Analysis of Fast Adaptive Traffic Engineering Using a Feedback Control Model

Ning Wu and Ao Tang

Abstract— **This paper analyzes fast adaptive traffic engineering from a feedback control perspective. We provide a model that characterizes the system parameters' effects on the performance of the dynamic routing system. This allows quantitative analysis of adaptive traffic engineering control laws and its design parameter choices. We then specialize the general framework in two representative network topologies and derive the stability conditions for their dynamic routing systems. Experiments are carried out to compare against our theoretical predictions considering two forms of adaptive traffic engineering control law as examples. Together they provide systematic insights on the relations among several network factors and the intrinsic tradeoff among different network control objectives.**

I. INTRODUCTION

Traffic engineering (TE) aims at improving the network performance by effectively mapping the traffic demand onto the network topology. In recent years, SDN [1] makes it more feasible to update routing frequently, and thus motivates the fast adaptive TE which gradually adjusts the routing decisions at high frequency according to the network states and loads in real time [2], [3]. Fast adaptive TE solutions [4], [6], [5], [7], [8], [9] become increasingly attractive because of their potential ability to fast react to the time-varying traffic demand as well as network failures and better utilize the network resources.

Any fast adaptive TE technique introduces more than one system parameters into their design. This is mainly due to the consideration of the intrinsic performance tradeoff between stability and responsiveness. On one hand, adaptive TE is expected to react to any network changes and traffic fluctuations as quickly as possible; on the other hand, it is supposed to keep the system stabilized in the steady state to prevent routing oscillation. In the real networks, the stability and responsiveness are affected by multiple factors, from design parameters in the control laws to engineering factors such as propagation delay, measurement noise. Moreover, a systematic understanding of how the factors interact with each other is nontrivial. Therefore, it is challenging to quantitatively analyze the impacts of the system parameters on the performance metrics for a given adaptive TE solution, let alone to best strike the performance tradeoff.

In this paper we provide a framework for analyzing the performance of any given adaptive TE solutions. We take a comprehensive approach and form a model that characterizes three critical parameters: (i) the step size in the TE control law which determines how much to move along the descending direction at each iteration; (ii) the update interval of the routing computation which defines how frequently the routing strategy changes; and (iii) the physical propagation delays on the network. Being reactive benefits from large step size and frequent update, whereas maintaining stability requires cautious steps and small system noise which is sensitive at high frequency. We analyze the intrinsic interactions of these parameters in two representative case studies. Our experimental results further provide insightful observations of their effects on the network performance.

The rest of the paper is organized as follows. In Section II, we describe the general model of the fast adaptive TE system, taking the system parameters into consideration. In Section III we specialize the model to two typical examples of network topology: single-link case and shared-link case, and analyze the internal relations of the parameters. In Section IV, the model and analysis are validated with the experimental results from Mininet [10] emulations, and based on the results we further discuss how the network performance is affected by the parameters. We conclude in Section V.

Ning Wu and Ao Tang are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, 14853, USA. Emails: {nw276, at422}@cornell.edu

II. MODEL

In this section, we propose the model that is used to analyze the fast adaptive TE system. In general, any real-time adaptive TE solutions can be viewed as a feedback control system (as is illustrated in Figure 1): the network itself is the plant which evolves with varying traffic demand input as well as the routing strategy being updated by some routing engine as the controller; periodically the metrics of the network state is measured and fed back to the routing engine; the routing engine then computes the TE routing solutions based on the state information obtained and configures the new routing strategy to the network.

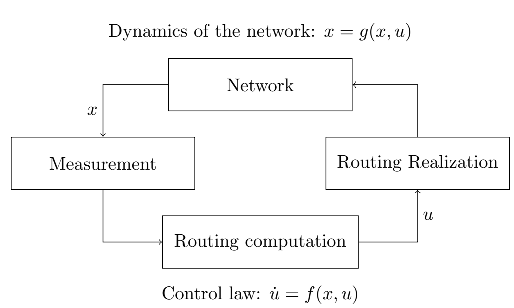


Fig. 1. Adaptive TE control diagram.

Suppose a network consists of a set of users (pairs of ingress-egress nodes) $\mathcal{I}$, and a set of links $\mathcal{L}$. Each link l has a capacity c_l. We assume the packets are routed based on source routing, meaning that the ingress router of a packet determines the specific path for the packet among all pre-defined paths. Each user $i \in \mathcal{I}$ has demand d_i to send from source to destination via a set of paths $\mathcal{P}$ where a subset $\mathcal{P}_i \subseteq \mathcal{P}$ consists of all feasible paths that belong to user i. Each path $k \in \mathcal{P}$ is composed by a set of directed links, denoted as $\mathcal{L}_k$. Suppose the number of paths and links in the network is K and L respectively. We further denote an $L \times K$ matrix $R_0 = (r_{lk})$ to represent the link composition of each path such that

$$r_{lk} = \begin{cases} 1, & \text{if } l \in \mathcal{L}_k, \\ 0, & \text{otherwise.} \end{cases}$$

For the traffic belonging to user i, the routing engine deployed in its ingress router decides on how much percentage of d_i is routed to each path $k \in \mathcal{P}_i$. The routing decisions are made according to the path metrics and the control law defined by the adaptive TE algorithm. Let the state vector be $x \in \mathbb{R}^{K \times 1}$ where the component x_k represents the path metric for path k. The control vector is denoted as $u \in \mathbb{R}^{K \times 1}$ where the component u_k is the split ratio for path k. The split ratios of paths for user i satisfy that $\sum_k u_k(t) = 1, \forall k \in \mathcal{P}_i$. The control law for adaptive TE follows some function $f(\cdot)$ such that

$$\dot{u}(t) = f\big(u(t), x(t)\big).$$

The path metrics are defined based on some performance objectives such as the sum of the utilization of all links on the path, the maximum link utilization on the path, or the total latency along the path, etc. The performance of a link is affected by the aggregated traffic from different user's paths that share the same link, which in turn determines the path metrics. Thus depending on the performance objective, the path metrics can be expressed as a function $g(\cdot)$ of the split ratios for each path:

