Congestion Control in Converged Ethernet with Heterogeneous and Time-Varying Delays

Wenxue Cheng*, Wanchun Jiang†, Tong Zhang*, Bo Wang*, Kun Qian*, Fengyuan Ren*
* Tsinghua National Laboratory for Information Science and Technology, Beijing, China
* Department of Computer Science and Technology, Tsinghua University, Beijing, China
† School of Information Science and Engineering, Central South University, Changsha, China
Email: {chengwx14, zhang-t14, wang-b14, qk15}@mails.tsinghua.edu.cn, jiangwc@csu.edu.cn, renfy@tsinghua.edu.cn

***Abstract*—Congestion control is an indispensable mechanism in the new trend of enhanced Ethernet as a unified fabric for traditional LAN, SAN, and high-performance computing networks. A congestion management framework for Converged Ethernet (CE) networks has been standardized by IEEE 802.1 Qau work group, and QCN is recommended as the congestion control scheme in the standard draft. QCN is heuristically designed for 1/10Gbps Ethernet without considering the impact of delays. Recent work find that QCN will encounter stability issues with feedback delays, and these issues will be more serious as Ethernet extends to 40/100Gbps and the delays become heterogeneous and time-varying. This work aims to mitigate the negative impact of delays on congestion control scheme in CE. Specially, considering the delays are heterogeneous and time-varying, we build a model for Converged Ethernet with the standard congestion management framework. The model provides a new congestion detector to estimate the real congestion status under the impact of delays and regards the heterogeneous and time-varying feature as disturbances. Leveraging the new congestion detector and tolerating the disturbance through the sliding mode control method, we design the Delay-tolerant Sliding Mode (DSM) congestion control scheme. Extensive simulations show that DSM outperforms other congestion control schemes when the Ethernet ranges from $1Gbps$ to $100Gbps$ and the delays are heterogeneous and time-varying.**

***Index Terms*—Congestion Control, Converged Ethernet, Heterogeneous and Time-Varying Delays**

I. Introduction

The IEEE 802.1 Data Center Bridging (DCB) task group has developed and standardized mechanisms to enhance Ethernet as a unified switch fabric for TCP controlled LAN, lossless SAN, low latency and low jitter IPC traffic in data centers [1]. Nowadays, some of these enhancements have been used to improve the performance of data center networks, such as reducing the flow completion time tail [2], avoiding packets dropping for overlay virtual networking [3], and supporting RDMA-based key-value storage [4].

Among these enhancements, congestion control is one of the indispensable mechanisms to ensure losslessness and to achieve high link utilization. Considering the special environment of link layer and learning the experience from congestion control investigations in history, the IEEE 802.1 Qau work group [5] has defined a congestion management framework for Converged Ethernet (CE) in the link layer, and developed a series of congestion control schemes, including BCN [6], FERA [7], E2CM [8], and Quantized Congestion Notification (QCN) [9]. Finally, QCN is recommended as the congestion control scheme in the standard draft. Nowadays, QCN has been implemented in devices such as [10] and [11]. Recently, QCN is extended as DCQCN [12] to support RoCEv2 in large-scale IP-routed data center networks.

The IEEE 802.1 Qau working group was launched in 2006, and focused on small or medium-scale data centers where 1/10Gps Ethernet is presupposed as the dominant. Since the network perimeter is limited, the delays among hosts and nodes are relatively small, and the differences are also imperceptible. Thus, to keep the simplicity of implementation, the heuristic QCN algorithms takes no account of delay.

However, driven by cloud computing and large-scale online applications, data center networks are becoming bigger and more complex [13]. Correspondingly, the network delay becomes non-negligible, and the diversity of end-to-end paths results in the heterogeneity of delay. Furthermore, the burst traffic intensifies frequent fluctuations of the queue length in switches. Consequently, the network delay including queuing waiting time becomes time-varying.

The historical experience from investigations on traditional Internet tells that delays definitely impose negative impacts on the stability of congestion control schemes[14–17], where the sources are obstructed from timely obtaining the information of load and congestion status, and then hardly making a proper rate adjustment. Recent work [18, 19] also find that QCN will also encounter stability issues in face of the delays. More seriously, this problem will be exacerbated when Ethernet extends to 40/100Gbps and delays become heterogeneous and time-varying.

In this work, we attempt to mitigate the negative impacts of heterogeneous and time-varying delays on the congestion control schemes in CE. The main contributions are twofold:

- Considering the heterogeneous and time-varying delays, we build a dynamic model for the Converged Ethernet with the standard congestion control framework. From the model, we obtain a new congestion detector which uses historical information to eliminate errors in load estimation introduced by network delays. And the heterogeneous and time-varying features are regarded as disturbances and proved to be limited.
- We design the Delay-tolerant Sliding Mode (DSM) congestion control scheme, which leverages the new conges-

978-1-5386-2704-4/17/$31.00

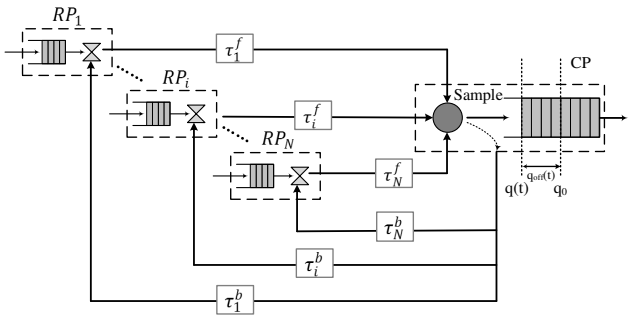


Fig. 1. Framework of Congestion Control

tion detector to mitigate the negative impact of delays. Specially, we follow the sliding mode control principle [20] in design of DSM to tolerate the disturbances caused by the heterogeneous and time-varying delays and adapt for Ethernet speeds ranging from 1Gbps to 100Gbps.

