# Analysis of DCTCP: Stability, Convergence, and Fairness

Mohammad Alizadeh, Adel Javanmard, and Balaji Prabhakar
Department of Electrical Engineering, Stanford University
{alizade, adelj, balaji}@stanford.edu

## ABSTRACT

Cloud computing, social networking and information networks (for search, news feeds, etc) are driving interest in the deployment of large data centers. TCP is the dominant Layer 3 transport protocol in these networks. However, the operating conditions—very high bandwidth links, low round-trip times, small-buffered switches—and traffic patterns cause TCP to perform very poorly. The Data Center TCP (DCTCP) algorithm has recently been proposed as a TCP variant for data centers and addresses these shortcomings.

In this paper, we provide a mathematical analysis of DCTCP. We develop a fluid model of DCTCP and use it to analyze the throughput and delay performance of the algorithm, as a function of the design parameters and of network conditions like link speeds, round-trip times and the number of active flows. Unlike fluid model representations of standard congestion control loops, the DCTCP fluid model exhibits limit cycle behavior. Therefore, it is not amenable to analysis by linearization around a fixed point and we undertake a direct analysis of the limit cycles, proving their stability. Using a hybrid (continuous- *and* discrete-time) model, we analyze the convergence of DCTCP sources to their fair share, obtaining an explicit characterization of the convergence rate. Finally, we investigate the "RTT-fairness" of DCTCP; i.e., the rate obtained by DCTCP sources as a function of their RTTs. We find a very simple change to DCTCP which is suggested by the fluid model and which significantly improves DCTCP's RTT-fairness. We corroborate our results with ns2 simulations.

**Categories and Subject Descriptors:**
C.2.2 [Computer-Communication Networks]: Network Protocols

**General Terms:**
Algorithms, Performance, Theory

**Keywords:**
Data center network, Congestion control, Analysis, TCP

*SIGMETRICS'11,* June 7–11, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0262-3/11/06 ...$10.00.

## 1. INTRODUCTION

### 1.1 Background and Motivation

TCP is the dominant transport protocol in the Internet and has proven to be scalable and robust to network operating conditions. It is also the dominant Layer 3 protocol in the data center environment, notably in data centers associated with cloud computing, social networking and information networks delivering services such as search, news, advertisement, etc. Its performance in data center environments, however, is quite poor. As detailed in [1], this is mainly because data centers have switches with very small buffers, high speed links, and low round-trip times. Moreover, the nature of traffic and the requirements of applications in data centers are quite different from that in the wide area Internet; data center applications generate a mixture of bursty query traffic, delay-sensitive short messages, and throughput-intensive long flows and often have strict deadlines for the completion of flows.

These differences in operating conditions have caused TCP to underperform in the following ways: (i) It requires large buffers so as to not under-run links, and these buffers can be expensive at high line rates. (ii) It uses all the available buffering; therefore, it induces unacceptably large queuing delays. (iii) It does not handle bursty traffic well, especially under 'Incast' [20] scenarios. See [1] for a more detailed discussion of TCP performance in data centers.

Data Center TCP (DCTCP) has been proposed in [1] to address these shortcomings of TCP. Figure 1, taken from [1], compares the operation of TCP and DCTCP in an actual hardware testbed. It is seen that DCTCP achieves full throughput (the buffer does not underflow) while maintaining a very low buffer occupancy compared to TCP. This allows DCTCP to simultaneously provide low latency and good burst tolerance for the short flows, and high throughput for the long flows.

The key mechanisms used by DCTCP are a simple active queue management scheme at the switch, based on Explicit Congestion Notification (ECN) [16], and a window control scheme at the source which reacts to ECN marks by reducing the window size in proportion to the *fraction* of packets that are marked (contrast this with TCP which *always* cuts the window by half if at least one packet is marked). The performance of DCTCP is determined by two parameters: (i) $K$, the marking threshold on the queue at the switch above which all packets are marked; and (ii) $g$, the weight used for exponentially averaging ECN mark values at the source. See Section 2.1 for details.

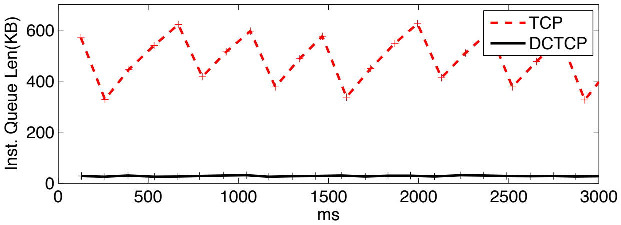

**Figure 1: Taken from [1]. Queue length measured on a Broadcom Triumph switch. Two long flows are launched from distinct 1Gbps ports to a common 1Gbps port. The switch allocates about 600KB of buffer to the congested port.**

## 1.2 Main Contributions and Organization

This paper undertakes a rigorous analysis of DCTCP. The following are our main contributions:

**1. DCTCP fluid model.** We briefly review the DCTCP algorithm and derive a fluid model for it in Section 2. The model comprises of a system of nonlinear delay-differential equations. Using ns2 [15] simulations, we find that the fluid model is very accurate and that it is more accurate across a wider range of parameters than a simple model presented in [1]. A key step in developing the fluid model is accurately capturing the bursty (0–1 type) marking at the switch which DCTCP employs.[1]

**2. Steady state.** We analyze the steady state behavior of DCTCP using the fluid model in Section 3. Due to its 0–1 style marking function, the fluid model does not have a fixed point. Instead, it exhibits a (periodic) limit cycle behavior in steady state. Theorem 1 provides a necessary and sufficient condition for local stability of the limit cycles using the so-called *Poincaré map* technique [4, 10, 14, 22]. We verify this condition (numerically) for a wide range of parameter values. We then explore the throughput and delay performance of DCTCP by explicitly evaluating the limit cycle solutions. Here we find that for DCTCP to achieve 100% throughput, the marking threshold, $K$, needs to be about 17% of the bandwidth-delay product. For smaller values of $K$, we determine the throughput loss; i.e., we obtain the throughput-delay tradeoff curve. A key result is that DCTCP's throughput remains higher than 94% even as $K \to 0$. This is much higher than the limiting throughput of 75% for a TCP source as the buffer size goes to zero [21].

**3. Convergence.** We analyze how quickly DCTCP flows converge to their fair equilibrium window sizes (equivalently, sending rates) in Section 4. This is important to determine since DCTCP reduces its sending rate by factors much smaller than TCP; therefore, DCTCP sources may take much longer to converge. Theorem 2 gives the following explicit characterization: For $N$ flows (with identical RTTs) sharing a single bottleneck link, the window sizes at the $n^{th}$ congestion episode, $W_i(T_n)$, converge to the fair share, $W^*$, as follows:

$$|W_i(T_n) - W^*| < O(n^2)\exp\left(-\beta_{DCTCP} T_n\right),$$

where an explicit expression is given for $\beta_{DCTCP}$. Using this, we compare the rate of convergence of TCP and DCTCP and obtain the following bounds

$$\beta_{DCTCP} < \beta_{TCP} < 1.4 \times \beta_{DCTCP}.$$

Thus, even though DCTCP converges slower than TCP, it is at most 1.4 times slower.

We note that the convergence results are obtained using a different model from the fluid model, since the fluid model is not suitable for conducting transient analyses. This new model of DCTCP, which we call the Hybrid Model, employs continuous- and discrete-time variables and is similar to the AIMD models used in the analysis of TCP–Drop-tail [5, 17, 18]. While the AIMD models are linear and can be used to determine convergence rates via an analysis of the eigenvalues of linear operators [17, 18], the Hybrid Model is more challenging because it is nonlinear.

**4. RTT-fairness.** We investigate the fairness properties of DCTCP for flows with diverse RTTs in Section 5. RTT-fairness is defined as the ratio of the throughput achieved by two groups of flows as a function of the ratio of their RTTs [23, 11, 6, 2]. Using ns2 simulations, we find that DCTCP's RTT-fairness is better than TCP–Drop-tail but worse than TCP with RED [7] marking. We identify a very simple change to the DCTCP algorithm, suggested by the fluid model, which considerably improves its RTT-fairness. The modified DCTCP is shown to have linear RTT-fairness ($Throughput \propto RTT^{-1}$) and achieve a better fairness than TCP–RED.

**5. DCTCP Parameter Guidelines.** Our analysis of DCTCP's steady state and convergence properties yields guidelines for choosing algorithm parameters. Let $C$ and $d$ respectively denote the bottleneck capacity (in packets/sec) and propagation delay (in sec). Then,

$$\mathbf{K \approx 0.17Cd}, \qquad (1)$$

$$\frac{\mathbf{5}}{\mathbf{Cd+K}} \lessapprox \mathbf{g} \lessapprox \frac{\mathbf{1}}{\sqrt{\mathbf{Cd+K}}}. \qquad (2)$$

For example, when $C = 10$Gbps and $d = 300\mu$s, assuming 1500Byte packets, $K$ needs to be about 42 packets and $0.02 \lessapprox g \lessapprox 0.06$.

# 2. DCTCP: ALGORITHM AND MODEL

## 2.1 The Algorithm

First, we briefly review the DCTCP algorithm. We focus on those aspects relevant to this paper and refer to [1] for more details.

**Switch Side.** DCTCP uses a very simple active queue management scheme: When a packet arrives at the switch buffer and the buffer occupancy is at least $K$ packets, the packet is "marked" using the ECN mechanism. Note that the arriving packet is marked if (and only if) it finds the *instantaneous* buffer occupancy to be larger than $K$ packets.

**Source Side.** DCTCP is designed to simultaneously achieve high throughput and very low queue occupancies. It does this by reducing its current window (hence, sending rate) in proportion to the *extent* of congestion. Specifically, a DCTCP source reduces its window by a factor that is proportional to the *fraction* of marked packets: the larger the fraction, the larger the decrease factor. This is in contrast to the behavior a TCP source which reacts to marked packets by *always halving* its window size.

[1] This is similar to the difficulty of modeling TCP–Drop-tail using fluid models.