$$x(t) = g\big(u(t)\big).$$

We consider a specific path metric that is defined as the maximum utilization of all the links along a path. In the delay-free case, the utilization of link l at time t, denoted as $w_l(t)$, is given by

$$w_l(t) = \frac{\sum_{j: l \in \mathcal{L}_j} u_j(t) d_i(t)}{c_l}, \quad i : j \in \mathcal{P}_i,$$

where user i is the owner of path j. So the metric for path k is given by

$$x_k(t) = \max_{l \in \mathcal{L}_k} \frac{w_l(t)}{c_l}.$$

We further consider a specific control law as follows

$$\dot{u}_k(t) = \alpha\bigg(\frac{\sum_{j \in \mathcal{P}_i} x_j(t)}{K_i} - x_k(t)\bigg), \forall k \in \mathcal{P}_i, \qquad (1)$$

where α is the step size as a tunable parameter, and K_i is the number of paths that belong to user i. We introduce the $K_i \times K_i$ symmetric matrix $M_i = (m^i_{j,k})$ for user i such that

$$m^i_{j,k} = \begin{cases} 1 - \frac{1}{K_i}, & \text{if } j = k, \\ -\frac{1}{K_i}, & \text{otherwise.} \end{cases}$$

To express the system in matrix form, let $P_0(t) = (p_{k,l}(t))$ be the $K \times L$ matrix of the path metric function, with $p_{k,l}(t) = 1$ if utilization of link l is the maximum along path k, and 0 otherwise. Furthermore, let $D_i(t) \in \mathbb{R}^{K_i \times K_i}$ be the $K_i \times K_i$ diagonal matrix of demand for user i's paths, i.e. $D_i(t) := diag(d_i(t))$, and $D(t) \in \mathbb{R}^{K \times K}$ be the $K \times K$ diagonal matrix of demand for all users' paths, i.e. $D(t) := diag(D_i(t))$. Finally, let $M := diag(M_i) \in \mathbb{R}^{K \times K}$, $C := diag(\frac{1}{c_l}) \in \mathbb{R}^{L \times L}$. Then the state space equations are in the following form:

$$\dot{u}(t) = -\alpha M x(t),$$
$$x(t) = P_0(t) C R_0 D(t) u(t).$$

To compute the path metrics, the ingress node of each user need to collect the information of link states by periodically sending the probing packets through all its paths. When the probing packet travels through each node along the path, the router will push the link states (the link utilization in our case) information in the probing packet. The destination router of the path then sends it back to the source router once receiving the probing packet. In real network, it takes a round-trip time (RTT) for the probing packet to come back, and thus we here incorporate delay into the model. Let δ^l_k denote the propagation delay from the source of path k to link l, and let β^l_k denote the backward delay from link l back to the source of path k for probing packets. The utilization of link l at time t is

$$w_l(t) = \frac{\sum_{j: l \in \mathcal{L}_j} u_j(t - \delta^l_j) d_i(t - \delta^l_j)}{c_l}. \qquad (2)$$

The path metric for any path $k \in \mathcal{P}$ at time t is determined by the feedback link utilization collected from each link, i.e.

$$x_k(t) = \max_{l \in \mathcal{L}_k} w_l(t - \beta^l_k), \quad \forall k \in \mathcal{P}. \qquad (3)$$

III. Analysis

In this section, we analyze the stability of the model, and specialize the general methodology in two representative network topologies [4], single-link and shared-link.

A. Stability analysis

One common feature for any adaptive TE method is that it can only be implemented in discrete time. This fact imposes an important factor into the practical system that the continuous-time model does not capture: the update interval of routing computation. In the following, we discretize the system and study the effect of step size, update interval and system delay, as well as their internal relations.

Let τ be the update interval of the routing computation that defines how frequently the routing strategy changes. We assume the link utilization values being collected at each router is measured during the period of τ. In Section III-B, we deeply investigate the internal relations of update interval τ and round-trip time T, and derive the stability condition for arbitrary value of τ/T in the case of single user with single bottleneck link. However, for the general network, we simplify the discrete-time model by assuming that $\beta_k^l = n_{\beta,k}^l \tau$ and $\delta_j^l = n_{\delta,j}^l \tau$, where $n_{\beta,k}^l$ and $n_{\delta,j}^l$ are integers. We assume the synchronization of all ingress nodes for each update at each time stamp $n\tau, n = 0, 1, 2, \cdots$. Furthermore, assuming that the demand matrix $D(t)$ is not changed within the time scale of our interest, i.e. $d_i(t) = d_i, \forall t$, we focus on the stability analysis in the steady state.

Discretizing Eq. 2 and Eq. 3, we have

$$w_l(n) = \frac{\sum_{j:l \in \mathcal{L}_j} u_j(n - n_{\delta,j}^l) d_i}{c_l}, \tag{4}$$

$$x_k(n+1) = \max_{l \in \mathcal{L}_k} w_l(n - n_{\beta,k}^l), \quad \forall k \in \mathcal{P}. \tag{5}$$

Taking the z transform of Eq. 4 and Eq. 5 yields

$$x_k(z) = \max_{l \in \mathcal{L}_k} \sum_{j:l \in \mathcal{L}_j} z^{-(n_{\delta,j}^l + n_{\beta,k}^l)} u_j(z) \frac{d_i}{c_l}, \quad i : j \in \mathcal{P}_i.$$

The control law follows from discretizing Eq. 1:

$$u_k(n) = u_k(n-1) + \alpha \left(\frac{\sum_{j \in \mathcal{P}_i} x_j(n)}{K_i} - x_k(n) \right), \forall k \in \mathcal{P}_i.$$

