue to achieve this goal is how to cope with dynamic and unpredictable changes in traffic demand. However, most of current Internet TEs rely on offline methods that use long-term average traffic demands [1]. Due to its offline nature, it might create inadequate traffic load distribution in the network if actual traffic demands differ from the long-term average traffic demands. Moreover, if the actual demands exceed the provisioned capacity, the following might occur: temporal overload(which can cause serious instability) and problems that pertain to QoS, such as longer delay, higher packet loss rate, reduced network throughput, and even router crashes. One approach that circumvents these intrinsic limitations of current TE is bandwidth over-provisioning, which has recently drawn much attention due to its simplicity. This approach upgrades bandwidth capacity when the maximum link utilization exceeds a particular threshold (about 40% utilization) [1]. However, it is obviously a cost-inefficient approach and opposed to the concept of TE.

In this paper, we present an alternative TE approach in which traffic engineering is carried out online by means of edge-to-edge multi-path aggregate flow control, responding to the actual demands of traffic. More specifically, the proposed scheme uses multiple paths to deliver aggregate traffic from an ingress to an egress router, splits traffic across multiple paths by exploiting path diversity for load balancing, and pushes out congestion if any to ingress edges by enforcing a certain bandwidth-sharing policy on edge-to-edge flows. The proposed scheme is hierarchical in that it solves the network-wide congestion control and bandwidth allocation problem on an aggregate and edge-to-edge basis, while individual flows that are multiplexed into the same edge-to-edge aggregate flow compete for and share their aggregate bandwidth by their own congestion control actions, such TCP traffic with AIMD (Additive Increase Multiplicative Decrease), at the network entrance. In addition, the proposed scheme provides a joint optimal solution that combines multi-path routing capability of the network with aggregate flow control optimally, for both underloaded and overloaded network conditions.

Several studies have been done in the area of aggregate flow control [1]–[3]. The scheme in [2] uses TCP trunking concept, according to which each edge-to-edge aggregate traffic is controlled by a single TCP that runs between ingress and egress edges. It is somewhat similar to ours in concept. However, it inherits the known drawbacks of TCP congestion control, such as persistent packet losses and throughput oscillation. Moreover, it would be extremely difficult to extend the scheme to the multi-path routing case unless multi-path TCP were to be developed. Recently, an edge-to-edge aggregate TE method called TeXCP [1] has been proposed for networks that use multi-path routing. In TeXCP, each load balancer at an ingress edge splits its edge-to-edge traffic across multiple routing paths in such a way that maximum link utilization in the network is minimized. In addition, TeXCP runs a closed-loop flow control for each path to avoid congestion and the authors claim that it yields path-wise max-min fair bandwidth allocation. In [3], an edge-to-edge aggregate TE method called MATE is proposed for MPLS networks with multi-path routing. MATE aims to minimize the sum of the link delays in the network. However, it has no flow control functionality to cope with overload. Our results are summarized as follows:

- The proposed edge-to-edge aggregate flow control scheme jointly optimizes multi-path routing and flow control such that the sum of the utilities of edge-to-edge aggregate flows is maximized whether the network is overloaded or underloaded. The distributed implementation of the scheme is also developed.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

- The proposed scheme has the advantage that it allows ISPs to apply various bandwidth-sharing policies to edge-to-edge flows as desired, including weighted proportional fairness, max-min fairness and throughput maximization, while fully utilizing the path diversity provided by multi-path routing.
- For the same traffic demands and even at half capacity, a network that uses the proposed scheme can support better loss and throughput performance than a network that does not use any TE schemes. This results in significant cost reduction of over-provisioning for the ISPs.
- The proposed scheme renders the network loss-free and provides stable throughput between edge pairs as the closed-loop control reaches steady state.

II. Problem Formulation

We consider an ISP network with a set L of links, and let c_l be the capacity of link l, $l \in L$. The network is shared by a set S of ingress-egress (IE) node pairs. Each IE pair s has a set P_s of disjoint routing paths and P_s, $s \in S$, are also disjoint sets. An IE pair s has an aggregate flow of rate r_s and routes x_{sp} amount of it on path $p \in P_s$. Let $x_s = (x_{sp}, p \in P_s)$ and $x = (x_{sp}, p \in P_s, s \in S)$. Associated with each IE pair s is a utility $U_s(r_s)$ as a function of r_s. Assume that $U_s(r_s)$ is an increasing and strictly concave function of r_s.

