

Pricing Link by Time

Chengdi Lai^{*}
Department of Electrical and
Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
laichengdi@eee.hku.hk

Ka-Cheong Leung
Department of Electrical and
Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
kcleung@eee.hku.hk

Steven H. Low
CMS & EE
California Institute of
Technology
Pasadena, 91125 CA, U.S.A.
slow@cms.caltech.edu

Victor O.K. Li
Department of Electrical and
Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
vli@eee.hku.hk

ABSTRACT

The combination of loss-based TCP and drop-tail routers often results in full buffers, creating large queueing delays. The challenge with parameter tuning and the drastic consequence of improper tuning have discouraged network administrators from enabling AQM even when routers support it. To address this problem, we propose a novel design principle for AQM, called the pricing-link-by-time (PLT) principle. PLT increases the link price as the backlog stays above a threshold β , and resets the price once the backlog goes below β . We prove that such a system exhibits cyclic behavior that is robust against changes in network environment and protocol parameters. While β approximately controls the level of backlog, the backlog dynamics are invariant for β across a wide range of values. Therefore, β can be chosen to reduce delay without undermining system performance. We validate these analytical results using packet-level simulation.

Categories and Subject Descriptors

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

Keywords

active queue management; congestion control; hybrid dynamical system.

^{*}The work was performed in part during visit to the Rigorous Systems Research Group, California Institute of Technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.
Copyright 2014 ACM 978-1-4503-2789-3/14/06 ...\$15.00.
<http://dx.doi.org/10.1145/2591971.2591974>.

1. INTRODUCTION

1.1 Motivation

Congestion control is a distributed feedback mechanism to share network resources among competing flows. It consists of a source algorithm that adapts the sending rate of a flow to the congestion level, measured by “congestion prices,” in its path and a router algorithm that updates its congestion prices, implicitly or explicitly, based on the local aggregate flow rate. On the Internet, the source algorithm is implemented in the transmission control protocol (TCP) of a host and is *loss-based*, where the congestion price takes the form of packet loss probability. The default router algorithm is implicit and called *drop-tail* where packets are dropped when, and only when, they arrive at a full buffer. The combination of loss-based TCP and drop-tail routers results in buffers constantly being filled, creating large queueing delays. The installation of large buffers in the network and the proliferation of applications with stringent delay requirements (e.g., IP telephony and video conferencing) have made this problem, recently termed “bufferbloat” [3, 14], a pressing issue.

In order to avoid persistent and high queueing delay, provide more timely congestion feedback, and desynchronize flow responses, active queue management (AQM) algorithms have been proposed where packets are probabilistically dropped before a buffer overflows. The most popular AQM is Random Early Detection (RED) proposed in 1993 [12]. Numerous variants have since been proposed, such as [5, 10, 11, 16, 20]. However, none of these AQM algorithms are widely deployed [9].¹

This is because the performance of these algorithms are usually very sensitive to parameter tuning. See Section 5.1 for an example with RED. These algorithms all compute a packet dropping probability based on some *parameterized* function of the backlog. For desirable performance, the parameters must incorporate network information such as capacity, delay, and the number of TCP flows sharing the link. Not only is it difficult for network administrators

¹Even though most routers implement RED, apparently most do not turn it on.

to determine the right parameter value if they know this information, more importantly, some of this information is constantly changing, rendering these AQM algorithms ineffective when the parameters are statically set. Adapting AQM parameters dynamically has proved to be very difficult due to the challenge in real-time estimation of network information. Besides, such adaptation process gives rise to an outer loop of feedback control and thus extra stability issue. Despite two decades of intense effort by the research and engineering communities, the challenge with parameter tuning and the drastic consequence of improper tuning continue to prevent network administrators from enabling AQM even when routers support it.

In this paper, we propose a method to address this problem through a core property of AQM design, we call the pricing-link-by-time (PLT) principle, for loss-based congestion control. We will show that an AQM algorithm that satisfies the PLT principle is:

- *Robust*: its performance does not degrade drastically even when its parameters are not well tuned.
- *Simple*: the average queueing delay is adjustable via a simple parameter tuning without side effects.

1.2 Our Contribution: PLT Principle

Recall that each link that implements an AQM algorithm computes a “link price.” For loss-based congestion control, the price corresponds to packet dropping (or marking) probability at the link. We say that *an AQM algorithm satisfies the PLT principle* if it has the following simple properties:

1. There is a preselected threshold β (in unit of packets or transmission time).
2. The link price monotonically increases with the duration of time for which the backlog $b(t)$ stays above β .
3. The link price is reset to a fixed value (e.g., zero) when the backlog $b(t)$ drops below β .

The PLT principle is defined mathematically in Section 2 by (3)–(4). It is illustrated and compared with RED in Fig. 1. PLT is *insensitive* to the magnitude of the backlog; only whether (and how long) it exceeds β matters. This stands in sharp contrast to existing AQM algorithms like RED, under which the link price depends critically on the *exact magnitude* of the backlog.

For the rest of the paper, we explore the implications of the PLT principle. Specifically, we make the following contributions:

- We present an analytical model of a congestion control system consisting of a TCP variant and an AQM algorithm that satisfies the PLT principle with a threshold β (Section 2).
- We *characterize completely* the behavior of the system when there is a single bottleneck link (Section 3). In particular, we prove that such a system has no equilibrium point. Instead, the backlog will go above and below β infinitely often. Surprisingly, starting from $b(t) = \beta$, the recurrent time for $b(t)$ to return to β is independent of the parameter β . Hence, β approximately controls the average level of backlog.

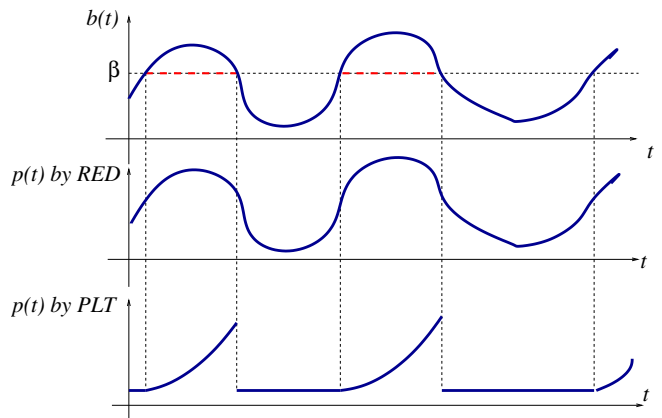


Figure 1: The PLT principle. The upper panel shows a backlog trajectory $b(t)$ and the threshold β . The middle panel shows the price $p(t)$ under RED that increases as $b(t)$ increases. The lower panel shows that price $p(t)$ under PLT that increases as the duration for which $b(t) > \beta$ increases.

The system possesses a family of cycles and we provide a sufficient condition for these limit cycles to be stable limit cycles. We prove a striking invariance property of these cycles with respect to β : the backlog trajectory $b(t)$ remains identical, except for a constant shift by $\tilde{\beta} - \beta$, if the threshold is changed from β to $\tilde{\beta}$.

- We explain why the recently proposed CoDel [25] and Data Center TCP (DCTCP) [1, 2] are AQM examples that satisfy the PLT principle (Section 4). CoDel has been experimentally shown to work well across various network settings with a single set of parameters. Our analysis explains the underlying reason for this robustness. The cyclic behavior of DCTCP has been reported in [2]. Our results provide additional insights on its behavior.
- We illustrate the PLT principle through ns-2 simulation of CoDel and DCTCP and compare their behavior with RED [12] and PI [16] that do not satisfy the PLT principle (Section 5). The simulation results confirm the robust cyclic behavior and verify the invariance of cycles against β .

Finally, we discuss related work in Section 6 and conclude in Section 7.

In summary, we have proposed the PLT principle and demonstrated both analytically and through simulation its effectiveness in solving the bufferbloat problem. Unlike traditional AQM algorithms that strive to attain a static equilibrium point, a PLT-based AQM algorithm has no equilibrium point but a family of cycles. *Nonetheless, these cycles are much more robust against changes in network environments and protocol parameters than the static equilibrium of traditional AQM algorithms. The threshold parameter β controls the level of delay without affecting other aspects of the system performance.*

On the technical aspect, our method is novel in applying the control-theoretic vehicle of cycles and a construct similar to Poincaré maps to study the congestion control system. This differs significantly from the popular stability analysis

Table 1: Notations

notation	meaning
$b(t)$	queue size at Time t
$b^{\mathcal{A}} / b^{\mathcal{B}}$	queue size under Mode \mathcal{A} / Mode \mathcal{B}
c	link capacity
$\mathbf{F}^{\mathcal{A}} / \mathbf{F}^{\mathcal{B}}$	mapping of flow rates from the start of Mode \mathcal{A} / Mode \mathcal{B} to the end of Mode \mathcal{A} / Mode \mathcal{B} ; see (11) / (16)
\mathbf{F}	$\mathbf{F}^{\mathcal{A}} \circ \mathbf{F}^{\mathcal{B}}$
$H^{\mathcal{A}} / H^{\mathcal{B}}$	hitting time under Mode \mathcal{A} / Mode \mathcal{B}
I_{β}	time elapsed since the queue size exceed β ; see (3)
$p(t)$	link price at Time t
U_f	utility function of Flow f
$x_f(t)$	flow rate of Flow f at Time t
\mathbf{x}	$(x_f(t) : f \in \mathcal{F})$
$\mathbf{x}^{\mathcal{A}} / \mathbf{x}^{\mathcal{B}}$	set of flow rates under Mode \mathcal{A} / Mode \mathcal{B}
$y^{\mathcal{A}} / y^{\mathcal{B}}$	aggregate flow rate under Mode \mathcal{A} / Mode \mathcal{B}

approach [18, 22, 28] that assumes the existence of equilibrium, and is necessitated by the absence of equilibrium in our interested system.

2. MODEL OF PLT PRINCIPLE

We use bold fonts to denote vectors and matrices. Unless otherwise specified, $\|\cdot\|$ denotes the 1-norm of a vector, i.e., $\|\mathbf{v}\| = \sum_i |v_i|$. We use \mathbb{R} , \mathbb{R}_+ , and \mathbb{R}_{++} to denote the sets of real numbers, nonnegative real numbers, and positive real numbers, respectively. A summary of frequently used notations is shown in Table 1.

We begin by considering a network with a single bottleneck link shared by a set \mathcal{F} of TCP flows.