Thus the discrete-time model with feedback delays in frequency domain is given by

$$u(z) = z^{-1} u(z) - \alpha M x(z), \tag{6}$$

$$z x(z) - z x(0) - x(1) = P(z) C R(z) D u(z), \tag{7}$$

where the matrices $R(z)$ and $P(z)$ in frequency domain are written as:

$$(R(z))_{jl} := \begin{cases} z^{-n_{\delta,j}^l}, & \text{if the path } j \text{ uses link } l, \\ 0, & \text{otherwise,} \end{cases}$$

$$(P(z))_{kl} := \begin{cases} z^{-n_{\beta,k}^l}, & \text{if } l \text{ is the bottleneck link for path } k, \\ 0, & \text{otherwise.} \end{cases}$$

The stability condition can be developed based on basic control theory. The system is stable if the poles of the closed-loop transfer function have magnitude less than 1, i.e. the roots of $det\big((z-1)(I + G(z))\big) = 0$ have magnitude less than 1, where

$$G(z) = \frac{\alpha M P(z) C R(z) D}{z - 1}.$$

Intuitively, the larger step size α and the smaller update interval τ make the system more responsive. However, the stability condition is also dependent of the value of α, as well as the relation of τ and delays. As a result, it is critical to determine the region of the parameter choices under the constraint of the stability condition so that the stability is guaranteed when seeking for the best solutions of achieving responsiveness.

B. Case study I: single link

In the first case study, we consider a topology consisting of a single pair of ingress-egress nodes with two paths as shown in Figure 2. The demand from ingress node to egress node is D. We assume all links are identical with the same capacity C and the same forwarding and backward delays. Suppose the link between the host and edge node is also restricted by capacity C, then we have $D \in [0, C]$. We further assume initially all traffic are routed through the upper path $l_1 \rightarrow l_2$, which we denote as path 1 and thus path 2 is the lower path. Either l_1 or l_2 can be viewed as the bottleneck link. In our following discussion, l_2 is assumed as the bottleneck link before the load on the two paths are balanced.

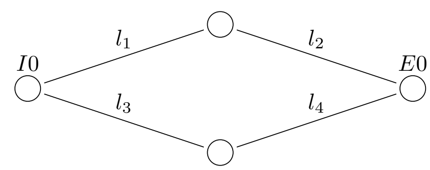


Fig. 2. Single user with two paths.

1) Delay-free: In the ideal case, the propagation delay and backward delay are negligible. In the discrete-time model, we still use α for the step size, while it is worthwhile noting that the step size α in continuous-time model is equivalent to $\alpha\tau$ when we discretize the system. Considering the fact that $u_1(n) + u_2(n) = 1$, the system can be simplified as

$$x(n+1) = \frac{D(n) u(n)}{C},$$

$$u(n) = u(n-1) + \alpha\left(\frac{D(n)}{2C} - x(n)\right),$$

where we let $x(n)$ and $u(n)$ be the state and control variable for path 1. For a given demand $D \in [0, C]$, the state evolution is given by

$$x(n+1) = \left(1 - \alpha \frac{D}{C}\right) x(n) + \frac{\alpha}{2} \left(\frac{D}{C}\right)^2.$$

To ensure that the closed-loop system is stable, the following condition need to hold:

$$\left|1 - \alpha \frac{D}{C}\right| < 1, \forall \frac{D}{C} \in [0, 1].$$

When $0 < \alpha < 2$, the system can be stablized and will converge to the state $\bar{x} = D/2C, \quad \bar{u} = 1/2$. In the single link case, when the delay is negligible, the stability constraint for the system is independent of any parameters other than the step size.

2) With delay: We now consider the delay of the closed-loop system. When the link propagation delay is not negligible, one significant variation from the ideal model is the fact that once the control decisions (i.e. the new split ratios) are updated at the source node, the state variables (the link utilization) in the downstream nodes do not change with the new update instantaneously. In other words, the link utilization measured by the nodes at each step within the measurement period τ may be affected by both the current and historical control variables. This fact is not reflected by the previous analysis of delay-free case. Furthermore, since any adaptive TE solution is implemented in discrete time, the effect of propagation delay on the system also depends on the update interval.

Let δ be the identical propagation delay from ingress node to link l_2. Let β be the identical backward delay that takes for the ingress node to receive the latest state information from link l_2. The instantaneous link utilization at time t for link l_1 and l_2 are given by

$$w_1(t) = \frac{u(t)D(t)}{C},$$
$$w_2(t) = w_1(t-\delta) = \frac{u(t-\delta)D(t-\delta)}{C}.$$

Since we assume the measurement time interval for the link utilization is τ, the actual link utilization of l_2 that is fed back to the ingress node is the averaged value $\bar{w}_2(t)$ over the past time period τ, i.e.

$$\bar{w}_2(t) = \frac{1}{\tau}\int_{t-\tau}^{t} w_2(\eta)d\eta = \frac{1}{\tau}\int_{t-\tau-\delta}^{t-\delta} u(\eta)\frac{D(\eta)}{C}d\eta.$$

To implement the control law in discrete time, we sample the state variable and compute the control law every τ, starting from $t = 0$. Once the control law is updated, the control variable $u(t)$ keeps unchanged within time interval τ until the next update is triggered, i.e.

$$u(t) = u(n\tau), \quad n = \lfloor\frac{t}{\tau}\rfloor.$$

As the bottleneck link, l_2's actual measured link utilization $\bar{w}_2$ will be the path metric. Since it takes β for the feedback message to travel back to the routing engine, the path metric computed at time $n\tau$ is $\bar{w}_2(n\tau-\beta)$, $n = 0, 1, 2, \cdots$, which is given by

$$\begin{aligned} x(n\tau) &= \bar{w}_2(n\tau-\beta) \\ &= \frac{1}{\tau}\int_{(n-1)\tau-\delta-\beta}^{n\tau-\delta-\beta} u(\eta)\frac{D(\eta)}{C}d\eta \\ &= \frac{1}{\tau}u\big((n-2-\lfloor\frac{T}{\tau}\rfloor)\tau\big)\int_{(n-1)\tau-T}^{(n-1)\tau-\lfloor\frac{T}{\tau}\rfloor\tau}\frac{D(\eta)}{C}d\eta \\ &+ \frac{1}{\tau}u\big((n-1-\lfloor\frac{T}{\tau}\rfloor)\tau\big)\int_{(n-1)\tau-\lfloor\frac{T}{\tau}\rfloor\tau}^{n\tau-T}\frac{D(\eta)}{C}d\eta, \end{aligned}$$

where $T = \delta + \beta$ is the RTT of the path.

