# Energy-Efficient Congestion Control

Lingwen Gan
EE, California Institute of Technology
lgan@caltech.edu

Anwar Walid
Alcatel-Lucent Bell Labs
anwar@research.bell-labs.com

Steven H. Low
CMS & EE, California Institute of Technology
slow@caltech.edu

## ABSTRACT

Various link bandwidth adjustment mechanisms are being developed to save network energy. However, their interaction with congestion control can significantly reduce throughput, and is not well understood. We firstly put forward an easily implementable link dynamic bandwidth adjustment (DBA) mechanism, and then study its iteration with congestion control. In DBA, each link updates its bandwidth according to an integral control law to match its average buffer size with a target buffer size. We prove that DBA reduces link bandwidth without sacrificing throughput—DBA only turns off excess bandwidth—in the presence of congestion control. Preliminary ns2 simulations confirm this result.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

## Keywords

bandwidth adjustment, congestion control, stability.

## 1. INTRODUCTION

Energy efficiency is becoming one of the main design criterias in the information and communication industry, from hardware design (devices to systems) to software configuration (e.g., virtualization), and to service deployment. While significant advances in semiconductor technology have led to decrease in power consumption per byte transmission, the power efficiency is starting to plateau and the demand for greater bandwidth is leading to an overall increase in energy consumption [10]. The level of energy consumption in information and communication industry is growing exponentially at an annual rate of 15%. This trend could lead to significant cost increase to network operators [10, 13]. Therefore, there is a need to reduce energy consumption through other power management technologies. In fact, there have been ongoing efforts to develop mechanisms to reduce network energy consumption by opportunistically turning off or slowing down network devices during periods of low traffic loads. Sleep-mode operations have been proposed at various network layers, and dynamic resource adjustment is being facilitated by voltage and frequency scaling techniques [12].

Today's core networks have large link bandwidths with low utilization levels (30-40%) [11], providing opportunities for turning off excess bandwidth and saving energy, especially when traffic loads follow diurnal pattern, and link utilization level is low during off-peak hours. Turning off excess bandwidth may be achieved by turning off one or more of the parallel links between routers, or slowing down a particular link. With current technology, power consumption in backbone routers and their line cards is essentially load-independent and therefore selective application of sleep mode is more feasible during periods of low utilization—[11, 13]. In core networks, pairs of routers are typically connected by multiple physical links, called a link bundle, and the bundle is used as a logical link for routing purposes. Link bundling, standardized in IEEE 802.1AX [11], facilitates incremental bandwidth upgrade and prolongs the lifetime of a router. Furthermore, individual link components of a bundle may be selectively turned off to provide graceful bandwidth adjustment and energy savings during low traffic loads without affecting the topology. In the future, application of frequency and voltage scaling could facilitate fine-grain adjustment of the bandwidth of each individual link component.

TCP is the dominant transport layer protocol on the Internet, and all energy saving mechanisms in the network can interact with TCP in intricate ways. Link bandwidth adjustment can lead to serious TCP throughput reduction and instability if it is not done in concert with the dynamics of TCP and active queue management, which is often represented by random early discard (RED). The objective of the paper is to study energy-saving bandwidth adjustment mechanisms in the presence of TCP/RED.

The contribution of this paper are mainly three folds. First, we propose a dynamic bandwidth adjustment (DBA) mechanism that updates the bandwidth with local buffer information, thus easy to implement. Each link uses an integral control law to match its average buffer size with some moving target buffer size. The target buffer size is a function of the bandwidth, and corresponds to a chosen queueing delay. Second, we prove the following results regarding the joint dynamic of TCP/RED/DBA.

- TCP/RED/DBA is globally stable if network delay is negligible. At steady state, excess bandwidth is turned off, and throughput is preserved.
- The steady-state TCP source rates and link bandwidths solve an optimization problem, which we call the network surplus maximization problem.
- TCP/RED/DBA is locally stable if network delay is bounded and link dynamics are sufficiently slow.

Third, we evaluate the performance of DBA through ns2 simulations, and confirm that DBA turns off excess bandwidth without reducing throughput.

The rest of the paper is structured as follows. In section

*SIGMETRICS'12,* June 11–15, 2012, London, England, UK.
Copyright 2012 ACM 978-1-4503-1097-0/12/06...$10.00.

2, we introduce some preliminaries including the network model and the utility maximization framework. In section 3, we propose a dynamic bandwidth adjustment (DBA) mechanism, and give its implementations. We analyze the steady state and transient stability of TCP/RED/DBA in section 4, and provide ns2 simulations in section 5.

# 2. PRELIMINARIES

Consider a network that consists of a set $\mathcal{L} := \{1, \ldots, L\}$ of unidirectional links and a set $\mathcal{N} := \{1, \ldots, N\}$ of sources. For each link $l \in \mathcal{L}$, let $\mathcal{N}_l$ denote the set of sources that traverse it. For each source $i \in \mathcal{N}$, let $\mathcal{L}_i$ denote the set of links on its path to the destination. Then $i \in \mathcal{N}_l$ is equivalent to $l \in \mathcal{L}_i$ for any $i \in \mathcal{N}$ and any $l \in \mathcal{L}$. We make the following assumptions on the links and sources.

**Assumption** 1. *Each link $l \in \mathcal{L}$ can set its bandwidth $c_l$ to any value in the range $\mathcal{C}_l := [c_l^{min}, c_l^{max}]$.*

Link bandwidth may be adjusted by assuming that the link is in fact a link bundle, as described in the introduction, and an individual link component may be put to sleep or turned off to control the total bandwidth of the bundle. Alternatively, the bandwidth of a link may be adjusted with voltage and frequency scaling technology. We assume continuous bandwidth levels in the theoretical framework, and discuss discrete bandwidth levels in the implementation in section 3.2, and in the simulations in section 5.

**Assumption** 2. *Each source $i \in \mathcal{N}$ can set its rate $x_i$ to any value in the range $\mathcal{X}_i := [0, x_i^{max}]$.*

Each source $i$ is associated with a transmission window $w_i$, and transmits roughly $w_i$ unacknowledged bits at any given time. Let $\tau_i$ denote the round-trip time (RTT) for source $i$, then source rate $x_i$ is roughly given by $x_i = w_i/\tau_i$. Though TCP only controls $x_i$ through adjusting $w_i$, we assume that TCP can control $x_i$ explicitly as in [1, 2].

Define routing entries

$$R_{li} := \begin{cases} 1 & \text{if } i \in \mathcal{N}_l \\ 0 & \text{otherwise} \end{cases}, \; l \in \mathcal{L}, \; i \in \mathcal{N},$$

then the traffic $y_l$ on link $l$ is

$$y_l = \sum_{i \in \mathcal{N}_l} x_i = \sum_{i \in \mathcal{N}} R_{li} x_i, \; l \in \mathcal{L}.$$

We use lower case letter without subscript to denote a column vector of the corresponding quantity. For example, $x$ denotes $(x_1, \ldots, x_N)^T$ and $c$ denotes $(c_1, \ldots, c_L)^T$.

**Table 1: List of Symbols**

| | |
|---|---|
| $i$ | a source, $i \in \mathcal{N} = \{1, \ldots, N\}$ |
| $l$ | a link, $l \in \mathcal{L} = \{1, \ldots, L\}$ |
| $c_l$ | capacity of link $l$ |
| $x_i$ | rate of source $i$ |
| $R_{li}$ | routine entry, $R_{li} = 1$ if source $i$ passes link $l$, and 0 otherwise |
| $y_l$ | traffic on link $l$ |
| $b_l$ | average buffer size on link $l$ |
| $p_l$ | packet drop probability on link $l$ |
| $q_i$ | end-to-end packet loss probability for source $i$ |
| $U_i$ | utility function of source $i$ |
| $\tilde{b}_l(c_l)$ | target buffer size of link $l$ |
| $d_l$ | target queueing delay of link $l$ |
| $\tilde{p}_l(c_l)$ | packet drop probability corresponding to $\tilde{b}_l$ |
| $V_l$ | cost function of link $l$ |

## 2.1 Traditional Congestion Control

There are two types of congestion control: loss-based congestion control and delay-based congestion control, and we focus on loss-based congestion control in this paper. In loss-based congestion control, links drop packets to reflect their buffer occupancy, and sources adjust their transmission rates to respond to their end-to-end packet loss.

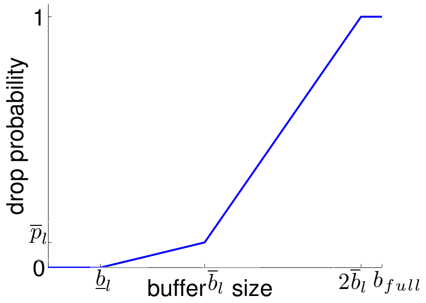

**Figure 1: Packet drop probability for RED.**

We consider active queue management link control, in particular the random early discard (RED) mechanism. In RED, a link $l$ keeps track of its average buffer size $b_l$,[1] and drops packets with probability (see Figure 1)

$$p_l = \begin{cases} 0 & \text{if } b_l \leq \underline{b}_l \\ \gamma_l(b_l - \underline{b}_l) & \text{if } \underline{b}_l < b_l \leq \overline{b}_l \\ \eta_l b_l - (1 - 2\overline{p}_l) & \text{if } \overline{b}_l < b_l \leq 2\overline{b}_l \\ 1 & \text{if } b_l > 2\overline{b}_l \end{cases},$$

where $\underline{b}_l$, $\overline{b}_l$, $\overline{p}_l$ are design parameters and $\gamma_l = \frac{\overline{p}_l}{\overline{b}_l - \underline{b}_l}$, $\eta_l = \frac{1-\overline{p}_l}{\overline{b}_l}$ are the corresponding slopes. As its buffer builds up, a link drops more and more packets, guiding the sources to reduce their transmission rates. In practice, TCP/RED often gets unstable when $b_l > \overline{b}_l$, and the parameters $\overline{p}_l$, $\overline{b}_l$, $\underline{b}_l$ are designed empirically or adaptively (ARED [14]) to avoid such instability. Hence, the model

$$p_l = \begin{cases} 0 & \text{if } b_l \leq \underline{b}_l \\ \gamma_l(b_l - \underline{b}_l) & \text{if } \underline{b}_l < b_l \leq \overline{b}_l \end{cases} \tag{1}$$

suffices for our study. In prior works (for example, [2]), the following simpler model

$$p_l = \gamma_l b_l \tag{2}$$

is often used. Model (2) neglects the region $[0, \underline{b}_l]$ on which $p_l = 0$, for the convenience of analysis. We will generally use model (2), but use the more precise model (1) to derive an important result: we can reduce the bandwidth without sacrificing throughput. Basically, after reducing the bandwidth, average buffer size $b_l$ at non-bottleneck links increases. But if $b_l$ is still below $\underline{b}_l$, the corresponding $p_l$ remains zero. Consequently, sources will not observe additional packet loss, and throughput will be preserved.