The rest of this paper is organized as follows. Section II presents the background and related work. In Section III, we we build a dynamic model for the Converged Ethernet with the standard congestion control framework. Subsequently, DSM is designed and its performance is evaluated in Section IV and Section V, respectively. Finally, Section VI concludes this paper.

II. Background and Related Work

A. Congestion Control in CE

The IEEE 802.1 Qau work group has standardized a congestion control framework for Converged Ethernet in the link layer. The framework is composed of two main parts: Congestion Point (CP) in switches and Reaction Point (RP) in sources, as illustrated in *Fig.*1. CP monitors the switch queue length and computes the congestion level as F_b. It samples the incoming packets with a probability p and sends a feedback packet involving F_b to the source of the sampled packet. After a backward delay τ^b, RP in the corresponding source receives the feedback packet and then adjusts its sending rate according to F_b. This rate adjustment will take effect on the switch queue length after a forward delay τ^f. With the cooperation of CP and RP, the switch buffer occupancy is expected to be controlled at a desired point q_0.

Following the above framework, several congestion control schemes have been proposed, from which QCN is the one recommended in the standard draft. SMCC [21] is another scheme that solves the configuration-sensitive issue of QCN. SMCC takes advantage of the slide mode motion to be robust to the changes of system parameters and network configurations. Therefore, SMCC can be easily implemented and scaled in most Ethernet networks.

B. Negative impact of Delays

Back to the congestion control framework, it is noteworthy that when the backward and forward delays are significantly large relative to the sampling period, the feedback congestion measure F_b and the rate adjustment may be ineffective. Moreover, due to the diversity of end-to-end paths, propagation and transmission delays become heterogeneous, and the network delay including queuing waiting time would be time-varying as a result of the burst traffic.

The historical experience from investigations on traditional Internet tells that the heterogeneous and time-varying delays definitely impose negative impacts on the stability of congestion control [14–17]. The relatively large network delays obstruct sources from timely obtaining the information of load and congestion status, and then hardly making a proper rate adjustment. And the heterogeneous and time-varying delays also pose an obstacle to taking coordinated actions in rate adjustments among different sources.

In addition, the impact of delays on the stability of congestion control system is strongly related to the link capacity. For example in practical Enhanced Ethernet of $10Gbps$, suppose the packet size is $1KB$ and the sampling probability $p = 0.01$, the sampling period T approximates to the time of transmitting 100 packets, i.e. about $80\mu s$. However, the feedback delays in 10Gbps Ethernet are about hundreds of microseconds [13], already larger than the sampling period. As the Ethernet speed extends to $40Gbps$ and $100Gbps$, the sampling period shrinks to $20\mu s$ and $8\mu s$, respectively. In this case, impacts of the large feedback delays will become more serious to a great extent.

Both QCN and SMCC are designed without considering such delays, thus suffer from the stability issues. Analytical results such as [18, 19] have shown the maximal tolerable delay of QCN is not beyond $500\mu s$ when the bottleneck link capacity is 10Gbps. Our simulation results in Section V also confirm this point and show that the maximal tolerable delay of SMCC cannot exceed $500\mu s$, either. And when the link capacity grows to 100Gbps, both QCN and SMCC fail to work normally with only $80\mu s$. Moreover, the situation becomes worse when the delays are heterogeneous and time-varying so that packet dropping and under-utilization are serious for both QCN and SMCC.

III. Model

In this section, we build a dynamic model for Converged Ethernet with the standard congestion control framework. Specially, the heterogeneous and time-varying delays are considered in this model. The main symbols are summarized in Table I.

A. Heterogeneous and Time-Varying Delays

Suppose that there are a total of N sources sharing the bottleneck link. For each source $S_i (i = 1, \cdots, N)$ and each sampling time $t_k (k = 1, 2, \cdots)$, $\tau_i^f(k)$ and $\tau_i^b(k)$ denote the forward and backward delay between source S_i and the congested switch respectively. Thus, $\tau_i(k) = \tau_i^f(k) + \tau_i^b(k)$ presents the total delay in the corresponding feedback loop. Note that the switch transmit $1/p$ packets during the sampling interval, the sampling period T is deterministic when the switch is congested. We can rewrite the sampling time t_k as

