Modeling and Analysis of the Latency-Based Congestion Control Algorithm DX

Wanchun Jiang, Lijuan Peng, Chang Ruan, Jia Wu, Jianxin Wang

Modeling and Analysis of the Latency-based Congestion Control Algorithm DX*

Wanchun Jiang, Lijuan Peng, Chang Ruan, Jia Wu, and Jianxin Wang

School of Computer Science and Engineering, Center South University,
Changsha, Hunan, China, 410083
{jiangwc, ruanchang, jxwang}@csu.edu.cn

Abstract. Nowadays, low latency has become one of the primary goals of congestion control in data center networks. To achieve low latency, many congestion control algorithms have been proposed, wherein DX is the first latency-based one. Specifically, DX tackles the accurate latency measurement problem, reduces the flow completion time and outperforms the *de facto* DCTCP algorithm significantly in term of median queueing delay. Although the advantages of DX have been confirmed by experimental results, the behaviors of DX have not been fully revealed. Accordingly, some drawbacks of DX under special environment are unexplored. Therefore, in this paper, we conduct fluid-flow analysis over DX, deducing sufficient condition for the stability of DX and revealing the behaviors of DX. Analytical results uncover two problems of DX: 1) it has poor throughput when either the base RTT is very large or the number of flows is relatively small; 2) it suffers from large queueing delay when either the base RTT is relatively small or the number of flows is very large. These results are instructive to the improvement and deployment of DX. Simulation results based on NS-3 verify our analytical results.

Keywords: Congestion control· Fluid-flow analysis· Stability· Latency

1 Introduction

Nowadays, low latency becomes one of the primary goals of designing congestion control algorithms for the data center network. To achieve low latency, accurate and fine-grained feedback signals are needed to represent the degree of congestion. Recently, many congestion control algorithms have been proposed [1, 10, 12, 5]. Generally speaking, most of them employ the following feedback signals: the packet loss, the explicit in-network feedback like ECN, and the latency-based feedback. Compared to the other two signals, latency-based feedback signals have the following advantages. The endpoint can detect the fine-grained degree of congestion, or even estimate the switch queue-size [10, 12] by measuring the Round Trip Time (RTT) and the base RTT. Moreover, the in-network support is never required.

* Supported by the Projects of Hunan Province Science and Technology Plan in China under Grant No. 2016JC2009.

However, the latency-based feedback signal is difficult to be measured accurately [1]. This is because most kernel implementations can only track RTTs at the granularity of $1ms$ [9], while the RTT is only a few hundreds of microseconds in the data center network. Recently, DX, the latency-based congestion control algorithm proposed for data center networks, tackles the measurement problem of RTT and has good performance. By setting its operating point close to zero, DX can reduce the flow completion time and outperform the *de facto* DCTCP algorithm significantly in term of median queueing delay.

In this paper, we model and analyze the DX algorithm because 1) DX is the up-to-date latency-based congestion control algorithm while other state-of-art algorithms such as ExpressPass and NDP are not. As a latency-based algorithm, DX has very good performance, which are validated by experimental results in [6]. 2) Although some advantages of DX have been confirmed by experimental results, the behaviors of DX have not been explored theoretically. 3) Moreover, existing analytical work on congestion control cannot be applied to the window-based latency-based DX algorithm. In details, we first model DX with the fluid-flow method and linearize the fluid-flow model such that the Nyquist stability criterion [6] can be applied to the model. Subsequently, we deduce the sufficient condition for the stability of DX. In this way, the influence of some parameters, such as the number of flows and the RTT, on the stability of DX can be exhibited. Moreover, we theoretically uncover a special behavior of DX under the condition of a large number of flows and small RTT. Finally, we implement DX in NS-3 simulator to confirm our analytical results.

In total, our analytical results mainly reveal two problems of DX. 1) DX has poor throughput when the DX system is unstable when either the base RTT is very large or the number of flows is relatively small. 2) DX suffers from large queueing delay when either the base RTT is relatively small or the number of flows is very large. Under these conditions, DX enters into the special stable state. It implies that DX should not be employed under these kinds of environments. We believe these results are instructive to the improvement and deployment of DX in practical data center network.