In the steady state where $D(t)$ is a constant value D, we derive the discrete time state space equations with feedback delays and update interval:

$$\begin{aligned} x(n) = \frac{D}{C}[&(k-\lfloor k\rfloor)u(n-2-\lfloor k\rfloor) \\ &+ (1+\lfloor k\rfloor-k)u(n-1-\lfloor k\rfloor)], \end{aligned} \tag{8}$$

$$u(n) = u(n-1) + \alpha(\frac{D}{2C} - x(n)), \quad k = \frac{T}{\tau}. \tag{9}$$

Figure 3a shows the block diagram of the system, where we define

$$H(z) = \frac{\alpha z}{z-1},$$
$$G(z) = \frac{D}{C}\left(\frac{k-\lfloor k\rfloor}{z^{2+\lfloor k\rfloor}} + \frac{1+\lfloor k\rfloor-k}{z^{1+\lfloor k\rfloor}}\right), R = \frac{D}{2C}.$$

The transfer function of the closed-loop system follows that

$$Q(z) = \frac{H(z)G(z)}{1+H(z)G(z)}.$$

To ensure that the system is stable, all the roots of the following equation should have magnitude less than 1:

$$z^{2+\lfloor k\rfloor} - z^{1+\lfloor k\rfloor} + \alpha\frac{D}{C}(1+\lfloor k\rfloor-k)z + \alpha\frac{D}{C}(k-\lfloor k\rfloor) = 0.$$

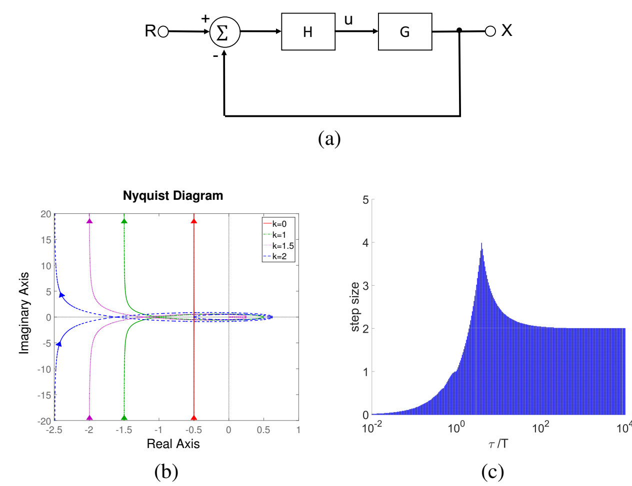


Fig. 3. (a) Feedback control block diagram. (b) Nyquist plot of $G(z)H(z)$ for different τ/T. (c) Stability region of step size when τ/T varies.

Figure 3b shows the Nyquist plot of $G(z)H(z)$ for some values of τ/T. Figure 3c shows the range of α as the value of τ/T varies when $D/C \in [0, 1]$. One way to understand why the stability region of step size changes in this form is to consider the effect of averaging state variables with different levels of historical states. When τ is much larger than T meaning that propagation delay is negligible, $x(n)$ depends on $u(n-1)$ only, so the region of α is independent of τ. This is also consistent with the result we have derived in the ideal delay-free case. With a certain level of averaging, the state variables are smoothed, thus enlarging the range of α. When $\tau = T$, the state variables obtained are out dated relying completely on historical data $u(n-2)$. As T becomes much larger than τ, i.e. τ/T becomes much smaller than 1, the system relies on older data. Therefore, it is more likely to oscillate and the stability region for α is narrower.

C. Case study II: shared links

In the following section, we study the case in which links are shared by multiple users. Consider the network topology shown in Figure 4, there are 4 users, namely $I1-E1$, $I1-E2$, $I2-E1$, $I2-E2$. Each user has two paths. The bottleneck links are l_5 and l_6 which are the common links

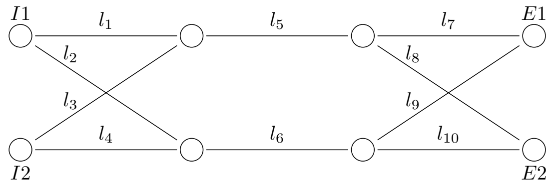


Fig. 4. Multiple users with shared links.

for four paths. $x(n)$ and $u(n)$ are 8×1 vectors. The state space equations of the system are described in Eq. 6 and Eq. 7, where the matrix M and $P(z)$ are specifically in the following forms:

$$M = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$P(z) = \begin{bmatrix} 0 & 0 & 0 & 0 & z^{-n^5_{\beta,1}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & z^{-n^6_{\beta,2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & z^{-n^5_{\beta,3}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-n^6_{\beta,4}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & z^{-n^5_{\beta,5}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-n^6_{\beta,6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & z^{-n^5_{\beta,7}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & z^{-n^6_{\beta,8}} & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The 1th to 4th column of $R(z)$, as an example due to the space limitation, is

$$R_{j,1-4}(z) = \begin{bmatrix} z^{-n^1_{\delta,1}} & 0 & z^{-n^1_{\delta,3}} & 0 \\ 0 & z^{-n^2_{\delta,2}} & 0 & z^{-n^2_{\delta,4}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ z^{-n^5_{\delta,1}} & 0 & z^{-n^5_{\delta,3}} & 0 \\ 0 & z^{-n^6_{\delta,2}} & 0 & z^{-n^6_{\delta,4}} \\ z^{-n^7_{\delta,1}} & 0 & 0 & 0 \\ 0 & 0 & z^{-n^8_{\delta,3}} & 0 \\ 0 & z^{-n^9_{\delta,2}} & 0 & 0 \\ 0 & 0 & 0 & z^{-n^{10}_{\delta,4}} \end{bmatrix}.$$

