Can Congestion Control and Traffic Engineering Be at Odds?

Jiayue He
Dept. of EE, Princeton University
Email: jhe@princeton.edu

Mung Chiang
Dept. of EE, Princeton University
Email: chiangm@princeton.edu

Jennifer Rexford
Dept. of CS, Princeton University
Email: jrex@cs.princeton.edu

***Abstract*— In the Internet today, traffic engineering is performed assuming that the offered traffic is inelastic. In reality, end hosts adapt their sending rates to network congestion, and network operators adapt the routing to the measured traffic. This raises the question of whether the joint system of congestion control and routing is stable and optimal. Using established optimization models for TCP and traffic engineering as a basis, we find the joint system is stable and typically maximizes aggregate user utility through simulation. The joint system may deviate from this solution when the topology is not uniform. A modification to the joint system will guarantee stability and optimality for applications that are sufficiently elastic, but at the cost of robustness.**

Keywords: Network utility maximization, Optimization, Congestion control, Routing, Traffic Engineering, Robustness.

I. Introduction

In the Internet today, end hosts running the Transmission Control Protocol (TCP) adapt their sending rates in response to network congestion. Separately, network operators monitor their networks for signs of overloaded links and adapt the routing of traffic to alleviate congestion, in a process known as traffic engineering. TCP congestion control assumes that the network paths do not change, and traffic engineering assumes that the offered traffic does not change. Due to the layered network architecture, congestion control and routing operate independently, though their individual decisions are inevitably coupled. In this paper, we investigate whether the joint system is stable and optimal and there is a good alternative system.

Traffic engineering and congestion control both solve, explicitly or implicitly, optimization problems defined for the entire network. Traffic engineering consists of collecting measurements of the traffic matrix—the observed load between each pair of entry and exit points—and performing a centralized minimization of a cost function that considers the resulting utilizations on all links (*e.g.*, [1], [2]). In contrast, TCP congestion control can be viewed as *implicitly* solving an optimization problem in a distributed fashion (*e.g.*, [3], [4], [5], [6]), where the many variants of TCP differ in the shape of user utility as a function of the source rate.

Previous analysis of congestion control and routing used congestion price as link weights (*e.g.*, [7], [8]) rather than modeling the current traffic engineering practices. In this paper, we use the established optimization models in (*e.g.*, [1], [2], [3], [4], [5], [6]) to study the interaction between traffic engineering and congestion control, and examine the following key questions through both analysis and simulation:

- **Stability**: Do the joint dynamics of congestion control and routing converge to an equilibrium?
- **Optimality**: If the joint system does converge, does the equilibrium maximize the aggregate user utility, over both the routing parameters and source rates?
- **Better design**: Can we modify the current system to guarantee stability and optimality?

In our joint congestion control and traffic engineering (CC-TE) model TCP converges under a fixed routing configuration, before any routing changes are made. From our analysis and simulation experiments, we obtain the following insights:

- **Confirming the intuition of network operators**: Our simulation results show the CC-TE model is stable for a variety of topologies.
- **Tension between performance and robustness**: A modification to the CC-TE model can guarantee stability and optimality (Theorem 1), but at the cost of robustness.

The rest of the paper is organized as follows. Section II introduces the network topology, congestion control and our joint congestion control and traffic engineering model. We simulate the CC-TE model in Section III and extend it with analysis in Section IV. Finally, Section V concludes the paper and points to future work.

II. Network Model

For analytic tractability, our study makes some simplifying assumptions. First, we focus on routing and congestion control in a single Autonomous System, where the operator has full view of the offered traffic load and complete control over routing. Second, we consider a routing model where traffic between source-destination pairs can be split arbitrarily across multiple paths. This is not the OSPF [9] or IS-IS [10] protocols used today, but can be implemented using the emerging MPLS [11] technology. Third, we assume that the sources have infinite backlog, i.e., long sessions modeling "elephant" traffic in the Internet. In other words, we ignore session level stochastic dynamics, and focus on the average behavior of TCP traffic profile, even though the actual sessions themselves are different. Our notation follows the work in [7], [8]: in general, small letters are used to denote vectors, *e.g.*, x with x_i as its i^{th} component; capital letters to denote matrices, *e.g.*, R, or constants, ***e.g.***, L, N. Also t is used to denote the iteration number, ***e.g.***, $x(t)$.