2 Background and Related work

In this section, we first introduce the DX algorithm in brief, and then present the related work on the theoretical analysis of congestion control algorithms.

2.1 The DX Algorithm

DX is a window-based congestion control algorithm, which uses the latency-based feedback signal to determine the congestion window should be increased or decreased. Similar to TCP, its congestion avoidance algorithm follows the Additive Increase Multiplicative Decrease (AIMD) style. The DX algorithm is characterized by dropping the queue size down to zero quickly as soon as it observes congestion. In the following, we introduce the DX algorithm in detail.

The main DX algorithm is composed of two parts: one is measuring the latency accurately, the other one is a congestion control algorithm for adjusting the congestion window. For accurately measure queueing delay, [10] exhibits sources of measurement errors and their magnitude and their elimination technique.

The congestion control algorithm of DX works as follows. In each RTT, DX measures the queueing delay, which is the difference between the base RTT and a sample RTT. If the queueing delay is not 0, DX considers the network is congested. Otherwise, DX considers that there is no congestion. Mathematically, the window adaption algorithm of DX is as follows:

$$W(t+1) = \begin{cases} W(t)+1, & if \quad Q(t)=0, \\ W(t)(1-\frac{Q(t)}{U(t)}), & if \quad Q(t)>0, \end{cases} \tag{1}$$

where $W(t)$ is the window size at time t, $Q(t)$ represents the average queueing delay measured by DX in current RTT. $U(t)$ is a self-updated coefficient.

$$U(t) = \frac{R_0 \cdot W(t)}{W(t)-1}, \tag{2}$$

where R_0 is the base RTT. The self-updated coefficient $U(t)$ is deduced in consideration of high utilization and the number of flows in the network.

According to equation (1), DX decreases the congestion window as soon as it detects the network congestion according to $Q(t)$. Therefore, DX keeps the near-zero queueing delay.

2.2 Related Work

Although there are many theoretical works on congestion control algorithms, such as those in [11, 7, 13], we focus on those works analyzing the state-of-art congestion control algorithms for data center networks in this paper.

Analysis on Non-Latency-based Algorithms DCTCP[1] is a famous congestion control algorithm using ECN. In [2], M. Alizadeh et al. develop a fluid-flow model of DCTCP and analyze its stability by the Bode Stability Criterion[6]. The analysis insights guide the configurations of design parameters like the threshold. DCQCN is the latest protocol which outperforms DCTCP in terms of reducing the flow completion time. In [14], the authors analyze its stability condition using the same method as DCTCP.

All these algorithms for data center works are based on non-latency congestion signals, while DX adopts the latency-based feedback signal. Therefore, the theoretical analysis of these algorithms cannot be directly applied to DX.

Analysis on Latency-based Algorithms TIMELY [12] is an end-to-end, rate-based congestion control algorithm that uses changes in RTT as a congestion signal. In [14], the author finds that TIMELY has no unique fixed point. To analyze the stability of TIMELY, they modify the algorithm. Its stability condition is analyzed through the Nyquist Stability Criterion [6].

Similar to TIMELY, DX is also a latency-based transport protocol. Different from TIMELY, DX is a window-based algorithm and adjusts the congestion

window according to the queueing delay. In [10], authors show that DX exhibits very good performance by extensive experiments. However, to the best of our knowledge, there is no theoretical work on the window-based latency-based DX up to now, which motivates us to perform this investigation.

3 Analysis of DX

In this section, we first build a fluid-flow model for the DX algorithm and then analyze its stability based on its linearized version.

3.1 Modeling

Considering the oversubscribed link and the applications like MapReduce [4], we assume that the sources are homogeneous and flows arrive according to the Poisson process, the same as [2], [3] and [8]. In other words, we assume that all sources have identical sending rates and RTTs, and the RTT equals to τ seconds.