According to the multivariable Nyquist criterion [11], the stability condition is equivalent to the following statement: the eigenvalues of $G(e^{j\omega})$, for ω from 0 to 2π, should not encircle the point -1. Let $\lambda\big(G(e^{j\omega})\big)$ denote any eigenvalue of $G(e^{j\omega})$. Suppose the RTT for all paths are identical so that $n^l_{\delta,j} + n^l_{\beta,k} = n_T$ for all users sharing link l, then the eigenvalues of $G(e^{j\omega})$ is given by

$$\lambda\big(G(e^{j\omega})\big) = \lambda\big(\alpha M P_0 C R_0 D\big)\frac{e^{-j\omega n_T}}{e^{j\omega}-1} = \alpha\lambda(H)\frac{e^{-j\omega n_T}}{e^{j\omega}-1},$$

where $H = MP_0CR_0D$. H has all eigenvalues of real number. Let $||A||_\infty$ denote the matrix ∞- norm $||A||_\infty = \max_i \sum_j |A_{ij}|$ which is the maximum row sum. The magnitude of any eigenvalue of H is upper-bounded by

$$|\lambda(H)| \le ||H||_\infty \le ||M||_\infty ||P_0CR_0||_\infty ||D||_\infty. \quad (10)$$

Note that

$$||M||_\infty = 2 - \frac{2}{K^{\max}}, \quad ||D||_\infty = d^{\max},$$

$$||P_0CR_0||_\infty \le \frac{N^{\max}}{c^{\min}},$$

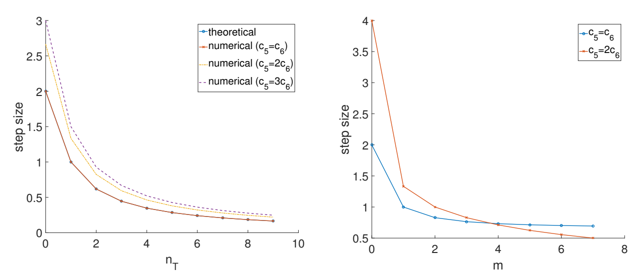
where $K^{\max} \ge 2$ and $d^{\max}$ are the maximum values of K_i and d_i among all users i, $N^{\max} \ge 1$ is the maximum number of users sharing a link, and $c^{\min}$ is the minimum link capacity.

Furthermore, when $Im\{\frac{e^{-j\omega n_T}}{e^{j\omega}-1}\} = 0$, $Re\{\frac{e^{-j\omega n_T}}{e^{j\omega}-1}\}$ is lower bounded by

$$Re\{\frac{e^{-j\omega n_T}}{e^{j\omega}-1}\} \ge \left(\frac{e^{-j\omega n_T}}{e^{j\omega}-1}\right)_{\omega=\pi/(2n_T+1)} = \frac{\sin\frac{\omega}{2}}{\cos\omega - 1} = -\frac{1}{2\sin\frac{\omega}{2}} = -\frac{1}{2\sin\frac{\pi}{4n_T+2}}.$$

Therefore the sufficient condition for the closed-loop system stability with homogeneous RTT is given by

$$\alpha < \frac{\sin\frac{\pi}{4n_T+2}K^{\max}c^{\min}}{(K^{\max}-1)d^{\max}N^{\max}}.$$



(a) (b)

Fig. 5. (a) The upper bound of step size for homogeneous RTT as n_T varies. (b) The upper bound of step size for heterogeneous RTT. $m = \frac{n_{\delta,5}}{n_{\delta,6}}$.

Figure 5a shows the theoretical upper bound of α derived from the sufficient condition above in homogeneous RTT case as n_T varies from 0 to 9, and the numerical results computed for the necessary condition of stability. In the theoretical computation, we set $c^{\min} = 20Mbps$, $d^{\max} =$

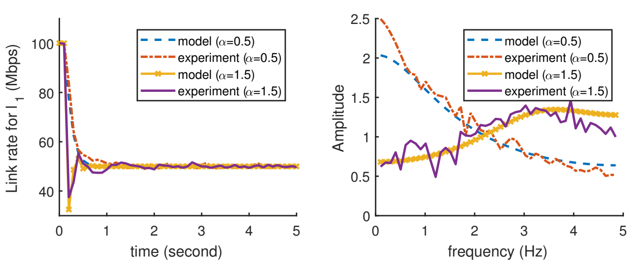
$5Mbps$, $N^{\max} = 4$ and $K^{\max} = 2$. In the numerical computation of homogeneous RTT case, we set the demand for all users to be $5Mbps$, all links capacity to be $20Mpbs$ except c_5. When $c_5 = c_6$, the numerical results are exactly the same as our theoretical derivation because $|\lambda(H)| = \frac{2(K^{\max}-1)d^{\max}N^{\max}}{K^{\max}c^{\min}}$. When $c_5 > c_6 = 20Mbps$, as is shown in the case of $c_5 = 2c_6$ and $c_5 = 3c_6$, $c^{\min} = 20Mbps$ still holds, but $|\lambda(H)|$ becomes strictly less than its upper bound. So in these two cases we can see α's upper bound derived from sufficient condition is less than the numerical computation for necessary condition.