TABLE I
SYMBOLS

Symbol	Definition
N	Total of sources
$r_i(k)$	Send rate of source RP_i during $((k-1)T, kT]$
C	Capacity of the bottleneck link
p	Sampling probability of CP
T	Sampling period of CP
$\tau_i(k)$	Feedback loop delays between RP_i and CP at time kT
$m_i(k)$	$\tau_i(k) = m_i(k)T$
m	Maximum value of $m_i(k)$
$Q_f(k)$	Switch queue length offset at time kT
$Q_v(k)$	Variance of queue length in a sampling interval
$\widehat{Q_f}(k)$	Estimation of $Q_f(k+m)$
$\widehat{Q_v}(k)$	Estimation of $Q_v(k+m)$
ξ	Disturbance implying heterogeneous and time-varying delays

kT and quantify the heterogeneous and time-varying delays $\tau_i^f(k)$, $\tau_i^b(k)$ and $\tau_i(k)$ with T, i.e.

$$\begin{cases} m_i^f(k) = \tau_i^f(k)/T \\ m_i^b(k) = \tau_i^b(k)/T \\ m_i(k) = \tau_i(k)/T \end{cases}$$

B. Congestion Control Rule

The congestion control scheme aims to maintain the switch buffer queue length q at a desired point q_0. Specially, at each sampling time kT, CP in the switch monitors both the offset of queue length $Q_f(k) = q(k) - q_0$ and the variance of queue length $Q_v(k) = q(k) - q(k-1)$ to compute the congestion measure $F_b(k)$. In fact, F_b can be computed utilizing the historical states such $Q_f(k-1), Q_v(k-1), Q_f(k-2), Q_v(k-2)$ as well, i.e.

$$F_b(k) = f(Q_f(k), Q_v(k), Q_f(k-1), Q_v(k-1), \cdots) \quad (1)$$

where f is a function that denotes the algorithm in CP.

After CP generates a feedback packet with $F_b(k)$ at sampling time kT, the source of the sampled packet RP_i will receive the feedback packet at time $(k + m_i^b(k))T$. At this time, RP_i can adjusts it sending rate r_i according to $F_b(k)$, that is

$$r_i\left(k + m_i^b(k) + 1\right) = r_i\left(k + m_i^b(k)\right) + u(F_b(k)) \quad (2)$$

where u is another function that denotes the rate adjustment algorithm in RP. According to equation (1)-(2), we can rewrite $u(F_b(k))$ as $u(k)$, i.e.

$$u(k) = u(f(Q_f(k), Q_v(k), Q_f(k-1), Q_v(k-1), \cdots)) \quad (3)$$

Finally, the algorithms in CP and RP are combined in u, which is called congestion control rule.

C. Evolution of Switch Queue

The evolution of the switch queue length can indicate the network performance, such as convergence and stability. In this subsection, we will show how a congestion control scheme impacts on the switch queue length, especially with heterogeneous and time-varying delays.

Note that the feedback packet with $F_b(k)$ is only sent to RP_i, the other RPs would not change their sending rates at time $(k + m_j^b(k))T$, i.e.

$$r_j\left(k + m_j^b(k) + 1\right) = r_j\left(k + m_j^b(k)\right), \forall j = 1, \cdots, N, j \neq i \quad (4)$$

Combining equations (2) and (4), we have

$$\begin{aligned} & \sum_{j=1}^{N} r_j\left(k + m_j^b(k) + 1\right) - \sum_{j=1}^{N} r_j\left(k + m_j^b(k)\right) \\ = \; & r_i\left(k + m_i^b(k) + 1\right) - r_i\left(k + m_i^b(k)\right) \\ = \; & u(k) \end{aligned} \quad (5)$$

Equation (5) implies that the evolution of the aggregated sending rate is totally determined by the feedback control rule and delays.

After RP_i adjusts its sending rate by the value of $u(k)$ at time $(k + m_i^b(k))T$, CP in the switch will not be aware of this rate adjustment until time $(k + m_i^b(k) + m_i^f(k))T$, i.e. $(k + m_i(k))T$. Note that for different i and k, the value of $m_i(k)$ are different, we define

$$m \triangleq \inf\{m \in \mathbb{N}^+ : m_i(k) \leq m, \forall i, k\} \quad (6)$$

Actually, m indicates the worst feedback delay in network. In the sampling interval of $[(k+m)T, (k+m+1)T]$, the variation of the switch queue length satisfies

$$\begin{aligned} & Q_v(k+m+1) \\ = \; & T\left[\sum_{j=1}^{N} r_j\left(k + m + 1 - m_j^f(k)\right) - C\right] \\ \triangleq \; & T\left[\sum_{j=1}^{N} r_j\left(k + m_j^b(k) + 1\right) + R(k) - C\right] \end{aligned} \quad (7)$$

where

$$R(k) \triangleq \sum_{j=1}^{N}[r_j\left(k + m - m_j^f(k) + 1\right) - r_j\left(k + m_j^b(k) + 1\right)] \quad (8)$$

$R(k)$ summarizes the influences of the heterogeneous and time-vary delays. Specially, if the delays are constant, i.e. $m_j(k) \equiv m$, then $R(k) \equiv 0$. Consequently, we can consider $R(k)$ as disturbance to approximate the network with heterogeneous and time-varying delays as that with constants delays.