Suppose that N sources share a single link of capacity C. Let $W(t)$ denote the congestion window, R_0 represent the fixed base RTT, and $Q(t)$ be the queueing delay. Let p denote the probability of $Q(t) > 0$. Although in practice, the probability p is time-varying. We find that p is close to a constant in the stable state, as shown in the simulation results under the condition of varying p in Section 4. Therefore, we assume that p is constant for the simplicity of analysis. With this assumption, we plug the equation (2) into equation (1), and can model the DX algorithm as follows by using the method of [11].

$$\frac{dW(t)}{dt} = \frac{seg*(1-p)}{R_0+Q(t-\tau)} - \frac{Q(t-\tau)(W(t)-seg)}{R_0(R_0+Q(t-\tau))}p, \tag{3}$$

$$\frac{dQ(t)}{dt} = \begin{cases} \frac{NW(t)}{C(R_0+Q(t-\tau))} - 1 & if \quad Q(t) > 0, \\ max\{0, \frac{NW(t)}{C(R_0+Q(t-\tau))} - 1\} & if \quad Q(t) = 0. \end{cases} \tag{4}$$

The equation (3) describes the dynamic evolution of the window size $W(t)$. The equation (4) models the evolution of the queueing delay $Q(t)$.

3.2 Stability Analysis

We analyze the stability of DX based on its fluid-flow model (3) and (4). Assume that the equilibrium point of DX is (W_0, Q_0). At the equilibrium point, we have $\dot{W}(t) = 0$ and $\dot{Q}(t) = 0$. Referring to equation (3) and equation (4), we have

$$seg * R_0(1-p) = pQ_0(W_0 - seg). \tag{5}$$

$$NW_0 = C(R_0 + Q_0). \tag{6}$$

Substituting equation (6) into equation (5), we can get the following expression of Q_0

$$Q_0 = \frac{p(N*seg - CR_0) + \sqrt{\Delta}}{2Cp}, \tag{7}$$

where

$$\Delta = (CpR_0 - pN * seg)^2 + 4CpNR_0 * seg * (1-p). \tag{8}$$

Next, we will linearize the fluid-flow model around the equilibrium point (W_0, Q_0) to obtain

$$\begin{aligned} \delta\dot{W} &= a_1\delta W + a_2\delta Q(t-\tau), \\ \delta\dot{Q} &= b_1\delta W + b_2\delta Q(t-\tau), \end{aligned} \tag{9}$$

where

$$\begin{aligned} \delta W &\doteq W - W_0, \\ \delta Q &\doteq Q - Q_0, \end{aligned} \tag{10}$$

and

$$\begin{aligned} a_1 &= -\tfrac{Q_0 p}{R_0(R_0+Q_0)},\ a_2 = \tfrac{2p*seg - pW_0 - seg}{(Q_0+R_0)^2}, \\ b_1 &= \tfrac{N}{C(R_0+Q_0)}, \qquad b_2 = -\tfrac{NW_0}{C(R_0+Q_0)^2}. \end{aligned} \tag{11}$$

To obtain the characteristic equation, we compute the Laplace transform of (9). Then we can obtain the transfer function of the linear time-delayed system

$$G(s) = e^{-s\tau}\frac{a_1 b_2 - a_2 b_1 - b_2 s}{s(s-a_1)}. \tag{12}$$

Then, we apply the Bode Stability Criteria [6] to the transfer function (12). Specifically, define the frequency characteristic function $G(j\omega) = G(s)|_{s=j\omega}$ of the system, we have

$$G(j\omega) = A(\omega)e^{j\varphi(\omega)}, \tag{13}$$

where

$$|A(\omega)|^2 = \frac{b_2^2[\omega^2 + (a_1 - \frac{a_2 b_1}{b_2})^2]}{\omega^2(\omega^2 + a_1^2)}, \tag{14}$$

$$\varphi(\omega) = -\frac{\pi}{2} - \omega\tau + arctan\frac{\omega}{a_1} + arctan\frac{\omega b_2}{a_1 b_2 - a_2 b_1}, \tag{15}$$

where $A(\omega)$ is amplitude - frequency characteristics and $\varphi(\omega)$ is phase-frequency characteristic. Assume that ω_c is the cross-over frequency which makes $L(\omega_c) = 0$, i.e., $A(\omega_c) = 1$. From equation (14), we have