Flow f adapts its rate according to:

$$\dot{x}_f(t) = \kappa_f \left(1 - \frac{p(t)}{U'_f(x_f(t))} \right) \quad (1)$$

where κ_f is a scaling constant, U_f is the utility function of Flow f , and $p(t)$ is the price of the bottleneck link. We assume $U_f : (0, \infty) \rightarrow \mathbb{R}$ to be strictly increasing, convex, and twice differentiable, with $\lim_{x_f \rightarrow 0} U'_f(x_f) = +\infty$. Thus, the utility U_f increases as the flow rate increases, but the marginal utility U'_f decreases. The flow dynamics aims to match the marginal utility to the link price.

This description of flow dynamics is customary in the literature and satisfied by most TCP variants. For example, TCP Reno exhibits such dynamics with $\kappa_f = \frac{1}{d_f}$ and $U_f(x_f) = -\frac{1}{d_f^2 x_f}$, where d_f is the round-trip time [21, 23].

At the bottleneck link, the backlog evolves according to the fluid model:

$$\dot{b}(t) = \sum_{f \in \mathcal{F}} x_f(t) - c \quad (2)$$

In reality, the backlog cannot be negative. Thus, a complete model should in addition enforce $\dot{b}(t) = 0$ when $b(t) = 0$ and $\sum_{f \in \mathcal{F}} x_f(t) - c < 0$. However, this will unnecessarily complicate our following analysis. We thus proceed with the simple model of (2). We postpone the discussion on the consequence of making such simplification to the performance validation section (Section 5.3).

Now, we are going to model the PLT principle. The time elapsed since the backlog exceeds β can be formally expressed as:

$$I_{\beta}(t) = \sup\{d \geq 0 : b(t-d) > \beta \forall \delta \in [0, d]\} \quad (3)$$

The link price is determined via:

$$p(t) = h(I_{\beta}(t)) \quad (4)$$

where h maps $I_{\beta}(t)$ to the link price.

$p(t)$ and $b(t)$ are thus related as follows in consistency with the PLT principle:

1. Once $b(t)$ exceeds β , the increase in $p(t)$ is determined by $h(\cdot)$ and independent of the magnitude of $b(t)$. In other words, $b(t)$ only determines when $p(t)$ will increase, but not how $p(t)$ increases.

2. When $b(t)$ goes below β , $I_{\beta}(t)$ becomes zero. It follows that $p(t)$ will be automatically reset as $h(0)$, regardless of its previous value.

We make the following mild assumption on $h(\cdot)$ to eliminate degenerate situations.

ASSUMPTION 1. $h(\cdot)$ is continuous and non-decreasing over $[0, \infty)$. Moreover,

$$\frac{\sum_{f \in \mathcal{F}} \kappa_f}{\max_{\|\mathbf{x}\|=c, \mathbf{x} \geq 0} \sum_{f \in \mathcal{F}} \frac{\kappa_f}{U'(x_f)}} > h(0) \quad (5)$$

$$\frac{\sum_{f \in \mathcal{F}} \kappa_f}{\min_{\|\mathbf{x}\|=c, \mathbf{x} \geq 0} \sum_{f \in \mathcal{F}} \frac{\kappa_f}{U'(x_f)}} < h(\infty) \quad (6)$$

(5) provides an upper bound for the link price when the backlog is under β . It ensures that the aggregate flow rate will increase with time when the backlog is below β . (6) provides a lower bound for the asymptotic link price when the backlog stays above β . It ensures that the aggregate data rate will eventually decrease with time when the backlog is above β .

DEFINITION 1. The system specified in (1)-(6) is referred to as the PLT(β) system.

Furthermore, we introduce the following concepts from the theory of dynamical systems [19].

DEFINITION 2. (Trajectory, cycle, and stable limit cycle) $\{(\mathbf{x}(t), b(t)) : t \geq 0\}$ is the trajectory of the PLT(β) system. When there exists $T > 0$ such that $(\mathbf{x}(t), b(t)) = (\mathbf{x}(t+T), b(t+T))$ for all $t \geq 0$, the trajectory is said to be a cycle. The minimum T satisfying this condition is the period of the cycle. A cycle is a stable limit cycle if all nearby trajectories asymptotically converge to it.

3. ANALYSIS OF PLT PRINCIPLE

In this section, we analyze the PLT principle rigorously by studying the PLT(β) system. Our main result is summarized by the following main theorem on the PLT(β) system:

THEOREM 1. The PLT(β) system has no equilibrium but a family of cycles, which are stable limit cycles under numerically checkable conditions. Consider $\beta \neq \tilde{\beta}$. A cycle of the PLT(β) system becomes a cycle of PLT($\tilde{\beta}$) after shifting $b(t)$ by $(\tilde{\beta} - \beta)$.

The cyclic behavior can be viewed as dynamic equilibrium. As a major consequence of the theorem, the behavior of a PLT(β) system is very robust against changes in parameters and the network environment. Such robustness is maintained across combinations of different network environments (U_f, c) and detailed protocol specifications and parameters (h, β). Moreover, the cycles are invariant across different values of β except for a shift of the backlog size. Therefore, the average backlog can be directly controlled via setting β , without affecting other aspects of the system performance.

We prove Theorem 1 in the rest of the section. Our analysis follows from the following initial observation:

PROPOSITION 1. (Recurrent return) *For any $t_0 \geq 0$, there exists $t > t_0$ such that $b(t) = \beta$.*

The proposition essentially says that the backlog $b(t)$ will recurrently return to β . The instances when $b(t) = \beta$ divide the time horizon into intervals of finite length. By capturing the dynamics over each such interval, we will have the full characterization of the system dynamics.

During these intervals, the PLT(β) system operates in either one of the two possible modes. When $b(t) > \beta$, $I_\beta(t)$ increases with time and so does the price. When $b(t) \leq \beta$, $I_\beta(t)$ becomes zero and the price will be fixed as $p(t) = h(0)$. We refer to these as Modes \mathcal{A} (bove) and \mathcal{B} (elow), respectively.

We study two continuous dynamical systems, namely, PLT- \mathcal{A} and PLT- \mathcal{B} systems, in Sections 3.1 and 3.2, respectively. With proper initialization, these two systems emulate the behavior of the PLT(β) system when it is in Mode \mathcal{A} and Mode \mathcal{B} , respectively. On this basis, we establish our main results on the PLT(β) system in Section 3.3.

3.1 PLT- \mathcal{A} System

We now consider the PLT- \mathcal{A} system that emulates the behavior of the PLT(β) system at Mode \mathcal{A} . The PLT(β) system starts operating at Mode \mathcal{A} when the backlog crosses β from below. Therefore, let $\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|}$ be the set of flow rates at that instant, we initialize the PLT- \mathcal{A} system as:

$$(\mathbf{x}(0), b(0)) = (\mathbf{R}, \beta) \quad (7)$$

It is easy to establish that $\|\mathbf{R}\| \geq c$, which is a necessary condition for the backlog to cross β from below.

Let $\tau \geq 0$ track the time elapsed since the system initialization. The dynamics of the PLT- \mathcal{A} system are specified by:

$$\dot{b}(\tau) = \sum_{f \in \mathcal{F}} x_f(\tau) - c \quad (8)$$

$$\dot{x}_f(\tau) = \kappa_f \left(1 - \frac{h(\tau)}{U'_f(x_f(\tau))} \right) \quad \forall f \in \mathcal{F} \quad (9)$$

These are almost identical to the dynamics of the PLT(β) system except that the link price becomes $h(\tau)$. This eliminates the discrete component in the dynamics of the PLT(β) system, making the analysis of the PLT- \mathcal{A} system easier. On the other hand, with the initialization in (7), (8)-(9) will exactly emulate the dynamics of the PLT(β) system at Mode \mathcal{A} up to when $b(\tau)$ goes below β .

We denote the solution to (7)-(9) by $(\mathbf{x}^{\mathcal{A}}(\mathbf{R}, \tau), b^{\mathcal{A}}(\mathbf{R}, \beta, \tau))$. We do not include β in the argument list of $\mathbf{x}^{\mathcal{A}}$ as the flow rates are obviously independent of β from (9).

To characterize the time instant after which $b^{\mathcal{A}}(\mathbf{R}, \beta, \tau)$ goes below β , we consider the following candidate, $H^{\mathcal{A}} : \{\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{R}\| \geq c\} \rightarrow [0, \infty]$, which is defined as:

$$H^{\mathcal{A}}(\mathbf{R}) \triangleq \inf\{\tau > 0 : b^{\mathcal{A}}(\mathbf{R}, \beta, \tau) = \beta\} \quad (10)$$

where we adopt the convention that $H^{\mathcal{A}}(\mathbf{R}) = \infty$ if the set on R.H.S. is empty.

The next result says that $H^{\mathcal{A}}(\mathbf{R})$ is the first time that the backlog returns to its initial level β .

PROPOSITION 2. *For $\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|}$ with $\|\mathbf{R}\| \geq c$:*

1. $H^{\mathcal{A}}(\mathbf{R})$ is independent of β .
2. $H^{\mathcal{A}}(\mathbf{R}) \in (0, \infty)$.
3. $b^{\mathcal{A}}(\mathbf{R}, \beta, \tau) > \beta$ for all $\tau \in (0, H^{\mathcal{A}}(\mathbf{R}))$.
4. $b^{\mathcal{A}}(\mathbf{R}, \beta, H^{\mathcal{A}}(\mathbf{R})) = \beta$.

REMARK 1. *Part 1 of Proposition 2 justifies our convention of excluding β from the argument list of $H^{\mathcal{A}}$. Combining with the rest of the proposition, it actually implies remarkably that the time taken for $b^{\mathcal{A}}(\mathbf{R}, \beta, \tau)$ to return to β is invariant across β .*

Now, we show that $H^{\mathcal{A}}(\mathbf{R})$ is actually the time instant after which $b^{\mathcal{A}}(\mathbf{R}, \beta, \tau)$ goes below β if we eliminate some degenerate cases.

When the backlog returns to β , the set of flow data rates $\mathbf{x}(\tau)$ will be mapped from the initial level to a new level. Formally, we define such a mapping as $\mathbf{F}^{\mathcal{A}} : \{\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{R}\| \geq c\} \rightarrow \{\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{r}\| \leq c\}$:

$$\mathbf{F}^{\mathcal{A}}(\mathbf{R}) \triangleq \mathbf{x}^{\mathcal{A}}(\mathbf{R}, H^{\mathcal{A}}(\mathbf{R})) \quad (11)$$

PROPOSITION 3. *For $\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|}$ with $\|\mathbf{R}\| \geq c$:*

1. $\|\mathbf{F}^{\mathcal{A}}(\mathbf{R})\| \leq c$.
2. If $\|\mathbf{F}^{\mathcal{A}}(\mathbf{R})\| < c$, $H^{\mathcal{A}}$ and $\mathbf{F}^{\mathcal{A}}$ are differentiable at \mathbf{R} .