We further show the step size upper bound for some heterogeneous RTT cases in Figure 5b. In the computation, we ignore all the backward delay and make the forwarding propagation delay identical for all the links from any path using this link, i.e. $n^p_{\delta,l} = n_{\delta,l}, \forall p \in \mathcal{L}_l$. We set $n_{\delta,6} = 1$ and vary $n_{\delta,5}$ to obtain the upper bound of α as the propagation delay difference between two bottleneck links changes. Comparing with Figure 5a, the theoretical result for sufficient condition will also hold for heterogeneous RTT cases with n_T being the maximum n_T, while the sufficient condition of stability for heterogeneous RTT remains to be further studied.

IV. Experiments

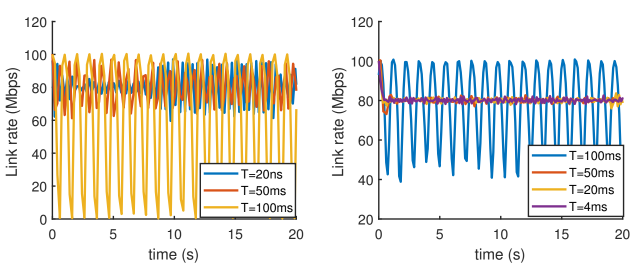
In this section, we describe our emulation setup, and demonstrate the validation results for our model and analysis.

A. Validating our model



(a) (b)

Fig. 6. Model validation. (a) Link rate for l_1 in time domain. (b) Amplitude spectrum of link rate for l_1. $\tau = 100ms$, $T = 10ms$.



(a) $\alpha = 1.3, \tau = 200ms$ (b) $\alpha = 1, \tau = 100ms$

Fig. 7. Validation of stability region for single-link.

A four-node topology, as is shown in Figure 2, is set up on Mininet for model validation. Each node is associated with an Open vSwitch [12]. The two paths are pre-configured in the network so that the switches forward the packets accordingly. A customized controller with our control law 1 is deployed on the ingress node. The routing computation is triggered every τ and then the updated routing rules will be installed to the ingress switch. The ingress switch uses group table to realize the multi-path routing with unequal weights. The switch in each node measures the link throughput within a measurement period τ. A probing packet is created and sent along each path to collect the link states every τ. We set $C = 100Mbps$ for all links.

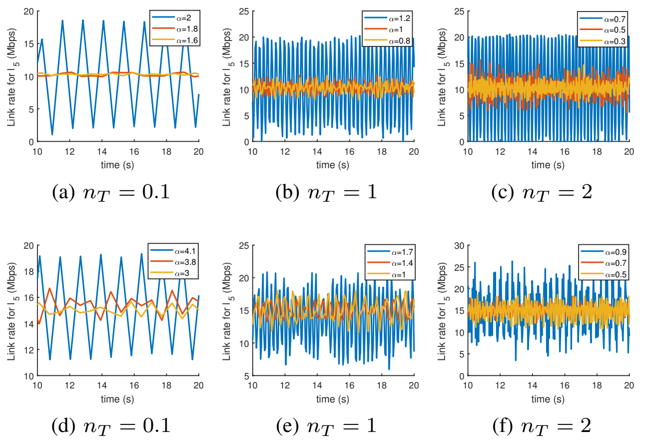
In the model validation experiments, $100\,\mathrm{Mbps}$ traffic destined to E_0 are generated at I_0 via iPerf. The initial routing strategy is configured to be forwarding all traffic through the upper path. Adaptive TE implemented in the controller collects the statistics of path metrics and dynamically adjusts the split ratio for each path. The actual evolution of transmission rate on link l_1 shown in Figure 6 validates the predictions from our model. With different values of the step size,the experimental curves for the measured link rate of l_1 in both time domain and frequency domain show good agreement with the ideal curve predicted by the model. On the experimental curve, the noise effect produced by the measurement can also be observed.

B. Validating our analysis

Our analysis is validated with respect to single-link and shared-link topologies. Two representative adaptive TE control laws are considered in the experiments: the general form specified in Eq. 1 and a particular routing strategy proposed by TeXCP [5].

On the testbed with four-node topology, we introduce $60Mbps$ background traffic on link l_4, and generate $100\,\mathrm{Mbps}$ traffic for user $I_0 - E_0$. Therefore, the split ratios for user $I_0 - E_0$ are expected to be stabilized around 0.8 and 0.2 for the upper path and the lower path respectively. To judge whether the system is stabilized, we record the measured transmission rate of link l_1. Since our analysis in Section III-B indicates that the stability is dependent of the step size, feedback delay and update interval, experiments are carried out with different values of the three parameters. The results in Figure 7a and Figure 7b show the agreement on the stability region of α with the theoretical results for single-link in Figure 3c. When $\alpha = 1.3$, the throughput of l_1 is stable in the case of $T = 20ns$. The oscillation is larger for $\tau/T = 2$ than $\tau/T = 4$. In Figure 7b, $\alpha = 1$ is not within the stability region of $\tau/T = 1$ ($T = 100\,\mathrm{ms}$). In both of the experiments, we fix the value of τ which is large enough to maintain the effect measurement noise input small and identical when T varies.

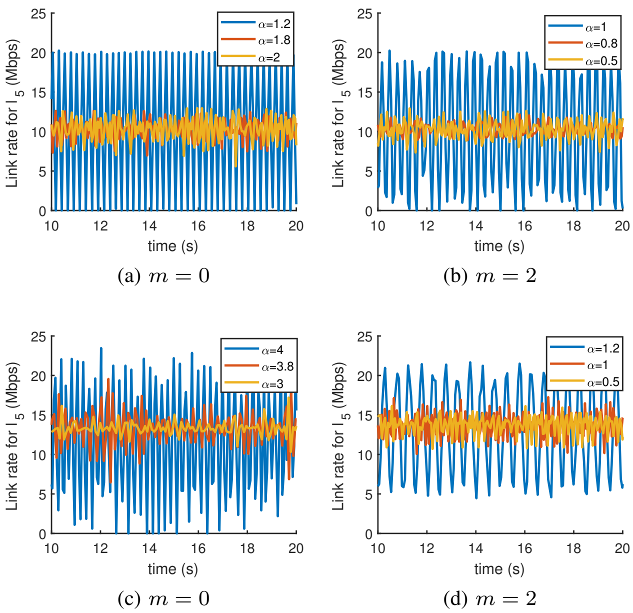
We proceed to validate our analysis for shared-link case with multi-user (Section III-C) by setting up an eight-node topology shown in Figure 4 on the same testbed. A centralized controller is implemented which controls the routing update for all users. Each user starts a session of