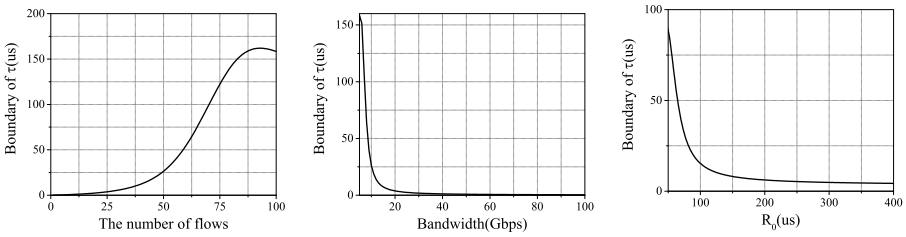
$$\omega_c = \sqrt{\frac{b_2^2 - a_1^2 + \sqrt{(a_1^2 - b_2^2)^2 + 4(a_1 b_2 - a_2 b_1)^2}}{2}}. \tag{16}$$

Note that $\varphi(0) = -\frac{\pi}{2}$. According to Bode Stability Criteria [6], the DX system is stable when $\varphi(\omega_c) > -\pi$, i.e., we have the following theorem in summary.

Theorem 1. *The DX system is stable if the delay satisfies*

$$\tau < \frac{1}{\omega_c}(arctan\frac{\omega_c}{a_1} + arctan\frac{\omega_c b_2}{a_1 b_2 - a_2 b_1} + \frac{\pi}{2}), \tag{17}$$

where ω_c is defined in (16), and a_1, b_1, a_2 and b_2 are defined in (11).



(a) Varying N (b) Varying C (c) Varying R_0

Fig. 1. The variation of the boundary of τ with different N, C, R_0.

Theorem 1 implies that the stability of the DX system holds just when τ is limited. The boundary of τ is associated with both the bottleneck bandwidth C and the number of flows N. In fact, according to equation (17), the boundary of τ decreases when either the bandwidth C increases or the number of flows decreases. In order to verify the result, we assume that the value of p is 0.95, the bandwidth C is 10Gbps, the number of flows is 50, the packet size seg is 1500 and the base RTT R_0 is $80\mu s$ by default. Fig. 1 shows the variation of the boundary of τ with different N, C, R_0 respectively. In Fig. 1(a), when N is small, the boundary of τ is small and accordingly Theorem 1 is probably not satisfied. Consider this condition, we do not know whether the DX system is stable. When DX becomes unstable, it will suffer from large queue-size oscillation and poor link utilization. However, when N is large, Theorem 1 is satisfied, i.e., the DX system is stable. In Fig. 1(b) and Fig. 1(c), when C or R_0 changes, similar results can be obtained according to Theorem 1. This is also why the evaluation of DX in [10] always shows good performance.

In total, Theorem 1 reveals the problem that DX may become unstable and have poor throughput when either the base RTT is very large or the number of flows is relatively small.

3.3 A Special Stable State

When we conduct the stability analysis of the DX algorithm, we do not consider the limitation on the congestion window size. In fact, the window size of the DX cannot be less than a segment in real networks. When there are too many flows, i.e., when $\frac{N*seg}{R_0} > C$, the aggregated sending rate of all flows are always larger than the bandwidth C. As a result, Q would be always greater than 0. Meanwhile, the congestion window of every flow is already at the minimum value 1 and cannot be decreased again. In other words, although the queueing delay is still greater than 0 in this scenario, the window size cannot be adjusted by the congestion control algorithm.