Defining $\xi(k) \triangleq R(k) - R(k-1)$ and referring to equations (5) and (7), we have

$$\begin{aligned} & Q_v(k+m+1) - Q_v(k+m) \\ = \; & T\sum_{j=1}^{N}\left[r_j\left(k + m_j^b(k) + 1\right) - r_j\left(k + m_j^b(k)\right)\right] \\ & + T\left[R(k) - R(k-1)\right] \\ = \; & Tu(k) + T\xi(k) \end{aligned}$$

Considering the evolution of Q_f and Q_v, there is

$$\begin{cases} Q_f(k+m+1) - Q_f(k+m) = Q_v(k+m+1) \\ Q_v(k+m+1) - Q_v(k+m) = T[u(k) + \xi(k)] \end{cases} \quad (9)$$

Equation (9) seemingly implies the evolution of switch queue length. However, $u(k)$ must be determined at time kT, while

$Q_f(k+m)$ and $Q_v(k+m)$ are unknown. Thus, the evolution of switch queue length is not evident. Subsequently, we can estimate $Q_f(k+m)$ and $Q_v(k+m)$ by recursion. Defining

$$X(k) \triangleq \left[\begin{array}{c} Q_f(k) \\ Q_v(k) \end{array} \right]$$

equations (9) can be written as follows

$$AX(k+m+1) - X(k+m) = Bu(k) + C\xi(k) \tag{10}$$

where

$$A = \left[\begin{array}{cc} 1 & -1 \\ 0 & 1 \end{array} \right],\ B = \left[\begin{array}{c} 0 \\ T \end{array} \right],\ C = \left[\begin{array}{c} 0 \\ T \end{array} \right] \tag{11}$$

From equation (10), we can do recursion as follows

$$\begin{aligned} & X(k+m) \\ = \ & A^{-1}X(k+m-1) + A^{-1}Bu(k-1) + A^{-1}C\xi(k-1) \\ = \ & A^{-2}X(k+m-2) + A^{-2}Bu(k-2) + A^{-1}Bu(k-1) \\ & +A^{-2}C\xi(k-2) + A^{-1}C\xi(k-1) \\ = \ & \cdots \\ = \ & A^{-m}X(k) + \sum_{i=1}^{m} A^{-i}Bu(k-i) + \sum_{i=1}^{m} A^{-i}C\xi(k-i) \end{aligned} \tag{12}$$

Substituting in (11) into (12), we can obtain

$$\left\{ \begin{array}{l} Q_f(k+m) = Q_f(k) + mQ_v(k) \\ \qquad\qquad +T\sum_{i=1}^{m} i[u(k-i) + \xi(k-i)] \\ Q_v(k+m) = Q_v(k) + T\sum_{i=1}^{m} [u(k-i) + \xi(k-i)] \end{array} \right. \tag{13}$$

Taking out the disturbances ξ from equation (13), we define

$$\left\{ \begin{array}{l} \widehat{\boldsymbol{Q_f}}(\boldsymbol{k}) \triangleq \boldsymbol{Q_f(k) + mQ_v(k) + T\sum_{i=1}^{m} iu(k-i)} \\ \widehat{\boldsymbol{Q_v}}(\boldsymbol{k}) \triangleq \boldsymbol{Q_v(k) + T\sum_{i=1}^{m} u(k-i)} \end{array} \right. \tag{14}$$

Comparing equation (13) and (14), we can find that $\widehat{Q_f}$ and $\widehat{Q_v}$ are estimations of $Q_f(k+m)$ and $Q_v(k+m)$, and the estimation errors rely on disturbances ξ. That is, when the delays are identical, i.e. $\xi(k) \equiv 0$ or $m_i(k) \equiv m$, we have $\widehat{Q_f}(k) = Q_f(k+m)$ and $\widehat{Q_v}(k) = Q_v(k+m)$. Referring to equation (13),

$$\begin{aligned} & \widehat{Q_f}(k+1) - \widehat{Q_f}(k) \\ = \ & [Q_f(k+m+1) - T\sum_{i=1}^{m} i\xi(k-i+1)] \\ & -[Q_f(k+m) - T\sum_{i=1}^{m} i\xi(k-i)] \\ = \ & \widehat{Q_v}(k+1) + mT\xi(k-m) \end{aligned} \tag{15}$$

and

$$\begin{aligned} & \widehat{Q_v}(k+1) - \widehat{Q_v}(k) \\ = \ & [Q_v(k+m+1) - T\sum_{i=1}^{m} \xi(k-i+1)] \\ & -[Q_v(k+m) - T\sum_{i=1}^{m} \xi(k-i)] \\ = \ & Tu(k) + T\xi(k-m) \end{aligned} \tag{16}$$

Combining equations (15) and (16), we can obtain an approximate evolution of the switch queue length,

$$\left\{ \begin{array}{l} \boldsymbol{\widehat{Q_f}(k+1) - \widehat{Q_f}(k) = \widehat{Q_v}(k+1) + mT\xi(k-m)} \\ \boldsymbol{\widehat{Q_v}(k+1) - \widehat{Q_v}(k) = Tu(k) + T\xi(k-m)} \end{array} \right. \tag{17}$$

where $\widehat{Q_f}(k)$ and $\widehat{Q_v}(k)$ are estimation of $Q_f(k+m)$ and $Q_v(k+m)$, $\xi(k-m)$ is the disturbance caused by the dynamic characteristics of heterogeneous and time-varying delays.

D. New Congestion Detector

From the approximate evolution of the switch queue length in equation (17), we find that the variable pair $(\widehat{Q_f}(k), \widehat{Q_v}(k))$ is a good congestion detector for the congestion control system with delays. The reason is twofold.

