2013 IEEE 33rd International Conference on Distributed Computing Systems

Ease the Queue Oscillation: Analysis and Enhancement of DCTCP

Wen Chen, Peng Cheng, Fengyuan Ren, Ran Shu, Chuang Lin
Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
Email: {chenwen, chengpeng5555, renfy, shuran, clin}@csnet1.cs.tsinghua.edu.cn

***Abstract*—Because of the terrible performance of TCP protocol in data center environment, DCTCP has been proposed as a TCP replacement, which uses a simple marking mechanism at switches and a few amendments at end hosts to adjust congestion window based on the extent of the congestion in networks. Thus, DCTCP can make a proper tradeoff between high throughput and low latency. However, through our observation, we discover that DCTCP causes severe oscillation of queue under some parameters and network configuration. Our perceptual analysis concludes that the rough single-threshold marking mechanism may be the essential reason. Therefore, we propose Double-Threshold DCTCP as an improvement of DCTCP. Then, by applying describing function method in nonlinear control theory, we analyze the stability of both DCTCP and Double-Threshold DCTCP, and theoretically explain why Double-Threshold DCTCP is more stable than DCTCP. At last, we validate theoretical analysis and conclude that the Double-Threshold DCTCP can achieve smaller queue, and the queue length of Double-Threshold DCTCP is less sensitive to the growing number of flows. Further, Double-Threshold DCTCP can postpone the throughput collapse caused by Incast traffic and reduce the tail latency in completion time experiment.**

***Keywords*-DCTCP; Double-Threshold DCTCP; DF approach; stability criterion.**

I. Introduction

As a prosperous industry in recent years, data center has been hosting diverse services which have different requirements of network performance in the same network. Some soft real-time online services, such as advertisement, retail and web search [16], require low predictable latency, while other large-scale, parallel-computing services, such as Mapreduce [6], Dryad [11] and Spark [17], require high sustained throughput. However, Transmission Control Protocol (TCP), the dominant congestion control mechanism in conventional Internet, does not fully satisfy all the service desires of network. Fast queue accumulation and emptying, which is aroused by coarse TCP window control scheme and delayed implicit congestion notification, lead to long delays, violent fluctuations in latency, and a large number of timeouts.

To address this problem, Data Center TCP (DCTCP) provides a more detailed window control scheme. Depending on the extent of the congestion informed by Explicit Congestion Notification (ECN) [13], DCTCP adjusts its send window properly to achieve high throughput while maintaining low buffer. It only needs a single parameter at the switch, and a few amends at the end hosts compared to TCP. Because of its simplicity both in algorithm and deployment, DCTCP has its dominant influence and receives the favor of many following protocols. In academic circles, D^2TCP [15] is proposed as a novel protocol, which builds on the top of DCTCP to achieve deadline-aware goal. Data Center Congestion Control [14] is another improvement of DCTCP, which uses a less compute intensive modification to ECN. In industry area, Microsoft introduces DCTCP in Windows Server 2012 [1].

Although DCTCP provides significant performance improvements both in throughput and latency, there is still a small defect of DCTCP algorithm in maintaining the queue length. As shown in Section III, through our observation from the simulation of DCTCP, with the growing number of flows, the bottleneck queue gradually oscillates with increasing amplitude, which strays away from its initial designing control objective. Through further analysis, we find that the single-threshold may result in the phenomenon of too late to inform increasing and decreasing congestion window, thus causing oscillation in DCTCP. In data center networks, because the burst of flows concurrently arrive [16], [3], [10], the queue length will increase rapidly in a short time, and a lot of packets will be marked with ECN. Then the marked packets return back to the senders and all the notified senders will decrease their congestion windows. As a result, after one RTT, the queue length will face a rapid decline. The marking process will continue until the queue length drops back to the settled threshold(e.g. K packets in DCTCP [3]), but the window size decline process will still last for a while because of the delay. We think this conventional single-threshold marking mechanism is not enough to start and end the marking process properly. Thus, we conclude that the nonlinear marking process of single-threshold is the essential reason for oscillation.

Therefore, we propose a new marking mechanism, the Double-Threshold DCTCP(DT-DCTCP). DT-DCTCP uses two parameters K_1 and K_2 to share the load of K, one is to start ECN marking in advance, and the other is to stop in advance. Then we analyze why oscillation happens in theory. Some papers have theoretically analyzed DCTCP system in framework of feedback control theory based on the continuous-time model or(and) discrete-time model(such as [4]). However, these linear control theoretic approaches are not appropriate to analyze the nonlinear component

1063-6927/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDCS.2013.22

IEEE computer society

of marking mechanism. Thus we introduce the conception of Describing Function (DF) approach, which is well-developed in nonlinear control theory. Using DF approach and stability criterion, we analysis the stability of DCTCP and DT-DCTCP, and prove that the latter is more stable than the former.

Next, we validate our analysis through simulations and experiments. The results show that, DT-DCTCP can achieve smaller queue, and the queue length of DT-DCTCP is less sensitive to the growing number of flows. Further, DT-DCTCP can postpone the throughput collapse caused by Incast traffic and reduce the tail latency in completion time experiment.

We make three major contributions in this paper:

i) We observe severe oscillation in DCTCP under some parameters and network configuration, and conclude that the nonlinear marking mechanism is the essential reason for oscillation.

ii) Through perceptual analysis, we propose an improved algorithm of DCTCP in data center network named DT-DCTCP, with only a small change of the marking mechanism, to get an obvious advantage on queue length control.

iii) Instead of conventional linear theoretic approaches, we use DF approach to theoretically analyze the stability of DCTCP and DT-DCTCP, and validate that DT-DCTCP performs better in queue control.

The rest of the paper is organized as follows: Section II presents the background of DCTCP and its fluid model. In section III, we show our observation of oscillation in DCTCP. Then we have a perceptual analysis of the marking mechanism, and propose an improved algorithm called DT-DCTCP. Section IV introduces DF approach in details and applies it to analyze the stability of a nonlinear system. Next in Section V, we linearize the fluid model, use the DF approach and stability criterion to analyze the stability of DCTCP and DT-DCTCP, and prove that DT-DCTCP performs better in stability. In Section VI, we corroborate our results with ns-2 simulator [2] and real experiments, we find that DT-DCTCP can achieve more stable queue length, smaller queue oscillation and lighter network congestion. What's more, DT-DCTCP can postpone the throughput collapse caused by Incast traffic and reduce the tail latency in completion time experiment. Finally, the conclusion is drawn.

II. Background

A. DCTCP

ECN is a well-known mechanism to track congestion at a switch. When congestion happens, according to the settled parameters, the switch marks the packets with the Congestion Encountered(CE) bit in the IP header. Then the sender will decrease its congestion window when it observes the CE bit.