REMARK 2. *For readers familiar with the dynamical systems, a remark on the subtle difference between the Poincaré map and $\mathbf{F}^{\mathcal{A}}$ is in order. In defining a Poincaré map $\tilde{\mathbf{F}}$, a hyperplane Σ tangential to a cycle of the system is selected as the Poincaré section. In (10), the constraint $b^{\mathcal{A}}(\mathbf{R}, \beta, \tau) = \beta$ determines such a Poincaré section. Let \mathbf{R} be the intersection of the cycle with the Poincaré section. $\tilde{\mathbf{F}}(\mathbf{R})$ is defined as the next intersection, which is exactly $\mathbf{F}^{\mathcal{A}}(\mathbf{R})$. However, the definition of $\tilde{\mathbf{F}}(\tilde{\mathbf{R}})$ for $\tilde{\mathbf{R}}$ near \mathbf{R} is then different. By the implicit function theorem, one can show that there exists a continuous mapping \tilde{H} such that $\mathbf{x}^{\mathcal{A}}(\tilde{\mathbf{R}}, \tilde{H}(\tilde{\mathbf{R}})) \in \Sigma$, with $\tilde{H}(\mathbf{R}) = H^{\mathcal{A}}(\mathbf{R})$. For $\tilde{\mathbf{R}}$ near \mathbf{R} , the Poincaré section is then defined as $\tilde{\mathbf{F}}(\tilde{\mathbf{R}}) \triangleq \mathbf{x}^{\mathcal{A}}(\tilde{\mathbf{R}}, \tilde{H}(\tilde{\mathbf{R}}))$. However, when $\tilde{\mathbf{R}} \neq \mathbf{R}$, there is no guarantee that the system trajectory starting from $\tilde{\mathbf{R}}$ will not intersect Σ during $(0, \tilde{H}(\tilde{\mathbf{R}}))$. When such intersection does happen, the Poincaré map $\tilde{H}(\tilde{\mathbf{R}})$ fails to capture the time for the first return.*

Now, the time derivative of the backlog at Time $H^{\mathcal{A}}(\mathbf{R})$ is $\|\mathbf{F}^{\mathcal{A}}(\mathbf{R})\| - c$. Following Proposition 3, this derivative is either negative or zero. In the former case, the backlog will go below β immediately after Time $H^{\mathcal{A}}(\mathbf{R})$. The latter

case is actually a degenerate case that the aggregate flow rate coincides with the link capacity upon the return of the backlog to β . We eliminate this degenerate case by making the following assumption in our subsequent discussion.

ASSUMPTION 2. $\|\mathbf{F}^{\mathcal{A}}(\mathbf{R})\| < c, \forall \|\mathbf{R}\| \geq c$.

REMARK 3. *The assumption actually imposes a constraint on $U_f(\cdot)$ and/or $h(\cdot)$. For example, it is satisfied when κ_f is invariant across all flows $f \in \mathcal{F}$ and $U_f(\cdot) = \log(\cdot)$. We also conjecture that it is satisfied when $h(t)$ increases fast enough in t .*

3.2 PLT- \mathcal{B} System

We now consider the PLT- \mathcal{B} system that emulates the behavior of the PLT(β) system at Mode \mathcal{B} . The PLT(β) system starts operating at Mode \mathcal{B} when the backlog crosses β from above. Therefore, let $\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|}$ be the set of flow rates at that instant (note that we actually have $\|\mathbf{r}\| \leq c$). We initialize the PLT- \mathcal{B} system as:

$$(\mathbf{x}(0), b(0)) = (\mathbf{r}, \beta) \quad (12)$$

Let $\tau \geq 0$ track the time elapsed since the system initialization. The dynamics of the PLT- \mathcal{B} system are specified by:

$$\dot{b}(\tau) = \sum_{f \in \mathcal{F}} x_f(\tau) - c \quad (13)$$

$$\dot{x}_f(\tau) = \kappa_f \left(1 - \frac{h(0)}{U_f(x_f(\tau))} \right) \quad \forall f \in \mathcal{F} \quad (14)$$

(12)-(14) will exactly emulate the dynamics of the PLT(β) system at Mode \mathcal{B} up to when $b(\tau)$ goes above β . We denote the solution to (12)-(14) by $(\mathbf{x}^{\mathcal{B}}(\mathbf{r}, \tau), b^{\mathcal{B}}(\mathbf{r}, \beta, \tau))$.

In general, the dynamics of the PLT- \mathcal{B} system are simpler than that of the PLT- \mathcal{A} system, since the link price is fixed as $h(0)$. We establish the following results in parallel to their counterparts for the PLT- \mathcal{A} system. Their interpretations are similar and will not be repeated.

The time taken for $b^{\mathcal{B}}(\mathbf{r}, \beta, \tau)$ to return to β is $H^{\mathcal{B}} : \{\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{r}\| \leq c\} \rightarrow \mathbb{R}$:

$$H^{\mathcal{B}}(\mathbf{r}) \triangleq \begin{cases} \inf\{\tau > 0 : b^{\mathcal{B}}(\mathbf{R}, \beta, \tau) = \beta\} & \|\mathbf{r}\| < c \\ 0 & \|\mathbf{r}\| = c \end{cases} \quad (15)$$

Again, this is due to the following proposition.

PROPOSITION 4. Consider $\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|}$. If $\|\mathbf{r}\| < c$, then:

1. $H^{\mathcal{B}}(\mathbf{r})$ is independent of β .
2. $H^{\mathcal{B}}(\mathbf{r}) \in (0, \infty)$.
3. $b^{\mathcal{B}}(\mathbf{r}, \beta, \tau) < \beta$ for all $t \in (0, H^{\mathcal{B}}(\mathbf{r}))$.
4. $b^{\mathcal{B}}(\mathbf{r}, \beta, H^{\mathcal{B}}(\mathbf{r})) = \beta$.

The mapping of data rates from when $\tau = 0$ to when $b^{\mathcal{B}}(\mathbf{r}, \beta, \tau)$ returns to β is defined as $\mathbf{F}^{\mathcal{B}} : \{\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{R}\| \leq c\} \rightarrow \{\mathbf{R} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{R}\| \geq c\}$:

$$\mathbf{F}^{\mathcal{B}}(\mathbf{r}) \triangleq \mathbf{x}^{\mathcal{B}}(\mathbf{r}, H^{\mathcal{B}}(\mathbf{r})) \quad (16)$$

PROPOSITION 5. For $\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|}$ with $\|\mathbf{r}\| \leq c$:

1. $\|\mathbf{F}^{\mathcal{B}}(\mathbf{r})\| > c$.
2. $H^{\mathcal{B}}$ and $\mathbf{F}^{\mathcal{B}}$ are continuous at \mathbf{r} . If $\|\mathbf{r}\| < c$, they are differentiable at \mathbf{r} .

3.3 Complete PLT(β) System

We now come back to the PLT(β) system. We first look at the system trajectory (Proposition 6). On this basis, we study the cycles (Propositions 7-8). These lead to the establishment of the main result, Theorem 1.

The next result says that the trajectory is periodic in which the behavior of the backlog alternates between Modes \mathcal{A} and \mathcal{B} deterministically. The time it stays in Mode \mathcal{X} is given by $H^{\mathcal{X}}$ ($\mathcal{X} = \mathcal{A}, \mathcal{B}$). Within Mode \mathcal{X} , the system state is determined by $(\mathbf{x}^{\mathcal{X}}, b^{\mathcal{X}})$. The set of flow rates when the system enters Mode \mathcal{X} is mapped to the set of flow rates when the system exits Mode \mathcal{X} via $\mathbf{F}^{\mathcal{X}}$.

PROPOSITION 6. (Trajectory) Let $s[1]$ be the first time instant when $b(t) = \beta$ with $\|\mathbf{x}(t)\| < c$. Denote the instant when $b(t)$ returns to β for the $(k-1)$ th time after $s[1]$ as $s[k]$. Let $\mathcal{M}[k] \in \{\mathcal{A}, \mathcal{B}\}$ be the mode of the PLT(β) system during $(s[k], s[k+1])$.

1. The mode of the PLT(β) system is:

$$\mathcal{M}[k] = \begin{cases} \mathcal{A} & k \text{ is even} \\ \mathcal{B} & k \text{ is odd} \end{cases} \quad (17)$$

2. The duration of the k th interval is:

$$s[k+1] - s[k] = H^{\mathcal{M}[k]}(\mathbf{x}(s[k])) \quad (18)$$

3. For $t \in (s[k], s[k+1])$,

$$\mathbf{x}(t) = \mathbf{x}^{\mathcal{M}[k]}(\mathbf{x}(s[k]), t - s[k]) \quad (19)$$

$$b(t) = b^{\mathcal{M}[k]}(\mathbf{x}(s[k]), \beta, t - s[k]) \quad (20)$$

4. $x(s[k+1]) = \mathbf{F}^{\mathcal{M}[k]}(x(s[k]))$. Moreover, $x(s[k]) > c$ if k is odd, and $x(s[k]) < c$ if k is even.

PROOF. We prove it by induction.

We first establish the case for $k = 1$ and 2. During $(s[1], s[2])$, we start with $\|\mathbf{x}(s[1])\| < c$. Thus, $b(t)$ has a negative derivative at $t = s[1]$, and will drop below β . Therefore, the system will operate in Mode \mathcal{B} and behaves as the PLT- \mathcal{B} system before $b(t)$ returns to β at $t = s[2]$. (18)-(20) then follow with $\mathcal{M}[1] = \mathcal{B}$. Moreover,

$$\mathbf{x}(s[2]) = \mathbf{F}^{\mathcal{B}}(\mathbf{x}(s[1])) \quad (21)$$

By Proposition 5, we have $\mathbf{x}(s[2]) > c$.

Thus, $b(t)$ has a positive derivative at $t = s[2]$, and will exceed β . Thus, during $(s[2], s[3])$, the system will operate in Mode \mathcal{A} and behaves as the PLT- \mathcal{A} system before $b(t)$ returns to β at $t = s[3]$. (18)-(20) then follow with $\mathcal{M}[2] = \mathcal{A}$. Moreover, by Assumption 2, we have $\|\mathbf{x}(s[3])\| < c$.