- The variable pair $(\widehat{Q_f}(k), \widehat{Q_v}(k))$ estimates the real evolution of the switch queue length with the impact of delay by utilizing the historical feedback adjustments $u(k-m), \cdots, u(k-1)$. It can be calculated immediately according to equation (14) when sampling a packet and sending a feedback message.
- When $(\widehat{Q_f}(k), \widehat{Q_v}(k)) = (0,0)$, the switch queue length is equal to the target value q_0 and the incoming rate of the congested switch is equal to the link capacity. That is, $(\widehat{Q_f}(k), \widehat{Q_v}(k)) = (0,0)$ is the stable state of the congestion control system.

Moreover, the heterogeneous and time-varying feature of delays is indicated in the disturbance $\xi(k-m)$. According to the definition of $\xi_(k)$ and equations (5)(8), we have

$$\begin{aligned} & \xi(k-m) \\ = \ & \sum_{j=1}^{N} \Big[r_j(k - m_j^f(k) + 1) - r_j(k - m_j^f(k)) \\ & -r_j(k - m + m_j^b(k) + 1) + r_j(k - m + m_j^b(k)) \Big] \\ = \ & \sum_{j=1}^{N} \Big[r_j(k - m_j^f(k) + 1) - r_j(k - m_j^f(k)) \Big] - u(k-m) \\ < \ & (N+1)\,|u|_{\max} \end{aligned} \tag{18}$$

where $|u|_{\max}$ denotes the largest value of one step rate adjustment. Therefore, the disturbance $\xi(k-m)$ is limited.

IV. The Delay-Tolerant Sliding Mode Congestion Control Scheme

This section presents the design of Delay-tolerant Sliding Mode (DSM) congestion control scheme, which leverages the new congestion detector to mitigate the negative impact of delays and utilizes the sliding mode control method to tolerate the disturbance caused by heterogeneous and time-varying delays. In addition, we will demonstrate that DSM has good properties in terms of complexity, stability, responsiveness and adaptability.

A. Congestion Control Rule

At first, we will design the congestion control rule u of DSM. To mitigate the negative impact of delays, the new congestion detector provided in our model is used. And to

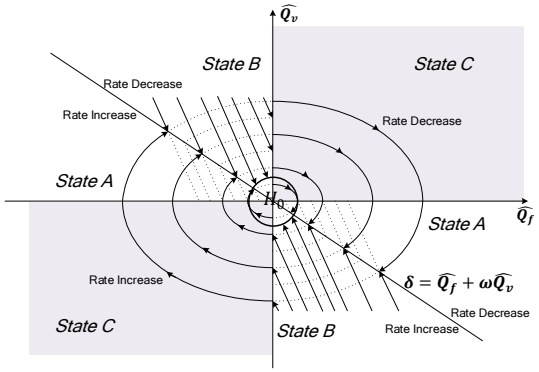


Fig. 2. DSM Design

tolerate the disturbance caused by the heterogeneous and time-varying delays, we follow the sliding mode control principle [20] in design, which has three steps.

Step 1: Determining Boundary Line. *Fig.*2 illustrates the state plane of the new congestion detector $(\widehat{Q_f}, \widehat{Q_v})$. We divide the state space into four parts by the axes of $\widehat{Q_f} = 0$ and $\widehat{Q_v} = 0$, which denote expected buffer queue and incoming rate, respectively. In quadrant I, we have $\widehat{Q_f} > 0$ and $\widehat{Q_v} > 0$, which implies that both switch queue length and incoming rate are excessive, thus the sources need to decrease their sending rates. On the opposite, in quadrant III, there is $\widehat{Q_f} < 0$ and $\widehat{Q_v} < 0$, which indicates that both switch queue length and incoming rate are lacking, hence the sources need to increase their sending rates. However, in the remaining parts of quadrant II and IV, we have $\widehat{Q_f} * \widehat{Q_v} < 0$, which means either of switch queue length and incoming rate is exceed and the other is lacking, as a result it is undermined for the sources whether to increase or decrease their sending rates. Consequently, we divide these parts of quadrant II and IV into areas of "rates increase" and "rates decrease" by a boundary line $\delta = 0$, where

$$\delta(k) = \widehat{Q_f}(k) + \omega\widehat{Q_v}(k) \tag{19}$$

and ω is a positive constant.

Step 2: Developing Rules Fulfilling Sliding Mode Motion. As shown in *Fig.*2, we obtain three parts named as $StateA$, $StateB$ and $StateC$, and a boundary line $\delta = 0$ to separate the "rates increase" and "rates decrease" areas. We can select this boundary line to construct the sliding mode motion. According to [20], the control rule u should have a variable structure among the boundary line and satisfy the following sufficient condition,

$$\delta(k)[\delta(k+1) - \delta(k)] < 0 \tag{20}$$

For simplicity, we assume $u(k)$ is a linear function of the new congestion detector $(\widehat{Q_f}(k), \widehat{Q_v}(k))$,

$$u(k) = \begin{cases} -a_1\widehat{Q_f}(k) - b_1\widehat{Q_v}(k), & in\ StateA \\ -a_2\widehat{Q_f}(k) - b_2\widehat{Q_v}(k), & in\ StateB \end{cases} \tag{21}$$

where a_1, b_1, a_2, b_2 are nonnegative coefficients. Actually, after some algebraic derivations in Appendix A, we have the following proposition.