Alizadeh et. al(DCTCP) explain that ECN is not sufficient to give the congestion information. The information should be the extent of the congestion, not only the presence. DCTCP slightly changes the mechanism both at the switch side and the sender side. At the switch side, DCTCP uses one threshold K to determine whether network congestion happens. When a packet arrives at the switch and the buffer occupancy is at least K packets at that moment, the arriving packet will be marked with ECN. At the sender side, the sender aggregates the one-bit ECN feedback from multiple packets to form multi-bit information in one RTT, then a weighted-average metric will be calculated, which will be used to adjust the sender's window size accordingly.

DCTCP can maintain a small queue size without impacting throughput. It performs well in a series of microbenchmarks like Incast, queue buildup and buffer pressure. Moreover, DCTCP is a TCP-friendly protocol, and it requires only 30 lines of code change to TCP and the setting of a single parameter on the switches, which makes it easy to deploy.

B. The fluid model

In [4], a dynamic model of DCTCP is developed using fluid model and nonlinear, delay-differential equation analysis. The fluid model of DCTCP considers the scenario that N flows traverse a single bottleneck with capacity C(in packets/sec). The dynamics of window size $W(t)$(in packets), congestion parameter $\alpha(t)$[1], and the queue size $q(t)$(in packets)at the switch, are described as follows:

$$\frac{dW}{dt} = \frac{1}{R(t)} - \frac{W(t)\alpha(t)}{2R(t)}p(t-R_0) \quad (1)$$

$$\frac{d\alpha}{dt} = \frac{g}{R(t)}(p(t-R_0)-\alpha(t)) \quad (2)$$

$$\frac{dq}{dt} = N\frac{W(t)}{R(t)} - C \quad (3)$$

where $p(t)$ indicates whether packet should be marked or not and is given by $p(t) = \mathbb{I}_{\{q(t)>K\}}$. $R(t)$ is the round-trip time, as a simplification, it is approximated as a fixed value $R_0 = d + K/C$(in secs).

III. Observation and perceptual analysis

We simulate DCTCP to observe its queue length at the switch with the variable number of flows using ns-2 simulator. Figure 1 shows the results of $N = 10$ and $N = 100$ long-lived DCTCP flows sharing a 10Gbps bottleneck, with $100\mu s$ RTT. When the number of flows reaches 100, the oscillation can't be ignored. The amplitude of queue in $N = 100$ is likely 3 or 4 times of that in $N = 10$. We can conclude from the simulation that DCTCP cannot maintain a stable queue length, and oscillates severely as the number of flows increases.

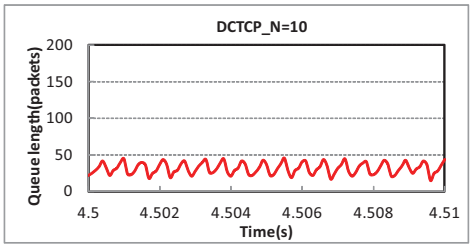
[1] α is the estimation of the fraction of packets that are marked, which is updated once for every window of data(roughly one RTT). It represents the congestion extent of the present network. See in [3].

We may look deeper into the reason of oscillation. DCTCP has one threshold K to determine both the start and the end of ECN marking. When the queue length climbs up to K, the switch marks the packets to inform the senders to decrease their congestion windows, and when the queue length falls back to K, the switch will release the congestion signal. However, one threshold may postpone the start as well as the end of marking. When a burst of flows arrive at the switch, the network becomes congested at once, but the senders will receive the congestion notice at least after one RTT. In this time period, the queue length will increase rapidly. Then the marked packets return back to the senders and all the notified senders will decrease their congestion windows. In the next few RTTs, the queue length will face a rapid decline. The marking process will continue until the queue length drops back to K, but the decline process will still last for a while because of the delay. Continuous increase and decrease in queue length cause the phenomenon of oscillation.

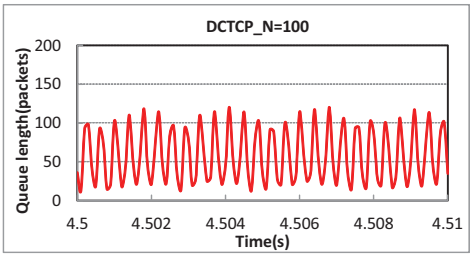
In order to reduce the average queue length, we expect K to be as low as possible, which will avoid Incast as well as queue buildup. In achieving this, we need ECN marking the earlier the better. However, if K is too small that, because of the oscillation, the queue length may drop to the bottom of the buffer, we may lose throughput. In other words, we also have to stop the marking process earlier when the queue length is at a decreasing state. Above all, we need such a mechanism that has one signal to inform congestion earlier to accelerate the marking process, and another signal to release the marking process earlier in order to prevent the queue length from dropping too much.

The original intention of Double-Threshold DCTCP is to improve the single-threshold mechanism of DCTCP. We are inspired from the discussion above that, one threshold K is not enough to control the total marking process. We can split K into two thresholds K_1 and K_2. One is to start ECN marking in advance, and the other is to stop in advance. In other words, the Double-Threshold model is designed to share the load of one threshold K. When the queue length increases beyond the lower threshold K_1, the network is having a potential congestion, and should set CE to inform the senders to decrease their window size. When the queue length decreases under the higher threshold K_2, the switch should release the message of congestion. This approach of "double-threshold" is more flexible than the "single-threshold" mechanism of DCTCP. The difference between these two marking mechanisms can be seen in Figure 2. The switch in DCTCP will mark the packet if the buffer occupancy is equal to or larger than K packets upon its arriving(Figure 2(a)). In DT-DCTCP, the packet will be marked when the queue length increases to K_1, and the marking process continues until the queue length falls back to K_2(Figure 2(b)).

In fact, we regard the single-threshold component as a relay. In control theory, this structure always corresponds to oscillation, which means the controlled objective always fluctuates up and down the settled threshold, and never converges to a certain value. What we do is to replace the relay with a hysteresis loop, which restricts the queue length between the two thresholds, thus the oscillation can be better controlled. So we need a tool to analyze the nonlinear component of both DCTCP and DT-DCTCP, which, however, few in existing analyses has theoretically focused on. In the next section, we will introduce the DF approach [5], which is mature in nonlinear control theory.

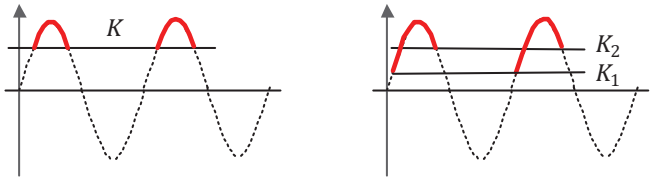


$(a) N = 10$



$(b) N = 100$

Figure 1: The oscillation of the queue at the switch. The DCTCP parameters are set to $K = 40$ packages, and $g = 1/16$.