Our objective is to maximize the total utility $\sum_{s\in S} U_s(r_s)$ by optimally throttling each IE flow r_s at the entrance and splitting it across paths in P_s:

$$\max_x \quad \sum_{s\in S} U_s(r_s) \tag{1}$$

$$\text{subject to} \quad \sum_{s\in S}\sum_{l\in p, p\in P_s} x_{sp} \le c_l, \text{ for all } l \in L \tag{2}$$

$$\sum_{p\in P_s} x_{sp} = r_s, \text{ for all } s \in S \tag{3}$$

$$0 \le r_s \le D_s, \text{ for all } s \in S \tag{4}$$

$$x_{sp} \ge 0, \text{ for all } p \in P_s, s \in S. \tag{5}$$

The finite (respectively, infinite) value of D_s in Eq. (4) represents the finite (respectively, infinite) traffic demand of IE pair s so that the optimization problem readily models both underloaded and overloaded cases.

The choice of $U_s(r_s)$ determines the bandwidth sharing among IE pairs. Following [4], we use

$$U_s(r_s) = \begin{cases} w_s \log r_s, & \text{if } \alpha = 1 \\ \frac{w_s}{1-\alpha} r_s^{1-\alpha}, & \text{if } \alpha \ge 0,\ \alpha \ne 1. \end{cases} \tag{6}$$

where w_s is the positive weight associated with each IE pair s. In particular, $\alpha = 0$, $\alpha = 1$ and $\alpha \to \infty$ correspond to weighted throughput maximization, proportional fairness, and max-min fairness, respectively. See [4] for details.

III. Multi-path Aggregate Flow Control

A. Distributed Algorithm

The objective function in Eq. (1) is strictly concave in $r = (r_s, s \in S)$ but not in x. Hence, its dual function is nondifferentiable [5] and we cannot directly apply a dual method to our problem for the development of a distributed algorithm. In order to circumvent this difficulty, we use an augmented Lagrangian method that can handle nondifferentiable dual functions [5]. After converting the inequality constraints in Eqs. (2) and (4) to equality constraints using the additional quadratic variables y_l^2, $\forall l \in L$, and z_s^2, $\forall s \in S$, respectively, the objective function in Eq. (1) is augmented as follows:

$$\sum_{s\in S} U_s(r_s) - \frac{\kappa_1}{2}\sum_{l\in L}\left(\sum_{s\in S}\sum_{l\in p,p\in P_s} x_{sp} - c_l + y_l^2\right)^2 - \frac{\kappa_2}{2}\sum_{s\in S}\left(r_s - D_s + z_s^2\right)^2 \tag{7}$$

where κ_1 and κ_2 are positive penalty parameters. Let $f(x,y,z)$ denote the function in Eq. (7) where $y = (y_l, l \in L)$ and $z = (z_s, s \in S)$. Then, the augmented Lagrangian function is given by

$$L_A(x,y,z,\mu,\lambda) = f(x,y,z) - \sum_{s\in S}\lambda_s\left(r_s - D_s + z_s^2\right) - \sum_{l\in L}\mu_l\left(\sum_{s\in S}\sum_{l\in p,p\in P_s} x_{sp} - c_l + y_l^2\right) \tag{8}$$

where $\mu = (\mu_l, l \in L)$ and $\lambda = (\lambda_s, s \in S)$ are Lagrange multipliers. By the method of multipliers [5], we have the following successive maximization of the form

$$x(t+1) = \arg\max_{x \,\, x\ge 0,y,z} L_A(x,y,z,\mu(t),\lambda(t)) \tag{9}$$

followed by updates of the vectors $\mu(t)$ and $\lambda(t)$ according to

$$\mu_l(t+1) = \left[\mu_l(t) + \kappa_1\left(\sum_{s\in S}\sum_{l\in p,p\in P_s} x_{sp}(t) - c_l\right)\right]^+, \tag{10}$$

$$\lambda_s(t+1) = \left[\lambda_s(t) + \kappa_2\left(\sum_{p\in P_s} x_{sp}(t) - D_s\right)\right]^+. \tag{11}$$