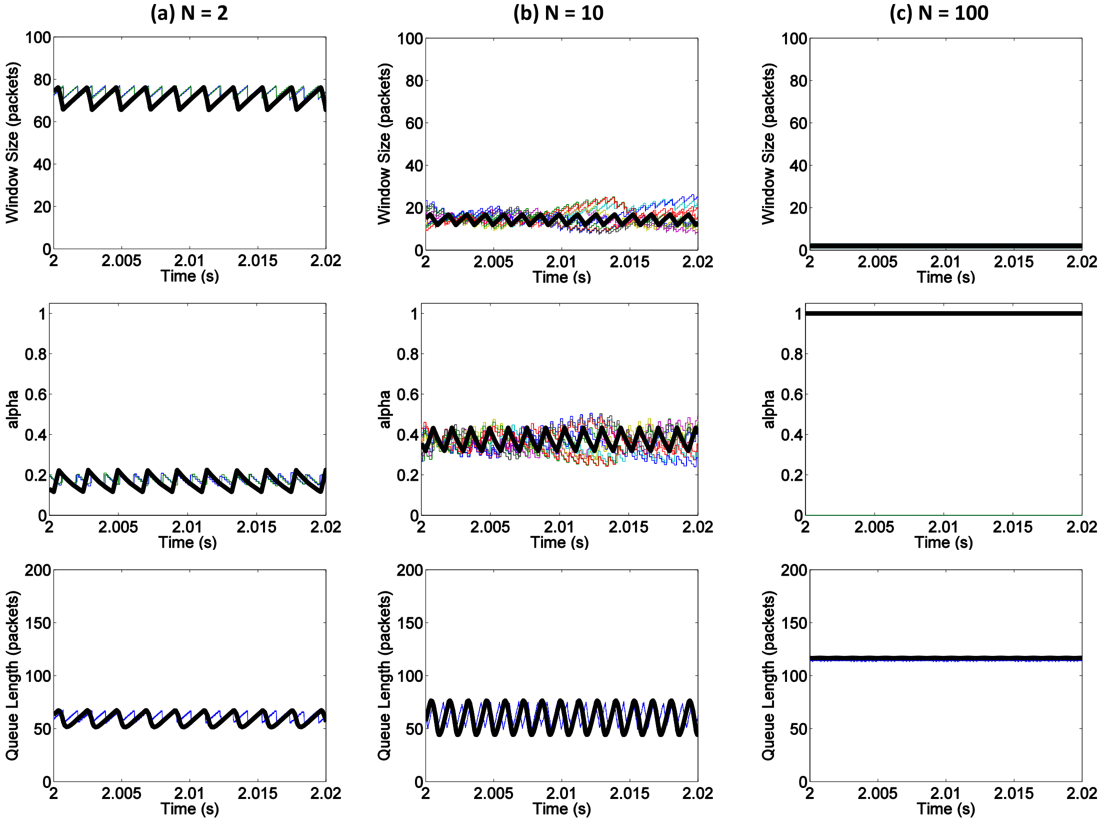

**Figure 2: Comparison between fluid model and ns2. The fluid model results are shown in solid black.**

Operationally, the source maintains a running estimate of the fraction of its packets that are marked. This estimate, $\alpha$, is updated once for every window of data (roughly each round-trip time) as follows:

$$\alpha \leftarrow (1-g)\alpha + gF, \tag{3}$$

where $F$ is the *fraction* of packets that were marked in the most recent window of data, and $g \in (0,1)$ is a fixed parameter. DCTCP uses $\alpha$ to cut its window size in response to a marked ACK as follows:

$$W \leftarrow (1-\frac{\alpha}{2})W. \tag{4}$$

Thus when $\alpha$ is close to 0 (low congestion), the window is only slightly reduced, whereas when $\alpha$ is close to 1 (high congestion), the window is cut nearly in half. It is important to note that, as in TCP, DCTCP cuts its window size at most once per window of data (or round-trip time) [16].

This is the only difference between a DCTCP source and a TCP source. Other aspects of the TCP algorithm, such as slow start, additive increase during congestion avoidance, or recovery from packet loss remain unchanged.

## 2.2 The Fluid Model

We now develop a fluid model for DCTCP by considering $N$ long-lived flows traversing a single bottleneck switch port with capacity $C$. The following non-linear, delay-differential equations describe the dynamics of $W(t)$, $\alpha(t)$, and the queue size at the switch, $q(t)$:

$$\frac{dW}{dt} = \frac{1}{R(t)} - \frac{W(t)\alpha(t)}{2R(t)}p(t-R^*), \tag{5}$$

$$\frac{d\alpha}{dt} = \frac{g}{R(t)}\left(p(t-R^*) - \alpha(t)\right), \tag{6}$$

$$\frac{dq}{dt} = N\frac{W(t)}{R(t)} - C. \tag{7}$$

Here $p(t)$ indicates the packet marking process at the switch and is given by

$$p(t) = \mathbb{1}_{\{q(t)>K\}}, \tag{8}$$

and $R(t) = d + q(t)/C$ is the round-trip time (RTT), where $d$ is the propagation delay (assumed to be equal for all flows), and $q(t)/C$ is the queueing delay.

Equations (5) and (6) describe the DCTCP source, while (7) and (8) describe the queue process at the switch and the DCTCP marking scheme. The source equations are coupled with the switch equations through the packet marking process $p(t)$ which gets fed back to the source with some delay. This feedback delay is the round-trip time $R(t)$, and varies with $q(t)$. However, as a simplification—and following [9] and [12]—we use the approximate fixed value $R^* = d + K/C$ for the delay. The approximation aligns well with DCTCP's attempt to strictly hold the queue size at around $K$ packets.

Equation (5) models the window evolution and consists of the standard additive increase term, $1/R(t)$, and a multiplicative decrease term, $-W(t)\alpha(t)/2R(t)$. The latter term models the source's reduction of its window size by a factor $\alpha(t)/2$ when packets are marked (i.e., $p(t-R^*)=1$), and this occurs once per RTT. Equation (6) is a continuous approximation of (3). Finally, equation (7) models the queue evolution: $N \cdot W(t)/R(t)$ is the net input rate and $C$ is the service rate.

**Remark 1.** In standard TCP fluid models [19, 13, 8, 12], the multiplicative decrease term is given by

$$-\frac{W(t)}{2}\frac{W(t-R^*)}{R(t-R^*)}p(t-R^*), \tag{9}$$

with the interpretation that $W(t-R^*)/R(t-R^*) \times p(t-R^*)$ is the rate at which marked ACKs arrive at the source, each causing a window reduction of $W(t)/2$. As explained in [13], this model is accurate when the packet marks can be assumed to occur as a Poisson process, for example, when the RED algorithm [7] marks packets. In DCTCP, the marking process is bursty, as captured by the function $p(t)$ at equation (8). Hence, the natural adaptation of (9):

$$-\frac{W(t)\alpha(t)}{2}\frac{W(t-R^*)}{R(t-R^*)}p(t-R^*),$$

is not valid for DCTCP.

**Simulation comparison of the model.** Fig. 2 compares the fluid model with packet-level simulations using the ns2 simulator [15]. The parameters used are: $C = 10$Gbps, $d = 100\mu$s, $K = 65$ packets, and $g = 1/16$ ($K$ and $g$ are chosen to match those of the 10Gbps experiments in [1]). The fluid model plots correspond to the evolution of $W(\cdot)$ and $\alpha(\cdot)$, whereas the ns2 plots show the evolutions of the window size and $\alpha$ of *each* of the $N$ sources, for $N = 2, 10, 100$. In the case $N = 100$, the queue size climbs to about 120 packets, even though the marking threshold, $K$, is 65. Thus *every* arriving packet is marked at the switch buffer (captured by $\alpha = 1$) to signal congestion to each of the 100 sources. The window size at each source is reduced to the minimum, equal to 2 packets, corresponding to a sending rate of 100 Mbps per source.[2]

As can be seen, the fluid model matches the simulations quite well. We have also compared the model with simulations for a wide range of line speeds, propagation delays, number of sources and DCTCP parameters. Our findings are that the model has very good fidelity and that, as is typical, the fidelity increases with the number of sources.

**Comparison with the Sawtooth model.** The DCTCP paper [1] presents a simplified model of the steady state behavior of long-lived DCTCP flows sharing a single bottleneck. We call this the 'Sawtooth' model. It is based on the assumption that the window sizes of the $N$ flows and the queue size can be thought of as being perfectly synchronized, described by periodic sawtooth processes. This makes it possible to compute the steady state fraction of packets marked at the switch, which can be used to completely specify the sawtooth processes in the model. See [1] for details.

The Sawtooth model provides simple closed form approximations to quantities of interest, such as the amplitude of queue oscillations, leading to some guidelines on how to set the DCTCP parameters. But the accuracy of the Sawtooth model rests heavily on the validity of the synchronization assumption. As mentioned in [1], this holds when $N$ is small, typically less than 10. Further, the Sawtooth model assumes that sources know the exact fraction of marked packets, and does not include the variations in $\alpha$ estimates resulting from the exponentially weighted moving average estimation at equation (3). In particular, the model is only accurate for small values of $g$ and does not capture the effect of $g$ on system dynamics.

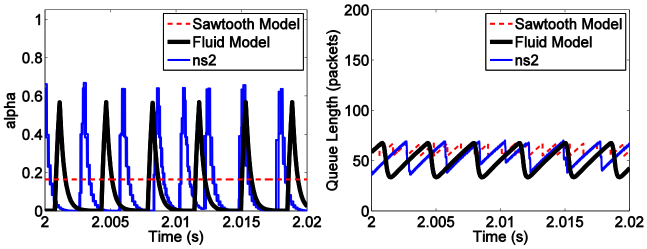

**Figure 3: Comparison of the fluid and Sawtooth models with ns2 simulations.**

This is demonstrated in Fig. 3, which repeats the previous simulation with $N = 2$, this time setting $g = 0.4$. Because the Sawtooth model assumes $\alpha$ is a constant, its prediction of the queue size evolution is very poor. The fluid model presented above captures the fluctuations in $\alpha$ and is, therefore, much more accurate.

# 3. ANALYSIS: STEADY STATE

## 3.1 The Normalized Fluid Model

We change variables

$$\tilde{W}(t) = W(R^*t),\ \tilde{\alpha}(t) = \alpha(R^*t),\ \tilde{q}(t) = \frac{q(R^*t)-K}{CR^*}, \tag{10}$$

and rewrite the fluid model equations (5)–(8):

$$\frac{d\tilde{W}}{dt} = \frac{1}{1+\tilde{q}(t)}\left(1 - \frac{\tilde{W}(t)\tilde{\alpha}(t)}{2}\tilde{p}(t-1)\right), \tag{11}$$

$$\frac{d\tilde{\alpha}}{dt} = \frac{g}{1+\tilde{q}(t)}\left(\tilde{p}(t-1) - \tilde{\alpha}(t)\right), \tag{12}$$

$$\frac{d\tilde{q}}{dt} = \frac{1}{\bar{w}}\frac{\tilde{W}(t)}{(1+\tilde{q}(t))} - 1. \tag{13}$$

Here

$$\tilde{p}(t) = \mathbb{1}_{\{\tilde{q}(t)>0\}}, \tag{14}$$

and $\bar{w} = (Cd+K)/N$ is the average per-flow window size. Henceforth, we refer to this as the normalized fluid model. The normalized model immediately reveals that while the DCTCP fluid model has 5 parameters ($C$, $N$, $d$, $K$ and $g$), the system dynamics is completely determined by: (i) the average per-flow window size $\bar{w}$, and (ii) the parameter $g$.

We now discuss the existence of possible fixed points for the normalized fluid model. A fixed point of the system, $(\tilde{W}, \tilde{\alpha}, \tilde{q})$, must satisfy the following equations:

$$1 - \frac{\tilde{W}\tilde{\alpha}}{2}\tilde{p} = 0, \quad \tilde{p} - \tilde{\alpha} = 0, \quad \frac{1}{\bar{w}}\frac{\tilde{W}}{(1+\tilde{q})} - 1 = 0, \tag{15}$$

where $\tilde{p} = \mathbb{1}\{\tilde{q} > 0\}$. The above equations have a solution only if $\bar{w} \leq 2$. Therefore, we have the following two operating regimes:

[2] Note that, when $N = 100$, the RTT equals 240 microseconds, since $d = 100$ microseconds and $q/C = 140$ microseconds. With these numbers, the sending *rate* of each source, $W/RTT$, equals 100 Mbps, as required.