A. Network Topology and Routing

A network is modeled as a set of L bidirectional links with finite capacities $c = (c_l, l = 1, \ldots, L)$, shared by a set of N source-destination pairs, indexed by i; we often refer to a source-destination pair simply as "source i."

The routing matrix R_{li} specifies the fraction of i's flow that traverses each link l. We consider a model of routing that closely reflects today's operational practice [1], [2]. The operators measure the offered load between each ingress-egress pair x_i. Based on the known network topology and the traffic matrix, the operators try to find the best routing matrix R to minimize network congestion.

For a given routing configuration, the utilization of link l is $u_l = \sum_i R_{li}x_i/c_l$. To penalize routing configurations that congest the links, candidate routing solutions are evaluated based on a cost function $f(u_l)$ that increases steeply as u_l approaches 1 (while staying finite): $f(u_l)$ is strictly convex and increasing. In addition, f should map zero to zero, since there should be no penalty for a link with 0% utilization. The routing update $R(t+1)$ is the solution to the following optimization problem over R for fixed x and c:

$$\text{minimize} \quad \sum_l f(\sum_i R_{li}x_i/c_l). \tag{1}$$

By considering the total link cost rather than trying to minimize a single bottleneck, the optimization framework would prefer a solution that utilizes a single link at 91% over one that loads many links at 90%. In practice, the network operators often use a piecewise-linear f for faster computation time [1], [2]. In this paper, we allow f to be any strictly convex, increasing, and continuous function that maps zero to zero.

B. TCP Model

While the various TCP congestion-control algorithms were originally designed based on engineering heuristics, recent work such as [5], [6] has shown through reverse engineering that they implicitly solve a convex optimization problem in a distributed fashion. Consider a network where each source i has a utility function $U_i(x_i)$ as a function of its total transmission rate x_i. The basic (concave) network utility maximization problem over source rate vector x, for a given *fixed* routing matrix R, is

$$\begin{array}{ll} \text{maximize} & \sum_i U_i(x_i) \\ \text{subject to} & Rx \preceq c. \end{array} \tag{2}$$

The goal is to maximize aggregate user utility by varying x (but not R), subject to the linear flow constraint that link loads cannot exceed capacity. TCP congestion-control algorithms implicitly solve (2), with different TCP variants maximizing different increasing and concave utility functions.

The utility function can be used to describe the user's degree of satisfaction with a particular throughput, and can also be viewed as a measure of the elasticity of the traffic. The aggregate utility capture both the efficiency and fairness of the system in allocating bandwidth to the traffic. A particular family of widely-used utility functions is parameterized by $\alpha \geq 0$ [12]:

$$U_\alpha(x) = \begin{cases} \log x, & \alpha = 1 \\ (1-\alpha)^{-1}x^{1-\alpha}, & \alpha \neq 1. \end{cases} \tag{3}$$

Maximizing these α-fair utilities over linear flow constraints leads to rate-allocation vectors that satisfy the definitions of α-fairness in the economics literature.

A utility function with $\alpha = 2$ was examined by [13], and then linked to TCP Reno [14]. TCP Vegas can be interpreted as $\alpha = 1$, see [4]. Other TCP variants that can be interpreted as $\alpha = 1$ are STCP [15] and FAST [16]. XCP has $\alpha \to \infty$ in the single-link case [17]. One widely-deployed TCP variant that is not modeled by an α-fair utility function is TCP Tahoe, which has been reverse engineered and shown to be maximizing the utility function $U(x) = \arctan x$ [5].

C. Joint Congestion Control and Traffic Engineering Model

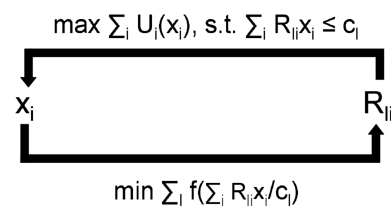


Fig. 1. A detailed view of the CC-TE model.