We consider TCP source control. In TCP Reno, a widely used loss-based TCP, a source generally increases its window size by 1 packet per round-trip time and halves its window size after a packet loss. For each source $i$, the end-to-end

[1] Average buffer size is the average of instantaneous buffer size over a moving time window of typically 500ms.

packet loss probability $q_i$ it observes is given by

$$q_i = 1 - \prod_{l \in \mathcal{L}_i} (1 - p_l) \approx \sum_{l \in \mathcal{L}_i} p_l = \sum_{l \in \mathcal{L}} R_{li} p_l.$$

The approximation is due to the fact that $\sum_{l \in \mathcal{L}_i} p_l \ll 1$, i.e., in practice, the end-to-end packet loss probability is very small. At steady state, source $i$ transmits at rate [7]

$$x_i = \begin{cases} x_i^{max} & \text{if } q_i < q_i^{min} \\ f_i(q_i) & \text{if } q_i \geq q_i^{min} \end{cases}, \tag{3}$$

where $f_i : [q_i^{min}, \infty) \rightarrow (0, x_i^{max}]$ is strictly decreasing, and determined by the TCP protocol source $i$ uses. For example,

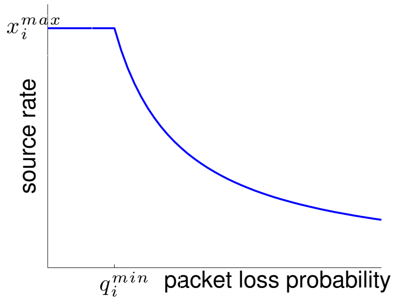

**Figure 2: Source rate.**

$f_i(q_i) = \frac{1}{\tau_i}\sqrt{\frac{3}{2q_i}}$ for TCP Reno, inversely proportional to the round-trip time $\tau_i$, and decreases as loss $q_i$ increases. Noting that (3) captures the stationary relationship between $q_i$ and $x_i$ for all TCP protocols, we use it for the sake of generality. Associate a utility function [7]

$$U_i(x_i) := \int^{x_i} f_i^{-1}(x) dx$$

with each source $i \in \mathcal{N}$, then (3) is equivalent to

$$x_i = \begin{cases} x_i^{max} & \text{if } q_i < U_i'(x_i^{max}) \\ U_i'^{-1}(q_i) & \text{if } q_i \geq U_i'(x_i^{max}) \end{cases}. \tag{4}$$

Since $f_i^{-1}$ is strictly decreasing and positive for $x_i \leq x_i^{max}$, $U_i$ is strictly concave and increasing. For instance, $U_i(x_i) = -\frac{3}{2\tau^2}\frac{1}{x_i}$ for TCP Reno [7].

## 2.2 Network Utility Maximization

For each link $l$, its average buffer size $b_l$ updates as[2]

$$\dot{b}_l = \begin{cases} 0 & \text{if } b_l = 0 \ \& \ y_l < c_l \\ y_l - c_l & \text{otherwise} \end{cases}. \tag{5}$$

Using model (2) for $p_l$, (5) is equivalent to

$$\dot{p}_l = \begin{cases} 0 & \text{if } p_l = 0 \ \& \ y_l < c_l \\ \gamma_l(y_l - c_l) & \text{otherwise} \end{cases}. \tag{6}$$

Discrete version of (6) and (4) is proven to solve the following *network utility maximization* (NUM) problem [2]

$$\textbf{NUM} \begin{cases} \max\limits_{x_i \in \mathcal{X}_i} & \sum\limits_{i \in \mathcal{N}} U_i(x_i) \\ \text{s.t.} & y_l \leq c_l, \ l \in \mathcal{L}. \end{cases}$$

Let $x^{\text{NUM}}$ denote the solution to NUM with $c_l = c_l^{max}$ for all $l$, then $x^{\text{NUM}}$ is the steady-state source rate for TCP/RED. Let $y^{\text{NUM}}$ denote the traffic corresponding to $x^{\text{NUM}}$.

# 3. BANDWIDTH ADJUSTMENT

In this section, we propose a dynamic bandwidth adjustment mechanism, and give its detailed implementations.

## 3.1 Algorithm

The objective of energy-efficient congestion control is to achieve the same throughput as traditional congestion control, while consuming less energy by turning off excess bandwidth. Conceptually, we can achieve this by finding $x^{\text{NUM}}$ and reducing the bandwidth $c$ to the traffic $y^{\text{NUM}}$. However, measuring the traffic on a link is expensive. In practice, a link only knows its own buffer size. To conclude, energy-efficient congestion control has the following objectives:

- achieves the source rate $x^{\text{NUM}}$;
- reduces link bandwidth $c$ to traffic rate $y^{\text{NUM}}$;
- uses local buffer information to update $c$.

We maintain the existing TCP/RED architecture, and introduce bandwidth updates. Noting that the only available information is its buffer size, a link $l$ compares its average buffer size $b_l$ with a moving target $\tilde{b}_l(c_l)$, and updates its bandwidth proportional to their difference according to the following *dynamic bandwidth adjustment* (DBA) mechanism:

$$\dot{c}_l = \begin{cases} 0 & \text{if } c_l = c_l^{min} \ \& \ b_l < \tilde{b}_l(c_l) \\ & \text{or } c_l = c_l^{max} \ \& \ b_l > \tilde{b}_l(c_l) \\ \alpha_l\left(b_l - \tilde{b}_l(c_l)\right) & \text{otherwise} \end{cases}. \tag{7}$$

Design parameter $\alpha_l > 0$ characterizes bandwidth dynamic speed, and its value will be discussed in section 3.2. We can interpret (7) as a truncated integral control law with a moving target. If $b_l$ exceeds $\tilde{b}_l$, bandwidth is increased to clear the buffer, and vice versa. At equilibrium, we expect $b_l = \tilde{b}_l(c_l)$. If the target $\tilde{b}_l$ is a constant, then queueing delay generally increases as we decrease bandwidth, and quality of service can be degraded. In order to limit the queueing delay, $\tilde{b}_l$ should decrease as $c_l$ decreases. If throughput $y_l$ is time invariant, the corresponding phase plot is shown in Figure 3: bandwidth $c_l$ converges to the throughput $y_l$, and buffer size $b_l$ equals its target $\tilde{b}_l(c_l)$ at convergence. However, $y_l$ will react to link buffer dynamics due to TCP, calling for more detailed study, which is given in section 4.

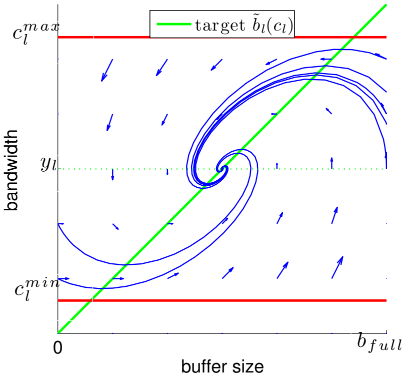

**Figure 3: Phase plot of $(b_l, c_l)$ for constant traffic $y_l$.**

[2] We do not distinguish between instantaneous buffer size and average buffer size in this paper.

## 3.2 Implementation

In this section, we give the implementations of DBA. For ease of presentation, we consider a single link $l$. Recall that $c_l$, $b_l$, and $\tilde{b}_l(c_l)$ denote the bandwidth, average buffer size, and target buffer size of link $l$ respectively. For notational brevity, we will drop the subscript $l$ in this section.

We set $\tilde{b}(c) = dc$, where $d$ is called *target queueing delay* and satisfies $d < \underline{b}/c^{max}$, so that the moving target $\tilde{b}$ is always below $\underline{b}$. Then, at steady state, packet drop probability $p_l = 0$ if link $l$ is not at its full bandwidth.[3] Consequently, either a link is at its full bandwidth, or it will not drop any packet. Hence, end-to-end loss probability $q_i$ is not increased after we reduce the bandwidth. As a result, source rates won't be reduced. A rigorous statement is provided in Theorem 1. Noting that queueing delay on non-bottleneck links increases by $d$, we should set $d$ small for quality of service consideration. A typical value is $d = 1ms$.

**Algorithm 1** Continuous DBA

```
At epoch $k \in \mathbb{Z}$, input $c[k-1]$, $b[k-1]$.
  1. Integral law.
      if $\left|\frac{b[k-1]}{dc[k-1]} - 1\right| \le \epsilon$ then
        $c \leftarrow c[k-1] + \delta_1 \frac{b[k-1] - dc[k-1]}{T}$;
      else
        $c \leftarrow c[k-1] + \delta_2 \frac{b[k-1] - dc[k-1]}{T}$;
      end if
  2. Lower and upper bounds.
      if $c > c^{max}$ then
        $c \leftarrow c^{max}$;
      else
        if $c < c^{min}$ then
          $c \leftarrow c^{min}$;
        end if
      end if
  3. Switching.
      $c[k] \leftarrow c$;
Return $c[k]$ as the link bandwidth during $[kT, (k+1)T)$.
```

Bandwidth $c$ should be updated periodically. Though current technology supports switching each physical interface (for instance, an optical cable) between idle and sleep states within 1ms, the updating period $T$ should be much larger than 1ms to reduce switching overhead. On the other hand, the period $T$ cannot be too large, in order to remain responsive to traffic increases. A typical value is $T = 100$ms. Assume bandwidth updates at epochs $t = kT, k \in \mathbb{Z}$, and define $c[k] := c(kT)$ as the bandwidth at epoch $k$ (then $c[k]$ is also the bandwidth during $[kT, (k+1)T)$). Similarly, define $b[k] := b(kT)$, $\tilde{b}[k] := \tilde{b}(kT) = dc(kT)$ as the average buffer size and target buffer size at epoch $k$. If we discretize (7) and neglect the lower and upper bounds on $c$, we get

$$c[k+1] = c[k] + T\alpha(b[k] - \tilde{b}[k]),\ k \in \mathbb{Z}.$$

Assuming static traffic $y$, static target buffer size $\tilde{b}$, and $c[k] = y$, then if we set $\alpha = \frac{1}{T^2}$, we have $b[k+2] = \tilde{b}$. Hence, $\alpha$ is on the order of $1/T^2$, and we update $c$ as

$$c[k+1] = c[k] + \delta \frac{b[k] - \tilde{b}[k]}{T},\ k \in \mathbb{Z},$$

where $\delta$ is a damping parameter and resides in $(0, 1]$. On one hand, we need small $\delta$ to guarantee local stability of TCP/RED/DBA (Theorem 4); on the other hand, we need big $\delta$ to avoid long transients. We balance this by using different $\delta$ in different buffer regions. We set a threshold $\epsilon > 0$. If the relative deviation $\left|b/\tilde{b} - 1\right| \le \epsilon$, we use a small $\delta_1$ for local stability. If $\left|b/\tilde{b} - 1\right| > \epsilon$, we use a big $\delta_2$ to shorten the transient. Typical values are $\epsilon = 0.1$, $\delta_1 = 0.04$, and $\delta_2 = 0.2$. Recall that $d$, $T$, $\epsilon$, $\delta_1$ and $\delta_2$ are design parameters, and we summarize the continuous dynamic bandwidth adjustment (continuous DBA) mechanism in Algorithm 1.