To obtain the stable point in this situation, $W(t)$ is kept invariant and its value is always a segment, which can be plugged into the equation (4). We can

get the new model.

$$\frac{dW(t)}{dt} = 0,$$

$$\frac{dQ(t)}{dt} = \begin{cases} \frac{N*seg}{C(R_0+Q(t-\tau))} - 1 & if \quad Q(t) > 0, \\ max\{0, \frac{N*seg}{C(R_0+Q(t-\tau))} - 1\} & if \quad Q(t) = 0, \end{cases}$$

we can get the fixed point (W^*, Q^*) as follows.

$$W^* = seg,$$

$$Q^* = \frac{N * seg - CR_0}{C}.$$

We find that the system is absolutely stable when $N \geq \frac{CR_0}{seg}$, and this special stable state is different from the stable state under $N < \frac{CR_0}{seg}$. In the stable state, the queueing delay will always drop to zero when $N < \frac{CR_0}{seg}$, so the stable state in this case still has the jitter. But if $N \geq \frac{CR_0}{seg}$, the window size does not change and the queueing delay will increase with the increasing number of flows. We can summarize this phenomenon as the following theorem.

Theorem 2. *When the condition $N \geq \frac{CR_0}{seg}$ is satisfied, the DX system enters a special stable state where*

(1)The system is stable;

(2)The congestion window of every flow is unchanged with size 1;

(3)The link is fully utilized.

Obviously, the queueing delay would increase under this case. In other words, Theorem 2 reveals the problem that DX would suffer from large queueing delay when either the base RTT is relatively small or the number of flows is very large.

4 Evaluation

In this section, we validate our theoretical analysis by NS-3 simulations. First, we evaluate the accuracy of our model by comparing the numerical solution of the model conclusion by Matlab 2014a with NS-3 simulation results. Subsequently, we validate our assumption about the probability p by simulations. Next, we examine the conclusion on the special stable state in Theorem 2. Finally, the theoretical conclusion in Theorem 1 is validated by several experiments with the changing parameter.

We use a many-to-one network topology with $10Gbps$ link capacity in our experiments. The switch buffer is set to be 256 KB. To validate the stability of a system, we use the metric of the link utilization. If a system is stable, the link utilization keeps a high level since the queue length at switch cannot be zero. We also show the queueing delay and queue size in a few experiments.

Note that in all experiments, we do not explore all values exhaustively for a parameter due to practical consideration. Specifically, the concurrent number

Table 1. Probability of decreasing windows

RTT \ N	10	20	30	40	50	60	70	80	90	100
$80\mu s$	0.79599	0.887787	0.934742	0.956067	0.973239	0.982844	0.995288	0.999902	0.999896	0.999891
$200\mu s$	0.782627	0.835583	0.869653	0.898591	0.914323	0.922147	0.933485	0.944501	0.953232	0.960271
$400\mu s$	0.743014	0.827145	0.876389	0.881062	0.889631	0.890585	0.898752	0.901909	0.913419	0.920201

of flows, which occupy the link fully, can not surpass the number of ports of a switch (often less than 96). The commonly deployed maximum bandwidth is not greater than $40Gbps$ in data center networks, and the base RTT is less than 500 μs [1].

4.1 Model Validation

Although we model the DX system in Section 3, how well the model can match the behavior of practical DX is yet unknown. We answer this question by comparing the queue length obtained by the model with that by running with the NS-3 code of DX. Before that, we first check the assumption that the probability of decreasing windows or $Q(t) > 0$, i.e., p, is constant in the stable state.

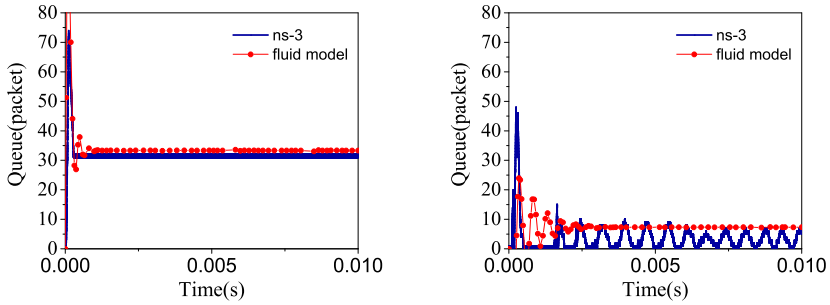
We select the scenario where the system enters a stable state and a special stable state, and test the change of p with N ranging from 10 to 100 when the base RTT (R_0) is $80\mu s$, $200\mu s$, $400\mu s$, as shown in Table 1. According to Theorem 1, we know that when R_0 is $80\mu s$, $200\mu s$, or $400\mu s$, the system stability conditions are $N > 30$, $N > 50$ or $N > 140$, respectively. Meanwhile, if the R_0 is $80\mu s$ and N is greater than 70, the system is in a special stable state. According to our measurement of p, all values of p are greater than 0.9 when the system is stable. When the system enters a special stable state, the value of p is even greater than 0.99. Using the average value 0.95, p represents those values in the two states basically. This is the reason why we set p as a constant.