The CC-TE model, as shown in Figure 1, has two steps in each iteration of the feedback loop. At time $t+1$, the congestion-control step first computes new source rates based on the routing configuration from time t:

$$x(t+1) = \text{argmax}_x \sum_i U_i(x_i), \text{ subject to } R(t)x \preceq c. \tag{4}$$

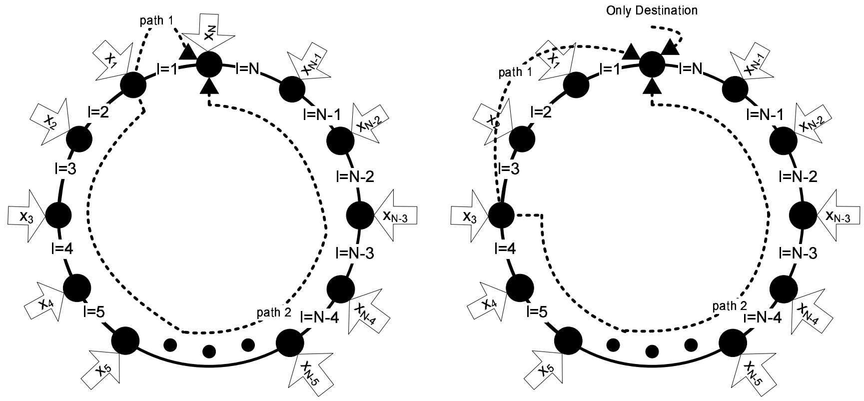
Then the routing step computes new paths based on the source rates:

$$R(t+1) = \text{argmin}_R \sum_l f\left(\sum_i R_{li}x_i(t+1)/c_l\right). \tag{5}$$

The iterations of (4,5) repeat over time, with congestion control adapting the source rates to the new routes, and traffic engineering adapting the routes to the measured traffic.

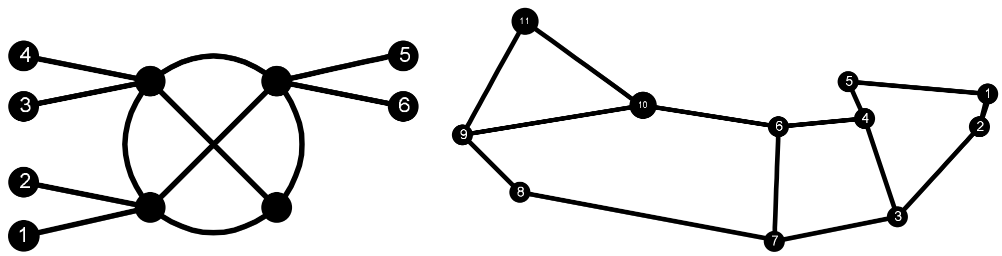
III. Simulation Results

We first illustrate some interesting numerical observations before presenting theorems on stability and optimality. Our numerical experiments use a combination of the Matlab and MOSEK [18] environments.



(a) N nodes, N sources (b) N nodes, 1 destination

Fig. 2. Two N-node ring topologies with different traffic patterns.



(a) Access-Core topology (b) Abilene topology

Fig. 3. Two realistic topologies.

A. Simulation Set-up

We evaluate two variants of TCP congestion control: $\alpha = 2$ (*e.g.*, TCP Reno) and $\alpha = 1$ (*e.g.*, TCP Vegas). For the cost-function $f(u_l)$, we use an exponential function, which is the continuous version of the function used in various studies of traffic engineering [1], [2].

Our initial experiments evaluate a simple N-node ring topology, where we can easily scale the size of the network. To evaluate the influence of the traffic patterns, we consider two scenarios. In the first scenario, each node is a source sending to its clockwise neighbor; each source has two possible paths: a direct one-hop path and an indirect $(N-1)$-hop path. In the second scenario, node 1 is the destination and the remaining $N-1$ nodes are sources; each source x_i has an i-hop path and an $(N-i)$-hop path. Our experiments vary the number of nodes N and the capacity of link 1 (between nodes 1 and N).

To study realistic topologies with greater path diversity, we also experiment with the two networks in Figure 3. On the left is a tree-mesh topology, which is representative of a common network structure. In the middle is a full mesh representing the core of the network with rich connectivity. On the edge are three access tree subnetworks. Of the twelve possible source-destination pairs, $1-3$, $1-5$, $2-4$, $2-6$, $3-5$, and $4-6$ are chosen, and for each source-destination pair, the three minimum-hop paths are chosen as possible paths. On the right is the Abilene backbone network [19]. Of the many possible source-destination pairs, we choose $1-6$, $3-9$, $7-11$, and $1-11$. For each source-destination pair, we choose the four minimum-hop paths as possible paths. For the access-core and Abilene topologies, the simulations assume the link capacities follow a truncated (so as to avoid negative values) Gaussian distribution, with an average of 100 and a standard deviation that varies from 0 to 50. We simulate twenty random configurations for each value of the standard deviation. In all experiments, we start with an initial routing configuration that splits traffic evenly among the K paths for each source-destination pair.