$(a)\ DCTCP$ $(b)\ DT-DCTCP$

Figure 2: The marking strategies of DCTCP and DT-DCTCP. (a) explains how DCTCP works while (b) explains how DT-DCTCP works.

IV. DF APPROACH AND STABILITY

A. Describing function

DF is an appropriate and powerful tool to analyze the nonlinear component. The system described in Figure 3 is a typical nonlinear control system, it consists of a single, static nonlinear element, and a linear dynamic plant having the transfer function $G(s)$.

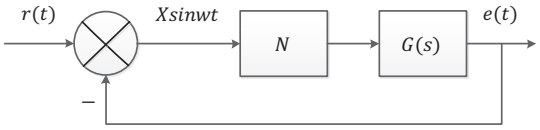


Figure 3: A typical nonlinear control system.

The output of the nonlinearity may be arbitrary with the input $x = Xsinwt$. We can represent it by a Fourier series:

$$\begin{aligned} y(t) &= \frac{A_0}{2} + \sum_{n=1}^{\infty}(A_n cosnwt + B_n sinnwt) \\ &= \frac{A_0}{2} + \sum_{n=1}^{\infty} Y_n sin(nwt + \phi_n) \end{aligned} \quad (4)$$

where

$$\begin{aligned} A_n &= \frac{1}{\pi}\int_0^{2\pi} y(t)cosnwtd(wt) \\ B_n &= \frac{1}{\pi}\int_0^{2\pi} y(t)sinnwtd(wt) \\ Y_n &= \sqrt{A_n^2 + B_n^2}, \phi_n = arctg\frac{A_n}{B_n} \end{aligned}$$

However, consider the low-passed characteristic of the plant, the harmonics of the input beyond the first(i.e. the fundamental) will be filtered out and do not appear in its output. Hence, the fundamental output from the nonlinearity is $A_1 coswt + B_1 sinwt$ approximately, and the DF is given by

$$N = \frac{Y_1}{X}\angle\phi_1 = \frac{\sqrt{A_1^2 + B_1^2}}{X}\angle arctg\frac{A_1}{B_1} = \frac{B_1}{X} + \frac{A_1}{X}j \quad (5)$$

The real and imaginary parts of the DF are both represented by the coefficients of the fundamental terms in the Fourier series expansion, divided by the amplitude X of the input. DF can be used for investigating self-oscillations of nonlinear systems.

B. Stability criterion

In this subsection, we will apply DF approach in stability analysis. If the DF of the nonlinear component is $N(X)$, and the linear dynamic plant $G(jw)$ is stable, the system will oscillate if the product of the DF and the open loop frequency response is equal to minus one.

$$1 + N(X)G(jw) = 0 \quad (6)$$

We call Eq. (6) the characteristic equation. It also can be written as:

$$G(jw) = -\frac{1}{N(X)} \quad (7)$$

As in general, $N(X)$ is a complex function of X, and $G(jw)$ is a complex function of w. The solution of Eq. (6) or Eq. (7) gives both the frequency and amplitude of the oscillation. Thus, the stability of a nonlinear system can be investigated using conventional linear approaches with the nonlinearity replaced by its DF.

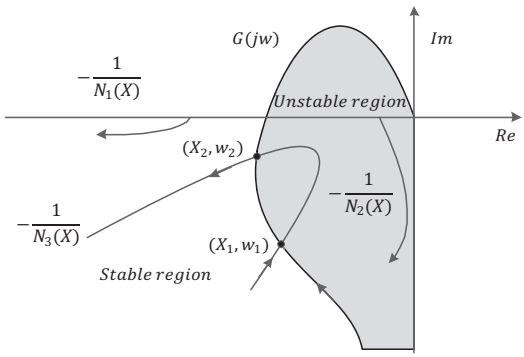


Figure 4: Illustration of stability

As a vivid view of the characteristic equation, we plot the frequency response $G(jw)$ and the negative reciprocal of the DF, $-1/N(X)$ on a Nyquist diagram as shown in Figure 4, any intersection in this figure is a solution of Eq. (6) or Eq. (7).

According to Nyquist stability criterion, the shadow region in Figure 4 is unstable, and the others are stable. The following conclusions are meeting:

i.If $-1/N(X)$ is not surrounded by the linear plant $G(jw)$, the system is stable, such as $-1/N_1(X)$.

ii.If $-1/N(X)$ is surrounded by $G(jw)$, the system is unstable, such as $-1/N_2(X)$.

iii.If $-1/N(X)$ intersects with $G(jw)$, the self-oscillation may exist, and the corresponding amplitude X and frequency w are solutions to the characteristic equation. The oscillation may be stable or unstable. Like $-1/N_3(X)$ shown in Figure 4, there are two intersections (X_1, w_1) and (X_2, w_2), hence two limit cycles are possible. If the system operates at the intersection (X_1, w_1), a slight increase will drive the system into the unstable region, causing a even larger oscillation. On the other hand, if the system operates at (X_2, w_2), a slight increase in gain will drive the system into the stable region. So the limit cycle predicted at (X_1, w_1) is unstable, while (X_2, w_2) is stable.

In practical applications, we usually introduce the conception of relative DF and the negative reciprocal of the relative DF. We can change the DF into the form as follows:

$$N(X) = K_0 N_0(X) \quad (8)$$

K_0 focuses on the characteristic parameter, which is K in DCTCP, and K_2 in DT-DCTCP. We define $N_0(X)$ the relative DF. Thus Eq. (7) can also be written in

$$K_0 G(jw) = -\frac{1}{N_0(X)} \quad (9)$$

We define $-1/N_0(X)$ the negative reciprocal of the relative DF. The details about DF and stability criterion can be found in [8].

V. Analysis

In most of the existed analysis works, the linear control theoretic approaches are frequently used. But for the complex nonlinear network system, they may not be the most appropriate choice. After introducing the DF approach, in this section, we will apply DF approach to analyze the stability of both DCTCP and DT-DCTCP.

We model DCTCP as a close-loop system, the nonlinear component N in Figure 3 can be expressed by the DF of the nonlinear marking mechanism. However, in order to get the linear dynamic plant $G(jw)$, we should linearize the equations of the fluid model first of all.