Proposition 1. *When the coefficients a_1, b_1, a_2, b_2 of the congestion control rule u in equation (21) satisfy*

$$\begin{cases} \left(1 + \frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_f}(k)}\right| < a_1 < \frac{2}{T} \\ b_1 = 0 \\ a_2 = 0 \\ \frac{1}{(\omega+1)T} + \left(1 + \frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_v}(k)}\right| < b < \frac{1}{T} \end{cases}$$

the system will "slide along the boundary line $\delta = 0$ under the control of u.

As drawn in *Fig.*2, the trajectory of system under the congestion control rule u in equation (21) follows clockwise ellipses and straight lines of slope $-\frac{b_2 T}{1-b_2 T}$ in $StateA$ and $StateB$, respectively.

Step 3: Developing Reaching Process. Finally, we will complement the congestion control rule u in $StateC$. From any initial point in $StateC$, the trajectory of system should be forced into $StateA$ or $StateB$ under the congestion control rule u. We set $u(k) = -c\widehat{Q_f}(k)$ with $0 < c < \frac{2}{T}$ in $StateC$, such that the trajectory of system follows a clockwise ellipse in $StateC$ and eventually enters into $StateA$.

In conclusion, based on our model result (17) and sliding mode control, we design the congestion control rule u of DSM

$$\boldsymbol{u(k) = \begin{cases} -a\widehat{Q_f}(k) & if\ \ \widehat{Q_v}(k)\delta(k) < 0 \\ -b\widehat{Q_v}(k) & if\ \ \widehat{Q_f}(k)\delta(k) < 0 \\ -c\widehat{Q_f}(k) & if\ \widehat{Q_f}(k)\widehat{Q_v}(k) > 0 \end{cases}} \tag{22}$$

where $\delta(k) = \widehat{Q_f}(k) + \omega\widehat{Q_v}(k)$, and coefficients satisfy

$$\begin{cases} \left(1 + \frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_f}(k)}\right| < a < \frac{2}{T} \\ \frac{1}{(\omega+1)T} + \left(1 + \frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_v}(k)}\right| < b < \frac{1}{T} \\ 0 < c < \frac{2}{T} \end{cases} \tag{23}$$

B. DSM Scheme

In this subsection, we will transform the congestion control rule u of DSM in equation (22) into algorithms of CP and RP.

The new congestion detector $(\widehat{Q_f}(k), \widehat{Q_v}(k))$ defined in (14) does not only rely on the current switch state, i.e. $Q_f(k)$ and $Q_v(k)$, but also depend on the historical feedback control values $u(k-1), \cdots, u(k-m)$. Note that $Q_f(k)$ and $Q_v(k)$ are monitored by CP and $u(k-1), \cdots, u(k-m)$ are based on the historical congestion measure $F_b(k-1), \cdots, F_b(k-m)$ computed by CP, the new congestion detector $(\widehat{Q_f}(k), \widehat{Q_v}(k))$ just can be computed in CP. Consequently, CP has to store the historical congestion measure $F_b(k-1), \cdots, F_b(k-m)$ and

calculate the RP algorithm $u(F_b)$. Hence, the CP algorithm in DSM is

$$F_b(k) = \begin{cases} -a\widehat{Q_f}(k) & if \quad \widehat{Q_v}(k)\delta(k) < 0 \\ -b\widehat{Q_v}(k) & if \quad \widehat{Q_f}(k)\delta(k) < 0 \\ -c\widehat{Q_f}(k) & if \ \widehat{Q_f}(k)\widehat{Q_v}(k) > 0 \end{cases} \tag{24}$$

where $\delta(k)$ is defined in (19), coefficients a, b, c satisfy (23), and $\widehat{Q_f}(k)$ and $\widehat{Q_v}(k)$ can be obtained as follows

$$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + T\sum\limits_{i=1}^{m} iF_b(k-i) \\ \widehat{Q_v}(k) = Q_v(k) + T\sum\limits_{i=1}^{m} F_b(k-i) \end{cases} \tag{25}$$

On the RP side, the algorithm is $u(F_b) = F_b$, i.e.

$$r \leftarrow r + F_b \tag{26}$$

C. Parameter Settings

The configurable parameters in DSM are m, a, b, c and ω.

m: According to equation (6), m is the smallest integer which is not less than any feedback loop delay $m_i(k)$ in the network. In implementation, m should be estimated in advance. If m is smaller than the ideal value, the unexpected large delay could harm the stability of the congestion control system. On the contrary, if m is larger than the ideal value, the congestion control system can always keep stable, as we designed the rate adjustment rules for the worst condition. However, the congestion control system may become less responsive. Consequently, m should be set at the ideal value or a little larger.

a, b, c: Parameters a, b, c are the coefficients of the congestion control rule u. On one hand, equation (23) indicates that a and b should be large enough to tolerate the disturbance $\xi(k-m)$. On the other hand, parameters a, b, c should also guarantee the largest value of one step rate adjustment not to exceed the link capacity C, i.e.