B. Suboptimality Gap Simulations

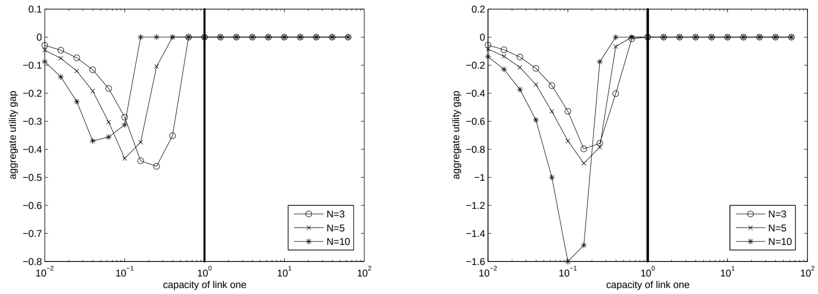
Given the structure of (2), it is natural to wonder if the interaction of congestion control and traffic engineering maximizes aggregate user utility. Previous work [20], [7] has proposed the following joint optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_i U_i(x_i) \\ \text{subject to} & Rx \preceq c,\ x \succeq 0 \end{array} \qquad (6)$$

where *both* R and x are variables.

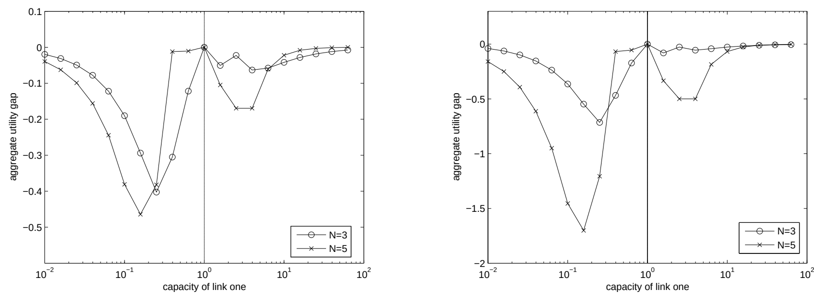
Our experiments quantify the gap in aggregate utility between the joint system and the optimal aggregate utility of (6). Table I summarizes the key results.

In Figure 4, we vary the capacity of link 1 and plot the gap in aggregate utility for ring topologies with three, five, and ten nodes, where each node communicates with its clockwise neighbor. The two graphs plot results for $\alpha = 1$ (*e.g.*, TCP Vegas) and $\alpha = 2$ (*e.g.*, TCP Reno), respectively.



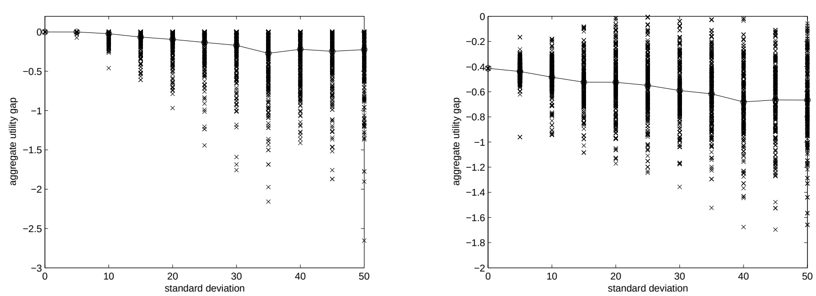
(a) $\alpha = 1$ (*e.g.*, TCP Vegas) (b) $\alpha = 2$ (*e.g.*, TCP Reno)

Fig. 4. Aggregate utility gap for the N-node, N-source ring.



(a) $\alpha = 1$ (*e.g.*, TCP Vegas) (b) $\alpha = 2$ (*e.g.*, TCP Reno)

Fig. 5. Aggregate utility gap for the N-node, 1-destination ring.



(a) Access-core topology (b) Abilene topology

Fig. 6. Aggregate utility gap for two realistic topologies (with $\alpha = 1$). A -x- marker denote an individual test point and a -o- marker denote the average.

Figure(s)	Key Message
4a versus 5a	Traffic pattern has a significant effect.
4a versus 4b 5a versus 5b	TCP variants give the same trend.
All figures	Relatively small suboptimality gap.
All figures	Homogeneity minimizes suboptimality gap.