Now, suppose that the result holds for all $k \leq K-1$. If K is odd, then $K-1$ is even. It follows from the supposition that $\mathcal{M}[K-1] = \mathcal{B}$, $\|\mathbf{x}(s[K-1])\| < c$. Thus, $x(s[K]) = \mathbf{F}^{\mathcal{A}}(\mathbf{x}(s[K-1]))$ and, by Proposition 5, $\|\mathbf{x}(s[K])\| > c$. Thus, during $(s[K], s[K+1])$, $b(t)$ will go above β and the PLT(β) system operate in Mode \mathcal{A} . (18)-(20) then follow with $\mathcal{M}[K] = \mathcal{A}$. Moreover,

$$\mathbf{x}(s[K+1]) = \mathbf{F}^{\mathcal{A}}(\mathbf{x}(s[K])) \quad (22)$$

Thus, all the results hold for $k = K$.

Similar arguments apply to the case when K is even. \square

Therefore, the system constantly switches between the two modes of operations, implying the *non-existence* of equilibrium for the system.

COROLLARY 1. *The $PLT(\beta)$ system has no equilibrium.*

Now, let $\mathbf{F} \triangleq \mathbf{F}^A \circ \mathbf{F}^B$, and \mathbf{F}^n denote the n th composites of \mathbf{F} . If $\mathbf{x}(t)$ starts off at a fixed point of \mathbf{F}^n , the system state $(\mathbf{x}(t), b(t))$ will have identical states at $t = s[1]$ and $t = s[2n+1]$, thereby exhibiting the cyclic behavior. It turns out that all cycles of the $PLT(\beta)$ system can be identified via the fixed points of \mathbf{F}^n .

A few definitions are in order.

DEFINITION 3. *Let*

$$\mathbb{I} \triangleq \{\mathbf{r} \in \mathbb{R}_+^{|\mathcal{F}|} : \|\mathbf{r}\| \leq c \text{ and } \mathbf{F}^n(\mathbf{r}) = \mathbf{r} \text{ for some } n\} \quad (23)$$

and, for $\mathbf{r} \in \mathbb{I}$,

$$\gamma(\mathbf{r}) \triangleq \min\{n : \mathbf{F}^n(\mathbf{r}) = \mathbf{r}\} \quad (24)$$

For $\mathbf{r}, \tilde{\mathbf{r}} \in \mathbb{I}$, \mathbf{r} is said to be equivalent to $\tilde{\mathbf{r}}$, written as $\mathbf{r} \sim \tilde{\mathbf{r}}$, if $\gamma(\mathbf{r}) = \gamma(\tilde{\mathbf{r}})$ and $\tilde{\mathbf{r}} = \mathbf{F}^m(\mathbf{r})$ for some m . The equivalent class of \mathbf{r} , written as $[\mathbf{r}]$, is a subset of \mathbb{I} that contains all elements equivalent to \mathbf{r} .²

PROPOSITION 7. *(Cycles and stable limit cycles)*

1. \mathbb{I} is non-empty.
2. There is one-to-one correspondence between the equivalent classes of \mathbb{I} and cycles of the $PLT(\beta)$ system.
3. A cycle corresponding to $[\mathbf{r}]$ is a stable limit cycle if the radius of the Jacobian matrix $D\mathbf{F}^{\gamma(\mathbf{r})}(\mathbf{r})$ is less than one.

PROOF. 1) Observe that \mathbf{F} is a continuous mapping over a convex compact subset of $\mathbb{R}^{|\mathcal{F}|}$. By the Brouwer fixed point theorem, \mathbf{F} has a fixed point. Thus, \mathbb{I} has at least a fixed point of \mathbf{F} .

2) Denote the set of all equivalent classes in \mathbb{I} by \mathbb{I}/\sim and the set of cycles of the $PLT(\beta)$ system by \mathbb{C} . We now construct a bijective mapping $\varphi : \mathbb{I}/\sim \rightarrow \mathbb{C}$.

Let $\varphi([\mathbf{r}])$ be the trajectory of the $PLT(\beta)$ system when $\mathbf{x}(0) = \mathbf{r}$ and $b(0) = \beta$. By Part 2 of Proposition 6, $\mathbf{x}(t)$ will return to $\mathbf{x}(0)$ when $b(t)$ returns to β for the $2\gamma(\mathbf{r})$ th time. Thus, $\varphi([\mathbf{r}])$ is a cycle.

Claim 1: φ is well defined. In other words, if $\mathbf{r} \sim \tilde{\mathbf{r}}$, $\varphi([\mathbf{r}]) = \varphi([\tilde{\mathbf{r}}])$.

By definition, $\tilde{\mathbf{r}} = \mathbf{F}^m(\mathbf{r})$ for some $m < \gamma(\mathbf{r})$. Thus, when the system starts with $\mathbf{x}(0) = \mathbf{r}$ and $b(0) = \beta$, $\mathbf{x}(t)$ will reach $\tilde{\mathbf{r}}$ when $b(t)$ returns to β for the $(2m)$ th time (say, at time T').

Now, suppose $(\bar{\mathbf{x}}, \bar{b}) \in \varphi([\tilde{\mathbf{r}}])$. When the system starts with $\mathbf{x}(0) = \tilde{\mathbf{r}}$ and $b(0) = \beta$, we have $(\mathbf{x}(t_0), b(t_0)) = (\bar{\mathbf{x}}, \bar{b})$ for some $t_0 > 0$. It follows that, when the system starts with $\mathbf{x}(0) = \mathbf{r}$ and $b(0) = \beta$, $(\mathbf{x}(t_0 + T'), b(t_0 + T')) = (\bar{\mathbf{x}}, \bar{b})$. This implies $(\bar{\mathbf{x}}, \bar{b}) \in \varphi([\mathbf{r}])$, and hence $\varphi([\tilde{\mathbf{r}}]) \subset \varphi([\mathbf{r}])$.

By the symmetry of equivalence relation, we can similarly establish $\varphi([\tilde{\mathbf{r}}]) \supset \varphi([\mathbf{r}])$. Thus, $\varphi([\tilde{\mathbf{r}}]) = \varphi([\mathbf{r}])$.

Claim 2: φ is injective. In other words, $\varphi([\mathbf{r}]) = \varphi([\tilde{\mathbf{r}}])$ implies $[\mathbf{r}] = [\tilde{\mathbf{r}}]$, or $\mathbf{r} \sim \tilde{\mathbf{r}}$.

Since $(\tilde{\mathbf{r}}, \beta) \in \varphi([\tilde{\mathbf{r}}])$, we have $(\tilde{\mathbf{r}}, \beta) \in \varphi([\mathbf{r}])$. Therefore, when starting with $(\mathbf{x}(0), b(0)) = (\mathbf{r}, \beta)$, $\mathbf{x}(t)$ reaches $\tilde{\mathbf{r}}$ when

²It is straightforward to verify that this is a well-defined equivalence relation. In other words, it satisfies reflexivity ($\mathbf{r} \sim \mathbf{r}$), symmetry ($\mathbf{r} \sim \tilde{\mathbf{r}} \Rightarrow \tilde{\mathbf{r}} \sim \mathbf{r}$), and transitivity ($\mathbf{r} \sim \tilde{\mathbf{r}} \wedge \tilde{\mathbf{r}} \sim \hat{\mathbf{r}} \Rightarrow \mathbf{r} \sim \hat{\mathbf{r}}$).

$b(t)$ returns to β for the k th time, for some k . Moreover, since $\|\tilde{\mathbf{r}}\| < c$, by Proposition 6, k is even and $\tilde{\mathbf{r}} = \mathbf{F}^{\frac{k}{2}}(\mathbf{r})$. Thus, $\mathbf{r} \sim \tilde{\mathbf{r}}$.

Claim 3: φ is surjective. In other words, any cycle of the system can be constructed from some $\mathbf{r} \in \mathbb{I}$ via the prescribed mapping.

Let $\Gamma = \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq 0\}$ be a cycle of the system with trajectory T . Pick an instant t_0 when $\|\mathbf{x}(t_0)\| < c$ and $b(t_0) = \beta$. The existence of such an instance is guaranteed by Propositions 1 and 5.

We now show that $\varphi(\mathbf{x}(t_0)) = \Gamma$.

For $t < t_0$, there exists a positive integer n such that $t + nT > t_0$. We have by periodicity of cycles of the system:

$$(\tilde{\mathbf{x}}(t), \tilde{b}(t)) = (\tilde{\mathbf{x}}(t + nT), \tilde{b}(t + nT)) \quad (25)$$

It follows that:

$$\{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : 0 \leq t < t_0\} \subset \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq t_0\} \quad (26)$$

Therefore,

$$\begin{aligned} \Gamma &= \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq 0\} \\ &= \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : 0 \leq t < t_0\} \cup \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq t_0\} \\ &= \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq t_0\} \end{aligned} \quad (27)$$

By definition and time-invariance of the system, we have:

$$\varphi(\mathbf{r}) = \{(\tilde{\mathbf{x}}(t), \tilde{b}(t)) : t \geq t_0\} \quad (28)$$

Thus, $\Gamma = \varphi(\mathbf{r})$ and the claim follows.

Combining Claims 1-3, we establish Part 2 of the theorem.

3) The condition implies that the mapping $\mathbf{F}^{\gamma(\mathbf{r})}$ is locally asymptotically stable, which in turn implies that the cycle is locally asymptotically stable, or a stable limit cycle. \square

REMARK 4. $D\mathbf{F}^{\gamma(\mathbf{r})}(\mathbf{r})$ can be expanded as:

$$D\mathbf{F}(\hat{\mathbf{x}}[\gamma(\mathbf{r})])D\mathbf{F}(\hat{\mathbf{x}}[\gamma(\mathbf{r}) - 1]) \cdots D\mathbf{F}(\hat{\mathbf{x}}[1]) \quad (29)$$

where $\hat{\mathbf{x}}[1] = \mathbf{r}$ and $\hat{\mathbf{x}}[k+1] = \mathbf{F}(\hat{\mathbf{x}}[k])$ for $k \geq 1$. $D\mathbf{F}(\hat{\mathbf{x}}[k])$ can in turn be written as:

$$D\mathbf{F}^A(\mathbf{F}^B(\hat{\mathbf{x}}[k]))D\mathbf{F}^B(\hat{\mathbf{x}}[k]) \quad (30)$$

All the matrices involved in (30) can be computed numerically by applying standard techniques for continuous dynamical systems. Interested readers can refer to Chapter 3 of [27] for details.