Bandwidth levels are far from continuous in practice. Assume that a link has $N$ bandwidth levels

$$c_1 < c_2 < \cdots < c_N.$$

To protect throughput, we choose a bandwidth $c_k$ such that $c_k \ge y^{\mathrm{NUM}} > c_{k-1}$. Since excess bandwidth $c_k - y^{\mathrm{NUM}}$ will probably make $b < \tilde{b}$, we set a hysteresis buffer region $[\utilde{b}, \tilde{b}]$ in which we do not reduce $c$. How to set $\utilde{b}$ is beyond the scope of this paper, and we pick $\utilde{b} = \tilde{b}/2$ in this paper for simplicity. The discrete dynamic bandwidth adjustment algorithm—given in Algorithm 2—exploits stochasticity to mimic continuous DBA. Recall that $d$, $T$, $\epsilon$, and $\delta_2$ are design parameters and we set $\utilde{b} = \frac{1}{2}\tilde{b}$.

**Algorithm 2** Discrete DBA

```
At epoch $k \in \mathbb{Z}$, input $c[k-1]$, $b[k-1]$.
  1. Integral law.
      $c \leftarrow c[k-1] + \delta_2 \frac{b[k-1] - dc[k-1]}{T}$;
  2. Lower and upper bounds.
      if $c > c^{max}$ then
        $c \leftarrow c^{max}$;
      else
        if $c < c^{min}$ then
          $c \leftarrow c^{min}$;
        end if
      end if
  3. Stochastic round off.
      $i \leftarrow \text{argmin}\,\{i \in \{1, \ldots, N-1\}\ :\ c_i \le c \le c_{i+1}\}$;
      $p \leftarrow \frac{c_{i+1} - c}{c_{i+1} - c_i}$;
      $c \leftarrow \begin{cases} c_i & \text{with probability } p \\ c_{i+1} & \text{with probability } 1-p \end{cases}$;
  4. Hysteresis.
      if $\frac{1}{2}dc[k-1] \le b[k-1] \le dc[k-1]$ then
        $c \leftarrow c[k-1]$;
      end if
  5. Switching.
      $c[k] \leftarrow c$;
Return $c[k]$ as the link bandwidth during $[kT, (k+1)T)$.
```

It can be verified that the expectation $E(c)$ does not change during step 3. Besides, $c$ coming from step 1 is equal to the result of continuous DBA when $\left|\frac{b[k-1]}{dc[k-1]} - 1\right| > \epsilon$. Hence, discrete DBA will hopefully mimic continuous DBA. We will further explore this in section 5.2.

# 4. ANALYSIS

There are mainly two questions to answer: will traffic $y$ oscillate, and how much throughput reduction will be incurred. Oscillation of $y$ leads to poor quality of service, and might

[3] At steady state, $c_l \ne c_l^{max}$ implies $b_l \le \tilde{b}_l$. Since $\tilde{b}_l < \underline{b}_l$, we have $b_l < \underline{b}_l$, which implies $p_l = 0$.

be induced by the interaction between DBA and TCP/RED. Supporting traffic is the primal goal of the network, hence it would be undesirable if DBA seriously reduce throughput. We answer these two questions in this section.

## 4.1 Global Stability without Network Delay

Recall that $p_l$, $b_l$, $x_i$ and $c_l$ denotes packet drop probability, average buffer size, source rate, and bandwidth respectively. We start with equations (1), (5), (4), and (7) to study the dynamic of TCP/RED/DBA, and call them the *synchronous model* (SM).

**Synchronous Model (SM):**

$$p_l = \begin{cases} 0 & \text{if } b_l \le \underline{b}_l \\ \gamma_l(b_l - \underline{b}_l) & \text{if } \underline{b}_l < b_l \le \bar{b}_l \end{cases} \tag{1}$$

$$\dot{b}_l = \begin{cases} 0 & \text{if } b_l = 0 \ \& \ y_l < c_l \\ y_l - c_l & \text{otherwise} \end{cases} \tag{5}$$

$$x_i = \begin{cases} x_i^{max} & \text{if } q_i < U_i'(x_i^{max}) \\ U_i'^{-1}(q_i) & \text{if } q_i \ge U_i'(x_i^{max}) \end{cases} \tag{4}$$

$$\dot{c}_l = \begin{cases} 0 & \begin{array}{l}\text{if } c_l = c_l^{min} \ \& \ b_l < \tilde{b}_l \\ \text{or } c_l = c_l^{max} \ \& \ b_l > \tilde{b}_l\end{array} \\ \alpha_l\left(b_l - \tilde{b}_l(c_l)\right) & \text{otherwise} \end{cases} \tag{7}$$

To model RED, we use the more precise equation (1) rather than (2), since the fact that we set $\tilde{b}_l < \underline{b}_l$ can only be captured by (1). We model the dynamics of average buffer size $b_l$ by the dynamics (5) of instantaneous buffer size for simplicity. The actual dynamics of $b_l$ should be a low-pass filtered version of (5). As mentioned in section 2.1, the model (4) we adopt for TCP captures the stationary relationship between source rate and end-to-end packet loss for a generic TCP protocol. To analyze the stability of a specific TCP protocol, we should change to its specific dynamic equation. The dynamic model (7) for $c_l$ is exact for continuous DBA. Note that Model SM does not capture network delay, which we will consider in section 4.3.

According to SM, source rate and link bandwidth converge globally to a unique equilibrium. At the equilibrium, excess bandwidth is completely turned off while source rate is preserved as if DBA has not been implemented. A rigorous statement is given in Theorem 1.

**Theorem** 1. *Let $(x(t), c(t), b(t), p(t))$ be an arbitrary trajectory generated by SM, then*

$$x(t) \to x^{NUM}, \ c(t) \to \max\{c^{min}, y^{NUM}\}$$

*as $t \to \infty$ provided the following conditions hold:*

*C1 $\tilde{b}_l'(c_l) = d_l > 0$ for $l \in \mathcal{L}$ and $c_l \in \mathcal{C}_l$.*

*C2 $\tilde{b}_l(c_l^{max}) < \underline{b}_l$ for $l \in \mathcal{L}$.*

*C3 $U_i''(x_i) \le -u < 0$, $\forall i \in \mathcal{N}$, $\forall x_i \in \mathcal{X}_i$.*

By the design of DBA, Conditions C1 and C2 hold automatically since we set $\tilde{b}_l(c_l) = d_l c_l$ with $d_l c_l^{max} < \underline{b}_l$. The utility function $U_i$ is strictly concave so that $U_i''(x_i) < 0$, and Condition C3 holds since $\mathcal{X}_i$ is bounded for each $i$.

The fact that $x(t) \to x^{\text{NUM}}$ implies no traffic oscillation and zero throughput reduction, and the fact that $c(t) \to \max\{c^{min}, y^{\text{NUM}}\}$ implies zero excess bandwidth at steady state. The proof of Theorem 1 is presented in Appendix A.

## 4.2 Network Surplus Maximization

The reason DBA reduces $c$ without sacrificing $x$ is that there is a nonempty interval $[0, \underline{b}_l]$ over which $p_l$ is zero. This feature is captured in (1). Without this feature, i.e., $\underline{b}_l = 0$, (1) reduces to (2), which we will use to simplify Model SM. Noting that (2) gives a bijective mapping between $b_l$ and $p_l$, we can substitute $b_l$ by $p_l$ to reduce the state space dimension. Define $\tilde{p}_l(c_l) := \gamma_l \tilde{b}_l(c_l)$ as the packet drop probability corresponding to $\tilde{b}_l$, and call its integral

$$V_l(c_l) := \int_{c_l^{min}}^{c_l} \tilde{p}_l(c)dc$$

the cost function of link $l$. Note that $V_l(c_l)$ comes from the moving target buffer size $\tilde{b}_l(c_l)$ we set, and is strictly convex and increasing. By substituting $b_l$ by $p_l$, the dynamics (7) for $c_l$ is equivalent to

$$\dot{c}_l = \begin{cases} 0 & \begin{array}{l}\text{if } c_l = c_l^{min} \ \& \ p_l < V_l'(c_l) \\ \text{or } c_l = c_l^{max} \ \& \ p_l > V_l'(c_l)\end{array} \\ \beta_l\left(p_l - V_l'(c_l)\right) & \text{otherwise} \end{cases} , \tag{8}$$

where $\beta_l := \alpha_l/\gamma_l$. Then, Model SM is simplified as below.

**Simplified Synchronous Model (SSM):**

$$\dot{p}_l = \begin{cases} 0 & \text{if } p_l = 0 \ \& \ y_l < c_l \\ \gamma_l(y_l - c_l) & \text{otherwise} \end{cases} \tag{6}$$

$$x_i = \begin{cases} x_i^{max} & \text{if } q_i < U_i'(x_i^{max}) \\ U_i'^{-1}(q_i) & \text{if } q_i \ge U_i'(x_i^{max}) \end{cases} \tag{4}$$

$$\dot{c}_l = \begin{cases} 0 & \begin{array}{l}\text{if } c_l = c_l^{min} \ \& \ p_l < V_l'(c_l) \\ \text{or } c_l = c_l^{max} \ \& \ p_l > V_l'(c_l)\end{array} \\ \beta_l\left(p_l - V_l'(c_l)\right) & \text{otherwise} \end{cases} \tag{8}$$

**Definition 1.** A point $(x, c, p)$ is an equilibrium for SSM, provided that at $(x, c, p)$, $\dot{p} = 0$, $\dot{c} = 0$ according to (6) and (8), and $(x, p)$ satisfy (4).

Then the steady states of TCP/RED/DBA are equilibrium points for SSM. Similar to NUM for traditional congestion control, we are able to show that steady state of TCP/RED/DBA solves the following *network surplus maximization* (NSM) problem

$$\textbf{NSM} \begin{cases} \max\limits_{x_i \in \mathcal{X}_i, c_l \in \mathcal{C}_l} & \sum\limits_{i \in \mathcal{N}} U_i(x_i) - \sum\limits_{l \in \mathcal{L}} V_l(c_l) \\ \text{s.t.} & y_l \le c_l, \ l \in \mathcal{L}. \end{cases}$$

A rigorous statement is given in Theorem 2. Let $(x^*, c^*)$ denote the unique solution to NSM,[4] and $y^*$ the corresponding traffic throughput. Since $V_l$ is strictly increasing, $c^* = y^*$ if there is no lower bound on $c$.

**Theorem** 2. *Equilibrium points of SSM exist, and a point $(x, c, p)$ is an equilibrium point for SSM if and only if $(x, c, p)$ is primal dual optimal for NSM.*

Theorem 2 connects NSM with TCP/RED/DBA, just like the relationship between NUM and TCP/AQM. It implies that we can study the steady-state traffic throughput and link bandwidth of TCP/RED/DBA by solving NSM. The proof of Theorem 2 is based on the duality theory [8, Chapter 5], and omitted due to page limit. Theorem 3 says that TCP/RED/DBA converges globally to its unique steady state $(x^*, c^*)$.