A. Linearization of fluid model

Because the coupled nonlinear differential Eq. (1), (2) and (3) prevent us from analyzing deeper into the algorithm, we now approximate the dynamics by their small-signal linearization about an operating point. Similar linearization of the fluid model of TCP has been studied in [9]. Taking p as the input and (α, W, q) as the state, we define the operating point (α_0, W_0, q_0, p_0) as the solution of $\dot{\alpha} = 0$, $\dot{W} = 0$ and $\dot{q} = 0$, so we obtain $p_0 = \alpha_0 = \sqrt{\frac{2}{W_0}}$, $W_0 = \frac{R_0 C}{N}$. Then we can represent α, W, q, p as follows:

$$\begin{aligned} \delta\alpha \doteq \alpha - \alpha_0 \quad &, \quad \delta W \doteq W - W_0 \\ \delta q \doteq q - q_0 \quad &, \quad \delta p \doteq p - p_0 \end{aligned}$$

We linearize Eq. (1), (2) and (3) at the operation point:

$$\dot{\delta\alpha(t)} = -\frac{g}{R_0}\delta\alpha(t) + \frac{g}{R_0}\delta p(t - R_0) \tag{10}$$

$$\dot{\delta W(t)} = -\frac{N}{R_0^2 C}\delta W(t) - \sqrt{\frac{C}{2NR_0}}\delta\alpha(t) - \sqrt{\frac{C}{2NR_0}}\delta p(t - R_0) \tag{11}$$

$$\dot{\delta q(t)} = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \tag{12}$$

Performing a Laplace transform on the linearized Eq. (10), (11) and (12), the process can be illustrated in Figure 5.

The feedback control depiction of any system based on AQM [7] can be modeled as a close-loop system like DCTCP. The action of the "Control Law" is to mark packets according to the measured queue length q, and the detailed action is based on different AQM schemes. In DCTCP, it has its own adjustments at end hosts, which will impact α, W and q as shown in blocks $P_\alpha(s)$, $P_{dctcp}(s)$ and $P_{queue}(s)$. $P_\alpha(s)$, $P_{dctcp}(s)$ and $P_{queue}(s)$ are expressed in terms of

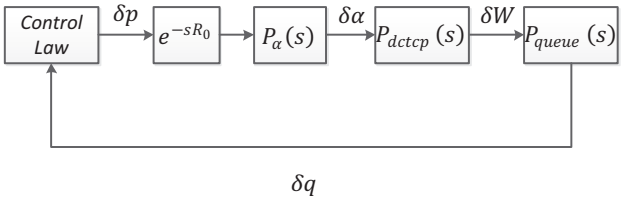


Figure 5: DCTCP as feedback control. The control law will impact the final queue, which in reverse will change the marking process according to the control law.

network parameters as follows:

$$P_\alpha(s) = \frac{\frac{g}{R_0}}{s + \frac{g}{R_0}} \tag{13}$$

$$P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \tag{14}$$

$$P_{dctcp}(s) = -\sqrt{\frac{C}{2NR_0}}\frac{1 + \frac{s + \frac{g}{R_0}}{\frac{g}{R_0}}}{s + \frac{N}{R_0^2 C}} \tag{15}$$

The plant transfer function $P(s)$ is:

$$P(s) = -P_\alpha(s)P_{dctcp}(s)P_{queue}(s) \tag{16}$$

The minus sign implies a negative feedback which relates to how the marking mechanism will affect the queue length.

Then substituting Eq. (13), (14) and (15) to Eq. (16), we get

$$\begin{aligned} P(s) &= -P_\alpha(s)P_{dctcp}(s)P_{queue}(s) \\ &= \frac{\sqrt{\frac{C}{2NR_0}}(\frac{2g}{R_0} + s)\frac{N}{R_0}}{(s + \frac{g}{R_0})(s + \frac{N}{R_0^2 C})(s + \frac{1}{R_0})} \end{aligned} \tag{17}$$

We represent the whole process except the "Control Law" in Figure 5 as $G(jw)$. $G(jw)$ is a function of network parameters capacity C(in packets/sec), number of flows N, fixed RTT time R_0(in secs) and weight g[2].

$$\begin{aligned} G(jw) &= P(s)e^{-jwR_0} \\ &= -\frac{\sqrt{\frac{C}{2NR_0}}(\frac{2g}{R_0} + jw)\frac{N}{R_0}e^{-jwR_0}}{(jw + \frac{g}{R_0})(jw + \frac{N}{R_0^2 C})(jw + \frac{1}{R_0})} \end{aligned} \tag{18}$$

We describe the plant transfer function $G(jw)$ as a linear process. However, the marking process of DCTCP at the switch, is nonlinear. In control theory, this nonlinear component may cause oscillation, which means the queue length may always oscillate and never convergence to a fixed value.

[2] g∈(0,1) is a fixed parameter, it is the weight given to new samples against the past. See in [3].

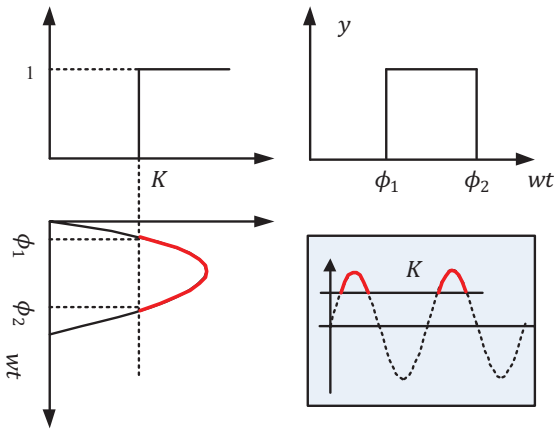


Figure 6: DF of DCTCP. The queue length oscillates above or under the marking line K as shown in the lower right corner. DCTCP marks the packets as long as the queue length is beyond K when they arrive(as the red line shows). To get its DF, we assume the queue behaves as sine, while the marking angel begins at ϕ_1 and ends at ϕ_2.

B. Stability analysis of DCTCP

According to the previous discussion, the queue in D-CTCP will oscillate under certain conditions.

Theorem 1. *If $K_{0dc}G(jw) > max(-\frac{1}{N_{0dc}(X)})$, the queue is stable. Otherwise if $X \geq K$, the queue in DCTCP may be influenced by the oscillation with amplitude X and frequency w which satisfy the equation*

$$K_{0dc}G(jw) = -\frac{1}{N_{0dc}(X)} \tag{19}$$

where $K_{0dc} = \frac{1}{K}$, $G(jw) = \frac{\sqrt{\frac{C}{2NR_0}}(\frac{2g}{R_0}+jw)\frac{N}{R_0}e^{-jwR_0}}{(jw+\frac{g}{R_0})(jw+\frac{N}{R_0^2C})(jw+\frac{1}{R_0})}$ *and* $N_{0dc}(X) = \frac{2}{\pi}(\frac{K}{X})\sqrt{1-(\frac{K}{X})^2}$.