$$|u|_{\max} \le C. \tag{27}$$

Substituting the feedback loop rule u (22) into constraint (27) the parameters a, b, c have to satisfy

$$\begin{cases} a \le \frac{C}{|\widehat{Q_f}(k)|_{max}} \\ b \le \frac{C}{|\widehat{Q_v}(k)|_{max}} \\ c \le \frac{C}{|\widehat{Q_f}(k)|_{max}} \end{cases} \tag{28}$$

Referring to the definition of $\widehat{Q_f}(k)$ and $\widehat{Q_v}(k)$ in (14), there is

$$\begin{cases} |\widehat{Q_f}(k)| \le \max(Q_f + mQ_v) + \frac{m(m+1)}{2}CT \\ |\widehat{Q_v}(k)| \le \max(Q_v) + mCT \end{cases} \tag{29}$$

Substituting (29) into (28) and noting that $T = \frac{1}{pC}$, there is

$$\begin{cases} a < \frac{1}{p\max(Q_f + mQ_v) + \frac{m(m+1)}{2}} \frac{1}{T} \\ b < \frac{1}{p\max(Q_v) + m} \frac{1}{T} \\ c < \frac{1}{p\max(Q_f + mQ_v) + \frac{m(m+1)}{2}} \frac{1}{T} \end{cases} \tag{30}$$

When the delays increase, m increases and the upper bounds of parameters a, b and c in (30) decrease, which implies that the system will evolve less violently. However, parameter c should be set large enough to accelerate the reaching process in $StateC$. And equation (23) shows that the delays have no impact on parameter c. Therefore, the item associated with m in (30) can be ignored for c, namely,

$$c < \frac{1}{p\max(Q_f)} \frac{1}{T}$$

ω: Parameter ω denotes the weight of $\widehat{Q_v}$ in boundary line δ of "rate increase" and "rate decrease". Combining (23) and (30), we can get

$$\frac{1}{(\omega+1)T} < b < \frac{1}{p\max(Q_v) + m} \frac{1}{T}$$

Therefore,

$$\omega > m + p\max(Q_v) - 1$$

In conclusion, the parameters of DSM should be set as follows

$$\begin{cases} m = \inf\{m \in \mathbb{N}^+ : m_i(k) \le m, \forall i, k\} \\ a \to \left[\frac{1}{p\max(Q_f + mQ_v) + \frac{m(m+1)}{2}} \frac{1}{T}\right]^- \\ b \to \left[\frac{1}{p\max(Q_v) + m} \frac{1}{T}\right]^- \\ c \to \left[\min\left\{\frac{1}{p\max(Q_f)} \frac{1}{T}, \frac{2}{T}\right\}\right]^- \\ \omega > m + p\max(Q_v) - 1 \end{cases} \tag{31}$$

where the symbol $x \to [y]^-$ means x approaches but is less than y.

D. Properties

Subsequently, we will explain the basic properties of DSM, including *complexity*, *stability*, *responsiveness* and *adaptability*. Moreover, these properties will be verified by simulations in Section V.

Complexity: In CP, the spatial complexity is $O(M)$ to memorize the last m feedbacks, and the computing complexity can be reduced to $O(1)$ by introducing two variables $S_1(k)$ and $S_2(k)$

$$\begin{cases} S_1(k) = \sum\limits_{i=1}^{m} F_b(k-i) \\ S_2(k) = \sum\limits_{i=1}^{m} iF_b(k-i) \end{cases}$$

into equation (25), i.e.

$$\begin{cases} \widehat{Q_f}(k) = Q_f(k) + mQ_v(k) + TS_2(k) \\ \widehat{Q_f}(k) = Q_v(k) + TS_1(k) \\ S_1(k+1) = S_1(k) + F_b(k) - F_b(k-m) \\ S_2(k+1) = S_1(k+1) + S_2(k) - mF_b(k-m) \end{cases}$$

In addition, the complexity in RP is $O(1)$ according to the RP algorithm (26). Therefore, the complexity of DSM is low.

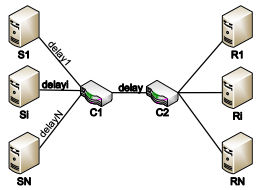
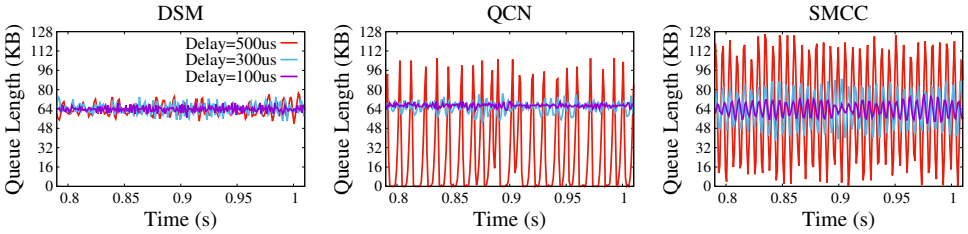


Fig. 3. Dumbbell Topology



(a) Queue length evolutions in DSM, QCN and SMCC in $10Gbps$ Ethernet.