[4] Since $U_i$ are strictly concave for all $i$ and $V_l$ are strictly convex for all $l$, solution to NSM is unique.

**Theorem** 3. *Let $(x(t), c(t), p(t))$ be an arbitrary trajectory generated by SSM. Then the source rate $x(t)$ and bandwidth $c(t)$ converge to $(x^*, c^*)$ as $t \to \infty$ if the following conditions hold:*

*C3* $U_i''(x_i) \le -u < 0$, $\forall i \in \mathcal{N}$, $\forall x_i \in \mathcal{X}_i$.

*C4* $V_l''(c_l) \ge v > 0$, $\forall l \in \mathcal{L}$, $\forall c_l \in \mathcal{C}_l$.

Condition C3 has already been justified in the discussion after Theorem 1. Condition C4 also holds since $V_l''(c_l) = d_l\gamma_l > 0$ for all $l$ and all $c_l$. Theorem 3 implies that SSM is globally stable, and solves NSM. The proof of theorem 3 is based on the uniqueness of $(x^*, c^*)$ and contraction-type analysis shown in appendix B. Contraction-type analysis can be found in many references including [3].

### 4.3 Local Stability with Delay

We proved global convergence to $(x^*, c^*)$ without considering network delay in section 4.2, and will study local stability of $(x^*, c^*)$ under network delay in this section. We add link to source and source to link delays to SSM. For each source link pair $(i, l) \in \mathcal{N} \times \mathcal{L}$, if $l \in \mathcal{L}_i$, let $\tau_{il}(t)$ and $\tau_{li}(t)$ denote the feedforward delay from source $i$ to link $l$ and the feedback delay from link $l$ to source $i$ at time $t$ respectively; if $l \notin \mathcal{L}_i$, define $\tau_{li}(t) = \tau_{il}(t) = 0$ for all $t$. Then link $l \in \mathcal{L}$ in fact observes traffic

$$\hat{y}_l(t) = \sum_{i \in \mathcal{N}_l} x_i(t - \tau_{il}(t)) = \sum_{i \in \mathcal{N}} R_{li} x_i(t - \tau_{il}(t)), \qquad (9)$$

and updates its drop probability according to

$$\dot{p}_l = \begin{cases} 0 & \text{if } p_l = 0 \ \& \ \hat{y}_l < c_l \\ \gamma_l(\hat{y}_l - c_l) & \text{otherwise} \end{cases}. \qquad (10)$$

Source $i \in \mathcal{N}$ in fact observes congestion level

$$\hat{q}_i(t) = \sum_{l \in \mathcal{L}_i} p_l(t - \tau_{li}(t)) = \sum_{l \in \mathcal{L}} R_{li} p_l(t - \tau_{li}(t)), \qquad (11)$$

and updates its rate as

$$x_i = \begin{cases} x_i^{max} & \text{if } \hat{q}_i < U_i'(x_i^{max}) \\ U_i'^{-1}(\hat{q}_i) & \text{if } \hat{q}_i \ge U_i'(x_i^{max}) \end{cases}. \qquad (12)$$

We do not model the delay from a link to itself since it is much shorter, and come to the following simplified asynchronous model (SAM).

**Simplified Asynchronous Model (SAM):**

$$\dot{p}_l = \begin{cases} 0 & \text{if } p_l = 0 \ \& \ \hat{y}_l < c_l \\ \gamma_l(\hat{y}_l - c_l) & \text{otherwise} \end{cases} \qquad (10)$$

$$x_i = \begin{cases} x_i^{max} & \text{if } \hat{q}_i < U_i'(x_i^{max}) \\ U_i'^{-1}(\hat{q}_i) & \text{if } \hat{q}_i \ge U_i'(x_i^{max}) \end{cases} \qquad (12)$$

$$\dot{c}_l = \begin{cases} 0 & \text{if } c_l = c_l^{min} \ \& \ p_l < \tilde{p}_l \\ & \text{or } c_l = c_l^{max} \ \& \ p_l > \tilde{p}_l \\ \beta_l \left(p_l - \tilde{p}_l(c_l)\right) & \text{otherwise} \end{cases} \qquad (8)$$

**Definition 2.** A point $(x, c, p)$ is an equilibrium for SAM, provided that at $(x, c, p)$, $\dot{p} = 0$, $\dot{c} = 0$ according to (10) and (8), and $(x, p)$ satisfy (12).

It is not difficult to show that SSM and SAM have the same equilibrium points. Hence, Theorem 2 holds for SAM as well as SSM. Let $(x^e, c^e, p^e)$ be an equilibrium point for SAM, then $(x^e, c^e) = (x^*, c^*)$—the solution to NSM—according to Theorem 2. We study local stability near $(x^*, c^*)$ through linearization. Neglect the dynamics of $\tau_{il}(t)$ and $\tau_{li}(t)$ when linearizing (9) and (11) around $(x^e, c^e, p^e)$ as in [5], i.e., treat them as static, and denote the respective values by $\tau_{il}$ and $\tau_{li}$. For each source $i \in \mathcal{N}$, its round-trip time $\tau_i$ is equal to the sum of source to link delay $\tau_{il}$ and link to source delay $\tau_{li}$, for all link $l$ on its path, i.e.,

$$\tau_{il} + \tau_{li} = \tau_i, \ i \in \mathcal{N}, l \in \mathcal{L}_i.$$

Then, we are able to show that $(x^*, c^*)$ is locally stable, if round-trip times are bounded and link price dynamics and link bandwidth dynamics are not too fast.

**Theorem** 4. *$(x^*, c^*)$ is locally asymptotically stable for SAM, provided the following conditions hold:*

*C3* $U_i''(x_i) \le -u < 0$, $\forall i \in \mathcal{N}$, $\forall x_i \in \mathcal{X}_i$.

*C4* $V_l''(c_l) \ge v > 0$, $\forall l \in \mathcal{L}$, $\forall c_l \in \mathcal{C}_l$.

*C5* $\tau_i \le \tau$ *for all* $i \in \mathcal{N}$.

*C6* $\gamma_l < \frac{u}{NL\tau}$ *for each* $l \in \mathcal{L}$.

*C7* $\beta_l < \frac{\pi(\pi-2)}{4\tau^2\gamma_l}$ *for each* $l \in \mathcal{L}$.

Conditions C3 and C4 have been justified in the discussions after Theorem 1 and Theorem 3. Condition C5 holds since network delay is typically below several hundred milliseconds in real networks. $\beta$ and $\gamma$ are design parameters we choose. A smaller $\gamma$ leads to a slower dynamic in $p$, and a smaller $\beta$ leads to a slower dynamic in $c$. By damping the dynamics, Conditions C6 and C7 can be satisfied. Hence, Theorem 4 implies that we can slow down link dynamics to guarantee local stability of $(x^*, c^*)$. Theorem 4 also implies that the larger the network delay $\tau$, the more likely instability occurs, since the upper bounds for $\beta$ and $\gamma$ are inversely proportional to $\tau$. The proof of Theorem 4 is based on the generalized Nyquist theorem [4], and shown in Appendix C.

## 5. SIMULATION

We evaluate the performance of the dynamic bandwidth adjustment (DBA) mechanisms using the network topology in Figure 4. There are 40 TCP Reno sources connected to a shared link through 1Mb/s access links. To investigate how DBA reacts to traffic changes, we assume that 20 of the TCP sources are long-lived, while the other 20 TCP sources are active only from 200s to 250s. The number of TCP connections change abruptly at 200s and 250s, making bandwidth adjustment more challenging. Since the analysis in section 4 only deals with TCP traffic, we also investigate the impact of non-TCP traffic by introducing 20Mb/s UDP constant-bit-rate traffic in the period 350 ∼ 400s.

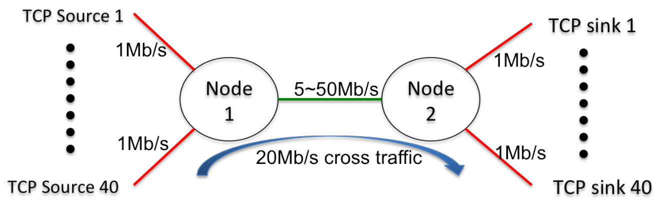

**Figure 4: There are** 20 **long-lived TCP connections and** 20 **short-lived TCP connections, which start at** 200s **and end at** 250s**. There is also** 20Mb/s **constant-bit-rate UDP cross traffic from** $350s$ **to** $400s$**.**

We compare the following settings for the shared link.

1. The shared link has static bandwidth of 50Mb/s.
2. The shared link has dynamic bandwidth with continuous levels ranging from 5Mb/s to 50Mb/s. Continuous

DBA, with parameters $T = 100\text{ms}$, $d = 1\text{ms}$, $\epsilon = 0.1$, $\delta_1 = 0.04$, and $\delta_2 = 0.2$, is implemented to adjust the bandwidth.

3. The shared link has dynamic bandwidth with discrete levels 5Mb/s, 10Mb/s, ..., 50Mb/s. Discrete DBA, with the same parameters as in setting 2, is implemented to adjust the bandwidth.

All the simulations in this paper use ns-allinone-2.34 on Mac OS X Lion 10.7.2.

## 5.1 Continuous DBA

Figure 5 shows the traffic throughput and link bandwidth on the shared link, in the static bandwidth setting (no DBA) and continuous bandwidth setting (continuous DBA). All TCP sources use TCP/Reno (loss-based), and have 50ms round-trip time. Due to the 1Mb/s access links (red links in Figure 4), each TCP connection takes up at most 1Mb/s bandwidth. Hence, throughput never reaches maximum bandwidth (50Mb/s) on the shared link. The low utilization level of the shared link provides an opportunity to save energy (by decreasing bandwidth) without reducing the throughput, and continuous DBA effectively exploits this opportunity.