Proof: Firstly, we try to get the DF of the nonlinear component of DCTCP(the "Control Law" in Figure 5). According to the definition of the DF and together with Figure 6, when $X \geq K$, we have:

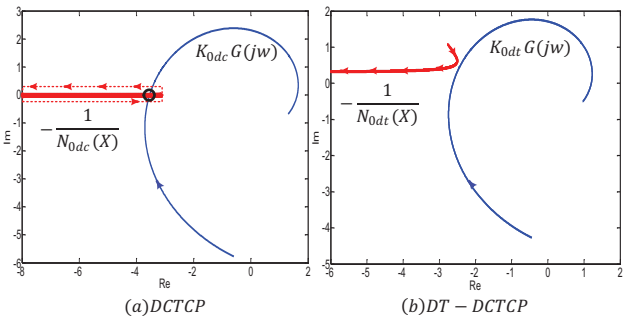
$$\phi_1 = arcsin\frac{K}{X}, \phi_2 = \pi - arcsin\frac{K}{X}$$

$$A_1 = \frac{1}{\pi}\int_{\phi_1}^{\phi_2} coswtd(wt) = 0 \tag{20}$$

$$B_1 = \frac{1}{\pi}\int_{\phi_1}^{\phi_2} sinwtd(wt) = \frac{2}{\pi}\sqrt{1-(\frac{K}{X})^2} \tag{21}$$

Substituting Eq. (20) and (21) into Eq. (5), we get the DF of DCTCP:

$$N_{dc}(X) = \frac{2}{\pi X}\sqrt{1-(\frac{K}{X})^2} \tag{22}$$



$(a) DCTCP$ $(b) DT-DCTCP$

Figure 7: Nyquist diagrams of DCTCP and DT-DCTCP

Then we substitute Eq. (22) together with $K_{0dc} = \frac{1}{K}$ into Eq. (8), we get the relative DF of DCTCP:

$$N_{0dc}(X) = \frac{2}{\pi}(\frac{K}{X})\sqrt{1-(\frac{K}{X})^2} \tag{23}$$

Plot the frequency response locus $K_{0dc}G(jw)$ and the negative reciprocal of the relative DF $-1/N_{0dc}(X)$ of D-CTCP in Figure 7(a). $-1/N_{0dc}(X)$ lies only in the real axis, and it is a convex function of amplitude X, which means when X increases to a certain value, $-1/N_{0dc}(X)$ will reach its maximum. If $K_{0dc}G(jw) > max(-\frac{1}{N_{0dc}(X)})$, the $-1/N_{0dc}(X)$ locus must not be surrounded by the $K_{0dc}G(jw)$ locus, the system will be stable according to the stability criterion.

Since the intersection represents the occurrence of oscillation, we may conclude that, if equation $K_{0dc}G(jw) = -\frac{1}{N_{0dc}(X)}$ has solutions, the queue may be influenced by oscillation. We substitute Eq. (18) and (23) into Eq. (9), thus the oscillation with amplitude X and frequency w will satisfy Eq. (19). Because $-1/N_{0dc}(X)$ is a convex function of amplitude X, for some values of N, there are actually two intersections of $K_{0dc}G(jw)$ and $-1/N_{0dc}(X)$. According to our analysis in stability criterion, the intersection which goes into $K_{0dc}G(jw)$ locus predicts an unstable limit cycle, while the other which goes out of $K_{0dc}G(jw)$ predicts a stable one. ■

C. Stability analysis of DT-DCTCP

For DT-DCTCP, the queue will also oscillate under certain conditions.

Theorem 2. *If $K_{0dt}G(jw) > max(-\frac{1}{N_{0dt}(X)})$, the queue is stable. Otherwise if $X \geq K_2$, the queue in DT-DCTCP may be influenced by the oscillation with amplitude X and frequency w which satisfy the equation*

$$K_{0dt}G(jw) = -\frac{1}{N_{0dt}(X)} \tag{24}$$

where $K_{0dt} = \frac{1}{K_2}$, $G(jw) = \frac{\sqrt{\frac{C}{2NR_0}}(\frac{2g}{R_0}+jw)\frac{N}{R_0}e^{-jwR_0}}{(jw+\frac{g}{R_0})(jw+\frac{N}{R_0^2C})(jw+\frac{1}{R_0})}$ *and* $N_{0dt}(X) = \frac{1}{\pi}(\frac{K_2}{X})[\sqrt{1-(\frac{K_1}{X})^2} + \sqrt{1-(\frac{K_2}{X})^2}] +$

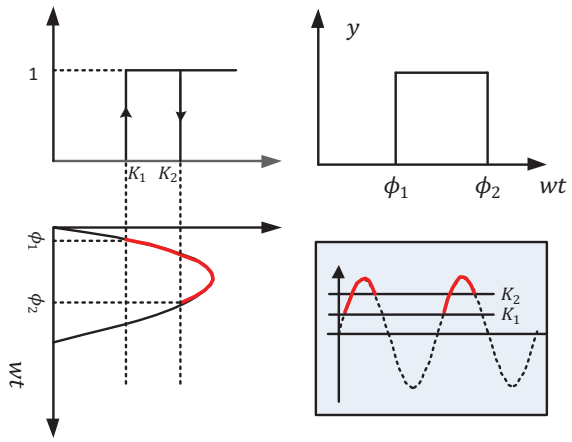


Figure 8: DF of DT-DCTCP. The marking process continues since the queue length increases to K_1 till it falls back to K_2(as the red line shows). The marking angel begins at ϕ_1 and ends at ϕ_2.

$j\frac{K_2^2}{\pi X^2}(1-\frac{K_1}{K_2})$.