The maximization in Eq. (9) at each time t is separable and can be solved by the gradient projection method as

$$x_{sp}(\tau+1) = \left[x_{sp}(\tau) + \gamma\left(U_s'\left(\sum_{p\in P_s} x_{sp}(\tau)\right) - \theta_{sp}(t)\right)\right]^+ \tag{12}$$

for given $\theta_{sp}(t) = \lambda_s(t) + \sum_{l\in p}\mu_l(t)$ where γ is a positive step size. $\mu_l(t)$ can be interpreted as the congestion price of link l and $\theta_{sp}(t)$ can be interpreted as the congestion price of path p if we view Eq. (4), modeling a finite amount of traffic demand D_s, as another constraint on link capacity. In addition, Eq. (12) reveals an interesting property that each IE pair s has identical congestion prices for all of its paths at optimal point $(r_s^*, \theta_{sp}^*, p \in P_s)$, i.e., $\theta_{sp}^* = U_s'(r_s^*)$, for all $p \in P_s$. We say that this is *equal cost load balancing*. The algorithm is fully distributed in that each link l computes Eq. (10) and each IE

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

pair s computes Eq. (11) and Eq. (12) for all $p \in P_s$ provided that each path p has round-trip control packets to keep track of $\sum_{l \in p} \mu_l(t)$.

B. Convergence

The convergence of the proposed distributed algorithm to a global optimum, which is not unique, can be readily proven for the case in which the algorithm is executed synchronously (see Chapter 4.2 in [5] and Chapter 3.4.4 in [6]). However, the convergence of its asynchronous version needs more work and we leave it for future study. Some variants of multi-path flow control can be found in the flow control literature [7] [8] but none of them have been proven mathematically to converge when executed asynchronously.

C. Path Selection and TCP Traffic Distribution

We assume that there is a pre-computed set of paths P_s for each IE pair s. A simple choice for P_s is disjoint K-shortest paths that connect the IE pair, where the path length is set to the number of hop counts. We use this choice in the simulation. Incoming traffic to an IE pair s can be distributed into the paths in P_s in a number of ways. One method is to distribute the traffic on a per-packet basis into the paths, responding to the available path rates x_{sp}, $p \in P_s$. However, the method suffers from potentially having to re-sequence a large number of packets at the receiving ends, which is especially undesirable for TCP applications [9]. The other method, which does not cause out-of-sequence packets, is to split aggregate flows among the paths on the basis of the granularity of a flow. For example, the ingress edge router constructs an aggregate flow by filtering the incoming traffics on a per-flow basis (e.g., based on <source IP address, source port, destination IP address, destination port, IP protocol> tuple) and distributes the aggregate flow to the paths on the basis of the filtered flows. We used this method in the simulation.

IV. Simulation Results

A. Throughput gain, fairness, and packet loss

Using NS-2, we simulated an ISP network scenario as shown in Fig. 1 where NSF topology is used with slight modification to create more number of disjoint paths for each IE pair. There are six IE pairs. Thirty-six long-lived TCP-NewReno flows are multiplexed into each IE pair at its ingress node. Each IE pair has a pre-computed set of disjoint paths. For example, the source group S4, which consists of 36 TCP sources that are destined for D4, enters the network at the ingress node E4 and departs the network at the egress node E7, and this IE pair has three disjoint paths between E4 and E7. The buffer sizes and link capacities are set equally to 140 packets and 24 Mbps, respectively. We compared four different cases for an overloaded network situation in which each TCP source always had some data to send. One case is not related to the proposed scheme. The other cases are related to the proposed scheme with different policies ($\alpha = 0, 1, 10$, respectively). For each case, we used K-shortest paths with $K \leq 4$ and set $w_s = 1$ for all s. When K paths were used for an IE pair, we distributed TCP traffic equally by associating $\frac{36}{K}$ TCP flows with each path.