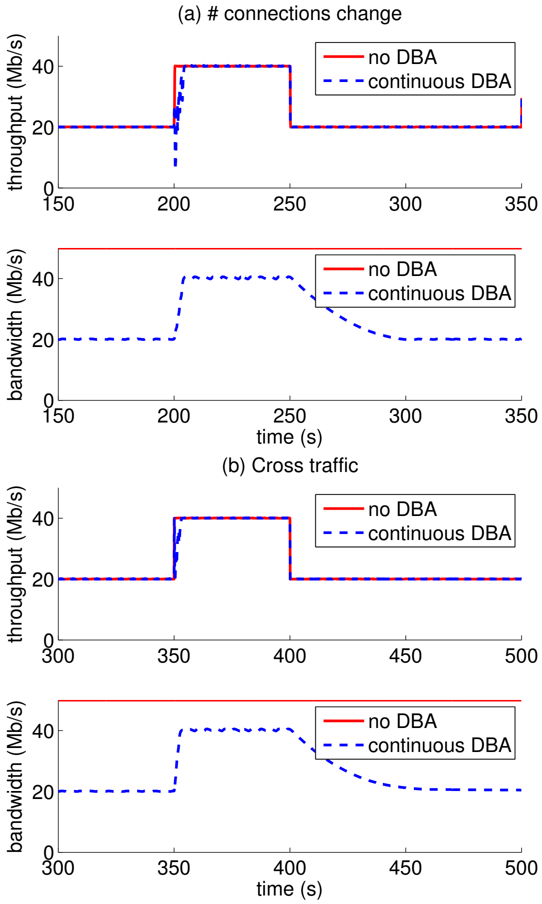

**Figure 5: There are 40 TCP connections with $50ms$ RTT. 20 of the TCP connections are long-lived while the other 20 are active only from $200s$ to $250s$. Additional 20Mb/s UDP constant-bit-rate cross traffic comes at $350s$ and leaves at $400s$.**

The following observations can be made from Figure 5.

- At steady state, the throughput is the same for both settings (no DBA and continuous DBA), even with the presence of non-TCP traffic.
- In the continuous DBA setting, when traffic (abruptly) increases (due to the increase in TCP connections or injection of cross traffic), the bandwidth increases fast. Hence, the effect of DBA is transparent to the TCP sources.
- In the continuous DBA setting, when traffic (abruptly) decreases (due to the decrease in TCP connections or removal of cross traffic), the bandwidth decreases slowly.
- In the continuous DBA setting, link bandwidth equals throughput at steady state.

As we observed, the bandwidth ramps up fast, but ramps down slowly. Fast ramping up makes DBA transparent to the TCP sources, and slow ramping down helps preserving network performance. We now give the reason for the fast ramping up and slow ramping down in DBA. When $c < y$, buffer $b$ can reach its full size $b_{full}$, and

$$c[k+1] - c[k] \le \delta_2 \frac{b_{full} - \tilde{b}[k]}{T}.$$

When $c > y$, $b$ can only decrease to 0, and

$$c[k] - c[k+1] \le \delta_2 \frac{\tilde{b}[k]}{T}.$$

Since $\tilde{b} < \underline{b} \ll b_{full}$, the ramping down speed has a much smaller upper bound than the ramping up speed. Furthermore, since $\tilde{b} = dc$, the smaller $d$, the slower $c$ can decrease. When cross traffic goes away at $400s$, excess bandwidth clears the buffer to 0 almost instantaneously. Taking average buffer size $b$ to be 0, the dynamics in (7) reduces to

$$\dot{c} = -\delta_2 \frac{\tilde{b}}{T^2} = -\frac{\delta_2 d}{T^2} c.$$

Bandwidth exponentially decays with time constant

$$\frac{T^2}{\delta_2 d} = \frac{(0.1s)^2}{0.2 \times 1ms} = 50s,$$

and it should take

$$50s \times \ln\left(\frac{40}{20}\right) = 34.7s$$

for $c$ to decrease from 40Mbps to 20Mbps, in accordance with the simulation result in Figure 5. We come to the following conclusions for the transients of continuous DBA:

- steady-state is (locally) stable;
- bandwidth ramps up very fast;
- bandwidth ramps down slowly and exponentially, with a time constant inversely proportional to the target queueing delay $d$.

We provide below two ways to decrease the bandwidth faster to save more energy.

- Increase $d$. This leads to larger RTT and affects quality of service. Note that $d < \underline{b}_l / c_l^{max}$, otherwise packet loss kicks in to reduce traffic throughput dramatically.
- Increase $\delta$. However, increasing $\delta$ potentially results in instability according to Theorem 4.

## 5.2 Discrete DBA

Section 5.1 evaluates the performance of continuous DBA, which provides an "upper bound" on what is achievable. In practice, bandwidth levels are discrete, and we may incur performance loss due to this restriction. This motivates us to evaluate discrete DBA in this section.

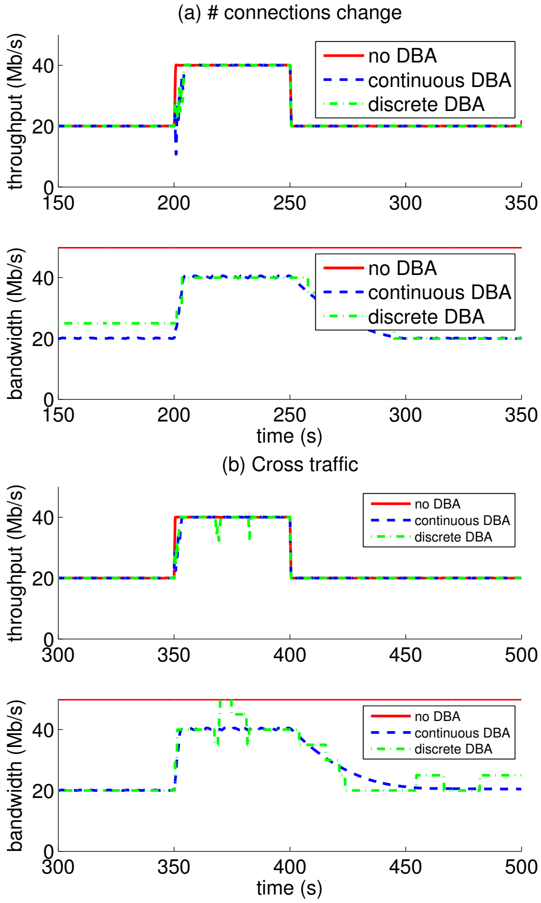

**Figure 6: There are 40 TCP connections with** 50$ms$ **RTT.** 20 **of the TCP connections are long-lived while the other** 20 **are active only from** 200$s$ **to** 250$s$**. Additional** 20**Mb/s UDP constant-bit-rate cross traffic comes at** 350$s$ **and leaves at** 400$s$**.**

Figure 6 shows the throughput and bandwidth for all three settings (no DBA, continuous DBA, and discrete DBA) under the same setup as Figure 5. It can be concluded that

- discrete DBA still preserves throughput;
- the bandwidth trajectory generated by discrete DBA roughly follows that of continuous DBA;
- the bandwidth trajectory generated by discrete DBA oscillates around that of continuous DBA—while staying above most of the time—at a low frequency.

Since discrete DBA only has few bandwidth levels to choose from, oscillation around optimal bandwidth is hardly avoidable. Spending most of the time above optimal bandwidth is what we prefer, since traffic protection is of primal concern. In fact, this is achieved by setting hysteric target buffer $[\underset{\sim}{b}, \tilde{b}]$.

On average, link bandwidth is updated only once every multiple seconds. Frequent link bandwidth updating will cause high overhead that prohibits implementation, and discrete DBA reduces the number of bandwidth updates by exploiting randomness. At each decision epoch, bandwidth is only updated with a small probability. For instance, at $25Mb/s$, the bandwidth switches to $20Mb/s$ with probability 1% (assuming empty buffer). Hence, it takes

$$\frac{1}{0.01}T \approx 10s$$

per update on average. Go on improving the performance of discrete DBA is a topic of our ongoing research.

## 5.3 Long Network Delay

As network delay increases, stability of flow control becomes more challenging, as reflected in Conditions C6 and C7. This motivates us to evaluate the impact of long network delay.

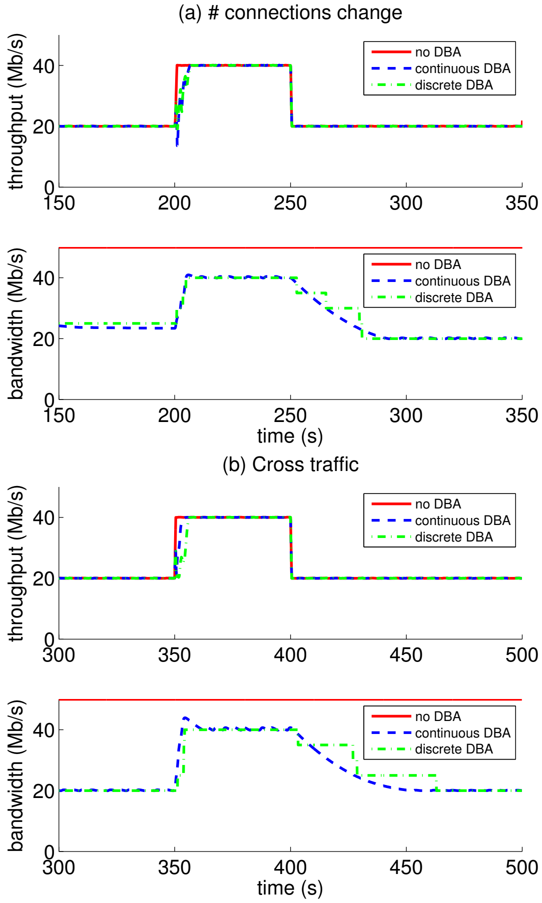

**Figure 7: There are 40 TCP connections with** 100$ms$ **RTT.** 20 **of the TCP connections are long-lived while the other** 20 **are active only from** 200$s$ **to** 250$s$**. Additional** 20**Mb/s UDP constant-bit-rate cross traffic comes at** 350$s$ **and leaves at** 400$s$**.**

Figure 7 shows the throughput and bandwidth for all three settings (no DBA, continuous DBA, and discrete DBA) under the same setup as Figure 6, except that the RTT for each flow is 100ms rather than 50ms. It can be seen that the performance of DBA is almost unaffected: the traffic throughput is still as if no DBA is implemented; excess bandwidth is completely turned off at steady state for continuous DBA; bandwidth of discrete DBA follows the bandwidth of continuous DBA.

# 6. CONCLUSION

We have proposed easily implementable dynamic bandwidth adjustment (DBA) mechanisms that update link bandwidth with local buffer information. We proved global stability of TCP/RED/DBA if network delay is negligible, and local stability if network delay is bounded and link dynamics are sufficiently slow. We also proved that DBA turns off excess bandwidth without reducing throughput, and this is confirmed by ns2 simulations. With a simplified model for RED, we proved that TCP/RED/DBA solves an optimization problem—which we call the network surplus maximization problem—that is similar to the well-known network utility maximization problem.

# 7. REFERENCES

[1] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, 1998, pp. 237-252.

[2] S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, 1999, pp. 861-874.

[3] W. Lohmiller and J. E. Lapsley, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, 1998, pp. 683-696.

[4] F. Paganini, Z. Wang, J. C. Doyle and S. H. Low, "Congestion control for high performance, stability, and fairness in general networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, 2005, pp. 43-56.

[5] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle, "Linear stability of TCP/RED and a scalable control," *Computer Networks Journal*, vol. 14, no. 5, 2003, pp. 633-647.

[6] D. X. Wei, C. Jin, S. H. Low and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, 2006, pp. 1246-1259.

[7] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, 2003, pp. 525-536.

[8] S. Boyd and L. Vandenberghe, "Convex optimization," Cambridge University Press, 2004.