Proof: For DT-DCTCP, we also calculate its DF(Figure 8) as what we do in DCTCP. When $X \geq K_2$, we have:

$$\phi_1 = arcsin\frac{K_1}{X}, \phi_2 = \pi - arcsin\frac{K_2}{X}$$

$$\begin{aligned} A_1 &= \frac{1}{\pi}\int_{\phi_1}^{\phi_2} coswtd(wt) \\ &= \frac{1}{\pi X}(K_2 - K_1) \qquad (25) \\ B_1 &= \frac{1}{\pi}\int_{\phi_1}^{\phi_2} sinwtd(wt) \\ &= \frac{1}{\pi}(\sqrt{1-(\frac{K_1}{X})^2} + \sqrt{1-(\frac{K_2}{X})^2}) \qquad (26) \end{aligned}$$

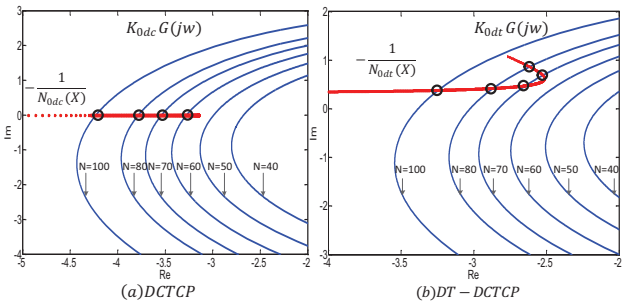
Substituting Eq. (25) and (26) into Eq. (5), we get the DF of DT-DCTCP:

$$\begin{aligned} N_{dt}(X) &= \frac{1}{\pi X}(\sqrt{1-(\frac{K_1}{X})^2} + \sqrt{1-(\frac{K_2}{X})^2}) \\ &\quad + j\frac{1}{\pi X^2}(K_2 - K_1) \qquad (27) \end{aligned}$$

Then we substitute Eq. (27) together with $K_{0dt} = \frac{1}{K_2}$ into Eq. (8), we get the relative DF of DCTCP:

$$\begin{aligned} N_{0dt}(X) &= \frac{1}{\pi}(\frac{K_2}{X})[\sqrt{1-(\frac{K_1}{X})^2} + \sqrt{1-(\frac{K_2}{X})^2}] \\ &\quad + j\frac{K_2^2}{\pi X^2}(1-\frac{K_1}{K_2}) \qquad (28) \end{aligned}$$

The analysis is the same with DT-DCTCP as in DCTCP. Plot the frequency response locus $K_{0dt}G(jw)$ and the negative reciprocal of the relative DF $-1/N_{0dt}(X)$ of DT-DCTCP in Figure 7(b). $-1/N_{0dt}(X)$ has real parts and imaginary parts and it is a convex function of amplitude X. If $K_{0dt}G(jw) > max(-\frac{1}{N_{0dt}(X)})$, the $-1/N_{0dt}(X)$ locus must not be surrounded by the $K_{0dt}G(jw)$ locus, the system will be stable.



$(a)DCTCP$ $(b)DT-DCTCP$

Figure 9: An example of Nyquist diagram. The x-axis is real axis, and y-axis is imaginary axis. $-1/N_{0dc}(X)$ and $K_{0dc}G(jw)$ intersects when N reaches 60. $-1/N_{0dt}(X)$ and $K_{0dt}G(jw)$ intersects when N reaches 70.

If equation $K_{0dc}G(jw) = -\frac{1}{N_{0dt}(X)}$ has solutions, which represent intersections, then the queue may be influenced by oscillation. We substitute Eq. (18) and Eq. (28) into Eq. (9), thus the oscillation with amplitude X and frequency w which will satisfy Eq. (24). ■

D. DT-DCTCP is more stable than DCTCP

We will substitute some certain parameters of DCTCP and DT-DCTCP into the Nyquist diagram(Eq. (19) of DCTCP and Eq. (24) of DT-DCTCP), to compare their stability deeply. For DCTCP, the configuration parameters are set as follows: $R = 0.0001$s, $C = 10$Gbps, $K = 40$packets, $g = 1/16$. As shown in Figure 9, $K_0G(jw)$ shifts to the left as the number of flows N increases. It means with more flows, the oscillation will more easily happen. For DCTCP in Figure 9(a), the $-1/N_{0dc}$ locus is not surrounded by the $K_{0dc}G(jw)$ locus before N reaches 60. In other words, depend on Theorem 1, after N exceeds 50, max(-$\frac{1}{N_{0dc}}$) is bigger than $K_{0dc}G(jw)$, which means DCTCP may occur oscillation.

In order to compare with DCTCP, we set configuration parameters in DT-DCTCP as $K_1 = 30$ packets, $K_2 = 50$ packets, which maintains an average of 40 packets, and keep other parameters unchangeable. The results are shown in Figure 9(b). The red dot matrix represents $-1/N_{0dt}(X)$, it intersects with $N = 70$ where $K_{0dt}G(jw)$ locates. In fact, if we plot $-1/N_{0dc}(X)$ and $-1/N_{0dt}(X)$ in the same Nyquist diagram, we will find that the max values in real axis of them are almost the same. So we can conclude that, besides the maximum values of $-1/N_{0dc}(X)$ and $-1/N_{0dt}(X)$, the shapes and locations of $-1/N_{0dc}(X)$, $-1/N_{0dt}(X)$ and $K_{0dt}G(jw)$ determine the intersections. Because $-1/N_{0dt}(X)$ has the positive imaginary part, which makes it far from $K_{0dt}G(jw)$ locus with smaller N, the

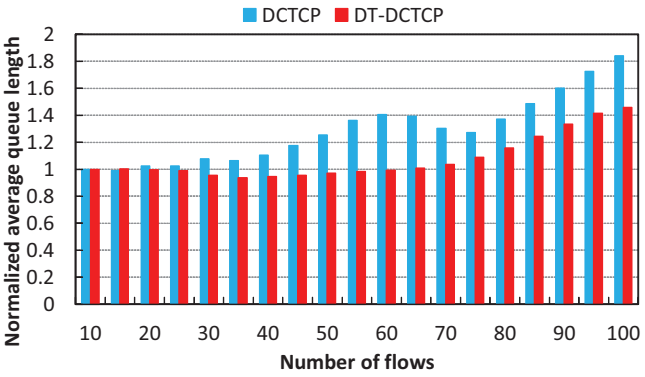


Figure 10: The average queue lengths of DCTCP and DT-DCTCP. Because the average queue lengths of DCTCP and DT-DCTCP are different, in order to compare fairly, we set the queue length at $N = 10$ as the baseline, and take the other queue length values the proportion to that of $N = 10$.

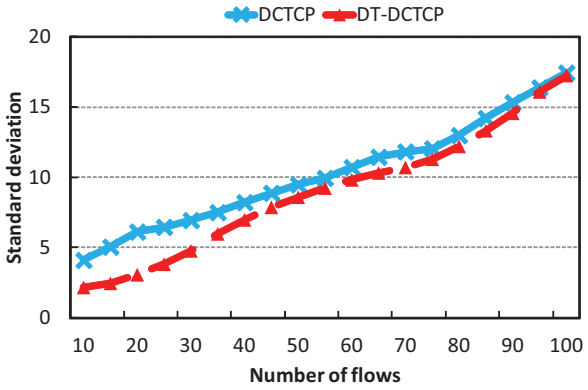


Figure 11: The standard deviation of DCTCP and DT-DCTCP.

intersection would occur with lower probability.