Next, we examine the accuracy of our whole model. Fig. 2(a) and Fig. 2(b) are respectively the evolution of the queue length under the condition of $N = 50, R_0 = 20\mu s$, where DX is in the special stable state, and $N = 50, R_0 = 80\mu s$, where the behaviors of DX are described by equations (12) and (13). The results of the fluid-flow model are close to the simulation results of NS-3. Therefore, the accuracy of our model for DX is good.

4.2 The Special Stable State

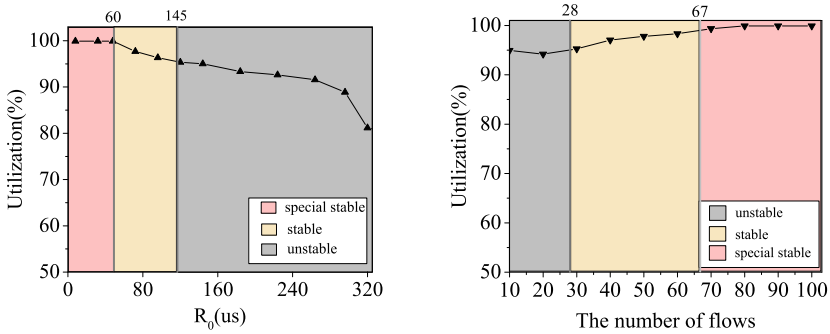
Through the stability analysis in Section 3.3, there is a special stable state under the condition of a large number of flows or small base RTT, according to $N < \frac{CR_0}{seg}$ in Theorem 2. When the DX system enters the special stable state, the utilization can even achieve 99.9% and the window size of each flow keeps 1. In this scenario, we will verify this conclusion.

We first set the number of flows to be 50 and the bottleneck bandwidth 10Gbps. Fig. 3(a) shows the three states of DX including the special stable,



(a) Special stable case (b) The stable case

Fig. 2. Comparison of the numerical results of fluid flow model with NS-3 simulation.



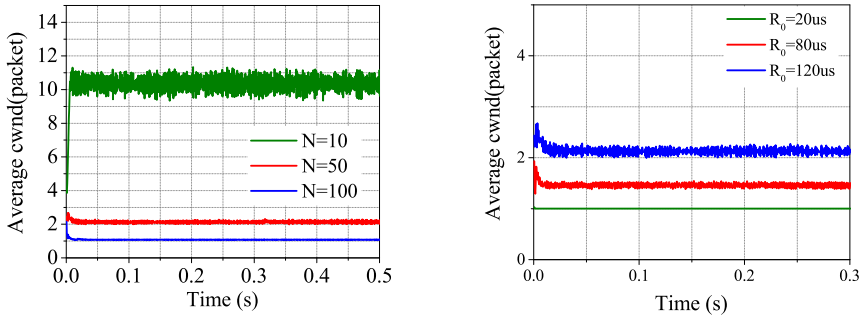
(a) $N = 50$. (b) $R_0 = 80\mu s$.

Fig. 3. The three states of DX.

the stable and the unstable states with varying R_0. In the special stable state, the link utilization is 99.9%. We observe that the transition from the stable state to the unstable state is smooth. In fact, the boundary between these two states is not absolute. This is because of we model and analysis DX with some assumptions, like homogeneous sources. In this case, the τ calculated according to Theorem 1, corresponding to the boundary line, is not the absolute upper bound of maintaining the stable state of DX.