[9] J. He, M. Chiang, and J. Rexford, "Can congestion control and traffic engineering be at odds," *in Proceedings of the IEEE GLOBECOM*, 2006.

[10] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, " Power awareness in network design and routing," *IEEE INFOCOM*, 2008.

[11] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," *in Proceedings of the first ACM SIGCOMM workshop on Green networking. ACM*, 2010, pp. 29-34.

[12] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," *in Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. USENIX Association*, 2008, pp. 323-336.

[13] L. Liu and B. Ramamurthy, "Rightsizing Bundle Link Capacities for Energy Savings in the Core Network", *IEEE GLOBECOM*, 2011.

[14] A. Francini, "Beyond RED: periodic early detection for on-chip buffer memories in network elements", *IEEE International conference on high performance switching and routing*, 2011.

# APPENDIX

# A. PROOF OF THEOREM 1

## *Convergence of $\dot{x}(t)$ to 0.*

**Lemma** 1. *For each $i \in \mathcal{N}$, if $U_i''(x_i) \leq -u_i \leq 0$ for all $x_i$ in $\mathcal{X}_i$, then the inequalities*

$$|\dot{q}_i(t)| \geq u_i |\dot{x}_i(t)|, \;\; \dot{q}_i(t)\dot{x}_i(t) \leq -u_i |\dot{x}_i(t)|^2. \qquad (13)$$

*hold for all $t$.*

Proof. $|\dot{q}_i| \geq |U_i''(x_i)\dot{x}_i| = u_i|\dot{x}_i|$, the rest is omitted. □

**Lemma** 2. *If condition C3 holds, then the inequality*

$$\sum_l \dot{p}_l(t)\dot{y}_l(t) \leq -u \sum_i \dot{x}_i(t)^2$$

*hold for all $t$.*

Proof.

$$\begin{aligned} \sum_l \dot{p}_l \dot{y}_l &= \sum_l \dot{p}_l \sum_i R_{li}\dot{x}_i = \sum_i \dot{x}_i \sum_l R_{li}\dot{p}_l \\ &= \sum_i \dot{x}_i \dot{q}_i \leq -\sum_i u_i \dot{x}_i^2. \qquad \square \end{aligned}$$

Define indicators $1_l$ as

$$1_l(t) := \begin{cases} 1 & \text{if } b_l(t) > \underline{b}_l \text{ or } b_l(t) = \underline{b}_l \;\&\; \dot{b}_l \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

for $l \in \mathcal{L}$, then $1_l = 0 \Rightarrow (p_l = 0 \;\&\; \dot{p}_l = 0)$ and $1_l = 1 \Rightarrow \dot{p}_l = \gamma_l \dot{b}_l$. We compute

$$\frac{d}{dt}\dot{p}_l^2 = 2\dot{p}_l\ddot{p}_l = 2\dot{p}_l\gamma_l\ddot{b}_l = 2\gamma_l\dot{p}_l(\dot{y}_l - \dot{c}_l),$$

$$\begin{aligned} 1_l\alpha_l\gamma_l\frac{d}{dt}(b_l - \tilde{b}_l)^2 &= 2\alpha_l\gamma_l(b_l - \tilde{b}_l)(\dot{b}_l - \tilde{b}_l'(c_l)\dot{c}_l)1_l \\ &= 2\alpha_l\gamma_l\dot{b}_l 1_l(b_l - \tilde{b}_l) - 2d_l\gamma_l\dot{c}_l^2 1_l \\ &= 2\alpha_l\dot{p}_l(b_l - \tilde{b}_l) - 2d_l\gamma_l\dot{c}_l^2 1_l. \end{aligned}$$

Then we have

$$\begin{aligned} &\frac{1}{\gamma_l}\frac{d}{dt}\dot{p}_l^2 + 1_l\alpha_l\gamma_l\frac{d}{dt}(b_l - \tilde{b}_l)^2 \\ = \;& 2\dot{p}_l(\dot{y}_l - \dot{c}_l) + 2\alpha_l(b_l - \tilde{b}_l)\dot{p}_l - 2d_l\gamma_l\dot{c}_l^2 1_l \\ = \;& 2\dot{p}_l\dot{y}_l + 2\dot{p}_l(\alpha_l(b_l - \tilde{b}_l) - \dot{c}_l) - 2d_l\gamma_l\dot{c}_l^2 1_l. \end{aligned}$$

For each link $l$, let

$$(t_{2k-1}^l, t_{2k}^l) \subseteq [0, \infty), k \geq 1$$

denote the time segments on which $1_l = 1$ ($t_k^l$ can be $\infty$). For convenience, we assume that $b_l(0) = \underline{b}_l$ for all $l \in \mathcal{L}$ in the rest of the proof. Extension to other initial points of $b$ is similar. We ignore the superscript $l$ when there is no confusion. Then 1) $t_1 = 0$; 2) $b_l = \underline{b}_l$ at $t_k$ if $t_k \neq \infty$; 3) $b_l \geq \underline{b}_l > \tilde{b}_l(c_l)$ on $(t_{2k-1}, t_{2k})$ and 4) $\dot{c}_l \geq 0$ on $(t_{2k-1}, t_{2k})$. Note that on $(t_{2k-1}, t_{2k})$, $\dot{c}_l = 0$ if and only if $c_l = c_l^{max}$, and $\dot{c}_l > 0$ otherwise. If $c_l$ reaches $c_l^{max}$ on $(t_{2k-1}, t_{2k})$, define $s_{2k}$ as the first time it attains $c_l^{max}$. Then we have 1) $c_l = c_l^{max}$ on $(s_{2k}, t_{2k})$, 2) $\dot{c}_l = \alpha_l(b_l - \tilde{b}_l)$ on $(t_{2k-1}, s_{2k})$, and

$$\begin{aligned} &\int_{t_{2k-1}}^{t_{2k}} 2\dot{p}_l(\alpha_l(b_l - \tilde{b}_l) - \dot{c}_l)dt \\ = \;& \int_{s_{2k}}^{t_{2k}} 2\gamma_l\dot{b}_l\alpha_l(b_l - \tilde{b}_l(c_l^{max}))dt \\ = \;& \alpha_l\gamma_l\left(b_l^2(t_{2k}) - b_l^2(s_{2k})\right) - 2\alpha_l\gamma_l\tilde{b}_l(c_l^{max})\left(b_l(t_{2k}) - b_l(s_{2k})\right) \\ = \;& \alpha_l\gamma_l\left(\underline{b}_l - b_l(s_{2k})\right)\left(\underline{b}_l + b_l(s_{2k}) - 2\tilde{b}_l(c_l^{max})\right) \leq 0. \end{aligned}$$

If $c_l$ does not reach $c_l^{max}$ on $(t_{2k-1}, t_{2k})$, the integration above is obviously 0 since $\dot{c}_l = \alpha_l(b_l - \tilde{b}_l)$ on $(t_{2k-1}, t_{2k})$. Hence,

$$\begin{aligned} & \int_0^\infty 2\dot{p}_l(\alpha_l(b_l - \tilde{b}_l) - \dot{c}_l)dt \\ = & \sum_{k\geq 1} \int_{t_{2k-1}}^{t_{2k}} 2\dot{p}_l(\alpha_l(b_l - \tilde{b}_l) - \dot{c}_l)dt \leq 0. \end{aligned}$$

We also have

$$\begin{aligned} & \int_{t_{2k-1}}^{t_{2k}} \left(-2d_l\gamma_l 1_l \dot{c}_l^2\right) dt \\ = & -2d_l\alpha_l\gamma_l \int_{t_{2k-1}}^{t_{2k}} \dot{c}_l(b_l - \tilde{b}_l)dt \\ \leq & -2d_l\alpha_l\gamma_l \int_{t_{2k-1}}^{t_{2k}} \dot{c}_l(\underline{b}_l - \tilde{b}_l)dt \\ = & d_l\alpha_l\gamma_l \left(d_l c_l^2 - 2\underline{b}_l c_l\right)|_{t_{2k-1}}^{t_{2k}} \\ = & \alpha_l\gamma_l \left(d_l c_l - \underline{b}_l\right)^2|_{t_{2k-1}}^{t_{2k}}. \end{aligned}$$

Hence,

$$\int_0^\infty \left(-2d\gamma_l 1_l \dot{c}_l^2\right) dt \leq \alpha_l\gamma_l \sum_{k\geq 1} \left(d_l c_l - \underline{b}_l\right)^2|_{t_{2k-1}}^{t_{2k}}.$$

We have

$$\begin{aligned} & \int_0^\infty \frac{d}{dt}\left(\frac{1}{\gamma_l}\dot{p}_l^2 + \alpha_l\gamma_l(b_l - \tilde{b}_l)^2\right) dt \\ = & \alpha_l\gamma_l \sum_{k\geq 1} (b_l - \tilde{b}_l)^2|_{t_{2k}}^{t_{2k+1}} \\ & + \int_0^\infty \left(2\dot{p}_l\dot{y}_l + 2\dot{p}_l(\alpha_l(b_l - \tilde{b}_l) - \dot{c}_l) - 2d\gamma_l\dot{c}_l^2 1_l\right) dt \\ \leq & \alpha_l\gamma_l \sum_{k\geq 1} (\underline{b}_l - \tilde{b}_l)^2|_{t_{2k}}^{t_{2k+1}} + 2\int_0^\infty \dot{p}_l\dot{y}_l dt \\ & + \alpha_l\gamma_l \sum_{k\geq 1} \left(d_l c_l - \underline{b}_l\right)^2|_{t_{2k-1}}^{t_{2k}} \\ = & 2\int_0^\infty \dot{p}_l\dot{y}_l dt - \alpha_l\gamma_l\left(\tilde{b}_l(0) - \underline{b}_l\right)^2. \end{aligned}$$

Sum up the previous inequality for all $l \in \mathcal{L}$ yields

$$\begin{aligned} & \int_0^\infty \frac{d}{dt}\sum_l \left(\frac{1}{\gamma_l}\dot{p}_l^2 + \alpha_l\gamma_l(b_l - \tilde{b}_l)^2\right) dt \\ \leq & 2\int_0^\infty \sum_l \dot{p}_l\dot{y}_l dt - \sum_l \alpha_l\gamma_l(\underline{b}_l - \tilde{b}_l(0))^2 \\ \leq & -\sum_l \alpha_l\gamma_l(\underline{b}_l - \tilde{b}_l(0))^2 - 2u\sum_i \int_0^\infty \dot{x}_l^2 dt. \end{aligned}$$

Since the left hand side is finite,

$$\int_0^\infty \dot{x}_i^2 dt$$

is finite for all $i$. Since $\dot{x}_i$ is Lipschitz, $\dot{x}_i \to 0$ as $t \to \infty$.