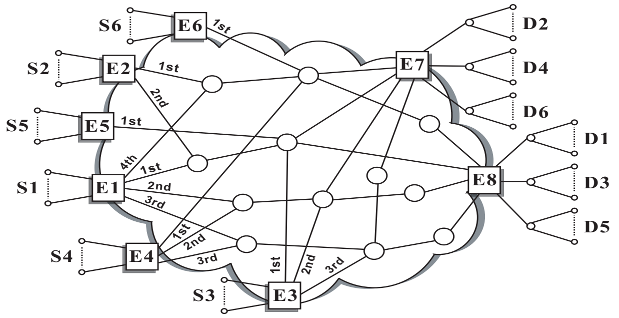


Fig. 1. Simulation scenario with modified NSF topology

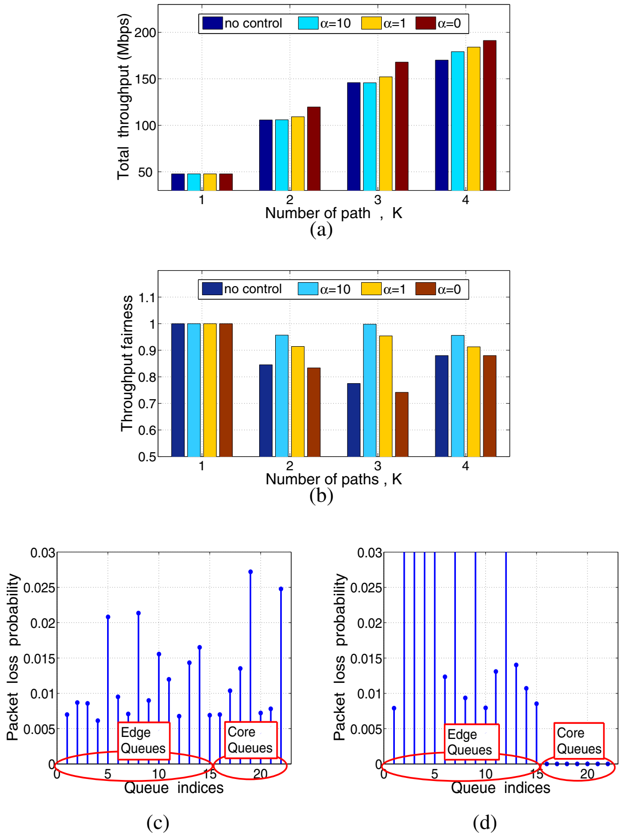


Fig. 2. Simulation results: (a) total throughput of IE flows, (b) throughput fairness of IE flows, (c) packet loss probability when the proposed scheme is not deployed, (d) packet loss probability when the proposed scheme is deployed with $\alpha = 1$ and $K = 4$.

Fig. 2(a) shows the throughput gain according to the number of paths. As we expected, as path diversity increases with increasing K, the total throughput of IE pairs increases but the increment decreases eventually, since either additional disjoint paths are not found or, if they are found, share some links with existing paths. This is a natural phenomenon, irrespective of the use of the proposed scheme or the choice of policy α. In addition to the throughput gain, we explored the efficiency-fairness property with respect to α and the number of paths. By comparing Fig. 2(a) with Fig. 2(b), we observe that for a given K, as α decreases, the total throughput of IE pairs tends to increase but the throughput fairness between IE pairs tends to degrade, and vice versa. This is the exact efficiency-fairness relationship shown by single-path networks [4]. However, if the proposed scheme is not applied, efficiency suffers due to lower throughput and fairness is supported less. Indeed, we can observe in the figures that as the number of paths increases, the throughput gain is always the lowest and fairness is never supported strictly. Finally, we investigated whether the proposed scheme can guarantee loss free service in the core networks. We argued that the proposed scheme renders the network virtually loss-free at the optimal point, even though all the inputs are TCP sources. As shown in Fig. 2(c) and Fig. 2(d), most of the packet losses occur at the ingress-edge buffers and only transient losses can occur inside the network.

Even though the scenario in the simulations used TCP traffic, we also simulated the same scenario with the condition that all IE pairs received persistent traffic without TCP. The throughput, fairness, and loss performance was about the same as above, although we do not present the result here due to limited space. What this implies is that as long as the ingress-edge buffers are maintained to always have something to send, whether the sources are TCP or deterministic, the traffic conditions do not matter to the proposed scheme from the standpoint of aggregate control.