Second, we set the base RTT to be $80\mu s$ and test the link utilization with varying N. When the number of flows exceeds the threshold (67 in Fig. 3(b)), the system enters the special stable state. Although the window sizes of these flows should be reduced due to the queueing delay, there is a limit on the window sizes, which cannot be lower than 1. As a result, the injected traffic may be greater than the bandwidth delay product, resulting in the queue at the switch cannot be drained up and high utilization.

Next, we inspect the special stable state further by taking deep study into the experiment detail. In Fig. 4(a) and Fig. 4(b), we show the dynamic change



(a) $R_0 = 120\mu s$. (b) $N = 50$.

Fig. 4. Comparison of the size of the average congestion window.

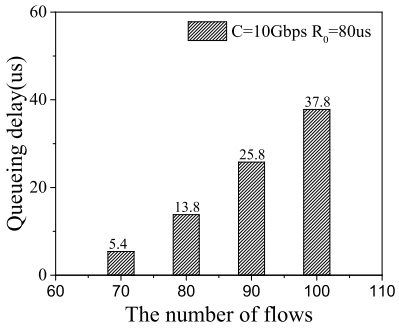


Fig. 5. The queueing delay changes with increasing N in the special stable state

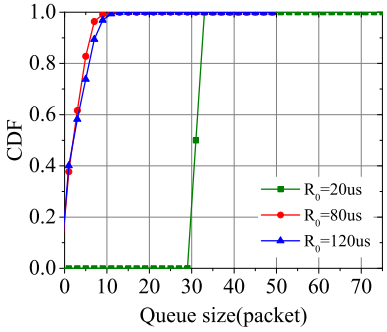


Fig. 6. The CDF of the queue size with the different base RTT

of the average congestion window (*cwnd*) of flows with increasing N when R_0 is 120 μs, and with increasing R_0 when N is 50. We calculate the corresponding conditions are $N \geq 100$ and $R_0 \leq 60$ for entering the special stable state, respectively. From Fig. 4, we can see that the average window size is indeed 1 when the conditions are satisfied, which means that the system enters the special stable state. Besides, according to our analysis, the queueing delay may increase with a larger number of flows. Fig. 5 shows the change of the queueing delay when N is larger than 70. We omit the result of $N < 70$ since the DX system enters a special stable state when $N \geq 67$ in this scenario. These simulation results verify our theoretical conclusions in Theorem 2, that is, the special stable state can lead to high network utilization but possible high queueing delay. Further, we plot the Cumulative Distribution Function (CDF) of the queue size in Fig. 6. When N is fixed, the system will enter the special stable state for smaller R_0,

resulting in that DX has the larger queue size or queueing delay for small R_0. In this figure, the queue size is constantly larger than 30 when $R_0 = 20\mu s$.

4.3 Stability Criterion

According to the analysis in Section 3 and Theorem 1, the system stability is affected by the number of flows N, R_0 and C. Further, the larger N or the smaller the R_0 or the smaller the C, the more stable the system. To verify this conclusion, we just change one parameter and keep other parameters invariant to investigate its sole influence on the stability of DX in our simulations.

Varying R_0 In this test, we fix the network parameter N as 50 and vary the base RTT R_0 from $20\mu s$, $120\mu s$ to $320\mu s$. According to Theorem 1, we calculate the upper bound of R_0 for keeping the DX system stable as 145 μs. We observe that the larger R_0 is, the lower the link utilization is, which ranges from 99.91%, 96.11% to 89.1%. The low link utilization means that the system becomes more unstable. This is consistent with the theoretical result.

Varying N In this test, we vary N from 10, 50 to 100 with fixed R_0 120 μs. The link utilization increases from 94.81%, 96.11% to 98.53% when N becomes larger and larger. Our theoretical conclusion is that when N is larger than 42, DX is stable according to Theorem 1. From the increase of the link utilization, our theoretical analysis is basically correct.