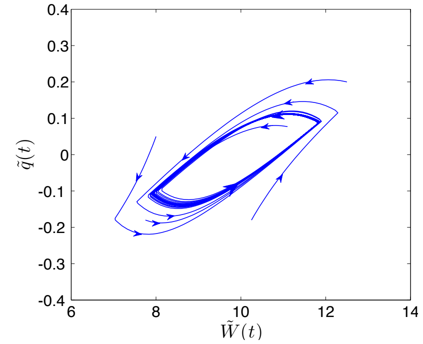

**Figure 4: Phase diagram showing occurrence of limit cycle for $\bar{w} = 10$, $g = 1/16$ (the projection on the $\tilde{W}$–$\tilde{q}$ plane is shown).**

**(i) $\bar{w} \leq 2$:** In this regime, the normalized model has a unique fixed point, namely $(\tilde{W}, \tilde{\alpha}, \tilde{q}) = (2, 1, \frac{2}{\bar{w}} - 1)$. Equivalently, (5)–(8) has the fixed point $(W, \alpha, q) = (2, 1, 2N - Cd)$. This regime corresponds to the large $N$ case, where the system has a very simple steady state behavior: each source transmits two packets per RTT, of which $Cd$ fill the link capacity, and the remaining $2N - Cd$ build up a queue. All packets are marked as the queue constantly remains larger than $K$. The $N = 100$ case of Fig. 2 is in this regime.
**(ii) $\bar{w} > 2$:** In this regime, the system does not have a fixed point. Rather, it has a periodic solution or limit cycle,[3] characterized by a closed trajectory in state space. Figure 4 shows a sample phase diagram of the limit cycle projected onto the $\tilde{W}$–$\tilde{q}$ plane for $\bar{w} = 10$ and $g = 1/16$. As shown, all trajectories evolve toward the orbit of the limit cycle. Both the $N = 2$ and $N = 10$ cases of Fig. 2 are in this regime.

The regime $\bar{w} \leq 2$ will no longer be discussed since it is trivial to show that the system converges to the unique fixed point. In the next two sections, we study the stability and structure of the limit cycle solution when $\bar{w} > 2$.

## 3.2 Stability of Limit Cycle

The analysis of the stability of limit cycles is complicated and few analytical tools exist. Fortunately, a wealth of computational tools are available [14], and it is possible to compute the periodic solution of the system and determine its stability properties numerically. A limitation of the computational approach is that it requires sweeping the different parameters of the model to completely characterize the dynamics. However, our task is greatly simplified by the fact that the dynamics of the normalized model is completely determined by the two parameters $\bar{w}$, and $g$. We proceed by defining stability of limit cycles.

Let $X^*$ denote the set of points in the state space belonging to the limit cycle. We define an $\epsilon$-neighborhood of $X^*$ by

$$U_\epsilon = \{x \in \mathbb{R}^n \mid dist(x, X^*) < \epsilon\},$$

where $dist(x, X^*)$ is the minimum distance from $x$ to a point in $X^*$; i.e., $dist(x, X^*) = \inf_{y \in X^*} ||x - y||$.

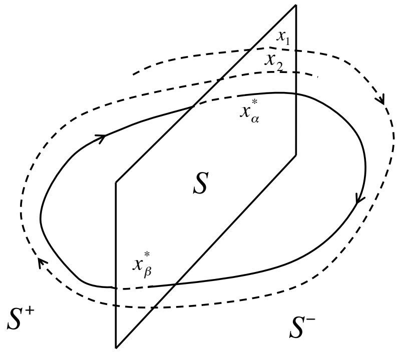

**Figure 5: Periodic system trajectory and the Poincaré map.**

**Definition** 1. *The limit cycle $X^*$ is*

(i) ***stable*** *if for any $\epsilon > 0$, there exists a $\delta > 0$ such that*

$$x(0) \in U_\delta \Rightarrow x(t) \in U_\epsilon, \ \forall t \geq 0.$$

(ii) ***locally asymptotically stable*** *if it is stable and $\delta$ can be chosen such that*

$$x(0) \in U_\delta \Rightarrow \lim_{t \to \infty} dist(x(t), X^*) = 0.$$

Figure 4 illustrates the local asymptotical stability of our system for a choice of parameters: orbits initiated from points close enough to the limit cycle are attracted to it.

To proceed, let $x(t) = (\tilde{W}(t), \tilde{\alpha}(t), \tilde{q}(t))^T$ denote the state space of the fluid model equations at (11)–(14). It is convenient to represent the fluid model as:

$$\dot{x}(t) = F(x(t), u(t-1)), \quad u(t) = \mathbb{1}_{\{cx(t) > 0\}}, \qquad (16)$$

where $c = [0, 0, 1]$ and $u(t) = \tilde{p}(t)$ is the system feedback.

**Poincaré map.** We analyze the stability of the limit cycle via the 'Poincaré map', which we introduce after making some definitions. Define the *switching plane* as $S = \{x \in \mathbb{R}^3 : cx = 0\}$ and let $S^+ = \{x \in \mathbb{R}^3 : cx > 0\}$ and $S^- = \{x \in \mathbb{R}^3 : cx < 0\}$ (see Fig. 5 for an illustration). Note that the switching plane, $\tilde{q} = 0$ (equivalently, $q = K$), corresponds to the DCTCP marking threshold. The limit cycle crosses the switching plane twice in each period, once from $S^+$ (at the point $x^*_\alpha$) and once from $S^-$ (at the point $x^*_\beta$).

The Poincaré map traces the evolution of the system at the times when its trajectory crosses the switching plane in a given direction. More precisely, let the successive intersections of the trajectory $x(t)$ with $S$ in direction $S^+$ to $S^-$ be denoted by $x_i$. The Poincaré map $x_{i+1} = P(x_i)$, maps the $i^{th}$ intersection to the subsequent one. Note that the Poincaré map is well-defined for our system, because equations (11)–(14) guarantee that starting from any point in $S^+$ or $S^-$, the trajectory will eventually intersect with $S$ (see also Lemma 3 in Appendix A).

The fixed point of the Poincaré map corresponds to the intersection of the limit cycle with the switching plane, say at $x^*_\alpha$. Therefore, local stability of the Poincaré map at $x^*_\alpha$ implies local stability of the limit cycle. We refer to [4, 10, 14, 22] for more details on the Poincaré map technique.

Next, there are two main steps to establish:
**Step 1.** The Poincaré map is locally stable. Theorem 1,

[3] Formally, a periodic solution is said to be a limit cycle if there are no other periodic solutions sufficiently close to it. In other words, a limit cycle corresponds to an isolated periodic orbit in the state space [14].

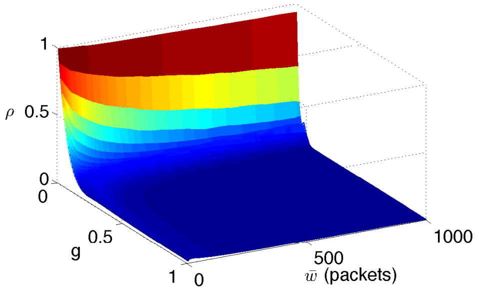

**Figure 6: Limit cycle stability (Theorem 1).**

which is an adaptation of Theorem 3.3.1 in [22] to our setting, provides a necessary and sufficient condition for local stability of the Poincaré map, and consequently, the limit cycle.

**Step 2.** Verify that our system satisfies the condition of Theorem 1 for a wide range of $\bar{w}$ and $g$.

Before proceeding with the statement of Theorem 1, we need the following definitions. Let $x^*(t)$ denote the trajectory of the limit cycle of system (16). Assume that $x^*(t)$ traverses the switching plane from $S^+$ to $S^-$ at time $t_0 = 0$; i.e., $x^*(0) = x^*_\alpha$, and that the period of the limit cycle is $T = (1 + h_\alpha) + (1 + h_\beta)$ with $h_\alpha > 0$ and $h_\beta > 0$, where $1 + h_\alpha$ (resp. $1 + h_\beta$) is the time taken for the trajectory to move from $x^*_\alpha$ to $x^*_\beta$ (resp. from $x^*_\beta$ to $x^*_\alpha$). For notational convenience, define $u_\alpha = 0$ and $u_\beta = 1$. Let

$$Z_1 = \left(I - \frac{F(x^*_\alpha, u_\beta)c}{cF(x^*_\alpha, u_\beta)}\right) \exp\left(\int_{1+h_\alpha}^{T} J_F(x^*(s), u(s-1))ds\right),$$

$$Z_2 = \left(I - \frac{F(x^*_\beta, u_\alpha)c}{cF(x^*_\beta, u_\alpha)}\right) \exp\left(\int_{0}^{1+h_\alpha} J_F(x^*(s), u(s-1))ds\right),$$

where $J_F$ is the Jacobian matrix of $F$ with respect to x, and $I$ is the identity matrix. The integral of the matrix $J_F$ is entry wise. We assume that $cF(x^*_\alpha, u_\beta) \neq 0$ and $cF(x^*_\beta, u_\alpha) \neq 0$; i.e., $x^*(t)$ is nontangent with the switching plane at the traversing points. Note that $Z_1$ and $Z_2$ are $3 \times 3$ matrices.

**Theorem** 1. *The Poincaré map (and its associated limit cycle) is locally asymptotically stable if and only if*

$$\rho(Z_1 Z_2) < 1.$$

*Here, $\rho(\cdot)$ is the spectral radius.*

A proof sketch is provided in Appendix A.

**Verification.** We use Theorem 1 to verify the stability of DCTCP's limit cycle. Since $Z_1$ and $Z_2$ do not have a closed form, we sweep the parameters $\bar{w}$ and $g$ in the ranges of interest and compute $\rho(Z_1 Z_2)$ numerically. This has been done in Fig. 6 for the range $g \in [0.001, 1]$, and $\bar{w} \in [2.01, 1000]$. Throughout this range, $\rho < 1$, and (local) stability is verified. We conjecture that the limit cycle is actually globally stable for all $g \in (0, 1]$, $\bar{w} > 2$.

## 3.3 Steady State Throughput & Delay

In this section, we study the key performance metrics of throughput and delay of DCTCP, by analyzing the limit cycle solution of equations (11)–(14).

A standard method for approximately determining the amplitude and frequency of limit cycles is the so-called 'Describing Function (DF) Method' [10]. Unfortunately, the DF method applied to system (11)–(14) yields very poor results. This is because a key assumption of the DF method—that the limit cycle can be well-approximated by a single frequency sinusoid—does not hold for this system. We therefore evaluate the exact limit cycle solutions numerically.

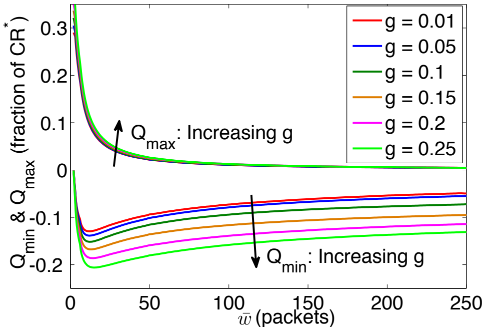

**Figure 7: Queue undershoot and overshoot.**

### 3.3.1 100% Throughput

The first question we consider is: How much buffering is required for DCTCP to achieve 100% throughput? Since queue underflow must be avoided for 100% throughput, we need to determine how large the queue size oscillations are about the operating point $K$.