B. Comparison with non-TE environment

We now investigate the improvement in performance in a more realistic network scenario by comparing our results with those for a non-TE environment network. For this purpose, we use a network topology shown in Fig. 3, which is based on the topology of a commercial IPS. The topology is composed of twenty edge routers and two core routers, so that each IE pair can have two paths. The capacity and the propagation delay of each link are set to values which are really configured and measured in the ISP network. However, we omit them with the proprietary reason which are contracted to the ISP company. The buffer sizes of all links are set equally to $100,000$ packets. The aggregate flow at each ingress edge node is modeled by a Pareto on-off source and the average on-period is set to 30secs to represent bursty traffic. We assume that the aggregate flows of all ingress edge nodes except $E17$ always have enough data to send and are delivered to $E17$ through each path by $C1$ and $C2$, respectively. The average rate of each aggregate flow is assigned as follows. Using the 40% of the bottleneck link capacity, i.e., links from $C1$ to $E17$ and $C2$ to $E17$, as a basis, the average rate of each flow is allocated in proportion to the outgoing link capacity of its ingress edge node. We set the policy parameter, α, of the proposed scheme to 1 throughout the simulation.

1) Loss rate, throughput, and delay: We first compare three performance metrics: throughput, loss rate, and delay. The results are shown in Fig. 4(a) and Fig. 4(b) in trace format to capture the characteristics of traffics. The figures show the aggregate source rate entering at $C1$ and the queue size of bottleneck link from $C1$ to $E17$. As shown in the figures, when the proposed scheme is applied, all of the aggregate sources are well controlled so as not to exceed the capacity of bottleneck link. As a result, the buffer always remains well below its limit. This constitutes further evidence that under the proposed scheme, the core network is loss-free. By contrast, if no TE scheme is applied, it is observed that uncontrolled aggregate traffic exceed the bottleneck link capacity and has a severe impact on throughput, loss rate, and delay. This may be seen from Table I, in which it is evident that the loss rate degrades exactly as much as throughput does, i.e., 19.22%, and the packet delay is almost thirty times longer on the average. We omit the other result for $C2$ to $E17$ due to space limitation. However, the results are almost the same.

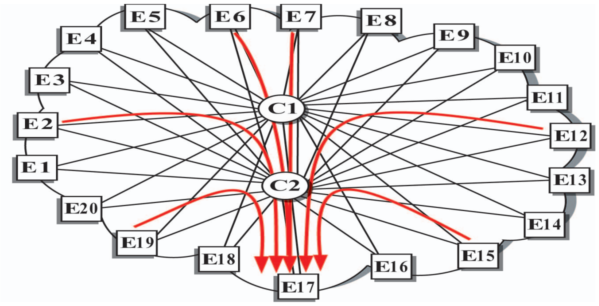


Fig. 3. Simulation scenario with the commercial ISP's network topology

2) Network efficiency: ISPs usually overprovision their network in order to cope with unpredictable traffic surges. However, this is not a cost-efficient approach from the point of view of TE. We now determine how much capacity allocation can be saved without affecting existing network performance significantly if the proposed scheme is applied. For this purpose, we simulate the above-stated scenario again with the bottleneck link capacity reduced to 50%. The reduced capacity eventually represents the amount by which the allocated capacity may be saved.

Fig. 4(c) and Fig. 4(d) show the result in trace format. As shown in the figures, even when the capacity is reduced to 50%, the proposed scheme works well. The sum of each aggregate flow can be maintained under the capacity of bottleneck link. Moreover, it can be seen from Table. I that the proposed scheme is able to support almost the same performance as before in this case, that is, it still keeps the network loss free and delivers the packets with a throughput of 99.53%. However, from the Fig. 4(c) and the table, we can see that this is achieved at the cost of *Delay*. The reason why the *Delay* becomes longer is that given that the traffic change is relatively severe due to reduced capacity, the proposed scheme tries to mitigate the effect of traffic surge by buffering as many packets as it can. This can be seen by comparing the queue traces between Fig. 4(a) and Fig. 4(c). Nonetheless, given an extreme case in which the bottleneck link capacity is reduced to 50% and the total performance gain achieved by

TABLE I
PERFORMANCE METRIC MEASURED AT EACH CASE

	Loss Rate	Throughput	Average Delay
With TE	0.0%	6.02 Gbps	0.34ms
Without TE	19.22%	4.86 Gbps	9.87ms
With TE (50% capacity)	0.05%	5.992 Gbps	58.2ms
Without TE (50% capacity)	40.1%	3.6 Gbps	50.96ms