TABLE I

SUMMARY OF RESULTS ON SUBOPTIMALITY GAP.

The graphs show trends that are very similar across a range of topology sizes, suggesting that the number of sources alone does not have a significant influence on the suboptimality gap. Similarly, the two TCP variants lead to very similar results.

The vertical line in the middle of the two graphs highlights the configuration where all links have unit capacity. The suboptimality gap is zero for a wide range of capacity configurations. When one link has much lower capacity than the other links, a suboptimality gap emerges. This occurs because the traffic-engineering step in the joint system stops making use of this low-capacity link, since the penalty for placing even a small amount of load on this link exceeds the cost of forcing the traffic on a longer path that places load on multiple links. When link 1 has an extremely low capacity, even the optimal solution cannot place much traffic on this link, leading to a small suboptimality gap.

The graphs in Figure 5 confirm that variations in link capacities affect the suboptimality gap. These graphs evaluate the N-node ring with one destination node, for two values of N and two TCP variants. In contrast to Figure 4, having either a smaller or a larger capacity on link 1 leads to a suboptimality gap. This is not surprising because link 1 is a bottleneck link for this traffic pattern. If the link has a small capacity, the traffic-engineering step does not make use of the link, making the left part of these curves closely resemble the plots in Figure 4. If the link has a high capacity, the traffic-engineering step tries to direct more sources through the link; however, this is not the best solution when the capacity of link 1 is just slightly larger than the other links because traffic traverses longer paths, placing load on a larger number of links. Comparing Figures 4 and 5 illustrates the important role the traffic pattern plays in determining whether the joint system successfully maximizes aggregate utility.

The graphs in Figure 6 illustrate the effects of a variation in link capacities on realistic topologies. We show how the suboptimality gap depends on the standard deviation of the link capacities, which are all varied according to a truncated Gaussian distribution; we plot separate points for each of the 500 experiments for each value of standard deviation, as well as a curve that highlights the mean values. The trend that a more homogeneous capacity distribution (smaller standard deviation) would lead to a smaller suboptimality gap exists, but it is much more subdued than in the ring topology and it is dominated by the variance. This suggests with realistic topologies, the relationship between link capacity and utility gap is more complex. One possible explanation is that the bottleneck link on each path is what matters and while that is easily correlated with varying a single link in the ring topology, the effect is coupled in a more complex topology. In addition, for the Abilene topology, a suboptimality gap exists even for a homogenous capacity distribution. While the results of the ring topology suggested network operators can favor certain configurations to improve network efficiency, it is more challenging when dealing with realistic topologies.

IV. Convergence and Optimality Analysis

Our simulations showed that the CC-TE model is stable and close to optimal for a range of topologies. We speculate, but cannot yet show it is provably stable for general topologies. In this section, we show how a change in the cost function can lead to a provably stable and optimal joint system. This comes at the cost of robustness, however, and is not recommended for implementation.

Theorem 1: If the cost function f is zero until $u_l = 1$, and positive afterwards, then the CC-TE model converges for sufficiently concave utilities (i.e., sufficiently elastic traffic): $U_i''(x_i) \leq -\frac{U_i'(x_i)}{x_i}$. In particular, it converges for α-fair utilities when $\alpha \geq 1$ and for $\arctan$ utility of TCP Tahoe.

Proof: The proof consists of three main steps. First we show that there exists an unconstrained optimization over both x and R such that the joint congestion control and routing system is equivalent to a successive, alternating optimization over x and then R. Then we provide a sufficient condition to guarantee convergence. Finally the condition is examined for α-fair utilities and arctan utility.

Consider the unconstrained minimization of

$$g(x,R) = -\sum_i U_i(x_i) + \gamma \sum_l f\left(\sum_i R_{li} x_i / c_l\right) \quad (7)$$

for some $\gamma \geq 0$. The two steps in the alternating optimization method of Gauss-Siedel algorithm [21] are as follows:

$$\begin{aligned} x(t+1) &= \text{argmin}_x -\sum_i U_i(x_i) + \gamma \sum_l f(\sum_i R_{li}(t) x_i / c_l) \\ R(t+1) &= \text{argmin}_R g(x(t+1), R(t)) \\ &= \text{argmin}_R \sum_l f(\sum_i R_{li}(t) x_i(t+1)/c_l). \end{aligned}$$