Varying C In this test, the bottleneck bandwidth C is changed from $1Gbps$, $10Gbps$ to $40Gbps$. We set N as 50 and the base RTT R_0 as 120 μs. . In particular, the link utilization decreases from 99.86%, 96.11% to 86.44%. when C is $40Gbps$, the utilization is lowest, which means that the system suffers from unstable. This confirms the theoretical analysis that the larger bandwidth will lead to the instability of the system in Section 3.

5 Conclusion

In this paper, we perform a theoretical analysis of DX, which is the up-to-date latency-based algorithm in data center network and has a better performance than the well-known DCTCP. Current investigations on DX are based on experiments and its theoretical analysis is spare. We establish the fluid-flow model of the DX system. By linearizing the fluid model and using the stability criterion of the linear system, we derive the stability condition of the DX system. According to our analysis, we found that the stability of the system is proportional to the number of flows, as well as inversely proportional to the propagation delay and the bottleneck bandwidth. In particular, there is a special stable state when N is too large or RTT is too small. Through the analysis, we find that DX has poor throughput when either the base RTT is very large or the number of flows is relatively small. Besides, DX suffers from large queueing delay when either the base RTT is relatively small or the number of flows is very large. Finally, we verify the conclusion in the NS-3 simulation. Our analysis takes a step forward for understanding DX deeply and can be helpful to deploy DX in the data center network or design new latency-based protocols built on DX.

References

1. Alizadeh, M., Greenberg, A., Maltz, D.A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., Sridharan, M.: Data center tcp (dctcp). In: ACM SIGCOMM computer communication review. vol. 40, pp. 63–74. ACM (2010)
2. Alizadeh, M., Javanmard, A., Prabhakar, B.: Analysis of dctcp: stability, convergence, and fairness. In: Proceedings of the ACM SIGMETRICS joint international conference on measurement and modeling of computer systems. pp. 73–84. ACM (2011)
3. Alizadeh, M., Kabbani, A., Atikoglu, B., Prabhakar, B.: Stability analysis of qcn: the averaging principle. In: Proceedings of the ACM SIGMETRICS joint international conference on measurement and modeling of computer systems. pp. 49–60. ACM (2011)
4. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
5. Gao, P.X., Narayan, A., Kumar, G., Agarwal, R., Ratnasamy, S., Shenker, S.: phost: Distributed near-optimal datacenter transport over commodity network fabric. In: Acm Conference on Emerging Networking Experiments & Technologies (2015)
6. Golnaraghi, F., Kuo, B.: Automatic control systems. Complex Variables **2**, 1–1 (2010)
7. Hollot, C.V., Misra, V., Towsley, D., Gong, W.: Analysis and design of controllers for aqm routers supporting tcp flows. IEEE Transactions on automatic control **47**(6), 945–959 (2002)
8. Jiang, W., Ren, F., Shu, R., Wu, Y., Lin, C.: Sliding mode congestion control for data center ethernet networks. IEEE Transactions on Computers **64**(9), 2675–2690 (2015)
9. Lee, C., Park, C.: Accurate latency-based congestion feedback for datacenters. In: USENIX ATC. pp. 403–415 (2015)
10. Lee, C., Park, C., Jang, K., Moon, S., Han, D.: Dx: Latency-based congestion control for datacenters. IEEE/ACM Transactions on Networking **25**(1), 335–348 (2017)
11. Misra, V., Gong, W.B., Towsley, D.: Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In: ACM SIGCOMM Computer Communication Review. vol. 30, pp. 151–160. ACM (2000)
12. Mittal, R., Dukkipati, N., Blem, E., Wassel, H., Ghobadi, M., Vahdat, A., Wang, Y., Wetherall, D., Zats, D., et al.: Timely: Rtt-based congestion control for the datacenter. In: ACM SIGCOMM Computer Communication Review. vol. 45, pp. 537–550. ACM (2015)
13. Srikant, R.: The mathematics of Internet congestion control. Springer Science & Business Media (2012)
14. Zhu, Y., Ghobadi, M., Misra, V., Padhye, J.: Ecn or delay: Lessons learnt from analysis of dcqcn and timely. In: Proceedings of the 12th international on conference on emerging networking experiments and technologies. pp. 313–327. ACM (2016)