PROPOSITION 8. *(Invariance property)* Consider $\beta > 0$ and $\tilde{\beta} > 0$. A cycle of $PLT(\beta)$ becomes a cycle of $PLT(\tilde{\beta})$ after shifting $b(t)$ by $(\tilde{\beta} - \beta)$. Both cycles correspond to the same equivalent class in \mathbb{I} .

PROOF. Let Γ be a cycle of $PLT(\beta)$ corresponding to the equivalence class $[\mathbf{r}]$. Similar to the proof of Proposition 7, Γ is in fact the trajectory of the $PLT(\beta)$ system starting from $\mathbf{x}(0) = \mathbf{r}$ and $b(0) = \beta$. Thus, by Proposition 6, we have:

$$\Gamma = \cup_{k=1}^{\infty} \Gamma[k] \quad (31)$$

$$\begin{aligned} \Gamma[k] &\triangleq \{(\mathbf{x}^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k], \tau), b^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k], \beta, \tau)) : \\ &\quad \tau \in [0, H^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k])]\} \end{aligned} \quad (32)$$

where $\hat{\mathbf{x}}[k] \triangleq \mathbf{x}(s[k])$. Since $\hat{\mathbf{x}}[k]$ returns to $\hat{\mathbf{x}}[1]$ when $k = \gamma(\mathbf{r}) + 1$, we have $\Gamma[k] = \Gamma[k + \gamma(\mathbf{r})]$. It follows that:

$$\Gamma = \cup_{k=1}^{\gamma(\mathbf{r})} \Gamma[k] \quad (33)$$

Let $\tilde{\Gamma}$ be a cycle of $\text{PLT}(\tilde{\beta})$ corresponding to the equivalence class $[\mathbf{r}]$. Similarly, we have:

$$\tilde{\Gamma} = \cup_{k=1}^{\gamma(\mathbf{r})} \tilde{\Gamma}[k] \quad (34)$$

$$\begin{aligned} \tilde{\Gamma}[k] \triangleq & \{(\mathbf{x}^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k], \tau), b^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k], \tilde{\beta}, \tau)) : \\ & \tau \in [0, H^{\mathcal{M}[k]}(\hat{\mathbf{x}}[k])]\} \end{aligned} \quad (35)$$

Now, noting that, for either $\mathcal{X} = \mathcal{A}$ or $\mathcal{X} = \mathcal{B}$,

$$\begin{aligned} & b^{\mathcal{X}}(\mathbf{R}, \tilde{\beta}, \tau) \\ = & \tilde{\beta} + \int_0^\tau (\|\mathbf{x}^{\mathcal{X}}(\mathbf{R}, \tilde{\tau})\| - c) d\tilde{\tau} \\ = & (\tilde{\beta} - \beta) + \underbrace{(\beta + \int_0^\tau (\|\mathbf{x}^{\mathcal{X}}(\mathbf{R}, \tilde{\tau})\| - c) d\tilde{\tau})}_{b^{\mathcal{X}}(\mathbf{R}, \beta, \tau)} \end{aligned} \quad (36)$$

It follows that:

$$\tilde{\Gamma}[k] = \Gamma[k] + (\mathbf{0}, \tilde{\beta} - \beta) \quad (37)$$

where $\mathbf{0}$ denotes the zero vector of dimension $|\mathcal{F}|$. Hence,

$$\begin{aligned} \tilde{\Gamma} &= \cup_{k=1}^{\gamma(\mathbf{r})} \tilde{\Gamma}[k] \\ &= \cup_{k=1}^{\gamma(\mathbf{r})} \Gamma[k] + (\mathbf{0}, \tilde{\beta} - \beta) \\ &= \Gamma + (\mathbf{0}, \tilde{\beta} - \beta) \end{aligned} \quad (38)$$

□

Theorem 1 follows directly from Propositions 6-8.

4. EXAMPLE AQM ALGORITHMS

In this section, we describe CoDel and DCTCP and show that their models are special cases of the $\text{PLT}(\beta)$ system. This demonstrates analytically that CoDel and DCTCP implement the PLT principle.

4.1 CoDel

CoDel [25] has been devised on the principle of differentiating between “good queues” and “bad queues” of packets at router buffers. Good queues are queues built up due to the arrival of bursty traffic. Such queues are in general transient. Bad queues refer to persistent queues due to the aggressive transmission behavior of TCP.

To differentiate, CoDel selects a target level, *target*, and monitors the queueing delay experienced by packets. When the queueing delay exceeds *target* for a time period *interval*, CoDel concludes that a bad queue is present and activates a control law for dropping packets. CoDel first drops a packet to induce a TCP backoff. If the queueing delay is persistently above *target*, the time period for the next packet drop is inversely proportional to the square root of the number of drops since entering the dropping period.

Let $\beta = c \cdot \text{target}$ and $I_0 = \text{interval}$. $I_\beta(t)$ then corresponds to the duration of time elapsed since the queueing delay exceeds target at Time t . Suppose that n packet drops have occurred over this time period $I_\beta(t)$, we have:

$$I_0 + \sum_{j=1}^{n-1} \frac{I_0}{\sqrt{j}} \approx I_\beta(t) \quad (39)$$

which gives $n \approx \rho(\frac{I_\beta(t)}{I_0})$ for some increasing function ρ . For simplicity, we may take fractional values for n and assume

that ρ is continuous. Since no drops occur when the backlog is below β (i.e. $I_\beta(t) = 0$), we can set $\rho(0) = 0$.

The time elapsed between the $(n-1)$ th drop and the n th drop is $\frac{I_0}{\sqrt{n}}$. During this interval, there are approximately $c \frac{I_0}{\sqrt{n}}$ arriving packets, one of which will be dropped. Therefore, the dropping rate (as the link price) at Time t is approximately:

$$p(t) = \frac{1}{c \frac{I_0}{\sqrt{n}}} \approx \frac{1}{c I_0} \sqrt{\rho(\frac{I_\beta(t)}{I_0})} \quad (40)$$

Therefore, the model of CoDel amounts to a special case of the $\text{PLT}(\beta)$ system with $h(\cdot) = \frac{1}{c I_0} \sqrt{\rho(\frac{\cdot}{I_0})}$.

4.2 DCTCP

Data center TCP (DCTCP) [1] adapts congestion control to the needs of data centers in keeping end-to-end delay minimal. It involves modifications to both TCP and AQM. Congestion notifications are conveyed from AQM to TCP by marking packets instead of actually dropping packets. Therefore, link price corresponds to the packet marking rate. When the number of packets buffered is K , every arrived packet will be marked. Otherwise, they are not marked. At the end of each round-trip time (RTT) round, a DCTCP source computes the fraction of packets marked (say α), and reduces the window by a factor of $(1 - \frac{\alpha}{2})$.

A model for DCTCP has been developed in [2]. With some simplifying assumptions, we present an equivalent form in the framework of the $\text{PLT}(\beta)$ system. Let S be the average packet size and $\beta = K \cdot S$. For a DCTCP flow f , let the RTT be d_f and the congestion window size be $w_f(t)$. It follows that $x_f(t) = \frac{w_f(t)}{d_f}$. In each round-trip time, $w_f(t)$ is increased by S . If the fraction of marked packets is $\alpha(t)$, $w_f(t)$ is reduced by a factor of $\frac{\alpha(t)}{2}$. Thus, the net change of $w_f(t)$ on every d_f is:

$$S - \frac{\alpha(t)w_f(t)}{2} \quad (41)$$

It follows that:

$$\begin{aligned} x_f(t) &= \frac{\dot{w}_f(t)}{d_f} \\ &\approx \frac{1}{d_f^2} (S - \frac{\alpha(t)w_f(t)}{2}) \end{aligned} \quad (42)$$

If $I_\beta(t) > d_f$, the number of packets at the router buffer is above K for the whole RTT. All packets will be marked and thus $\alpha(t) = 1$. Otherwise, around a proportion $\frac{I_\beta(t)}{d_f}$ of the packets will be marked and $\alpha(t) \approx \frac{I_\beta(t)}{d_f}$. Therefore,

$$\alpha(t) \approx \sigma(\frac{I_\beta(t)}{d_f}) \quad (43)$$

where

$$\sigma(x) \triangleq \begin{cases} x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases} \quad (44)$$

We can approximate σ with an arbitrarily small error by a non-decreasing, smooth function, say, $\tilde{\sigma}$. It follows that:

$$\alpha(t) \approx \tilde{\sigma}(\frac{I_\beta(t)}{d_f}) \quad (45)$$

Therefore, the model of DCTCP amounts to a special case of the PLT(β) system with $U_f(\cdot) = \frac{2}{d_f} \log(\cdot)$, $\kappa_f = \frac{1}{d_f^2}$, and $h(\cdot) = \tilde{\sigma}(\frac{\cdot}{d_f})$.

5. PERFORMANCE VALIDATION

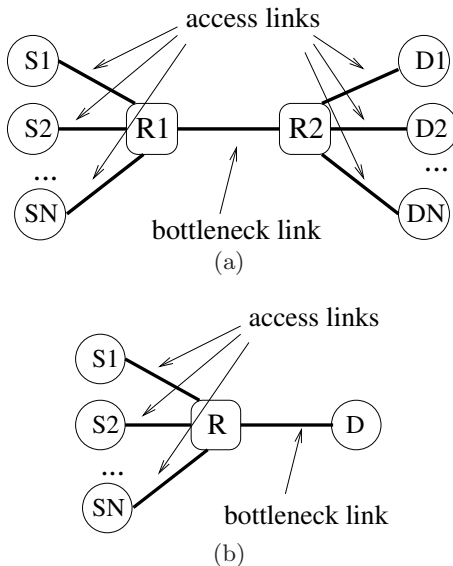


Figure 2: Simulation topologies.

In this section, we present our simulation results. Recall that, in Section 3, our analysis predicts that 1) congestion control systems implementing the PLT principle will exhibit the cyclic behavior, and 2) the cycles are independent of β except for a shift in the backlog level. In this section, we conduct our simulation using Network Simulator (ns) Version 2.35 to validate these results in realistic network settings, complementing the theoretical study in the following manners:

1. The PLT(β) system approximates the network dynamics at the flow level by modeling the data traffic as a continuous stream of data with time-varying rates. Our simulation has been conducted at the finer packet level.
2. In congestion control systems, the source observes time-lagged link price and transmits data on the link with further time lag. Such delays are pertinent to the system stability but hard to be incorporated with nonlinear analysis that involves the cycles. ns-2 automatically simulates such delays as an event-driven simulator.
3. We include some classical AQM algorithms in the literature for comparison. This helps identify the unique features of the PLT principle.