In conclusion, $-1/N_{dc}(X)$ is earlier to intersect with the $G(jw)$ locus, and $-1/N_{dt}(X)$ is less likely to have intersections. The earlier the intersections occur, the more unstable the system is. Therefore, DT-DCTCP algorithm is more stable.

VI. Evaluation

This section is divided into two parts. We first examine the stability of DCTCP and DT-DCTCP using ns-2 simulator and compare some basic properties including queue length, queue oscillation and α described in Section II. Next, we reproduce some important experiments to validate how DT-DCTCP ameliorates the specific performance of DCTCP, including Incast experiment in [12] and completion time experiment in [3].

A. Simulation

We verify our Theorem 1 and Theorem 2 via simulations using ns-2 simulator. N flows share a single 10Gbps bottleneck link. All the flows have a fixed RTT of $100\mu s$. The parameters are $K = 40$ packets and $g = 1/16$ for DCTCP, and $K_1 = 30$ packets, $K_2 = 50$ packets and $g = 1/16$ for DT-DCTCP.

1) Self-oscillation in queue: We assume a scenario that N servers send messages to one client at the same time. We do the simulation from $N = 10$ to $N = 100$ with an increase of every 5 flows. Figure 10 shows the average queue length of DCTCP and DT-DCTCP. We set the baseline of DCTCP as the average queue length of 32 packets when $N = 10$, and the baseline of DT-DCTCP as the average queue length of 42 packets when $N = 10$. For DCTCP, the queue length stays stable from $N = 10$ to $N = 30$. However, at around $N = 35$, the queue length of DCTCP strays away. The queue length floats from 1.10 times to 1.83 times of the baseline, and reaches a local maximum value when $N = 60$. This phenomenon is not the same with the analysis of the stability of DCTCP in Section V, that the queue length will maintain stable until $N = 60$. We analyze that the nonlinear marking mechanism is not the only reason for the average queue length deviation for DCTCP. The single-threshold is too weak to control the queue length within a region, even if the number of flows is small, the average queue length may float above its initial design. In comparison, DT-DCTCP maintains a stable queue length before $N = 70$. The fluctuation is so obscure that the queue length changes only from 0.94 times to 1.01 times of the baseline before $N = 70$. From $N = 70$ to $N = 100$, the average of queue length of DT-DCTCP strays away from its baseline because of the oscillation, which is correspondent to our analysis of DT-DCTCP in Section V.

Figure 11 gives us a vivid description of queue oscillations of both DCTCP and DT-DCTCP. The standard deviations of DCTCP and DT-DCTCP both increase as the number of flows increases, which indicates that the oscillation becomes heavier. In comparison, at each number of flows, the standard deviation of DCTCP is larger than that of DT-DCTCP. Thus we conclude that, the oscillation of queue in DT-DCTCP is lighter than that of DCTCP.

Therefore, we can prove that DT-DCTCP maintains a more horizontal queue length, and smaller standard deviation than those of DCTCP.

2) Performance of α: α is an important parameter to represent the extent of congestion in the network. We set the scenario that N flows inject into the single bottleneck at the same time. We take 5000 samples in 5 minutes, and compare the average of α in both DCTCP and DT-DCTCP in Figure 12.

Essentially, α close to 0 indicates low level of congestion, and α close to 1 indicates high level. We want the value of α the lower the better. In Figure 12, both the values of α in DCTCP and DT-DCTCP increase as the number of

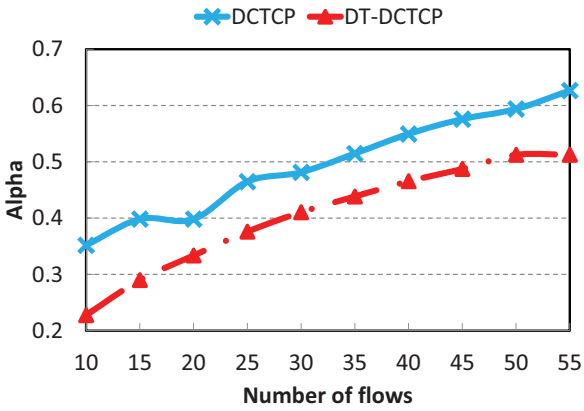


Figure 12: Parameter α of DCTCP and DT-DCTCP. The blue part represents the proportion that α_{dc} is larger than α_{dt}, red part represents equal, and green part represents smaller.

flows increases, which means the network becomes more congestive. However, the line of α of DT-DCTCP is always lower than that of DCTCP. α of DCTCP is always bigger than that of DT-DCTCP about 0.1, and as the number of flows increases, α of DT-DCTCP has a smooth increasing tendency, while α of DCTCP increases strictly and has a potential to increase higher. So we conclude that the network in DT-DCTCP is less congestive.

Overall, DT-DCTCP performs a better control of the queue length, queue oscillation and network congestion.

B. Experiment

In this section, we carry out some experiments, compare the results between DCTCP and DT-DCTCP, and show how DT-DCTCP improves the performance in real scenarios.

The basic topology of our testbed is made up of four switches and ten end hosts as shown in Figure 13. All switches in our experiments are NetFPGA cards with four 1Gbps Ethernet ports. Switch 2, Switch 3 and Switch 4 are all connected to Switch 1, and each of them connects to three hosts with Intel Celeron Dual-Core 2930MHz CPU, 4GB RAM and 1Gbps NIC , while Switch 1 only connects to one host as the aggregator. The implementations of DCTCP and DT-DCTCP are downloaded to NeFPGA cards in Switch 1, and the buffer sizes of each port of Switch 1 is set 128KB. Switch 2, Switch 3 and Switch 4 are implemented with Drop-Tail mechanism with buffer sizes 512KB, which makes the bottleneck centralized on the link between Switch 1 and the client. All hosts in our testbed are running CentOS5.5 with Linux kernel 2.6.38 which the patches of DCTCP and DT-DCTCP are applied in. The propagation RTT without queuing delay is approximately $100\mu s$ between two end hosts connected to the same switch. In our experiments, each packet is about 1.5KB. We set the parameter $K = 32$KB for DCTCP , and two groups of parameters for DT-DCTCP: $K_1 = 34$KB, $K_2 = 28$KB and $K_1 = 34$KB, $K_2 = 30$KB.

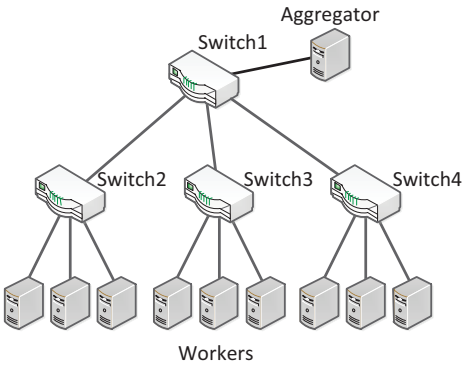


Figure 13: Testbed topology.