Assume $\{(\tilde{W}(t), \tilde{\alpha}(t), \tilde{q}(t)) | 0 \leq t < T\}$ is the limit cycle of our system (with period $T$). Define $Q_{min} = \min_{0 \leq t < T} \tilde{q}(t)$, and $Q_{max} = \max_{0 \leq t \leq T} \tilde{q}(t)$ to be the maximum and minimum excursions of the (normalized) queue size during a period. In Fig. 7, we plot $Q_{min}$ and $Q_{max}$ against $\bar{w}$ for some values of $g$.

There are three key observations:

**(i)** The queue overshoot is not sensitive to $g$ and increases as $\bar{w}$ decreases. This follows because the queue overshoot is primarily determined by the rate at which flows increase their window size, and as $\bar{w}$ decreases, the window increase rate of 1 packet/RTT per source becomes increasingly large compared to the bandwidth-delay product.

**(ii)** There is a worst case $\bar{w}$ (about 12-16 packets for the range of $g$ values shown in Fig. 7) at which the queue undershoot is maximized. This implies an interesting property of DCTCP: as per-flow window sizes increase—for example, due to higher and higher link speeds—DCTCP requires less buffers as a fraction of the bandwidth-delay product to achieve high utilization.

**(iii)** The amplitude of queue undershoot increases as $g$ increases. This is to be expected: high values of $g$ cause large fluctuations in $\alpha$ which inhibit DCTCP's ability to maintain a steady sending rate (see Fig. 3 for an example.).[4]

**Choosing K.** As seen in Fig. 7, $Q_{min} \gtrapprox -0.15$ when $g$ is sufficiently small (the choice of $g$ is discussed next). Therefore, to avoid queue underflow, we require $K > |Q_{min}|CR^* \gtrapprox 0.15CR^*$. Substituting $R^* = d + K/C$ in this inequality gives the following guideline:

$$\mathbf{K} \approx \mathbf{0.17Cd}. \tag{17}$$

[4] It is important to note that although $\alpha$ will cease to oscillate as $g \to 0$, queue size oscillations cannot be made arbitrary small by lowering $g$. In fact in Fig. 7, all $g \leq 0.01$ values basically produce the same curve.

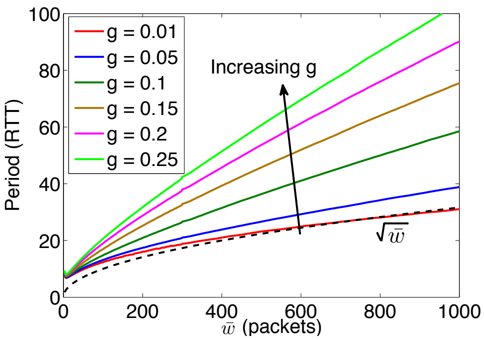

**Figure 8: Limit cycle period of oscillations.**

In words, about 17% of the bandwidth-delay product of buffering is needed for 100% throughput; any more available buffer can be used as headroom to absorb bursts. Note that this value for $K$ is quite close to the guideline $K > (1/7)Cd$, suggested in [1]. This also confirms the validity of the simple sawtooth model for small $g$.

**Limit cycle period & an upper limit for g.** Figure 8 plots the period of oscillations of the limit cycle. The figure suggests that the period grows as $\sqrt{\bar{w}}$ (for small $g$). Now, the marking process $\{p(t-1)\}$ is a periodic 'signal' (with period equal to the period of the limit cycle), which is input to the low-pass filter defined at equation (12). Since the filter has a cutoff frequency of about $g$, for it to be effective, it is necessary that $g$ be smaller than the primary oscillation frequency of the signal, $1/\sqrt{\bar{w}}$. But $\bar{w} = (Cd + K)/N$ and is maximized for $N = 1$. Therefore, we get the following bound:

$$\mathbf{g} \lessapprox \frac{\mathbf{1}}{\sqrt{\mathbf{Cd + K}}}. \qquad (18)$$

Appendix C provides different justification for (18) based on the Hybrid Model introduced in the Section 4. We will revisit the matter of choosing $g$ in Section 4.2, where we provide a lower limit for $g$.

### 3.3.2 Throughput-Delay Tradeoff

We have seen that if $K$ is at least 17% of the bandwidth-delay product, DCTCP achieves 100% throughput. However, when we require very low queueing delays, or when switch buffers are extremely shallow, it may be desirable to choose $K$ even smaller than this. Inevitably, this will result in some loss of throughput. We are interested in quantifying how much throughput is lost, and in effect, deriving a throughput-delay tradeoff curve for DCTCP.

A more accurate model of the switch queue is needed to study throughput loss. Equation (13) ignores the fact that a real queue will never become negative. To account for this and capture the correct behavior when queues underflow, we define $\psi \triangleq K/(Cd)$, and replace (13) with:

$$\frac{d\tilde{q}}{dt} = \begin{cases} \frac{1}{\bar{w}}\frac{\tilde{W}(t)}{(1+\tilde{q}(t))} - 1 & \tilde{q}(t) > \frac{-\psi}{1+\psi}, \\ \max\left(\frac{1}{\bar{w}}\frac{\tilde{W}(t)}{(1+\tilde{q}(t))} - 1, 0\right) & \tilde{q}(t) = \frac{-\psi}{1+\psi}. \end{cases}$$

We can now explore the limit cycle solution as we vary $\psi$, and compute the average throughput (over period $T$):

$$Throughput = \frac{1}{T}\int_0^T \frac{\tilde{W}(t)}{\bar{w}(1+\tilde{q}(t))}dt.$$

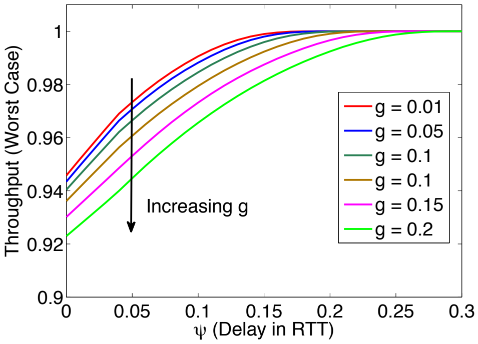

**Figure 9: Worst Case Throughput vs Delay.**

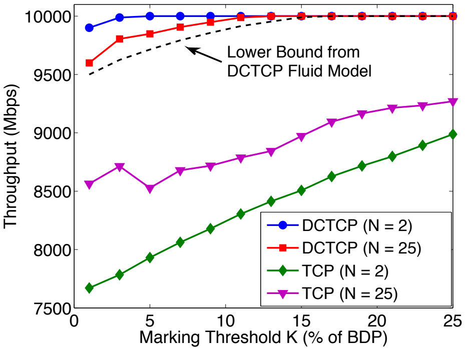

**Figure 10: Throughput vs marking threshold $K$ in ns2 simulation; $K$ is varied from 4 to 100 packets (1-25% of the BDP). Note that as predicted by the analysis, the throughput remains higher than 94% even for $K$ as small as 1% of the BDP.**

In Fig. 9, we plot the worst case throughput as $\psi$ is varied. At each value of $\psi$, the value of $\bar{w}$ which yields the lowest throughput is used. It should be noted that since the queue size is maintained near the marking threshold, the $\psi$ axis also (roughly) corresponds to the average queueing delay. As expected, when $\psi \gtrapprox 0.17$, 100% throughput is achieved (for small $g$). As $\psi$ is lowered, the throughput decreases, but **is always at least 94%** for $g < 0.1$. This indicates that very small marking thresholds can be used in DCTCP, with only a minor loss in throughput. We verify this next through ns2 simulations.

### 3.3.3 Simulations

We use ns2 simulations to evaluate the throughput achieved as the marking threshold $K$ is varied. We choose $C$ = 10Gbps and $d = 480\mu s$, for a bandwidth-delay product of 400 packets (each 1500 Bytes), and set $g = 0.05$. We consider two cases with $N = 2$ and $N = 25$ long-lived flows. The results are shown in Fig. 10. For reference, we also report the results for TCP in the same scenarios.

The simulations clearly show that DCTCP achieves high throughput, even with marking threshold as low as 1% of the bandwidth-delay product. In all cases, we find the throughput is indeed higher than the worst case lower bound predicted by the fluid model. Of course, we see a much worse throughput loss with TCP. An interesting observation is that unlike TCP, whose throughput improves as we increase the number of flows, DCTCP gets lower throughput with $N = 25$ flows than $N = 2$ flows. This is expected from our analysis, because with DCTCP, the worst case queue size fluctuations (and throughput loss) occur when the per-flow window size is around 10-20 packets (see Fig. 7 and observation (ii) in Section 3.3.1).

# 4. ANALYSIS: CONVERGENCE

DCTCP uses the multi-bit information derived from estimating the fraction of marked packets to reduce its window by factors smaller than two. As we have seen in the previous section, this allows it to very efficiently utilize shallow buffers, achieving both high throughput and low queueing delays. However, the reduced multiplicative decrease factors mean slower convergence times: it takes longer for a flow with a large window size to relinquish bandwidth to a flow with a small window size.

Since the convergence time is proportional to the RTT for window-based algorithms and the RTTs in a data center are only a few 100s of microseconds, it is argued in [1] the actual time to converge is not substantial relative to the transfer time of large data files. But an analysis of convergence time has not been conducted in [1]. Based on simulations, [1] reports that the convergence time of DCTCP is a factor 2-3 slower than TCP. The results of this section show that this is, indeed, correct in general.

Our aim in this section is to derive rigorous bounds for the rate of convergence of DCTCP. We consider how fast $N$ DCTCP flows with identical RTTs[5], starting with arbitrary window sizes and values of $\alpha$, converge to their share of the bottleneck bandwidth. Since this is an analysis of the system in transience, the fluid model of the previous section is inadequate. Instead, we use a hybrid (continuous- and discrete-time) model based on the AIMD models introduced in [5, 17]. A key difference between our model and the AIMD models is that ours is non-linear because it models the DCTCP-style multiplicative decrease, whereas the models in [5, 17] and are linear since they correspond to a constant decrease factor (equal to 2 for TCP).

## 4.1 Hybrid Model

Consider $N$ DCTCP flows whose window size and value of $\alpha$ at time $t$ (measured in units of RTT) are denoted by $W_i(t)$ and $\alpha_i(t)$. Assume the window sizes of the $N$ flows are synchronized; i.e., each flow reduces its window at every congestion event. See Fig. 11 for an illustration.