(a) $n_T = 0.1$ (b) $n_T = 1$ (c) $n_T = 2$

(d) $n_T = 0.1$ (e) $n_T = 1$ (f) $n_T = 2$

Fig. 8. Validation of stability region for shared-link with identical RTT $= 60\,\text{ms}$. $c_5 = c_6 = 20\,\text{Mbps}$ in (a) (b) and (c); $c_5 = 3c_6 = 60\,\text{Mbps}$ in (d) (e) and (f).



(a) $m = 0$ (b) $m = 2$

(c) $m = 0$ (d) $m = 2$

Fig. 9. Validation of stability region for shared-link with heterogeneous RTT. $c_5 = c_6 = 20\,\text{Mbps}$ in (a) and (b); $c_5 = 2c_6 = 40\,\text{Mbps}$ in (c) and (d).

$5\,\text{Mbps}$ traffic transfer to its destination asynchronously. The capacity for each link except l_5 is fixed at $20\,\text{Mbps}$. In the case of homogenous RTT, the RTT for each path is $60\,\text{ms}$. To justify the stability region in Figure 5a with different n_T, we consider three values of the update interval τ ($600\,\text{ms}$, $60\,\text{ms}$ and $30\,\text{ms}$) and two values of link capacity c_5 ($20\,\text{Mbps}$ and $60\,\text{Mbps}$). Figure 8 shows in each case, when the transmission rate of l_5 exhibits periodic oscillation, the corresponding value of the step size is outside the stability region predicted in Figure 5a. For the experiments with heterogenous RTT, the paths using link l_6 have RTT $= 60\,\text{ms}$, and the paths using link l_5 have RTT of either 0 or $120\,\text{ms}$. The update interval is configured to be $\tau = 60\,\text{ms}$ so that m is either 0 or 2 with $n_{\delta,6} = 1$. We consider two values of m (0 and 2) and two values of c_5 ($20\,\text{Mbps}$ and $40\,\text{Mbps}$). Figure 9 provides the validation for the stability region in Figure 5b.

We finally validate the stability region analysis of TeXCP [5] using our proposed methodology. TeXCP mainly consists of two layers of control functions: the outer layer is the adaptive TE update, and inside there is a feedback congestion control loop operating at least 5 times within one TE update. We simplify the process by ignoring the inner loop of congestion control and only consider the feedback loop of TE updates. Since the congestion control strategy guarantees the state variables $x(n)$ obtained have been stabilized to at least 95% of the latest control variables $u(n)$, we wait for long enough time to make sure the transmission rates have become stable. Therefore, we set $\tau = 5T$ in all of the following experiments. According to the method of routing computation described in TeXCP, its TE control law for path $k \in \mathcal{P}_i$ is in the following form

$$u_k(n) = u_k(n-1) + u_k(n-1)\Big[\sum_{j\in\mathcal{P}_i} u_j(n-1)x_j(n) - x_k(n)\Big].$$

In the single-link case, the routing rule can be written as

$$u(n) = u(n-1) + 2u(n-1)\big(1 - u(n-1)\big)\Big(\frac{D}{2C} - x(n)\Big).$$

Compared with Eq. 9, the step size of TeXCP follows that $\alpha = 2u(n)(1-u(n))$, which is a varying number in the range of $[0, 0.5]$ because $u(n) \in [0, 1]$. Based on the stability region we derived in Figure 3c, the range of step size $[0, 0.5]$ will guarantee the stability regardless of the feedback delay and update interval, as is shown in Figure 10a. By imposing an additional parameter a on the TE control law such that

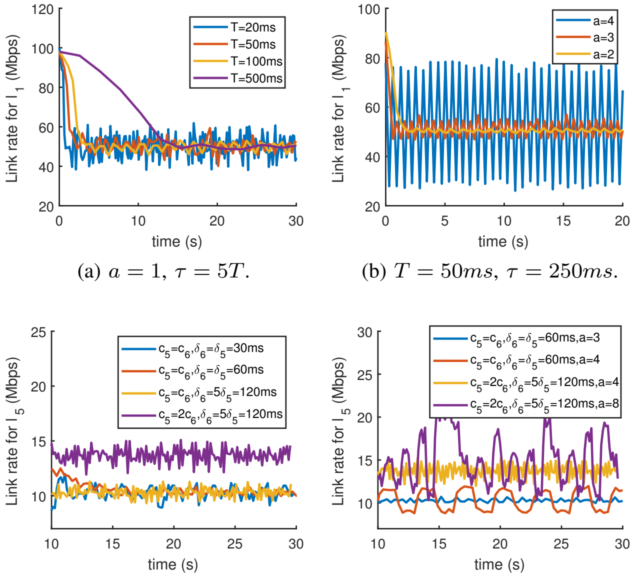
$$u(n) = u(n-1) + a \cdot 2u(n-1)\big(1 - u(n-1)\big)\Big(\frac{D}{2C} - x(n)\Big),$$

we can investigate the impact of step size on stability. In this case, the step size is written as $\alpha = a \cdot 2u(n)(1-u(n))$ and it is easy to compute its boundary $\alpha \in [0, 0.5a]$. Note that with the parameter a adding in the control law, the original design in TeXCP is a special case where $a = 1$. In Figure 10b, when $a = 4$ indicating that the step size could be as large as 2, the system becomes unstable. It is consistent with our analysis in Figure 3c that when $\tau/T = 5$, the upper bound of step size is less than 2.