### *Convergence of* $x(t)$ *to* $x^{\mathrm{NUM}}$.

Define the set

$$\mathcal{S} := \{(x, c, p, b) \mid \dot{x} = 0 \text{ according to Model SM}\},$$

and $\mathcal{M}$ its largest invariant set. Then starting from any element $(x, c, p, b) \in \mathcal{M}$, $x$ and $y$ are time-invariant. If $y_l > c_l^{max}$ for some $l \in \mathcal{L}$, then $b_l(t)$ blows up, driving $x_i$ to 0 for $i \in \mathcal{N}_l$, contradict with $\dot{x} = 0$. Hence, $y_l \leq c_l^{max}$ for $l \in \mathcal{L}$ on $\mathcal{M}$. Each link $l$ has a state $(c_l, b_l)$ and constant control input $y_l$. From the phase plot in Figure 3, it is not difficult to show that $(c_l, b_l)$ will converge to some equilibrium point $(c_l^e, b_l^e)$. Let $p_l^e$ be the packet drop probability corresponding to $b_l^e$. By checking the KKT conditions, $(x, p^e)$ is primal dual optimal for Problem NUM with $c_l = c_l^{max}$ for all $l \in \mathcal{L}$. Hence, $x = x^{\mathrm{NUM}}$. Now we have shown that $x \in \mathcal{M} \Rightarrow x = x^{\mathrm{NUM}}$. Similar to Lasalle's theorem, we can show that $(x(t), c(t), p(t), b(t)) \to \mathcal{M}$. Consequently, $x(t) \to x^{\mathrm{NUM}}$ as $t \to \infty$.

### *Convergence of* $c(t)$ *to* $\max\{c^{min}, y^{\mathrm{NUM}}\}$.

Consider each link $l$ as an autonomous system with state $(c_l, b_l)$ and input $y_l$. According to the phase plot in Figure 3, the system of link $l$ is globally asymptotically stable with constant input $y_l$.[5] Since $x(t) \to x^{\mathrm{NUM}}$, $y_l \to y_l^{\mathrm{NUM}}$ for all $l$. If the input is $y_l^{NUM}$, the unique equilibrium point $(c_l^e, b_l^e)$ satisfies that $c_l^e = \max\{c_l^{min}, y_l^{\mathrm{NUM}}\}$. Consequently, $c_l(t) \to \max\{c_l^{min}, y_l^{\mathrm{NUM}}\}$ as $t \to \infty$ for all $l \in \mathcal{L}$.

This completes the proof of Theorem 1.

## B. PROOF OF THEOREM 3

Let $(x^*, c^*, p^*)$ be an equilibrium for Model SSM, $(x(t), c(t), p(t))$ be an arbitrary trajectory generated by SSM. Define $\delta x(t) := x(t) - x^*$, $\delta c(t) := c(t) - c^*$, $\delta p(t) := p(t) - p^*$ as the deviation of the trajectory from the equilibrium.

**Lemma** 3. *For each $i \in \mathcal{N}$, if*

$$U_i''(x_i) \leq -u_i \leq 0, \ \forall x_i \in \mathcal{X}_i,$$

*then*

$$|\delta q_i(t)| \geq u_i|\delta x_i(t)|, \ \delta q_i(t)\delta x_i(t) \leq -u_i(\delta x_i(t))^2 \qquad (14)$$

*for all $t$.*

Proof. For brevity, we abbreviate the argument $t$. We only show proof for the case $\delta x_i \leq 0$. Proof for the case $\delta x_i \geq 0$ is similar. If $\delta x_i = 0$, (14) is satisfied. Otherwise,

$$x_i^{min} \leq x_i < x_i^* \leq x_i^{max}.$$

It follows that

$$q_i \geq U_i'(x_i) > U_i'(x_i^*) \geq q_i^*,$$

and

$$\begin{aligned} \delta q_i \geq & U_i'(x_i) - U_i'(x_i^*) = \int_{x_i}^{x_i^*} -U_i''(x)dx \\ \geq & \int_{x_i}^{x_i^*} u_i dx = -u_i\delta x_i. \end{aligned}$$

It follows that (14) is satisfied. □

Define $B := \mathrm{diag}(\beta_1, \ldots, \beta_L)$, $\Gamma := \mathrm{diag}(\gamma_1, \ldots, \gamma_L)$. Since $\gamma_l > 0$, $\beta_l > 0$ for all $l \in \mathcal{L}$, diagonal matrices $B$ and $\Gamma$ are positive definite.

**Lemma** 4. *For each $l \in \mathcal{L}$,*

$$\begin{aligned} \frac{d}{dt}\left(\frac{(p_l - p_l^*)^2}{\gamma_l}\right) \leq & 2(p_l - p_l^*)(y_l - c_l) \qquad (15) \\ \leq & 2\delta p_l(\delta y_l - \delta c_l). \qquad (16) \end{aligned}$$

[5] When the input $y_l$ is time-invariant, the dynamic of link $l$ can be shown in a 2-dimensional phase plot, from which its global asymptotical stability is obvious.

Proof. For each $l \in \mathcal{L}$, define

$$[a]_0 := \begin{cases} 0 & \text{if } p_l = 0 \ \& \ a < 0 \\ a & \text{otherwise} \end{cases},$$

then

$$\begin{aligned} \frac{d}{dt}\left(\frac{(p_l - p_l^*)^2}{\gamma_l}\right) &= \frac{2}{\gamma_l}(p_l - p_l^*)\dot{p}_l \\ &= \frac{2}{\gamma_l}(p_l - p_l^*)\left[\gamma_l(y_l - c_l)\right]_0 \\ &= 2(p_l - p_l^*)\left[y_l - c_l\right]_0 \end{aligned}$$

If $\left[y_l - c_l\right]_0 = y_l - c_l$, (15) is satisfied with equality. Otherwise, $\left[y_l - c_l\right]_0 = 0$, $p_l = 0$ and $y_l < c_l$. It follows that the left hand side of (15) is zero, and the right hand side of (15) is non-negative. Hence, (15) is always satisfied. Besides,

$$\begin{aligned} & 2(p_l - p_l^*)(y_l - c_l) \\ = \ & 2\delta p_l(y_l - y_l^* + c_l^* - c_l) + 2\delta p_l(y_l^* - c_l^*) \\ = \ & 2\delta p_l(\delta y_l - \delta c_l) + 2p_l(y_l^* - c_l^*) - 2p_l^*(y_l^* - c_l^*) \\ \leq \ & 2\delta p_l(\delta y_l - \delta c_l). \end{aligned}$$

The last inequality holds since $p_l^*(y_l^* - c_l^*) = 0$, $p_l \geq 0$, and $y_l^* \leq c_l^*$. □

**Lemma** 5. *For each $l \in \mathcal{L}$, if*

$$V_l''(c_l) \geq v_l \geq 0, \ \forall c_l \in \mathcal{C}_l,$$

*then*

$$\begin{aligned} \frac{d}{dt}\left(\frac{(c_l - c_l^*)^2}{\beta_l}\right) &\leq 2(c_l - c_l^*)(p_l - V_l'(c_l)) && (17) \\ &\leq 2\delta c_l \delta p_l - 2v_l(\delta c_l)^2. && (18) \end{aligned}$$

Proof. For each $l \in \mathcal{L}$, define

$$[a]_{c_l^{min}}^{c_l^{max}} := \begin{cases} 0 & \text{if } c_l = c_l^{min} \ \& \ a < 0 \\ & \text{or } c_l = c_l^{max} \ \& \ a > 0 \\ a & \text{otherwise} \end{cases},$$

then,

$$\begin{aligned} \frac{d}{dt}\left(\frac{(c_l - c_l^*)^2}{\beta_l}\right) &= \frac{2}{\beta_l}(c_l - c_l^*)\dot{c}_l \\ &= \frac{2}{\beta_l}(c_l - c_l^*)\left[\beta_l(p_l - V_l'(c_l))\right]_{c_l^{min}}^{c_l^{max}} \\ &= 2(c_l - c_l^*)\left[p_l - V_l'(c_l)\right]_{c_l^{min}}^{c_l^{max}} \end{aligned}$$

If $\left[p_l - V_l'(c_l)\right]_{c_l^{min}}^{c_l^{max}} = p_l - V_l'(c_l)$, (17) is satisfied with equality. Otherwise, $\left[p_l - V_l'(c_l)\right]_{c_l^{min}}^{c_l^{max}} = 0$, the left hand side of (17) is zero. Besides, either $p_l > V_l'(c_l)$, $c_l = c_l^{max}$ or $p_l < V_l'(c_l)$, $c_l = c_l^{min}$. For both cases, the right hand side of (17) is non-negative. Hence, (17) is always satisfied. Besides,

$$\begin{aligned} & 2(c_l - c_l^*)(p_l - V_l'(c_l)) \\ = \ & 2\delta c_l(p_l - p_l^* + V_l'(c_l^*) - V_l'(c_l)) + 2\delta c_l(p_l^* - V_l'(c_l^*)) \\ \leq \ & 2\delta c_l \delta p_l + 2\delta c_l(V_l'(c_l^*) - V_l'(c_l)) \\ = \ & 2\delta c_l \delta p_l + 2\delta c_l \int_{c_l}^{c_l^*} V_l''(c)dc \\ \leq \ & 2\delta c_l \delta p_l + 2\delta c_l \int_{c_l}^{c_l^*} v_l dc \\ = \ & 2\delta c_l \delta p_l - 2v_l(\delta c_l)^2. \end{aligned}$$

Hence, (18) holds. □

**Lemma** 6. *If Conditions C3 and C4 hold, then*

$$\begin{aligned} & \frac{d}{dt}\left[\begin{pmatrix} \delta p \\ \delta c \end{pmatrix}^T \begin{pmatrix} \Gamma^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta c \end{pmatrix}\right] \\ \leq \ & -2\sum_{i \in \mathcal{N}} u(\delta x_i)^2 - 2\sum_{l \in \mathcal{L}} v(\delta c_l)^2. \qquad (19) \end{aligned}$$

Proof. If Conditions C3 and C4 hold, then

$$\begin{aligned} & \frac{d}{dt}\left[\begin{pmatrix} \delta p \\ \delta c \end{pmatrix}^T \begin{pmatrix} \Gamma^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} \delta p \\ \delta c \end{pmatrix}\right] \\ = \ & \sum_{l \in \mathcal{L}} \frac{d}{dt}\left(\frac{(p_l - p_l^*)^2}{\gamma_l}\right) + \sum_{l \in \mathcal{L}} \frac{d}{dt}\left(\frac{(c_l - c_l^*)^2}{\beta_l}\right) \\ \leq \ & \sum_{l \in \mathcal{L}} 2\delta p_l(\delta y_l - \delta c_l) + \sum_{l \in \mathcal{L}} 2\left(\delta c_l \delta p_l - v(\delta c_l)^2\right) \\ = \ & 2\sum_{l \in \mathcal{L}} \delta p_l \delta y_l - 2\sum_{l \in \mathcal{L}} v(\delta c_l)^2 \\ = \ & 2\sum_{i \in \mathcal{N}} \delta x_i \delta q_i - 2\sum_{l \in \mathcal{L}} v(\delta c_l)^2 \\ \leq \ & -2\sum_{i \in \mathcal{N}} u(\delta x_i)^2 - 2\sum_{l \in \mathcal{L}} v(\delta c_l)^2. \end{aligned}$$