Let $W_{max} = Cd + K$. When $\sum_{i=1}^{N} W_i(t) < W_{max}$, all window sizes increase linearly with slope 1 packet/RTT. Once $\sum_{i=1}^{N} W_i(t) = W_{max}$, a congestion event occurs and each flow cuts its window size according to (4) using its current value of $\alpha$. Note that packets are marked ($p(t) = 1$) for 1 RTT after the window reductions because of the feedback delay. Assume the $k^{th}$ congestion event occurs at time $T_k$. Since the total reduction in $\sum_{i=1}^{N} W_i(t)$ at $T_k$ is equal to $\sum_{i=1}^{N} W_i(T_k)\alpha_i(T_k)/2$, which is regained at the rate of $N$ packets/RTT, the duration of the $k^{th}$ "congestion epoch" is

$$\Delta T_k \triangleq T_{k+1} - T_k = 1 + \frac{\sum_{i=1}^{N} W_i(T_k)\alpha_i(T_k)}{2N}, \qquad (19)$$

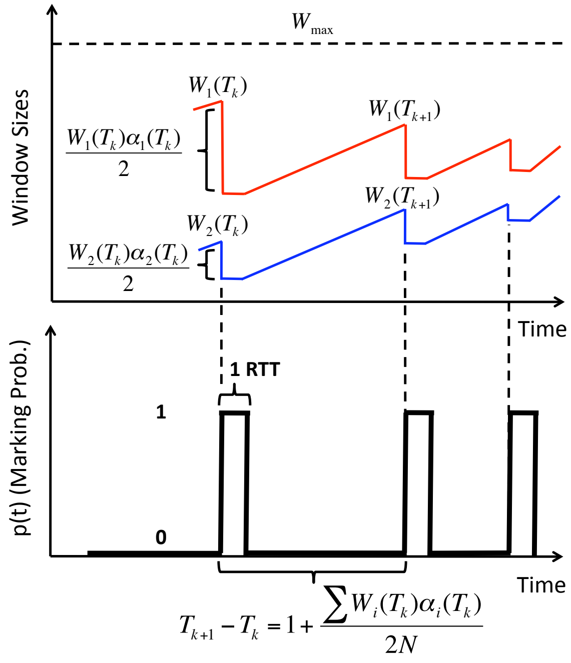

**Figure 11: Hybrid Model. The window sizes are shown for 2 flows. Note that $\sum W_i(T_k) = W_{max}$ for all $k$.**

and we have:

$$W_i(T_{k+1}) = (1 - \frac{\alpha_i(T_k)}{2})W_i(T_k) + \Delta T_k - 1. \qquad (20)$$

It only remains to specify the evolution of $\alpha_i(t)$. This is simply given by:

$$\frac{d\alpha_i}{dt} = g(p(t) - \alpha_i(t)).$$

In particular, $\alpha_i(T_{k+1})$ is the solution of the following initial value problem at time $\Delta T_k$:

$$\frac{dx}{dt} = g(p(t) - x(t)), \quad x(0) = \alpha_i(T_k)$$

$$p(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & t \ge 1 \end{cases}$$

Using this, it is not difficult to show:

$$\alpha_i(T_{k+1}) = e^{-g\Delta T_k}(e^g - 1 + \alpha_i(T_k)). \qquad (21)$$

## 4.2 Rate of Convergence

We make the following assumptions regarding the system parameters:

$$N \ge 2, \quad W_{max} \ge 2N, \quad g \le \frac{1}{\sqrt{W_{max}}}. \qquad (22)$$

The first two assumptions are natural for studying convergence, and the third is in accordance with the guidelines of the previous section regarding steady state. Let $W^* = W_{max}/N$. The main result of this section is given by the following theorem.

[5] This ensures that in equilibrium, each flow gets $(1/N)^{th}$ of the bottleneck bandwidth. See Section 5 for a discussion of the bias against flows with longer RTTs.

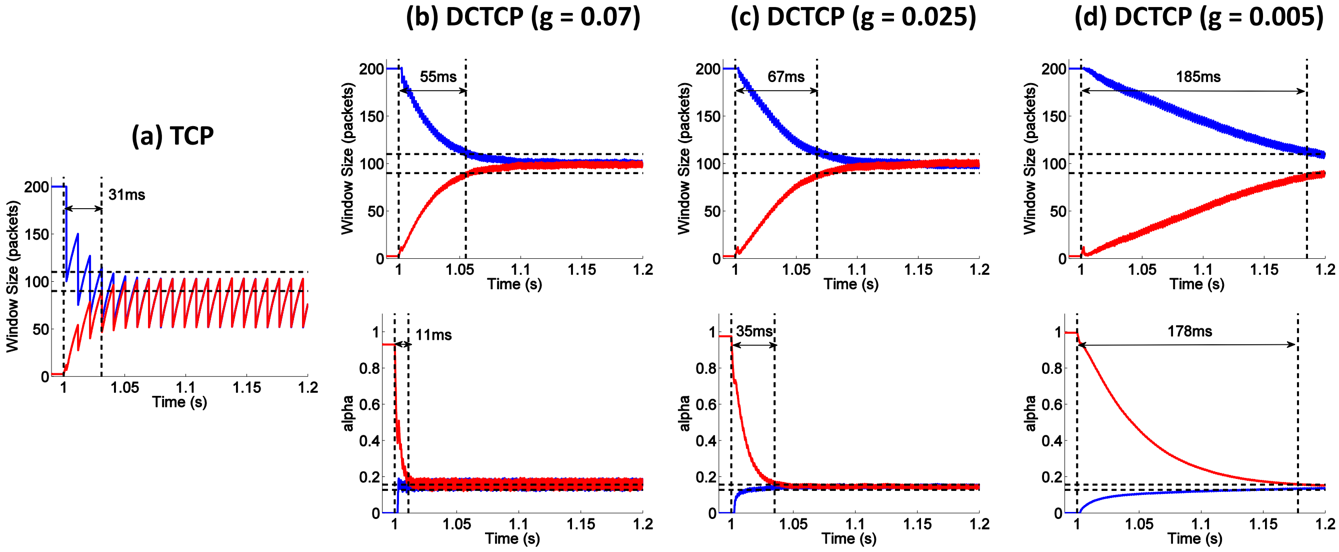

**Figure 12: ns2 simulation of convergence time for 2 flows. The chosen $g$ values for DCTCP correspond to: (b) $1/\sqrt{W_{max}}$, (c) $5/W_{max}$, and (d) $1/W_{max}$.**

**Theorem** 2. *Consider $N$ DCTCP flows evolving according to* (20) *and* (21)*, with parameters satisfying* (22)*. Suppose that $W_i(0)$ and $\alpha_i(0)$ are arbitrary. Let $0 < \alpha^* \le 1$ be the unique positive solution of*

$$\alpha^* = e^{-g(1+W^*\alpha^*/2)}(e^g - 1 + \alpha^*). \tag{23}$$

*Then $W_i(T_n) \to W^*$ (defined above), and $\alpha_i(T_n) \to \alpha^*$ for all $1 \le i \le N$ as $n \to \infty$. Moreover:*

$$|W_i(T_n) - W^*| < 2W_{max}\Big(1 + g^6 n^2\Big)e^{-\beta(T_n - T_{P2})}, \tag{24}$$

*for all $1 \le i \le N$, where:*

$$\beta = \min\Big(g, \frac{-\log(1-\alpha^*/2)}{1 + W^*\alpha^*/2}\Big), \tag{25}$$

$$T_{P2} = \frac{\log(2W^*/g^6)}{g} + 2(1 + W^*/2).$$

The proof of Theorem 2 is given in Section 4.4. Here, we make a few remarks about the convergence rate. The crucial term in (24) is the exponential decay $e^{-\beta T_n}$. The convergence rate is therefore chiefly determined by (25), which has the following interpretation. Two things occur (simultaneously) during convergence of DCTCP: (i) the $\alpha_i(T_n)$ converge to $\alpha^*$, and (ii) the $W_i(T_n)$ converge to $W^*$. It is shown in the proof of Theorem 2 that (i) happens with rate $g$, and (ii) with rate $\gamma = -\log(1-\alpha^*/2)/(1+W^*\alpha^*/2)$. The overall convergence rate is determined by the slower of the two.

**Lower limit for g.** We have seen that $g$ should be smaller than $1/\sqrt{W_{max}}$ to not adversely affect steady state performance. Equation (25) suggests a lower bound for $g$, in order to not slow down convergence. Note that:

$$\frac{1}{W^*} < \gamma \approx \frac{-2\log(1-\alpha^*/2)}{W^*\alpha^*} < \frac{2\log 2}{W^*}. \tag{26}$$

Therefore if $g > (2\log 2)/W^*$, it will not limit the convergence rate. Of course, in practice, we don't know the number of flows $N$, and so don't know $W^*$. However, in data centers, there are typically a small number of large active flows on a path[6]. Therefore, we mainly need to focus on the small $N$ case, for which convergence is also slowest. With these considerations, we propose the following guideline:

$$\frac{\mathbf{5}}{\mathbf{W_{max}}} \lessapprox \mathbf{g} \lessapprox \frac{\mathbf{1}}{\sqrt{\mathbf{W_{max}}}}. \tag{27}$$

**Comparison with TCP.** As mentioned in the beginning of this section, DCTCP converges slower than TCP. But how much slower is DCTCP? It is straight forward to show that the convergence rate of TCP is given by:

$$\beta_{TCP} = \frac{\log 2}{1 + W^*/2} \approx \frac{2\log 2}{W^*}.$$

Therefore, (25) and (26) imply that for $g$ properly chosen large enough:

$$\beta_{DCTCP} < \beta_{TCP} < (2\log 2)\beta_{DCTCP} \approx 1.4 \times \beta_{DCTCP}.$$

This means that the DCTCP convergence rate is at most 40% slower than TCP. It is important to note however that this is a statement about the *asymptotic* rate of convergence. In practice, all the terms present in (24) affect the convergence time. Therefore, in simulations, the actual time to converge is about a factor 1.5-2 larger (see also [1]).

**Bounds on $\alpha^*$.** The following bounds are proven in Appendix C.

$$\frac{1}{2}\sqrt{\frac{2}{W^*}} - g < \alpha^* < \sqrt{\frac{2}{W^*}} + g.$$

## 4.3 Simulations

We have verified the results of our analysis using extensive ns2 simulations. Figure 12 shows a representative example

[6] For example, measurements reported in [1] show at most 4 flows larger than 1MB concurrently active at a server in any 50ms period. Note that only the large flows need to be considered, as only they can possibly converge.

with 2 flows. The choice of parameters $C = 10$Gbps, $d = 200\mu$s, and $K = 35$ (or 17% of the BDP) gives $W_{max} \approx 200$ packets. One flow starts at the beginning of the simulation, and grabs all the available capacity (its window size reaches 200). At time 1sec, a second flow begins with a window of 1 packet, and we are interested in how long it takes for both window sizes to get within 10% of the fair share value (100 packets) for the first time. In order to have the worst case convergence time with DCTCP, we begin the second flow with $\alpha = 1$, whereas the first flow has $\alpha = 0$ before time 1sec; so the second flow actually cuts its window by much larger factors initially.

We test three values of $g$ for DCTCP. With $g = 1/\sqrt{W_{max}}$, as expected from the analysis, the $\alpha$ variables converge much quicker than the window sizes (about 5x faster). Reducing $g$ to $5/\sqrt{W_{max}}$, the convergence times of the $\alpha$ and window sizes get closer, with the $\alpha$ still converging about twice as fast. Here, the increase in convergence time is small (67ms up from 55ms). But when $g$ is further reduced to $1/W_{max}$, the convergence of $\alpha$ and window sizes take about the same amount of time, showing that the limiting factor is now the convergence of $\alpha$. In this case, the small value of $g$ significantly increases the total convergence time (185ms). A final observation is that when $g$ is appropriately chosen according to (27), the convergence time of DCTCP is indeed up to about a factor of 2 longer than TCP.

## 4.4 Proof of Theorem 2

The key idea in proving Theorem 2 is to consider convergence as happening in three separate phases.