Rather than specifying a particular AQM algorithm, the PLT principle gives rise to a family of congestion control algorithms. While the analysis of the PLT(β) system studies the general PLT principle, a simulation-based study necessarily focuses on its special realizations. Therefore, our simulations examine CoDel and DCTCP that have been shown in Section 4 to be two realizations of the PLT principle.

The classical AQM algorithms compared are RED [7] and the proportional-integral controller (PI) [16]. RED is the first AQM algorithm and by far the most popular one. It maintains a running average of the backlog size, b_{aver} , and

compares it to two threshold parameters, namely, $minThresh$ and $maxThresh$. The packet dropping rate is then set as:

$$\begin{cases} 0 & b_{aver} \leq minThresh \\ 1 & b_{aver} \geq maxThresh \\ \frac{p_{max}(b_{aver} - minThresh)}{maxThresh - minThresh} & \text{otherwise} \end{cases}$$

where the parameter p_{max} is the maximum packet dropping rate when b_{aver} falls between $minThresh$ and $maxThresh$, the intended operational region of RED. Obviously, it is a challenge for setting p_{max} properly as there is a trade off between exerting prompt response to congestion and avoiding over-control.³

PI is another popular AQM algorithm. Along with REM [5], PI is representative of a class of AQM algorithms that track the backlog to a preselected threshold value. This is done by adjusting the packet dropping rate based on the deviation of the backlog from the threshold value. Such scheme bears some resemblance to the PLT principle. However, PLT essentially interprets the deviation in a discrete manner by only looking at whether the deviation is positive or negative. In contrast, PI and similar AQM algorithms take into account the actual magnitude of the deviation. This makes their performance highly sensitive to parameters relating the magnitude of backlog deviation to the packet dropping rate.

Specifically, PI records the backlog size in the previous and current time steps, say, as q_{old} and q_{new} , respectively. In each time step, it adds to the packet dropping rate:

$$a(q_{new} - q_{ref}) + b(q_{new} - q_{old}) \quad (46)$$

where a and b are weighting parameters, and q_{ref} is the preselected threshold value. For PI to function properly, a and b need to be carefully selected based on the network settings [16].

Two simulation topologies are investigated, as shown in Fig. 2. In Fig. 2(a), the sources and destinations of N TCP connections are connected to $R1$ and $R2$, routers at two ends of a bottleneck link, via access links, respectively. In Fig. 2(b), the sources of N TCP connections are connected to a router at one end of a bottleneck link via access links. These connections share a common destination at the other end of the bottleneck link. The configuration settings are summarized in Table 2. C_b , d_b , C_a , and d_a denote the bandwidth and delay of the bottleneck link, and the bandwidth and the delay of an access link, respectively. The final column of the table corresponds to $target$ for CoDel, K for DCTCP, and q_{ref} for PI. Recall that, for CoDel, the parameter $target$ is essentially β divided by the bottleneck bandwidth C_b (Section 4.1). For DCTCP, the parameter K is essentially β normalized by the packet size (Section 4.2). Therefore, changes in $target$ and K induce proportional changes in the effective value of β for CoDel and DCTCP, respectively.

Section 5.1 examines the cyclic behavior of CoDel and compares it with RED and PI. Section 5.2 examines the invariance of cycles of CoDel and DCTCP. While the simulation results validate most of our analytic results, we do identify one major discrepancy due to modeling simplifications, which we examine in Section 5.3.

³We have also tried adaptive RED (ARED) [11], an RED variant that dynamically adapts the parameters, but it behaves similarly as RED.

Table 2: Network configurations

Figure	TCP:AQM	Topology	N	C_a	d_a	C_b	d_b	$target / K / q_{ref}$
Fig. 3(a)	Reno:CoDel	Fig. 2(a)	60	150 Mbps	40 ms	15 Mbps	20 ms	50 ms
Fig. 3(b)	Reno:RED	Fig. 2(a)	60	150 Mbps	40 ms	15 Mbps	20 ms	-
Fig. 3(c)	Reno:CoDel	Fig. 2(a)	30	750 Mbps	40 ms	75 Mbps	20 ms	50 ms
Fig. 3(d)	Reno:RED	Fig. 2(a)	30	750 Mbps	40 ms	75 Mbps	20 ms	-
Fig. 4(a)	Reno:CoDel	Fig. 2(a)	60	150 Mbps	40 ms	15 Mbps	20 ms	50 ms
Fig. 4(b)	Reno:PI	Fig. 2(a)	60	150 Mbps	40 ms	15 Mbps	20 ms	93.75 Kbytes
Fig. 4(c)	Reno:CoDel	Fig. 2(a)	60	750 Mbps	40 ms	75 Mbps	20 ms	50 ms
Fig. 4(d)	Reno:PI	Fig. 2(a)	60	750 Mbps	40 ms	75 Mbps	20 ms	468.75 Kbytes
Fig. 5(a)	Reno:CoDel	Fig. 2(a)	15	1 Gbps	20 ms	100 Mbps	10 ms	{20 ms, 30 ms, 40 ms}
Fig. 5(b)	Reno:CoDel	Fig. 2(a)	15	1 Gbps	20 ms	100 Mbps	10 ms	{20 ms, 30 ms, 40 ms}
Fig. 6(a)	Reno:CoDel	Fig. 2(a)	30	1 Gbps	20 ms	100 Mbps	10 ms	{20 ms, 30 ms, 40 ms}
Fig. 6(b)	Reno:CoDel	Fig. 2(a)	15	1 Gbps	20 ms	50 Mbps	10 ms	{20 ms, 30 ms, 40 ms}
Fig. 6(c)	Reno:CoDel	Fig. 2(a)	15	1 Gbps	10 ms	100 Mbps	5 ms	{20 ms, 30 ms, 40 ms}
Fig. 7(a)	DCTCP	Fig. 2(b)	8	11 Gbps	0.025 ms	10 Gbps	0.025 ms	{45, 65, 85}
Fig. 7(b)	DCTCP	Fig. 2(b)	8	11 Gbps	0.05 ms	10 Gbps	0.05 ms	{45, 65, 85}

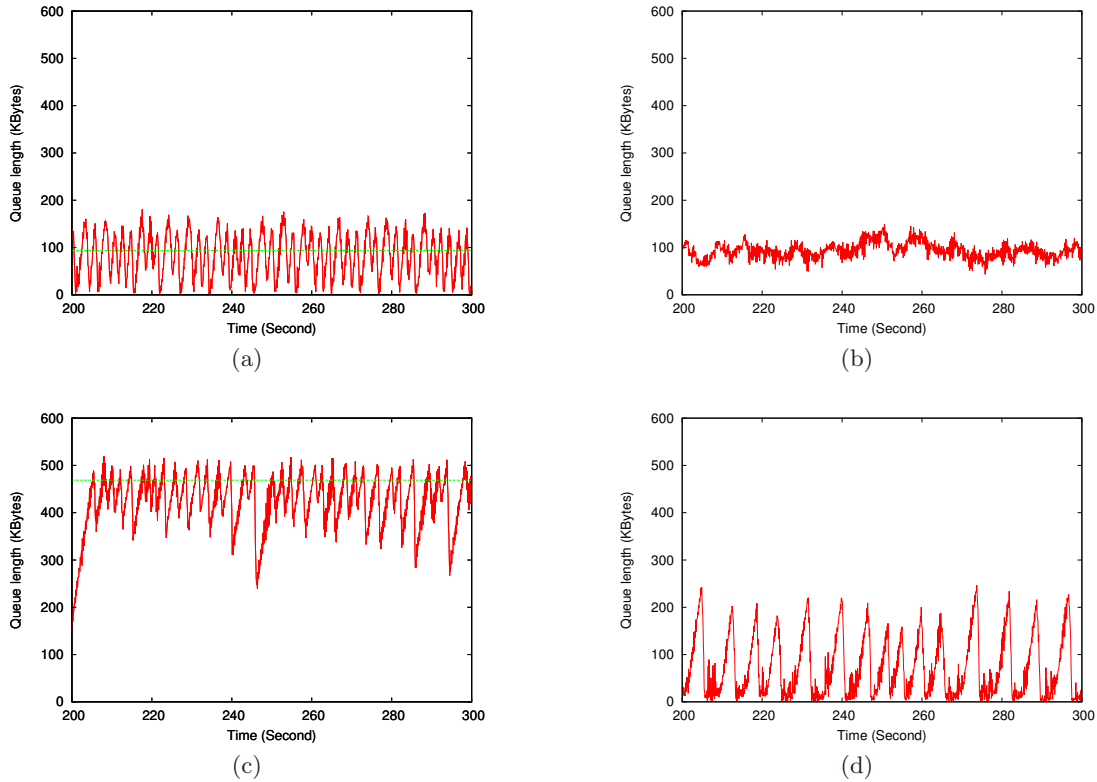


Figure 3: Comparison between CoDel and RED.

5.1 Cyclic Behavior

Fig. 3(a)-(b) compare the queue dynamics of CoDel and RED. The effective value of β for CoDel is plotted horizontally in Fig. 3(a) as well. The network configuration and the parameters of RED follow those in [16]. Cyclic pattern can be clearly identified in the backlog with CoDel (Fig. 3(a)) as predicted.⁴ On the other hand, no cyclic pattern can

⁴Indeed, this pattern is approximate but not exact. In loss-based congestion control, a TCP connection backs off when a packet loss occurs. On the other hand, the flow rate adap-

be identified in the backlog with RED (Fig. 3(b)). In this case, RED essentially keeps the backlog fluctuation small by computing the link price as a well-tuned function of the *magnitude* of the backlog.

tation process of the $PLT(\beta)$ system in (1) adopts the well-established fluid model [23], which eliminates the randomness by looking at the *expected* flow rate. The queueing dynamics, which are determined by the aggregate flow rates, in turn inherit similar modeling errors.