The first inequality is due to (16) and (18), and the second inequality is due to (14). □

### *Proof of Theorem 3.*

For brevity, define $z := (\delta p^T, \delta c^T)^T$, $M := \text{diag}(\Gamma^{-1}, B^{-1})$. Then (19) implies that

$$\begin{aligned} & \int_0^\infty \frac{d}{dt}\left(z^T M z\right) dt \\ \leq \ & -2u\sum_{i \in \mathcal{N}} \int_0^\infty (\delta x_i)^2 dt - 2v\sum_{l \in \mathcal{L}} \int_0^\infty (\delta c_l)^2 dt. \end{aligned}$$

Since the left hand side is bounded and $\delta x_i$ is Lipschitz, we have $\delta x_i(t) \to 0$ as $t \to \infty$ for all $i \in \mathcal{N}$. Similarly, $\delta c_l(t) \to 0$ as $t \to \infty$ for all $l \in \mathcal{L}$. As $\delta x(t)/\delta c(t)$ denotes the deviation from $x(t)/c(t)$ to $x^*/c^*$, source rate $x(t)$ and link bandwidth $c(t)$ converge to the unique solution $(x^*, c^*)$ to NSM. □

## C. PROOF OF THEOREM 4

Let $(x^*, c^*, p^*)$ be an equilibrium for Model SAM, then it's also an equilibrium for Model SSM, and primal dual optimal point for Problem NSM. Let $(x(t), c(t), p(t))$ be an arbitrary trajectory generated by SAM, and define $\delta x(t) := x(t) - x^*$, $\delta c(t) := c(t) - c^*$, $\delta p(t) := p(t) - p^*$ as the deviation from the trajectory to the equilibrium point $(x^*, c^*, p^*)$.

We use the generalized Nyquist theorem to show local stability of $(x^*, c^*)$. We first linearize dynamic equations (8,10,12) near $(x^*, c^*, p^*)$, and then derive the loop transfer matrix. In the end, we evaluate the eigenvalues of the loop transfer matrix and show that they will not encircle the $-1$ point as frequency goes from $-\infty$ to $\infty$, if the conditions in Theorem 4 holds.

### *Linearization.*

For each link $l \in \mathcal{L}$, $c_l$ is restricted to the region $[c_l^{min}, c_l^{max}]$. If the constraint is active, $c_l$ will be fixed at either $c_l^{min}$ or $c_l^{max}$. Effectively, it is equivalent to $c_l$ being static, and we only need to linearize (8) for the links that $c_l$ is not static. For the same reason, we only linearize (10) and (12) for the links and sources that $p_l$ and $x_i$ are not effectively static. Without loss of generality, we assume that all links and sources are not effectively static, leading us to

$$\begin{aligned} \dot{\delta c_l}(t) &= \beta_l(\delta p_l(t) - v_l \delta c_l(t)),\ l \in \mathcal{L}, && (20) \\ \dot{\delta p_l}(t) &= \gamma_l \left( \sum_{i \in \mathcal{N}} R_{li} \delta x_i(t - \tau_{il}) - \delta c_l(t) \right),\ l \in \mathcal{L}, && (21) \\ \delta x_i(t) &= -u_i \sum_{l \in \mathcal{L}} R_{li} \delta p_l(t - \tau_{li}),\ i \in \mathcal{N}, && (22) \end{aligned}$$

where $v_l := V_l''(c_l^*)$ for $l \in \mathcal{L}$ and $u_i := -1/U_i''(x_i^*)$ for $i \in \mathcal{N}$.

### *Loop Transfer Matrix.*

Let $R(s)$ denote the $L \times N$ matrix whose $(l,i)$ entry is given by $R_{li}(s) := R_{li} \exp(-s\tau_{il})$. Let $X(s)$, $P(s)$, $C(s)$ denote the Laplacian transformation of $\delta x(t)$, $\delta p(t)$, and $\delta c(t)$ respectively. Define $U := \operatorname{diag}(u_1, \ldots, u_N)$, $V := \operatorname{diag}(v_1, \ldots, v_L)$. Equation (20) implies

$$C(s) = \operatorname{diag}\left( \frac{\beta_l}{s + \beta_l v_l} \right) P(s).$$

Equation (21) implies

$$sP(s) = \Gamma R(s) X(s) - \Gamma C(s).$$

Equation (22) implies

$$\begin{aligned} X_i(s) &= -u_i \sum_{l \in \mathcal{L}} R_{li} P_l(s) \exp(-s\tau_{li}) \\ &= -u_i \exp(-s\tau_i) \sum_{l \in \mathcal{L}} R_{li} P_l(s) \exp(s\tau_{il}) \end{aligned}$$

for $i \in \mathcal{N}$. Hence,

$$X(s) = -\operatorname{diag}\left(u_i e^{-s\tau_i}\right) R^T(-s) P(s).$$

It is not difficult to show that the loop transfer matrix is

$$L(s) = \frac{1}{s} \Gamma R(s) \operatorname{diag}\left(u_i e^{-s\tau_i}\right) R^T(-s) + \operatorname{diag}\left( \frac{\beta_l \gamma_l}{s(s + \beta_l v_l)} \right).$$

### *Eigenvalues.*

Define matrices

$$A := \operatorname{diag}\left( \frac{u_i e^{-jw\tau_i}}{jw} \right),\ D := \operatorname{diag}\left( \frac{\beta_l \gamma_l}{jw(jw + \beta_l v_l)} \right).$$

Let $\lambda$ denote an arbitrary eigenvalue of $L(jw)$, and $z$ its corresponding eigenvector. To this end, abbreviate $R(jw)$ as $R$, and use the superscript $H$ to denote conjugate transpose, then

$$\begin{aligned} \lambda z &= (\Gamma R A R^H + D) z, \\ \lambda z^H \Gamma^{-1} z &= z^H (R A R^H + \Gamma^{-1} D) z, \\ \lambda &= \frac{z^H R A R^H z}{z^H \Gamma^{-1} z} + \frac{z^H \Gamma^{-1} D z}{z^H \Gamma^{-1} z}. \end{aligned}$$

Define

$$\lambda_1 := \frac{z^H R A R^H z}{z^H \Gamma^{-1} z},\ \lambda_2 := \frac{z^H \Gamma^{-1} D z}{z^H \Gamma^{-1} z},$$

then $\lambda = \lambda_1 + \lambda_2$. Denote $\xi := R^H z$, then

$$\|\xi\|^2 = z^H R R^H z \le LN \|z\|^2.$$

Condition C3 implies that $u_i \le 1/u$. Define $\gamma := \max_{l \in \mathcal{L}} \{\gamma_l\}$, $f_i := \frac{e^{-jw\tau_i}}{jw\tau_i}$, then

$$\begin{aligned} \lambda_1 &= \frac{\xi^H A \xi}{z^H \Gamma^{-1} z} \\ &\in [0,1] * \gamma \frac{\xi^H A \xi}{z^H z} \\ &\subseteq [0,1] * \gamma LN \frac{\xi^H A \xi}{\xi^H \xi} \\ &= [0, \gamma LN] * \frac{\sum_{i \in \mathcal{N}} u_i \tau_i f_i \|\xi_i\|^2}{\sum_{i \in \mathcal{N}} \|\xi_i\|^2} \\ &= \left[0, \frac{\gamma LN\tau}{u}\right] * \sum_{i \in \mathcal{N}} \frac{u u_i \tau_i \|\xi_i\|^2}{\sum_{j \in \mathcal{N}} \tau \|\xi_j\|^2} f_i \\ &\subseteq [0,1] * \sum_{i \in \mathcal{N}} \frac{u u_i \tau_i \|\xi_i\|^2}{\sum_{j \in \mathcal{N}} \tau \|\xi_j\|^2} f_i. \end{aligned}$$

Since $\tau_i \le \tau$, $uu_i \le 1$ for all $i \in \mathcal{N}$, $\lambda_1$ lies in the convex hull of $\{0, f_1, \ldots, f_N\}$. Define $d_l := \frac{\beta_l \gamma_l}{jw(jw + \beta_l v_l)}$ for all $l \in \mathcal{L}$, then $d_l$ lies in the third orthant for all $w > 0$, and

$$\lambda_2 = \sum_{l \in \mathcal{L}} \frac{\|z_l\|^2 / \gamma_l}{\sum_{k \in \mathcal{L}} \|z_k\|^2 / \gamma_k} d_l$$

lies in the convex hull of $\{d_1, \ldots, d_L\}$.

### *Proof of Theorem 4.*

When $0 < w < \frac{\pi}{2\tau}$, $f_i$ is in the third orthant for all $i \in \mathcal{N}$. Hence, $\lambda_1$ is in the third orthant. Since $d_l$ is also in the third orthant for all $l \in \mathcal{L}$ and all $w > 0$, $\lambda_2$ is in the third orthant. Hence, $\lambda = \lambda_1 + \lambda_2$ is in the third orthant when $0 < w < \frac{\pi}{2\tau}$.

When $w \ge \frac{\pi}{2\tau}$, Condition C7 implies that

$$|d_l| \le \frac{\beta_l \gamma_l}{\frac{\pi}{2\tau} \sqrt{\left(\frac{\pi}{2\tau}\right)^2 + (\beta_l v_l)^2}} \le \frac{4\tau^2}{\pi^2} \beta_l \gamma_l < 1 - \frac{2}{\pi}.$$

Hence, $|\lambda_2| < 1 - \frac{2}{\pi}$. Note that $\lambda_1$ lies in the shaded area in Figure 8, and $\lambda_2$ lies in the third orthant. $|\lambda_2| \ge 1 - \frac{2}{\pi}$ is necessary for $\lambda$ to wind the -1 point. Hence, $\lambda$ does not wind the -1 point when $w \ge \frac{\pi}{2\tau}$.

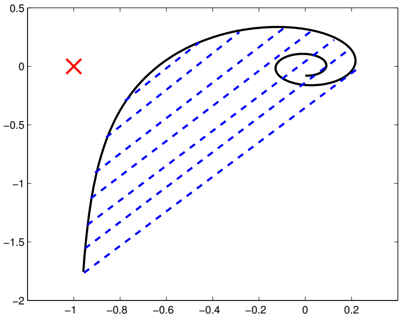

**Figure 8: Nyquist plot for $f_i, i \in \mathcal{N}$.**

Hence, $\lambda$ does not wind the -1 point as $w$ goes from 0 to $\infty$. Similarly, $\lambda$ does not cross the -1 point as $w$ goes from $-\infty$ to 0. Since $\lambda$ is an arbitrary eigenvalue of the loop transfer matrix, the equilibrium point $(x^*, c^*, p^*)$ is locally asymptotically stable by the generalized Nyquist theorem.□