**Phase 1.** Initially, the $\alpha_i$ values get close to each other. In fact (21) implies that for any $i$ and $j$:

$$|\alpha_i(T_n) - \alpha_j(T_n)| \leq e^{-gT_n}|\alpha_i(T_0) - \alpha_j(T_0)| \leq e^{-gT_n} \quad (28)$$

We have the following simple Lemma.

**Lemma** 1. *Let* $\bar{\alpha}(T_k) = \sum_{i=1}^{N} \alpha_i(T_k)/N$*. Then:*

$$|\alpha_i(T_n) - \bar{\alpha}(T_n)| \leq e^{-gT_n}$$

*for all* $1 \leq i \leq N$*.*

Proof. This follows by applying the triangle inequality and using (28). □

We take Phase 1 to last until time $T_{P1} \triangleq log(2W^*)/g$, so that:

$$|\alpha_i(T_k) - \bar{\alpha}(T_k)| \leq e^{-gT_k} = \frac{1}{2W^*}e^{-g(T_k - T_{P1})}. \quad (29)$$

**Phase 2.** The second phase begins with $T_k \geq T_{P1}$. In this phase, the $\alpha_i$ converge to a positive constant $\alpha^*$. The following proposition is the main convergence result for Phase 2 and is proved in Appendix B.

**Proposition** 1. *For* $n \geq 1$*, and all* $1 \leq i \leq N$*:*

$$|\alpha_i(T_n) - \alpha^*| \leq A_0 n e^{-g(T_n - T_{P1})}, \quad (30)$$

*where* $A_0 = e^{2g(1+W^*/2)}$*.*

We take Phase 2 to last until time

$$T_{P2} \triangleq T_{P1} + 2(1 + W^*/2) - 6\log(g)/g,$$

so that:

$$|\alpha_i(T_k) - \alpha^*| \leq A_0 k e^{-g(T_k - T_{P1})} = g^6 k e^{-g(T_k - T_{P2})}. \quad (31)$$

In particular, it is easy to check that given (22), $g^6 T_{P2} \leq g$, which implies that for $T_k \geq T_{P2}$:

$$\zeta_k \triangleq g^6 k e^{-g(T_k - T_{P2})} \leq g^6 T_k e^{-g(T_k - T_{P2})} \leq g. \quad (32)$$

**Phase 3.** The third and final phase begins with $T_k \geq T_{P2}$. In Phase 3, the $\alpha_i$ values are all close to $\alpha^*$, and the sources essentially perform AIMD with a decrease factor of $\alpha^*/2$.

The following Lemma is the key ingredient for convergence in Phase 3.

**Lemma** 2. *For* $T_k \geq T_{P2}$*, and any* $1 \leq i, j \leq N$*:*

$$|W_i(T_{k+1}) - W_j(T_{k+1})| \leq e^{-\gamma \Delta T_k}|W_i(T_k) - W_j(T_k)| + 2W_{max}\zeta_k, \quad (33)$$

*where* $\gamma = -\log(1 - \alpha^*/2)/(1 + W^*\alpha^*/2)$*.*

Proof. Using (20), (31), and (32), we have:

$$\begin{aligned} |W_i(T_{k+1}) - W_j(T_{k+1})| &\leq (1 - \frac{\alpha^*}{2})|W_i(T_k) - W_j(T_k)| + \\ &\frac{|W_i(T_k)|}{2}|\alpha_i(T_k) - \alpha^*| + \frac{|W_j(T_k)|}{2}|\alpha_j(T_k) - \alpha^*|, \\ &\leq (1 - \frac{\alpha^*}{2})|W_i(T_k) - W_j(T_k)| + W_{max}\zeta_k. \end{aligned}$$

Noting that

$$\Delta T_k \leq 1 + W^*\alpha^*/2 + W^*\zeta_k/2, \quad (34)$$

the result follows from:

$$(1 - \alpha^*/2) \leq e^{-\gamma(\Delta T_k - W^*\zeta_k/2)} \leq e^{-\gamma \Delta T_k}(1 + \zeta_k),$$

where the last inequality is true because:

$$\begin{aligned} e^{\gamma W^* \zeta_k/2} &\leq e^{-\zeta_k \log(1-\alpha^*/2)/\alpha^*} \\ &\leq e^{\zeta_k \log 2} = 2^{\zeta_k} \leq 1 + \zeta_k. \quad (35) \end{aligned}$$

This holds for $\zeta_k \leq 1$, which we have from (32). □

We can now prove Theorem 2. Let $r \triangleq \inf\{k | T_k \geq T_{P2}\}$ and $\beta \triangleq \min(g, \gamma)$. Similar as we did for Proposition 1, we iterate (33) backwards starting from $n \geq r$ to get:

$$\begin{aligned} &|W_i(T_n) - W_j(T_n)| \\ &\quad \leq W_{max}\Big(e^{\beta(T_r - T_{P2})} + 2g^6 \sum_{k=r}^{n-1} k e^{\beta \Delta T_k}\Big) e^{-\beta(T_n - T_{P2})}. \end{aligned}$$

But:

$$e^{\beta(T_r - T_{P2})} \leq e^{\gamma(1+W^*/2)} \leq e^{-\log(1-\alpha^*/2)/\alpha^*} \leq e^{\log 2} = 2,$$

and using (34), (35), and (32):

$$\begin{aligned} \sum_{k=r}^{n-1} k e^{\beta \Delta Tk} &\leq \sum_{k=r}^{n-1} k e^{\gamma(1+W^*\alpha^*/2)} e^{\gamma W^* \zeta_k/2} \\ &\leq (1 - \alpha^*/2)(1 + g) \sum_{k=r}^{n-1} k < n^2. \end{aligned}$$

Noting that $|W_i(T_n) - W^*| \leq \frac{1}{N}\sum_{j=1}^{N} |W_i(T_n) - W_j(T_n)|$, we have established (24) for $n \geq r$. The result is trivial for $n < r$, completing the proof. □

## 5. RTT-FAIRNESS

It is well-known that TCP has a bias against flows with long round-trip times; i.e., flows with longer RTTs get a lower share of the bottleneck bandwidth when competing with flows with shorter RTTs [11, 6, 2, 3, 23]. This is due to the fact that the rate at which flows increase their window size is inversely proportional to their RTT (one packet per RTT). Therefore, flows with short RTTs grab bandwidth much more quickly than flows with long RTTs and settle at a higher rate in steady-state. In fact, it has been shown that the throughput achieved by a TCP flow is inversely proportional to $RTT^{\theta}$ with $1 \le \theta \le 2$ [11].

Another important factor which affects RTT-fairness is synchronization between flows. Typically, higher synchronization, in terms of detecting a loss or mark event, leads to worse RTT-fairness. In fact, it has been argued that active queue management (AQM) schemes like RED [7] which avoid synchronization by probabilistically dropping (or marking) packets improve RTT fairness compared to Drop-tail queues [2, 23].

Since DCTCP employs the same additive increase mechanism used by TCP, it is expected that DCTCP also exhibits a bias against flows with longer RTTs. Moreover, the active queue management of DCTCP is (by design) similar to Drop-tail (albeit with marking instead of dropping) and prone to causing synchronization. This makes a study of how well DCTCP handles RTT diversity important.

### 5.1 Simulations

We consider the following ns2 simulation to investigate RTT-fairness. Four flows share a single 10Gbps bottleneck link in a "dumbbell" topology. The first two of these flows have a fixed RTT of 100$\mu$s. The RTT of the other two flows is varied from 100$\mu$s to 1ms. We measure the throughput for all flows in each test, and compute the ratio of the throughput of the first two flows to that of the second two flows, as a function of the RTT ratio.

The DCTCP parameters chosen are $K = 35$ packets and $g = 1/16$. For reference, we compare the results with TCP using the DCTCP style "Drop-tail" (0–1) marking, and also TCP with RED marking[7]. The results are shown in Fig. 13. The algorithm labeled "DCTCP–Improved Fairness" is described in the next section.

As expected, DCTCP does exhibit a bias against flows with longer RTTs. The simulations indicate that DCTCP's RTT-fairness is better than TCP–Drop-tail, which has approximately squared RTT-fairness ($Throughput \propto RTT^{-2}$), but worse than TCP–RED, which has approximately linear RTT-fairness ($Throughput \propto RTT^{-1}$). In this and other simulations not reported here, we observed that when the RTT ratio is moderate, DCTCP exhibits slightly worse than linear RTT-fairness, but tends to squared RTT-fairness as the RTT ratio becomes large. Apparently, the smooth window adjustments made by DCTCP help alleviate some of the synchronization effects caused by the 0–1 marking strategy, thereby improving DCTCP's RTT-fairness.

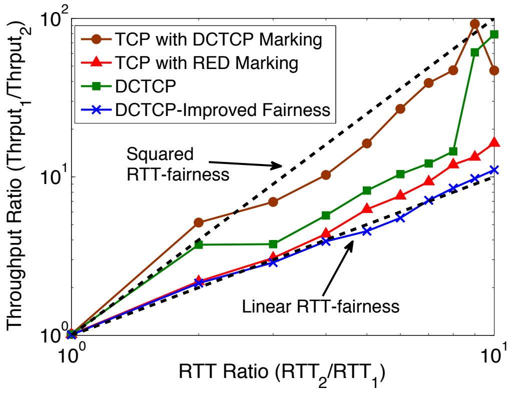

**Figure 13: RTT-fairness in ns2 simulation: 2 groups each with 2 flows are activated. Flows in group 1 have $RTT_1 = 100\mu s$. The $RTT$ for flows in group 2 ($RTT_2$) is varied from 100$\mu s$ to $1ms$. Note the log-log scale.**

### 5.2 DCTCP–Improved Fairness

The DCTCP fluid model (Section 2.2) suggests a very simple change which considerably improves the RTT-fairness of DCTCP. We will first state this change, contrasting it with the DCTCP algorithm defined in Section 2.1. We will then explain how the fluid model suggests this change.

Recall that the DCTCP algorithm reduces the window size according to

$$W \leftarrow W(1 - \alpha/2)$$

in response to a marked ACK, and that this is done at most once for each window of data. Instead, we propose subtracting $\alpha/2$ from the window size for each marked ACK, resulting in the following simple window update equation:

```
for each received ACK:
```

$$W \leftarrow W + \begin{cases} 1/W & \text{if ECN} = 0 \\ 1/W - \alpha/2 & \text{if ECN} = 1 \end{cases} \tag{36}$$

Note that a full window of marked ACKs will cause a reduction of about $W\alpha/2$, which is the same amount DCTCP would reduce the window size upon receiving a mark. A nice feature is that since (36) applies to every ACK, it does not require maintaining extra state to prevent window reductions from happening more than once per window of data.

The simulation results in Fig. 13 confirm that this simple change significantly improves RTT-fairness (especially at high RTT ratios), and does indeed achieve linear RTT-fairness. In fact, it even achieves a slightly better RTT-fairness than TCP–RED.