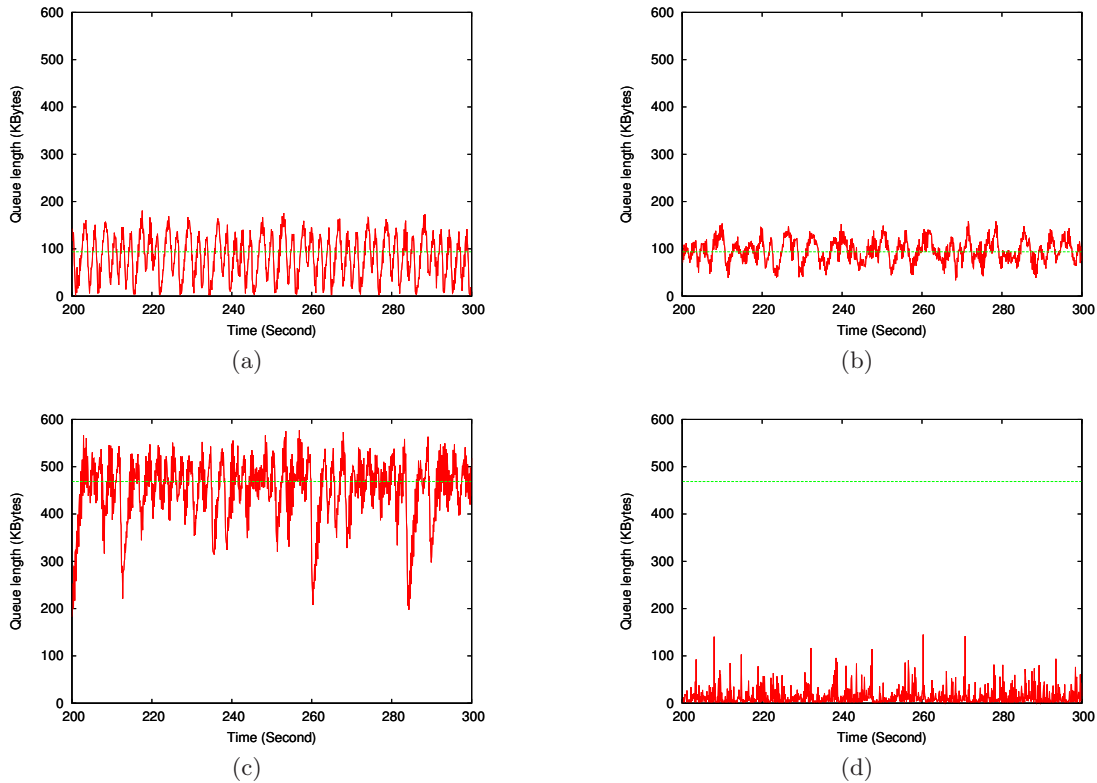


Figure 4: Comparison between CoDel and PI.

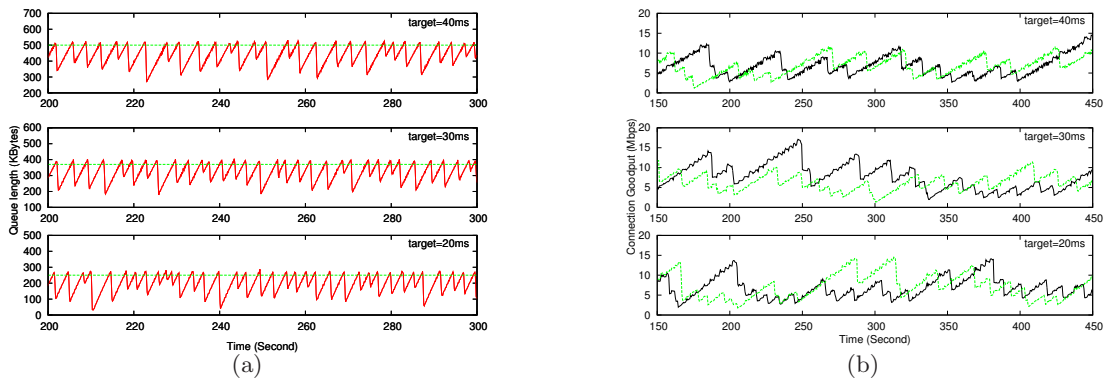


Figure 5: Invariance of cycles with CoDel: base case.

Fig. 3(c)-(d) plot the queue dynamics of CoDel and RED, respectively, when the bandwidth of the bottleneck link is increased by five-fold and the number of connections is reduced by half with the parameters of CoDel and RED unchanged. The backlog with CoDel continues to exhibit the regular cyclic pattern. Moreover, the backlog is non-empty, keeping the bottleneck link fully utilized. On the other hand, without parameters well tuned to the new network environment, RED fails to maintain its intended operational behavior. Rather, the backlog is often driven to become empty due to excessive packet dropping when the link is backlogged. This severely undermines the link utilization.

Fig. 4(a)-(b) compare the queue dynamics of CoDel and PI. In addition, the effective value of β for CoDel and the

value of q_{ref} for PI are plotted horizontally in Fig. 4(a)-(b), respectively. The network configuration and the parameter of PI follow those in [16]. We set $target$ of CoDel such that the resulting β is equal to q_{ref} of PI for better comparison. Again, the cyclic pattern can be clearly identified in the backlog with CoDel (Fig. 4(a)) but not in the backlog with PI (Fig. 4(b)). PI demonstrates a smaller magnitude of the backlog deviation. This is because PI explicitly counteract the deviation by taking its magnitude and derivative into account when deciding the packet dropping rate. It is intended for the backlog to closely track q_{ref} .

Fig. 4(c)-(d) plot the queue dynamics of CoDel and PI, respectively, when the bandwidth of the bottleneck link is increased by five-fold with the parameters of CoDel and PI

unchanged. CoDel continues to exhibit the regular cyclic behavior in keeping the backlog non-empty and evolve following the cyclic pattern. On the other hand, without parameters well tuned to the new network environment, PI fails to maintain its intended operational behavior. Rather, the backlog is almost always empty.

Therefore, the cyclic behavior of CoDel is in general robust across different network environments *without* careful tuning of parameters. While the fluctuation in the backlog with CoDel is more severe than those with carefully tuned conventional AQM algorithms, CoDel is nevertheless able to maintain a low buffer occupancy in the long run. In practice, this gives network administrators incentives to simply “turn it on” rather than leaving the buffers unmanaged. In general, the AQM and congestion control algorithms implementing the PLT principle are expected to exhibit such robust cyclic behavior. For example, such similar cyclic behavior has been reported in the case of DCTCP in [2].

5.2 Invariance of Cycles

We now examine the invariance of the cycles against the threshold β .

Fig. 5(a) plots the queue dynamics of CoDel under three different values of *target*, which results in three different values of β as plotted horizontally. Fig. 6(b) plots the corresponding dynamics in goodput of two TCP connections. In Fig. 5(a), the dynamic deviation of the queue dynamics from their respective levels of β are remarkably similar across different levels of β . In Fig. 5(b), the goodput dynamics are also similar across different values of β , although the cyclic pattern is not clear in these dynamics due to the randomness in the backoffs of a single connection. Thus, we have clearly observed the invariance property, that the cycles of $PLT(\beta)$ system is invariant across different values of β , except for a shift in the backlog size.

Fig. 6(a)-(c) plot the queue dynamics of CoDel when the network is changed by doubling the number of TCP flows, halving the bottleneck link capacity, and halving the delays of all links, respectively. It can be seen that, in each resulting network scenario, similar inferences to those from Fig. 5(a) can be made. Moreover, the dynamics in these figures are obviously different from those of Fig. 5(a). Hence, the queue dynamics around β are determined by the network characteristics and invariant across different values of β .

Fig. 7 plots the queue dynamics of DCTCP under different levels of K , which result in three different values of β horizontally plotted. All other configurations follow that in [1]. We observe that the queue dynamics across different values of β are similar except for the shift. In comparison with CoDel, the cycling period is much shorter and the magnitude deviation is much smaller. This is due to the high-speed, low-latency links in the simulated data center environment, which makes the round-trip time minimal.

Therefore, the invariance properties of CoDel and DCTCP are verified. This property greatly simplifies their parameter tuning. When configuring β , one only needs to worry about the desirable level of delay, and avoid overly small β that leads to idle links.

5.3 Discussion: Lower Bound for β

In addition to the plotted results, we have explored various levels of β . It is found that there is a lower bound on β ,

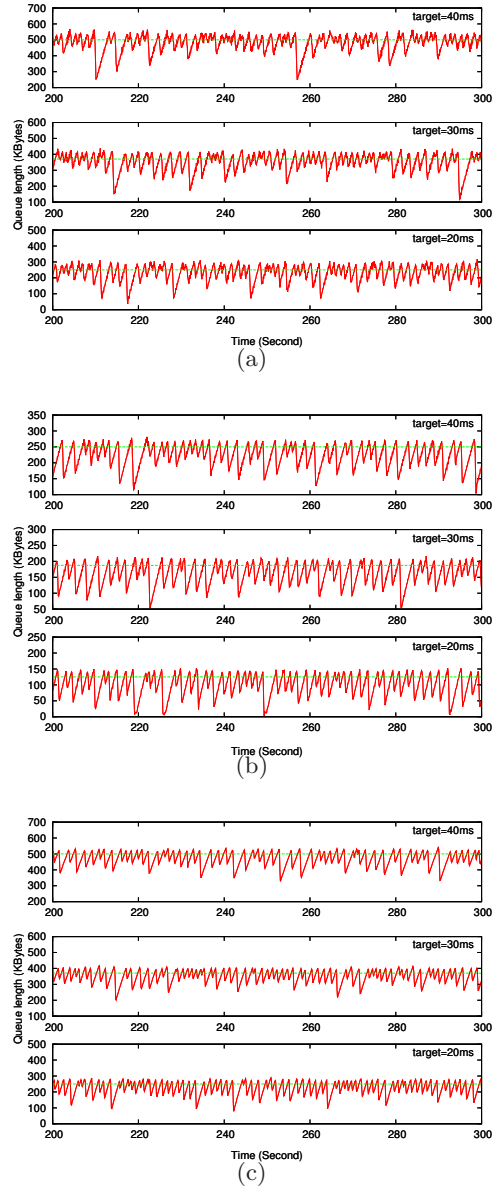


Figure 6: Invariance of cycles with CoDel: extended cases.

below which the predicted invariance property breaks down. This is because our model assumes that the bottleneck link is always backlogged. When β is small, it is possible that backoffs in TCP connections will leave the link idle for some time. The smaller the value of β is, the more frequent a link will become idle.