The minimization of $g(x,R)$ over R is clearly equivalent to (1). We need to further show that minimizing $g(x,R)$ over x (an unconstrained problem) is equivalent to the utility-maximization problem implicitly solved by TCP congestion control (2) over x (a constrained problem), for sufficiently large γ. By the penalty function method (see [22] for details), there exists a penalty function P and a constant γ so that (2) is equivalent to (8):

$$\text{maximize}_x \sum_i U_i(x_i) - \gamma \sum_l P\left(\sum_i R_{li} x_i / c_l - 1\right), \quad (8)$$

provided that γ is sufficiently large and P is convex, increasing, and zero for $Rx \preceq c$ (positive otherwise). Essentially, in (8) $-\gamma \sum_l P\left(\sum_i R_{li} x_i / c_l - 1\right)$ in the objective function replaces the constraint $Rx \preceq c$ when γ is sufficiently large. We now just need to establish a mapping between the link cost function f and penalty function P while preserving the desired properties. Indeed, convexity of P implies convexity of f (convexity is preserved through a linear operation). If P is increasing, so is f. If operators chooses a cost function f which is zero until $u_l = 1$, then it can match P exactly.

So far we have constructed an optimization problem (minimization of $g(x,R)$) whose Gauss-Siedel solution algorithm is equivalent to the system model of joint congestion control and routing as described in the previous subsection. Now we will examine the conditions for convergence of this Gauss-Siedel Algorithm. From [21], the Gauss-Siedel Algorithm will converge to the minimizer of g if g is bounded from below, differentiable, marginally strictly convex in x and R, and jointly convex in x and R.

The first three conditions are already satisfied through the constraints placed in the system model definition. Condition 1 is satisfied since $x \succeq 0$, $R \succeq 0$ by definition. Condition 2 is satisfied since U and f are differentiable, so is g. The third condition is satisfied since U is strictly concave in x, and f is marginally strictly convex in x and R. The last condition is not satisfied in general since the function $f(\sum_l R_{li} x_i / c_l)$ is not jointly convex in R and x.

In order to satisfy the condition on joint convexity in x and R, consider a log change of variable. Let $\tilde{x}_i = \log x_i, \tilde{R}_{li} = \log R_{li}$, then $R_{li} x_i = \exp(\tilde{R}_{li} + \tilde{x}_i)$. With the change of variable, it can be readily verified that f is still jointly convex in $\tilde{x_i}$ and $\tilde{R_{li}}$, but the utility function may no longer be concave in $\tilde{x}$. If the utility function is concave in $\tilde{x}$, then g would be strictly convex in $\tilde{x}$ since f is strictly convex in $\tilde{x}$. Denote the new utility function (after the log change of variable) as $W_i(\tilde{x_i})$. A sufficient condition for convergence of the Gauss-Siedel algorithm is for W to be concave in $\tilde{x}$. A simple derivation shows that such a condition reduces to the following simple bound on the curvature of the utility function: $U_i^{''}(x_i) \leq -U_i'(x_i)/x_i$.

Now we specialize to the α-fairness model for U which covers TCP Reno (currently deployed) and several proposed variants. In this case, $W_\alpha(\tilde{x})$ can be written as follows:

$$W_\alpha(\tilde{x}) = \begin{cases} \tilde{x}, & \alpha = 1 \\ (1-\alpha)^{-1}\exp(x)^{1-\alpha}, & \alpha \neq 1. \end{cases} \qquad (9)$$

Examining $W^{''}(\tilde{x})$ shows that $W(\tilde{x})$ is concave for $\alpha \geq 1$.

Finally, TCP Tahoe is examined. Recall that $U(x) = \arctan(x)$ for TCP Tahoe, and $W^{''}(\tilde{x}) = \arctan(x)$. It follows that $W^{''}(\tilde{x}) = (\exp(\tilde{x}) - \exp(3\tilde{x}))/(1+\exp(2\tilde{x}))^2$ and W is concave. Therefore, convergence of the system model is guaranteed for TCP algorithms with $\alpha \geq 1$ and TCP Tahoe. ■

Following a similar argument as in this proof, Theorem 1 can be extended to utility functions W which are not concave in $\tilde{x}$, as long as link-cost functions f are sufficiently convex.