**Connection with Fluid Model.** Consider $N$ flows with round-trip times $RTT_i$ ($1 \le i \le N$) in steady-state. Recall the source side equations:

$$\frac{dW_i}{dt} = \frac{1}{RTT_i} - \frac{W_i(t)\alpha_i(t)}{2RTT_i} p(t - RTT_i), \tag{37}$$

$$\frac{d\alpha_i}{dt} = \frac{g}{RTT_i} \left( p(t - RTT_i) - \alpha_i(t) \right), \tag{38}$$

where, for simplicity, we neglect the contribution of the (time-varying) queueing delay to the RTT. Note that each flow sees a time-delayed version of the common marking

[7] For RED, the marking probability increases linearly from 0 to 10% as the *average* queue length (EWMA with weight 0.1) increases from 30 to 100 packets. These parameters are chosen to ensure stability of RED in this configuration and yield roughly the same queue lengths as DCTCP.

process $p(\cdot)$. Therefore, the $\alpha_i(\cdot)$ processes—each given by passing $p(\cdot)$ through a low pass filter—all oscillate around the same value, namely, the duty cycle of $p(\cdot)$. This leads to the following key observation based on (37): the *average* window size for flow $i$ does not depend on $RTT_i$; i.e., the fluid model suggests that all flows should on average have the same window size, thereby achieving linear RTT-fairness (because $Throughput_i = W_i/RTT_i$).

However, as seen in Fig. 13, simulation results indicate that the actual RTT-fairness is worse than linear. The key source of discrepancy between the fluid model and simulations lies in the *continuous* dynamics present in the fluid model.[8] In particular, let us consider the manner in which the window size decreases in (37). While $p(t - RTT_i) = 1$, the window steadily decreases with rate $W_i(t)\alpha_i(t)/(2RTT_i)$. In contrast, the packet-level algorithm reduces its window *instantaneously* upon receiving a mark. This suggests the change previously discussed to the DCTCP algorithm in equation (36), which bridges the gap between the packet-level and fluid dynamics.

Intuitively, this change improves the RTT-fairness by allowing flows with long RTTs to reduce their window sizes (on average) by a smaller factor compared to flows with short RTTs. This is because in a typical congestion event, flows with short RTTs receive marks earlier and reduce their window sizes, thereby relieving congestion. In such cases, less than a full window of packets from a flow with a long RTT will be marked, and therefore, the net decrease to its window size based on (36) will be smaller than the standard DCTCP window reduction, $W\alpha/2$.

# 6. FINAL REMARKS

We analyzed DCTCP, a TCP variant for data centers which has recently been proposed in [1]. Our analysis shows that DCTCP can achieve very high throughput while maintaining low buffer occupancies. Specifically, we found that with a marking threshold, $K$, of about 17% of the bandwidth-delay product, DCTCP achieves 100% throughput, and that even for values of $K$ as small as 1% of the bandwidth-delay product, its throughput is at least 94%. While DCTCP converges slower than TCP, we found that its convergence rate is no more than a factor 1.4 slower than TCP. We also evaluated the RTT-fairness of DCTCP, and found a simple change to the algorithm, which considerably improves its RTT-fairness.

In future, it is worth understanding the behavior of DCTCP in general networks. The work in this paper has focused on the case where there is a single bottleneck.

## Acknowledgments

Mohammad Alizadeh and Adel Javanmard are supported by Caroline and Fabian Pease Stanford Graduate Fellowships. We thank the anonymous reviewers whose comments helped us improve the paper.

[8] We emphasize that the mentioned discrepancy affects the accuracy of the fluid model only for heterogenous RTTs. As shown in Section 2.2, the fluid model is very accurate when sources have identical RTTs.

# 7. REFERENCES

[1] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center TCP (DCTCP). In *Proceedings of SIGCOMM '10*, pages 63–74, New York, NY, USA, 2010. ACM.

[2] E. Altman, C. Barakat, E. Laborde, P. Brown, and D. Collange. Fairness analysis of TCP/IP. In *Proceedings of the 39th IEEE Conference on Decision and Control, 2000.*, volume 1, pages 61 –66 vol.1, 2000.

[3] E. Altman, T. Jiménez, and R. Núñez Queija. Analysis of two competing TCP/IP connections. *Perform. Eval.*, 49:43–55, September 2002.

[4] K. Astrom, G. Goodwin, and P. Kumar. *Adaptive Control, Filtering, and Signal Processing.* SpringerVerlag, 1995.

[5] F. Baccelli and D. Hong. AIMD, fairness and fractal scaling of TCP traffic. In *INFOCOM*, 2002.

[6] P. Brown. Resource sharing of TCP connections with different round trip times. In *INFOCOM*, 2000.

[7] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Netw.*, 1(4):397–413, 1993.

[8] C. V. Hollot, V. Misra, D. Towsley, and W. bo Gong. A control theoretic analysis of RED. In *Proceedings of IEEE INFOCOM*, pages 1510–1519, 2001.

[9] 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*, 47:945–959, 2002.

[10] H. Khalil. *Nonlinear Systems.* Prentice Hall, 2002.

[11] T. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *Networking, IEEE/ACM Transactions on*, 5(3):336 –350, June 1997.

[12] H. Low, O. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22:28–43, 2002.

[13] 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 Comput. Commun. Rev.*, 30(4):151–160, 2000.

[14] A. H. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods.* Wiley-VCH, 2007.

[15] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[16] K. Ramakrishnan, S. Floyd, and D. Black. RFC 3168: the addition of explicit congestion notification (ECN) to IP.

[17] R. Shorten, D. Leith, J. Foy, and R. Kilduff. Analysis and design of AIMD congestion control algorithms in communication networks. *Automatica*, 41(4):725 – 730, 2005.

[18] R. Shorten, F. Wirth, and D. Leith. A positive systems model of TCP-like congestion control: asymptotic results. *IEEE/ACM Trans. Netw.*, 14(3):616–629, 2006.

[19] R. Srikant. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications).* SpringerVerlag, 2004.

[20] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, G. A. Gibson, and B. Mueller. Safe and effective fine-grained TCP retransmissions for datacenter communication. In *Proceedings of SIGCOMM '09*, pages 303–314, New York, NY, USA, 2009. ACM.

[21] A. Vishwanath, V. Sivaraman, and M. Thottan. Perspectives on router buffer sizing: recent results and open problems. *SIGCOMM Comput. Commun. Rev.*, 39:34–39, March 2009.

[22] Q.-G. Wang, T. H. Lee, and C. Lin. *Relay Feedback: Analysis, Identification and Control.* SpringerVerlag, 2003.

[23] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks. In *INFOCOM*, 2004.

# APPENDIX

## A. PROOF SKETCH OF THEOREM 1

The proof proceeds by computing the Jacobian of the Poincaré map at its fixed point $x^*_\alpha$ and requiring it to be stable. Define $\mathcal{B}_\epsilon(x^*_0) \triangleq \{\xi \in \mathcal{R}^3 : ||\xi - x^*_0|| \leq \epsilon\}$. We need the following lemma to verify that starting from a neighborhood of $x^*_\alpha$, successive switching can occur and thereby the Poincaré map is well-defined.

**Lemma** 3. *Assume that $F(\mathrm{x}, u)$ is infinitely differentiable with respect to* x. *For any given $\delta > 0$, there exists $\epsilon_\delta > 0$ such that the trajectory of $x(t)$ starting from any traversing point in $B_{\epsilon_\delta}(x^*_\alpha) \cap S$ (resp. $B_{\epsilon_\delta}(x^*_\beta) \cap S$ ) at time $t$ will traverse the plane $S$ again and the traversing time $t+1+t_{trav}$ satisfies $h_\alpha - \delta < t_{trav} < h_\alpha + \delta$ (resp. $h_\beta - \delta < t_{trav} < h_\beta + \delta$).*

Intuitively, this lemma is true because of continuity and can be proved following a very similar argument as that presented in the proof of lemma 3.3.3 of [22]. In the next lemma, we characterize the trajectory of $x(t)$ starting from a region close to $x^*_\alpha$.

**Lemma** 4. *There exists $\epsilon > 0$ such that any trajectory of the system described by* (16)*, starting from $x_\alpha = x^*_\alpha + \delta x_\alpha \in B_\epsilon(x^*_\alpha) \cap S$ will intersect and traverse $S$, and the traversing point $x(t_1)$ satisfies*

$$x(t_1) - x^*_\beta = Z_2 \delta x_\alpha + O(\delta^2).$$

Proof. From lemma 3, there exists some $\epsilon > 0$ such that the trajectory of $x(t)$ starting from $B_\epsilon(x^*_\alpha) \cap S$ will traverse $S$ at some instant $t_1 = 1 + h_\alpha + \delta h_\alpha$. Define $\delta x(t) = x(t) - x^*(t)$. Then,

$$\begin{aligned}\dot{\delta x}(t) &= F(x(t), u(t-1)) - F(x^*(t), u(t-1)) \\ &= F(x^*(t) + \delta x(t), u(t-1)) - F(x^*(t), u(t-1)) \\ &= J_F(x^*(t), u(t-1))\ \delta x(t) + O(\delta^2). \qquad (39)\end{aligned}$$

where we have used the series expansion of $F(\mathrm{x}^* + \delta\mathrm{x}, u)$ around $(\mathrm{x}^*, u)$. Since $\delta x(0) = \delta x_\alpha$, from (39) we get:

$$\begin{aligned}&x(1 + h_\alpha + \delta h_\alpha) - x^*(1 + h_\alpha + \delta h_\alpha) \\ &= \exp\left(\int_0^{1+h_\alpha+\delta h_\alpha} {}_{J_F(x^*(s),u(s-1))ds}\right) \delta x_\alpha + O(\delta^2) \\ &= \exp\left(\int_0^{1+h_\alpha} {}_{J_F(x^*(s),u(s-1))ds}\right) \delta x_\alpha + O\left(\delta_{h_\alpha}\delta_{x_\alpha}\right) + O(\delta^2) \\ &= \exp\left(\int_0^{1+h_\alpha} {}_{J_F(x^*(s),u(s-1))ds}\right) \delta x_\alpha + O(\delta^2) \qquad (40)\end{aligned}$$

Furthermore, making a series expansion in $\delta h_\alpha$, we get:

$$x^*(1 + h_\alpha + \delta h_\alpha) - x^*_\beta = F(x^*_\beta, u_\alpha)\ \delta h_\alpha + O(\delta^2), \qquad (41)$$

where we use the fact that $x^*(1 + h_\alpha) = x^*_\beta$. Using (40) and (41), we have:

$$\begin{aligned}x(1 + h_\alpha + \delta h_\alpha) - x^*_\beta =& \exp\left(\int_0^{1+h_\alpha} {}_{J_F(x^*(s),u(s-1))ds}\right) \delta x_\alpha \\ &+ F(x^*_\beta, u_\alpha)\delta h_\alpha + O(\delta^2). \qquad (42)\end{aligned}$$

Since $x(1+h_\alpha+\delta h_\alpha)$ and $x^*_\beta$ are on the switching plane $S$, we have $cx(1+h_\alpha+\delta h_\alpha) = cx^*_\beta = 0$. After some manipulations, we get:

$$\delta h_\alpha = -\frac{c\ \exp\left(\int_0^{1+h_\alpha} {}_{J_F(x^*(s),u(s-1))ds}\right) \delta x_\alpha}{c\ F(x^*_\beta, u_\alpha)} + O(\delta^2). \quad (43)$$

Plugging (43) in (42) gives the desired result. □

Similarly, if we define $t_2$ to be the time taken for trajectory of $x(t)$ to traverse $S$ again (at a traversing point close to $x^*_\alpha$), we have:

$$x(t_2) = x^*_\alpha + Z_1 \delta x_\beta + O(\delta^2)$$

with initial condition $x^*_\beta + \delta x_\beta = x^*_\beta + Z_2 \delta x_\alpha + O(\delta^2)$. Replacing $\delta x_\beta$ with $Z_2 \delta x_\alpha + O(\delta^2)$ in the above equality yields:

$$x(t_2) = x^*_\alpha + Z_1 Z_2 \delta x_\alpha + O(\delta^2).$$

Note that $x(t_2) = P(x_\alpha)$ and $x^*_\alpha = P(x^*_\alpha)$. Thus,

$$P(x_\alpha) = P(x^*_\alpha) + Z_1 Z_2 \delta x_\alpha + O(\delta^2).$$

Consequently, the Jacobian of the Poincaré map at $x^*_\alpha$ is $Z_1 Z_2$. Therefore, the Poincaré map is locally stable at $x^*_\alpha$ (respectively, the limit cycle $x^*$ is locally stable) if and only if all the eigenvalues of $Z_1 Z_2$ are inside the unit circle; i.e., $\rho(Z_1 Z_2) < 1$.