In obtaining a lower bound for practical settings of β , we propose to leverage existing results on the right size for a router buffer, such as [4]. Under the assumption that the buffer is managed by the passive drop-tail algorithm, these results generally establish that the buffer size can be made much smaller than the bandwidth delay product without undermining the bandwidth utilization. While the PLT principle guides the design of AQM, it is similar to the drop-tail algorithm in that no packet is dropped when the backlog is below a threshold value (which is β in the case of PLT,

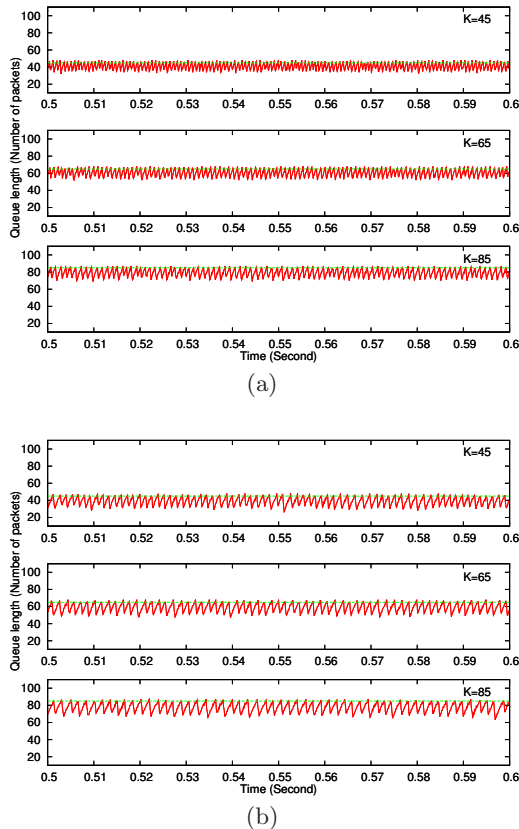


Figure 7: Invariance of cycles with DCTCP.

and the buffer size in the case of drop-tail).⁵ On the other hand, drop-tail exerts more abrupt control than PLT when the backlog exceeds its respective threshold value. Therefore, the lower bound for β should be lower than the derived buffer size. We leave a complete examination of this issue as part of future work.

6. RELATED WORK

Solutions for the bufferbloat problem can be network-based or end-to-end. The network-based solutions aim to enhance the robustness of AQM against protocol parameters. Inheriting the idea of adaptive RED [11], [26] proposes that AQM should automatically estimate global information and adapt parameters to the estimation. [13] proposes only updating packet drop rate at the end of fixed time intervals instead of at every packet arrival. In general, it remains to conduct comprehensive evaluation of these novel AQM proposals. Most notably, the period at which the packet dropping rate/protocol parameter is updated form a new parameter itself, and should be carefully selected. This is out of the scope of the present paper and left as our future work.

The end-to-end solutions propose to improve the responsiveness TCP to incipient network congestion, so that buffer occupancy level can be reduced at routers. Several TCP variants, such as [8, 30, 31], suggest that TCP could incorporate delay information in inferring network loading con-

⁵Arguably, this simple structure leads to the common robustness of drop-tail and PLT.

dition to attain congestion avoidance at a finer granularity, and in particular attain more prompt congestion response. Alternatively, [17] suggests that TCP receiver could exert extra control by adapting the size of advertised window *awnd* based on the estimated queueing delay. However, a TCP connection with these enhancements in general tends to yield its fair share of bandwidth to the prevalent loss-based TCP due to earlier backoff [24]. Moreover, the resulting system may have multiple equilibrium [29]. These side effects hinder the incremental deployment of the end-to-end solutions in the heterogeneous Internet.

As a link pricing scheme, the $PLT(\beta)$ system guides the update of packet dropping rate in AQM and thus mostly belongs to network-based solutions. Nevertheless, it can also be (partly) implemented end-to-end, as in the case of DCTCP.

Exhibiting both discrete and continuous dynamics, the $PLT(\beta)$ system belongs to the general class of the hybrid dynamical systems [15]. Existing studies of such systems, however, mostly consider systems that possesses equilibrium. They cannot apply to the $PLT(\beta)$ system, which possess no equilibrium but limit cycles. In [2], the limit cycle behavior of DCTCP is reported. The stability of the limit cycle is studied via checking stability of the Poincaré maps. Our work considers the more general $PLT(\beta)$ system, and shows that any congestion control system implementing the PLT principle will exhibit the limit cycle behavior. Moreover, we formally establish the invariance property, and experimentally validate it in the cases of CoDel and DCTCP.

7. CONCLUSION

In this paper, we have presented the pricing-link-by-time (PLT) principle. We have developed a model of a congestion control system designed under the PLT principle, the $PLT(\beta)$ system. We have shown that the $PLT(\beta)$ system exhibits a family of cycles instead of equilibrium, and have provided numerically checkable conditions under which such cycles become stable limit cycles. We further establish the invariance property, that the cycles are invariant across different values of β except for a shift of the backlog trajectory. Our simulation results show that CoDel and DCTCP, two realizations of the PLT principle, exhibit the cyclic behavior and their cycles are invariant across different values of β . Thus, when tuning β for an AQM algorithm implementing the PLT principle, one needs only focus on the desirable level of queueing delay. This makes the parameter tuning for AQM an optional and simple process, and provides an integral solution to combatting bufferbloat.

There are several possible extensions to our work, including analyzing PLT in networks with multiple bottleneck links and comparing various specific forms of the pricing function h .

8. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their useful comments. This research is supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant No. HKU 714510E, ARO MURI grant W911NF-08-1-0233, NSF NetSE grant CNS 0911041, and Bell Labs, Lucent-Alcatel.

9. REFERENCES

- [1] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data Center TCP (DCTCP). *ACM SIGCOMM Computer Communication Review*, Vol. 40, No. 4, pp. 63-74, October 2010.
- [2] M. Alizadeh, A. Javanmard, and B. Prabhakar. Analysis of DCTCP: Stability, Convergence, and Fairness. *ACM SIGMETRICS Performance Evaluation Review*, Vol. 39, No. 1, pp. 73-84, June 2011.
- [3] M. Allman. Comments on Bufferbloat. *ACM SIGCOMM Computer Communication Review*, Vol. 43, No. 1, pp. 31-37, January 2013.
- [4] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 4, pp. 281-292, October 2004.
- [5] S. Athuraliya, V.H. Li, S.H. Low, and Q. Yin. REM: Active Queue Management. *IEEE Network*, Vol. 15, No. 3, pp. 48-53, May 2001.
- [6] C. Berge. Topological Spaces. *Oliver & Boyd*, 1963.
- [7] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. *IETF RFC 2309*, April 1998.
- [8] L.S. Brakmo and L.L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 8, pp. 1465-1480, October 1995.
- [9] M. Dischinger, A. Haeberlen, K.P. Gummadi, and S. Saroiu. Characterizing Residential Broadband Networks. *Proc. of ACM SIGCOMM IMC*, pp. 43-56, October 2007.
- [10] W.-C. Feng, K.G. Shin, D.D. Kandlur, and D. Saha. The BLUE Active Queue Management Algorithms. *IEEE/ACM Transactions on Networking*, Vol. 10, No. 4, pp. 513-528, August 2002.
- [11] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. *Technical Report, ACIRI*, August 2001.
- [12] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August 1993.
- [13] A. Francini. Periodic Early Detection for Improved TCP Performance and Energy Efficiency. *Computer Networks*, Vol. 56, No. 13, pp. 3076-3086, September 2012.
- [14] J. Gettys and K. Nichols. Bufferbloat: Dark Buffers in the Internet. *Communications of the ACM*, Vol. 55, No. 1, pp. 57-65, January 2012.
- [15] R. Goebel and R.G. Sanfelice, and A.R. Teel. Hybrid Dynamical Systems. *IEEE Control Systems*, Vol. 29, No. 2, pp. 28-93, April 2009.
- [16] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows. *IEEE Transactions on Automatic Control*, Vol. 47, No. 6, pp. 945-959, June 2002.
- [17] H. Jiang, Y. Wang, K. Lee, and I. Rhee. Tackling Bufferbloat in 3G/4G Networks. *Proc. of ACM IMC*, pp. 329-342, November 2012.
- [18] F. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, Vol. 49, No. 3, pp. 237-252, March 1998.
- [19] H.K. Khalil. Nonlinear Systems. *Upper Saddle River, Prentice-Hall*, 2002.
- [20] S. Kunniyur and R. Srikant. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. *ACM SIGCOMM Computer Communication Review*, Vol. 31, No. 4, pp. 123-134, August 2001.
- [21] S.H. Low. A Duality Model of TCP and Queue Management Algorithms. *IEEE/ACM Transactions on Networking*, Vol. 11, No. 4, pp. 525-536, August 2003.
- [22] S.H. Low, F. Paganini, and J.C. Doyle. Internet Congestion Control. *IEEE Control Systems Magazine*, Vol. 22, No. 1, pp. 28-43, February 2002.
- [23] V. Misra, W.-B. Gong, and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with An Application to RED. *SIGCOMM Computer Communication Review*, Vol. 30, No. 4, pp. 151-160, August 2000.
- [24] J. Mo, R.J. La, V. Anantharam, and J. Walrand. Analysis and Comparison of TCP Reno and Vegas. *Proc. of IEEE INFOCOM'99*, Vol. 3, pp. 1556-1563, March 1999.
- [25] K. Nichols and V. Jacobson. Controlling Queue Delay. *ACM Queue*, Vol. 10, No. 5, pp. 1-15, May 2012.
- [26] R. Pan, P. Natarajan, C. Piglione, M.S. Prabhu, V. Subramanian, F. Baker, and B.V. Steeg. PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem. *Proc. of IEEE HPSR*, pp. 148-155, July 2013.
- [27] T.S. Parker and L.O. Chua. Practical Numerical Algorithms for Chaotic Systems. *Springer-Verlag*, 1989.
- [28] R. Srikant. *The Mathematics of Internet Congestion Control*. Cambridge, MA: Birkhauser, 2004.
- [29] A. Tang, J. Wang, S.H. Low, and M. Chiang. Equilibrium of Heterogeneous Congestion Control: Existence and Uniqueness. *IEEE/ACM Transactions on Networking*, Vol. 15, No. 4, pp. 824-837, October 2007.
- [30] D.X. Wei, J. Cheng, S.H. Low, and S. Hegde. FAST TCP: Motivation, Architecture, Algorithms, Performance. *IEEE/ACM Transactions on Networking*, Vol. 14, No. 6, pp. 1246-1259, December 2006.
- [31] K. Winstein and H. Balakrishnan. TCP ex Machina: Computer-Generated Congestion Control. *ACM SIGCOMM Computer Communication Review*, Vol. 43, No. 4, pp. 123-134, Oct. 2013.