While this change to the objective function of traffic engineering can guarantee stability and optimality, we do not recommend it in practice for robustness reasons. By letting the cost function be zero up until link utilization hits capacity, there is a risk of having multiple links operate at capacity. This is a fragile point of operation for the network since a small burst in traffic would cause the traffic on certain links to exceed capacity. Once the traffic exceeds capacity, congestion is inevitable and so is the subsequent packet loss and delay increase. So here we have a trade-off between performance metrics on one hand (stability and optimality) and robustness on the other.

V. Conclusion

In this paper, we have studied the interaction of congestion control and routing from a network operator's perspective. TCP and traffic engineering both try to make efficient use of link bandwidth to improve network performance for end users. In today's IP networks, however, these two mechanisms operate independently, though they are coupled because they both adapt to network congestion. In this paper, we find through simulation that TCP and traffic engineering work effectively together to reach a stable equilibrium that maximizes aggregate user utility under most network configurations. A modification to the operator's cost function leads to a provably stable and optimal system, but at the cost of robustness. This highlights the potential tension between performance and non performance metrics.

In our ongoing work, we have defined an optimization problem where the objective is a weighted difference of end-user utilities and network operator penalty function. A distributed solution to this problem and its implementation over existing TCP and traffic-engineering systems have recently been presented [23]. This helps to balance the tension between robustness and optimality in two ways. First, by incorporating the operator's penalty function into the objective, it protects the network from short traffic bursts. Second, by finding a distributed solution, the algorithm can react to traffic shifts on a smaller timescale.

Acknowledgment

We would like to thank Ma'ayan Bresler for her help with the simulation re-runs and her suggestions to the proof. This work has been supported in part by NSF grants CNS-0519880 and CCF-0448012, and a Cisco grant GH072605.

References

[1] B. Fortz and M. Thorup, "Optimizing OSPF weights in a changing world," *IEEE JSAC*, vol. 20, pp. 756–767, May 2002.

[2] J. Rexford, "Route optimization in IP networks," in *Handbook of Optimization in Telecommunications*, Springer Science + Business Media, February 2006.

[3] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. of Operational Research Society*, vol. 49, pp. 237–252, March 1998.

[4] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: A duality model," *J. of the ACM*, vol. 49, pp. 207–235, March 2002.

[5] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, pp. 525–536, August 2003.

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

[7] J. Wang, L. Li, S. H. Low, and J. C. Doyle, "Cross-layer optimization in TCP/IP networks," *IEEE/ACM Trans. Networking*, vol. 13, pp. 582–595, June 2005.

[8] J. He, M. Chiang, and J. Rexford, "TCP/IP Interaction Based on Congestion Price: Stability and Optimality," in *Proc. International Conference on Communications*, June 2006.

[9] J. Moy, "OSPF Version 2." RFC 2328, April 1998.

[10] R. Callon, "Use of OSI IS–IS for Routing in TCP/IP and Dual Environments." RFC1195, December 1990.

[11] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture." RFC 3031, January 2001.

[12] J. Mo and J. C. Walrand, "Fair End-to-end Window-based Congestion Control," *IEEE/ACM Trans. Networking*, vol. 8, pp. 556–567, October 2000.

[13] L. Massoulié and J. Roberts, "Bandwidth sharing: Objectives and Algorithms," *IEEE/ACM Trans. Networking*, vol. 10, pp. 320–328, June 2002.

[14] S. Kunniyur and R. Srikant, "End-to-End Congestion Control: Utility Functions, Random Losses and ECN Marks," *IEEE/ACM Trans. Networking*, pp. 689–702, October 2003.

[15] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, pp. 83–91, February 2003.

[16] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," in *Proc. IEEE INFOCOM*, March 2004.

[17] S. H. Low, H. Andrew, and B. P. Wydrowski, "Understanding XCP: Equilibrium and Fairness," in *Proc. IEEE INFOCOM*, March 2005.

[18] MOSEK Optimization Software. `http://www.mosek.com/`.

[19] Abilene Backbone. `http://abilene.internet2.edu/`.

[20] X. Lin and N. B. Shroff, "Utility Maximization for Communication Networks with Multi-path Routing," *IEEE Trans. Automatic Control*, 2006. To appear.

[21] D. P. Bersekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, second ed., 1997.

[22] D. P. Bersekas, *Nonlinear Programming*. Athena Scientific, second ed., 1999.

[23] J. He, M. Chiang, and J. Rexford, "DATE: Distributed Adaptive Traffic Engineering." Poster session at INFOCOM 2006.