## B. PROOF OF PROPOSITION 1

Recall that the second phase begins with $T_k \geq T_{P1}$. Since the $\alpha_i$ are all close their mean $\bar{\alpha}$ after Phase I, it suffices to study the evolution of $\bar{\alpha}$. This is very useful, as it reduces an $N$-dimensional problem into a 1-dimensional one.

Define the function $f : [0, \infty] \longrightarrow [0, \infty]$:

$$f(\alpha) = e^{-g(1+W^*\alpha/2)}(e^g - 1 + \alpha).$$

Using (21) and (19) we have:

$$\begin{aligned}\bar{\alpha}(T_{k+1}) &= e^{-g\Delta T_k}(e^g - 1 + \bar{\alpha}(T_k)) \\ &= f(\bar{\alpha}(T_k))e^{g/2N \sum_i W_i(T_k)(\bar{\alpha}(T_k) - \alpha_i(T_k))}.\end{aligned}$$

Invoking (29), we can bound the exponent term

$$\Big|\frac{g}{2N} \sum_i W_i(T_k)(\bar{\alpha}(T_k) - \alpha_i(T_k))\Big| \leq \frac{g}{4} e^{-g(T_k - T_{P1})},$$

to get:

$$\bar{\alpha}(T_{k+1}) = f(\bar{\alpha}(T_k))(1 + \delta_k), \qquad (44)$$

where $|\delta_k| \leq \epsilon_k \triangleq g e^{-g(T_k - T_{P1})}/2$. Since the $\delta_k$ vanish exponentially fast, (44) implies that $\bar{\alpha}$ essentially evolves according to the iteration $\bar{\alpha}(T_{k+1}) \approx f(\bar{\alpha}(T_k))$.

Before proceeding, we state a few useful properties of $f$.

**Lemma** 5. *For parameters satisfying* (22)*:*

(i) *$f$ maps $[0, 1]$ to $[0, 1]$; i.e., $f([0, 1]) \subseteq [0, 1]$.*

(ii) *$f$ has a global maximum at $\alpha_{max} = 2/(gW^*) - (e^g - 1)$, and is strictly increasing for $\alpha < \alpha_{max}$, and strictly decreasing for $\alpha > \alpha_{max}$.*

(iii) *$f(\alpha_{max}) < 4\alpha_{max}/5$.*

(iv) *$f$ has a unique fixed point $0 < \alpha^* < \min(\alpha_{max}, 1)$.*

(v) *for any $0 \leq \alpha \leq \alpha_{max}$,*

$$|f(\alpha) - \alpha^*| \leq e^{-g(1+W^*\alpha/2)}|\alpha - \alpha^*|.$$

PROOF. (i), (ii), and (iii) follow after some simple algebraic manipulations.

(iv) Since $f(0) > 0$, $f(1) < 1$, and $f(\alpha_{max}) < \alpha_{max}$, the continuity of $f$ implies the existence of a fixed point in $(0, \min(\alpha_{max}, 1))$. The derivative of $f$ is given by:

$$f'(\alpha) = -\frac{gW^*}{2} f(\alpha) + e^{-g(1+W^*\alpha/2)}.$$

Because $f(\alpha) \geq 0$, this implies:

$$|f'(\alpha)| \leq \max(\frac{gW^*}{2} f(\alpha_{max}), e^{-g}) \leq \max(4/5, e^{-g}) < 1.$$

Thus, $f$ is a contraction, and the uniqueness of the fixed point follows by the contraction mapping principle.

(v) By the Mean-Value Theorem:

$$f(\alpha) - \alpha^* = f'(c)(\alpha - \alpha^*),$$

for some $c \in (0, \alpha_{max})$. Note that $f'(c) > 0$ by part (ii). Assume $\alpha > \alpha^*$. Then $f(\alpha) > \alpha^*$, and we have:

$$\alpha^* = f(\alpha^*) > e^{-g(1+W^*\alpha/2)}(e^g - 1 + \alpha^*),$$
$$\Longrightarrow f(\alpha) - \alpha^* < e^{-g(1+W^*\alpha/2)}(\alpha - \alpha^*).$$

An identical argument for $\alpha < \alpha^*$ completes the proof. □

**Lemma** 6. *Let* $m \triangleq \inf\{k | T_k \geq T_{P1}\}$. *Then for* $k \geq m+1$*:*

$$|\bar{\alpha}(T_{k+1}) - \alpha^*| \leq e^{-g\Delta T_k} |\bar{\alpha}(T_k) - \alpha^*| + 2\epsilon_k. \quad (45)$$

PROOF. Using (44) and Lemma 5 (iii), for $k \geq m+1$:

$$\bar{\alpha}(T_k) < (4\alpha_{max}/5)(1 + g/2) < \alpha_{max}.$$

The last inequality uses $g < 1/2$, which follows from (22). Hence, using (44) and Lemma 5 (i) and (v):

$$|\bar{\alpha}(T_{k+1}) - \alpha^*| \leq |f(\bar{\alpha}(T_k)) - \alpha^*| + f(\bar{\alpha}(T_k))\delta_k$$
$$\leq e^{-g(1+W^*\bar{\alpha}(T_k)/2)} |\bar{\alpha}(T_k) - \alpha^*| + \epsilon_k.$$

The result follows by noting that:

$$e^{-g(1+W^*\bar{\alpha}(T_k)/2)} = e^{-g\Delta T_k} e^{g/2N \sum_i W_i(T_k)(\alpha_i(T_k) - \bar{\alpha}(T_k))}$$
$$\leq e^{-g\Delta T_k}(1 + \epsilon_k). \quad \square$$

We are now ready to prove Proposition 1. Let $m$ be defined as in Lemma 6. Iterating (45) backwards from $n \geq m+1$:

$$|\bar{\alpha}(T_n) - \alpha^*|$$
$$\leq e^{-g(T_n - T_{m+1})} |\bar{\alpha}(T_{m+1}) - \alpha^*| + 2 \sum_{k=m+1}^{n-1} \epsilon_k e^{-g(T_n - T_{k+1})},$$
$$\leq \Big(e^{g(T_{m+1} - T_{P1})} + g \sum_{k=m+1}^{n-1} e^{g(T_{k+1} - T_k)}\Big) e^{-g(T_n - T_{P1})}.$$

Using $\Delta T_k \leq 1 + W^*/2$ and Lemma 1, we have

$$|\alpha_i(T_n) - \alpha^*| \leq \Big(1 + e^{2g(1+W^*/2)} + g e^{g(1+W^*/2)} n\Big) e^{-g(T_n - T_{P1})},$$

for all $1 \leq i \leq N$. It is easy to show that for $n \geq 2$:

$$1 + e^{2g(1+W^*/2)} + g e^{g(1+W^*/2)} n \leq e^{2g(1+W^*/2)} n,$$

and the result (equation (30)) follows for $n \geq m+1$. For $1 \leq n \leq m$, the result is trivial because the right side of (30) is larger than 1.

# C. BOUNDS ON $\alpha^*$

Theorem 2 shows that $\alpha_i(T_n)$ converge to the constant $\alpha^*$ which solves (23). Here, we provide bounds on the value of $\alpha^*$. Recall that $W^* = (Cd + K)/N$.

**Theorem** 3. *If* $W^* \geq 2$*, then* $\alpha^*$ *satisfies:*

$$\frac{1}{2}\sqrt{\frac{2}{W^*}} - g < \alpha^* < \sqrt{\frac{2}{W^*}} + g. \quad (46)$$

Theorem 3 implies that for very small $g$, $\alpha^* \approx c/\sqrt{W^*}$ for a constant $1/\sqrt{2} < c < \sqrt{2}$. As $g$ increases, the $\alpha_i(t)$ fluctuate more widely resulting in weaker bounds on $\alpha^*$. To stop the fluctuations from becoming large compared to the "ideal" value, $c/\sqrt{W^*}$, an upper limit is required for $g$. This provides another justification for choosing $g \lessapprox 1/\sqrt{Cd + K}$, suggested using the fluid model in Section 3.3.

PROOF. Assume that system starts with initial condition $W_i(0) = W^*$ and $\alpha_i(0) = \alpha^*$. Its easy to see that starting from this initial condition, the marking process $p(t)$ is a periodic square wave with period $\Delta T = 1 + W^*\alpha^*/2$ and duty cycle $1/\Delta T$. Also, $W_i(t)$ and $\alpha_i(t)$ do not depend on $i$, and so dropping the subscript, the common $\alpha(t)$ process evolves according to:

$$\frac{d\alpha}{dt} = g(p(t) - \alpha(t)). \quad (47)$$

The low pass filter described by (47), has frequency response $H(s) = g/(s+g)$ and impulse response $h(t) = ge^{-gt}$. Let $\alpha_{dc}$ denote the DC value of the $\alpha(t)$ process. Since the DC value of $p(t)$ is equal to its duty cycle, $1/\Delta T$, $\alpha_{dc} = H(0)/\Delta T = 1/\Delta T$. Furthermore,

$$\alpha(t) - \alpha_{dc}$$
$$= (p(t) - \frac{1}{\Delta T}) * (ge^{-gt})$$
$$= \int_0^{min\{1,t\}} (1 - \frac{1}{\Delta T}) g e^{-g(t-\tau)} d\tau + \int_{min\{1,t\}}^{t} (-\frac{1}{\Delta T}) g e^{-g(t-\tau)} d\tau,$$

where $*$ is the convolution operator. Using this, we have the following bounds:

$$\alpha(t) - \alpha_{dc} < g \int_0^1 (1 - \frac{1}{\Delta T}) d\tau = g(1 - \frac{1}{\Delta T}),$$
$$\alpha(t) - \alpha_{dc} > g \int_1^{\Delta T} (-\frac{1}{\Delta T}) d\tau = -g(1 - \frac{1}{\Delta T}).$$

Substituting $\alpha_{dc} = 1/\Delta T$ in the above bounds, and taking t to be a congestion instant (so $\alpha(t) = \alpha^*$) gives:

$$-g + \frac{1}{\Delta T} \leq \alpha^* \leq g + \frac{1}{\Delta T}.$$

Finally, substituting $\Delta T = 1 + W^*\alpha^*/2$, and using $W^* \geq 2$, the result follows after some basic algebra. □