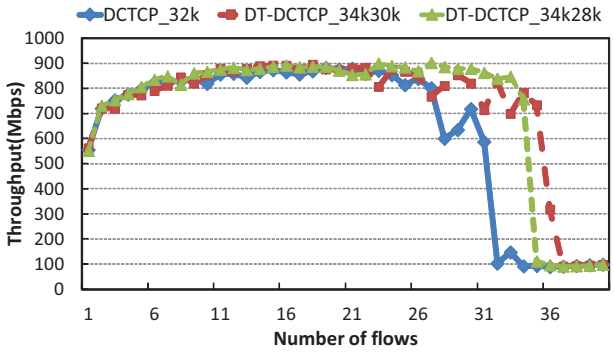


Figure 14: Experiment results of Incast impairment.

1) Incast impairment: In this section, we examine the Incast impairment. The aggregator generates one query from each worker, and each of them responses 64KB data and sends them simultaneously to the aggregator. This experiment has been repeated for 100 times.

Figure 14 shows the experiment results of both DCTCP and DT-DCTCP. When the number of flows increases to 32, DCTCP endures an obvious throughput collapse, while DT-DCTCP maintains a relative high value of throughput until 37 synchronized flows. Thus DT-DCTCP postpones the throughput collapse of 5 flows.

In DCTCP, because of the severe queue oscillation, when the queue length is near the buffer size, some packets will be dropped. So the aggregator should wait until timeout happens. However, in DT-DCTCP, we control the queue oscillation smaller than that of DCTCP, so the throughput collapse is postponed. Therefore, the benefit of smaller oscillation in DT-DCTCP improves its performance in Incast.

2) Oscillate completion time impairment: The aggregator requests 1MB from n different workers, and each worker responds with the requested 1MB/n data. The aggregator waits until it receives all the responses, so the overall completion time matters. We repeat the experiment for 100 times.

Figure 15 shows the average query completion time. The minimum completion time is around 10ms. Because the in-

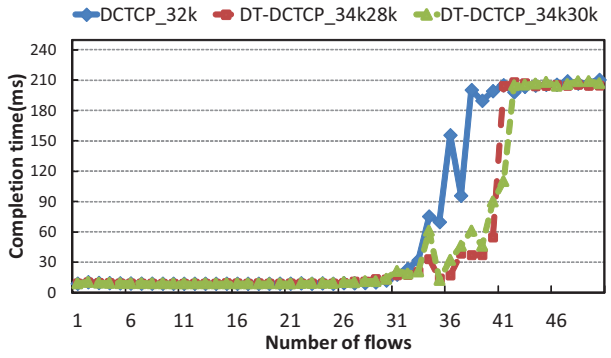


Figure 15: Experiment results of the oscillating completion time impairment.

coming link to the client is the bottleneck, so it takes around 10ms to deliver 1MB data. With the number of synchronized flows increases, the completion time of both DCTCP and DT-DCTCP may have a sudden burst of 20 times higher, where Incast happens. Incast of DCTCP happens when the number of flows reaches 40, however, from 34 synchronized flows to 40 synchronized flows, the completion time of DCTCP has severe oscillation. In comparison, DT-DCTCP does not suffer timeout until the number of flows increases to 42, and the completion time climbs smoothly from 35 synchronized flows to 42 synchronized flows.

In our experiment, the longest transmission time of the flows determines the completion time. In DCTCP, because of the queue oscillation, the longest queueing delay changes severely, which makes DCTCP suffered from the oscillating completion time. DT-DCTCP eases the queue oscillation, so the longest queueing delay does not change much. Thus, we conclude that DT-DCTCP solves the problem of oscillating completion time of DCTCP.

VII. Conclusion

In data center networks, DCTCP is an effective protocol to maintain low latency and high burst tolerance for short flows, as well as high utilization for long flows. The simplicity of operating DCTCP is very attractive, but the queue oscillation degrades the performance. In this paper, we put forward a novel algorithm DT-DCTCP to amend the shortcomings of DCTCP. We use the DF approach to analyze the stability of both DCTCP and DT-DCTCP algorithms. We prove that the oscillation is caused by the nonlinear control scheme at the switches, and conclude that DT-DCTCP can achieve better stability in queue than DCTCP. This argument is validated through simulations and experiments.

VIII. Acknowledgement

The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by the National Natural Science Foundation of ChinaNSFC under Grant No. 61225011, National Basic Research Program of China (973 Program) under Grant No. 2012CB315803 and 2009CB320504, and National Science and Technology Major Project of China (NSTMP) under Grant No.2011ZX03002-002-02.

References

[1] http://technet.microsoft.com/en-us/library/hh997028.aspx.

[2] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[3] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. DCTCP: Efficient Packet Transport for the Commoditized Data Center. In *SIGCOMM*, Augest 2010.

[4] M. Alizadeh, A. Javanmard, and B. Prabhakar. Analysis of DCTCP: Stability, Convergence, and Fairness. In *SIGMETRICS*, June 2011.

[5] D. P. Atherton. *Nonlinear Control Engineering*. 1982.

[6] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI*, 2004.

[7] V. Firoiu and M. Borden. A Study of Active Queue Management for Congestion Control. In *INFOCOM*, 2000.

[8] B. Friedland. Advanced Control System Design. In *Prentice Hall. Englewood Cliffs*, 1996.

[9] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong. A Control Theoretic Analysis of RED. In *INFOCOM*, 2001.

[10] C.-Y. Hong, M. Caesar, and P. B. Godfrey. Finishing Flows Quickly with Preemptive Scheduling. In *SIGCOMM*, 2012.

[11] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed Data-parallel Programs from Sequential Building Blocks. In *EuroSys*, March 2007.

[12] D. Nagle, D. Serenyi, and A. Matthews. The Panasas ActiveScale Storage Cluster: Delivering scalable hig bandwidth storage. In *SC*, 2004.

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

[14] R. R. Stewart, M. Tüxen, and G. V. Neville-Neil. An Investigation into Data Center Congestion Control with ECN. In *BSDCan*, 2011.

[15] B. Vamanan, J. Hasan, and T. N. Vijaykumar. Deadline-Aware Datacenter TCP(D^{2}TCP). In *SIGCOMM*, Augest 2012.

[16] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowstrown. Better Never than Late: Meeting Deadlines in Datacenter Networks. In *SIGCOMM*, Augest 2011.

[17] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster Computing with Working Sets. In *HotCloud*, March 2010.