In the shared-link case, the specific control law for the given network topology in Figure 4 follows that

$$u(n) = u(n-1) - 2U(n-1)\big(I - U(n-1)\big)Mx(n),$$

where $U(n) := diag(u_k(n)) \in \mathbb{R}^{K\times K}$. Note that the step size in the control law is considered as a scalar in Eq. 6, whereas TeXCP attaches a specific step size $\alpha_k(n) = 2u_k(n)\big(1 - u_k(n)\big)$ to each control variable u_k. We run a set of experiments on the testbed with 8-node topology, including the cases of homogeneous/heterogenous RTT and link capacities. δ_5 and δ_6 are defined as the RTT of the paths that use link l_5 and l_6 respectively. The update interval remains to be $\tau = 5\delta_5$, so that $n_T = 0.2$ holds in homogeneous RTT case, while in heterogenous RTT case



(a) $a = 1$, $\tau = 5T$. (b) $T = 50ms$, $\tau = 250ms$.

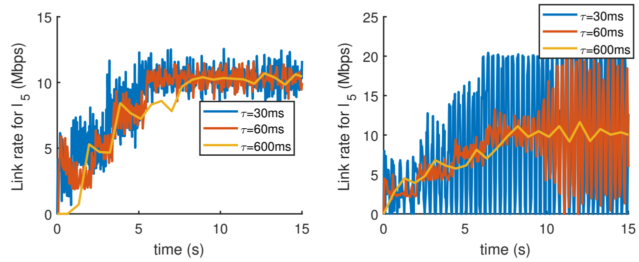
(c) $a = 1$, $\tau = 5\delta_5$. (d) $\tau = 5\delta_5$.

Fig. 10. Validation of stability region for TeXCP. (a) (b) Single-link case; (c) (d) Shared-link case.

$n_{\delta,5} = 0.2, n_{\delta,6} = 1$. Figure 10c demonstrates that the original TeXCP control law ensures the routing update to be stabilized in all cases. We again impose the parameter a on the original control law and experimentally seek for the new stability region by tuning the value of a. Figure 10d shows that with homogeneous RTT, $a = 4$ makes the system unstable, which validates the upper bound of step size for $n_T = 0.2$ in Figure 5a. Furthermore, it also provides justification of the stability region for heterogeneous RTT in Figure 5b when $m = 0.2$ and $c_5 = 2c_6$.

C. Stability versus responsiveness

Our comprehensive experiments above have shown that the critical system parameters and their internal relations play significant roles on the system stability. Figure 10 for single-link case and Figure 11 for shared-link case are good examples of illustrating how the choices of step size α, update interval τ and feedback delay T affect the performance tradeoff between stability and responsiveness. Observation from Figure 10a indicates that the system is more responsive when the feedback delay is smaller. Figure 10b shows that a larger step size leads to faster convergence to the set point, however, too large a step size will cause the routing oscillation. In Figure 11a, more frequent update can make the system respond and converge more quickly, while it turns out in Figure 11b that smaller update interval may result in a smaller stability region for choices of the step size. Our quantitative analysis for the model helps to restrict the boundary of the parameters based on stability condition, and thus facilitates the systematic evaluation of parameters' effects on the performance.



(a) $\alpha = 0.5$ (b) $\alpha = 1.2$

Fig. 11. Tradeoff between stability and responsiveness.

V. CONCLUSIONS AND FUTURE WORKS

A control theoretical model is proposed for fast adaptive traffic engineering. The model incorporates several important factors including physical propagation delay, routing update interval and step size. In particular, it formalizes the intrinsic tradeoff between being reactive, hence tending to use large step size and frequent sampling, and maintaining stability, which benefits from adopting more cautious steps as well as larger measurement duration to better filter out the high frequency noise. We analyze the model to provide guiding insight for network operators to set those important parameters. Furthermore, experiments are carried out to quantitatively verify the predictions from the model. Our model covers related existing work as special cases and therefore can serve as a basis for further combination with different traffic models.

REFERENCES

[1] McKeown, Nick. "Software-defined networking." *INFOCOM keynote talk* 17.2 (2009): 30-32.
[2] Agarwal, Sugam, et al. "Traffic engineering in software defined networks." *INFOCOM*. IEEE, 2013.
[3] Akyildiz, Ian F, et al. "A roadmap for traffic engineering in SDN-OpenFlow networks." *Computer Networks* 71 (2014): 1-30.
[4] Elwalid, Anwar, et al. "MATE: MPLS adaptive traffic engineering." *INFOCOM*. IEEE, 2001.
[5] Kandula, Srikanth, et al. "Walking the tightrope: Responsive yet stable traffic engineering." *ACM SIGCOMM Computer Communication Review*. ACM, 2005.
[6] Benson, Theophilus, et al. "MicroTE: Fine grained traffic engineering for data centers." *CoNEXT*. ACM, 2011.
[7] Jain, Sushant, et al. "B4: Experience with a globally-deployed software defined WAN." *ACM SIGCOMM Computer Communication Review*. ACM, 2013.
[8] Alizadeh, Mohammad, et al. "CONGA: Distributed congestion-aware load balancing for datacenters." *ACM SIGCOMM Computer Communication Review*. ACM, 2014.
[9] Michael, Nithin, and Ao Tang. "Halo: Hop-by-hop adaptive link-state optimal routing." *IEEE/ACM Transactions on Networking (TON)* 23.6 (2015): 1862-1875.
[10] Lantz, Bob, Brandon Heller, and Nick McKeown. "A network in a laptop: rapid prototyping for software-defined networks." *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010.
[11] Desoer, C., and Yung-Terng Wang. "On the generalized Nyquist stability criterion." *IEEE Transactions on Automatic Control* 25.2 (1980): 187-196.
[12] Open vSwitch – An Open Virtual Switch. *http://www.openvswitch.org*, 2014.