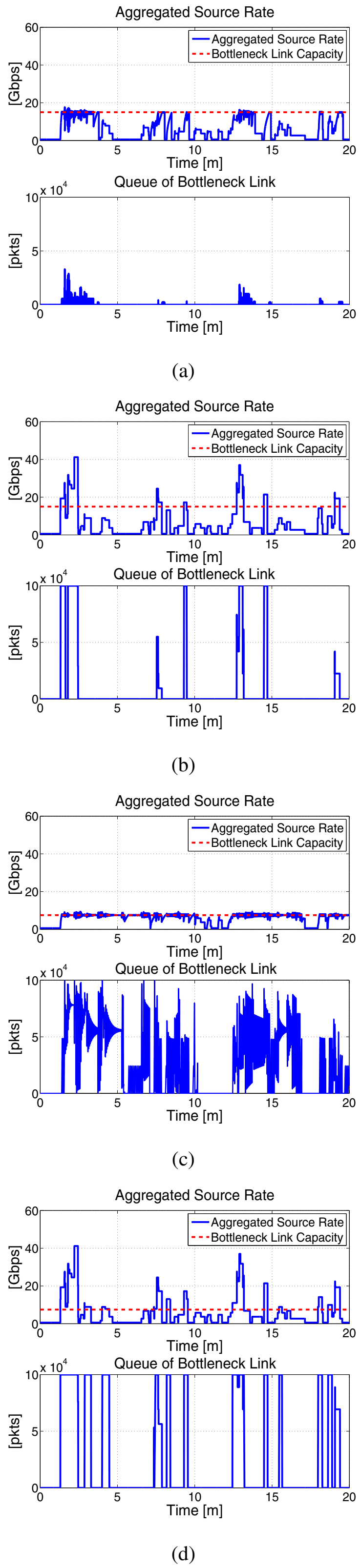
the proposed scheme, the degradation is moderate rather than severe. This may be seen by comparing our results with those of non-TE case in Table. I, where at the almost same cost of delay, packet loss and throughput degrade nearly twice as before and it almost looks as though one of the two packets is not delivered or lost. From the results, it is evident that in current ISP networks, a certain amount of the allocated capacity is unused and wasted. By using the proposed scheme, that unused amount can be reduced without performance being degraded significantly. Hence, we renders the network more cost-efficient.

V. CONCLUSION

We presented an online TE method for multi-path networks, which dynamically adapts the traffic load and controls the flow on the basis of real-time changes in traffic, rather than long-term average traffic demand or a pre-selected set of traffic matrices. The method is implemented in a fully distributed manner and requires only local knowledge of the queue backlog to reach the global optimal point. The method provides equal cost load balancing and keeps the network loss free at an optimal point, regardless of source types. It also provides almost the same performance with reduced network resources. This should be beneficial for many ISPs, since they can make the network more cost-effective by reducing over-provisioning. It remains to prove mathematically the convergence in an asynchronous environment in order to complete the performance measure.

REFERENCES

[1] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *ACM SIGCOMM'05*, Aug. 2005, pp. 253–264.
[2] H. T. Kung and S. Y. Wang, "Tcp trunking: Design, implementation and performance," in *ICNP'99*, Oct. 1999, pp. 222–231.
[3] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "Mate: Mpls adaptive traffic engineering," in *IEEE INFOCOM'01*, April. 2001, pp. 1300–1309.
[4] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. on Networing*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
[5] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.
[6] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
[7] W. Wang, M. Palaniswami, and S. Low, "Optimal flow control and routing in multi-path networks," *Performance Evaluation*, vol. 52, pp. 119–132, 2003.
[8] X. Lin and N. Shroff, "The multi-path utility maximization problem," in *Proc. of 41st Annual Allerton Conference on Communication, Control and Computing*, Oct 2003.
[9] S. Kandula, D. Katabi, S. Shnha, and A. Berger, "Dynamic load balancing without packet reordering," in *ACM SIGCOMM Computer Communication Review*, vol. 37, April. 2007, pp. 53–62.



(a)

(b)

(c)

(d)

Fig. 4. Simulation results: (a) traces of sum of all aggregate flow rates and queue with the proposed scheme, (b) traces of sum of all aggregate flow rates and queue without the proposed scheme, (c) traces of sum of all aggregate flow rates and queue with the proposed scheme (50% capacity), (d) traces of sum of all aggregate flow rates and queue without the proposed scheme (50% capacity).