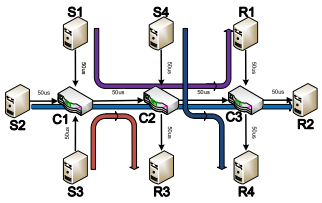
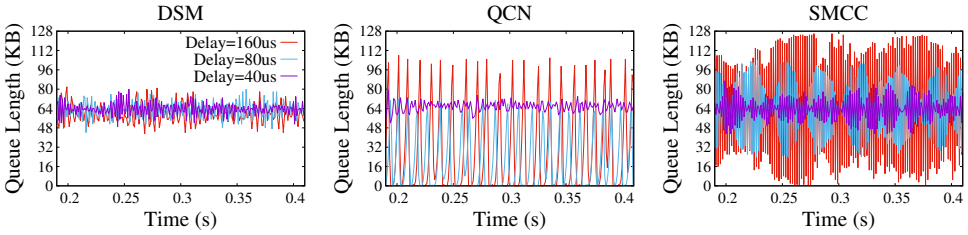


Fig. 4. Parking-lot Topology



(b) Queue length evolutions in DSM, QCN and SMCC in $100Gbps$ Ethernet.

Fig. 5. Stability.

Stability: DSM leverages the new congestion detector $(\widehat{Q}_f(k), \widehat{Q}_v(k))$ to mitigate the negative impact of delays and regards the heterogeneous and time-varying features as limited disturbance $\xi(k-m)$. The disturbance $\xi(k-m)$ can be tolerated. As shown in *Fig.*2, there exists an area H_0 in the neighborhood of the stable state $(\widehat{Q}_f(k), \widehat{Q}_v(k)) = (0,0)$, where the coefficients constraints (23) hardly hold due to the disturbance $\xi(k-m)$. However, the system can always enter into H_0 via the sliding mode motion and stay in H_0. Moreover, the range of H_0 is quite small when the parameters a and b are large enough. Hence, DSM has a good stability with heterogeneous and time-varying delays.

Responsiveness: When the system is in serious overflow or underflow, i.e. in the $StateC$ in *Fig.*2, DSM will force the system in to the sliding mode motion through the reaching process. Since the sliding mode motion has been proved to converge rapidly in [20], the responsiveness of DSM is mainly based on parameter c, which is the rate of the reaching process. According to equation (31), c is set as large as possible and will not decrease when the delays increase. Therefore, DSM is always responsive regardless of the delays.

Adaptability: Taking advantages of sliding mode motion [20], DSM is naturally adaptive for various network configurations. Simulations in next section will show that DSM always guarantees 100% utilization of the bottleneck link regardless of the link capacity and the feedback delays. In addition, DSM involves a CPID flag in the feedback packets to adapt for multiple bottlenecks scenario. The CPID flag has been used in BCN [6] to explicitly identify the congestion point. The sources will store the CPID when receiving a "rate-decreased" feedback, and only react to the "rate-increase" feedbacks which match the stored CPID. Consequently, DSM DSM adapts to various scenarios.

V. Performance Evaluation

In this section, we will evaluate the performance of DSM by simulations on NS2 platform, as well as comparing with QCN and SMCC. Specially, both the single bottleneck scenario and the multiple bottlenecks scenario are considered.

A. Default Configurations

The default network configuration of our simulations is as follows: the buffer size is $128KB$ and the target queue length is set to be $q_0 = 64KB$, the size of a packet is $1KB$, the link capacity is $10Gbps$ and the feedback loop delay is $300\mu s$.

For DSM, the sampling probability p is set to be 0.01, i.e. $100KB$ data will pass CP during a sampling interval. Consequently, the sampling period is $T = 80\mu s$ and $\frac{1}{T} = 12.5KHz$ when links are fully utilized. Substituting this default network configuration into the parameter constraints in (31), we have

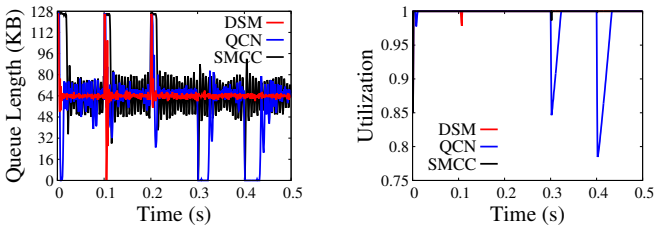
$$\begin{cases} a \to \left[\frac{1}{m^2+4m+2} * \frac{2}{T}\right]^- \\ b \to \left[\frac{1}{2m+3} * \frac{2}{T}\right]^- \\ c \to \left[\frac{1}{2} * \frac{2}{T}\right]^- \end{cases}$$

Following this guideline, parameters a, b, c should be set such that $H_a \triangleq (m^2+4m+2)a$, $H_b \triangleq (2m+3)b$ and $H_c \triangleq 2c$ are smaller than $\frac{2}{T} = 25KHz$. By default, we set $m = \max_{i,k}\{m_i(k)\} = 4$, $H_a = H_b = H_c = 20KHz$ and $\omega = 5$.

In addition, QCN is configured as [22] and SMCC follows the parameter settings in [21].

B. Single Bottleneck Scenario

Firstly, the N-sources dumbbell topology shown in *Fig.*3 is used to show the basic properties of DSM, especially stability, responsiveness, adaptability and tolerance for heterogeneous and time-varying delays in the single bottleneck scenario.



(a) Queue Length (b) Utilization

Fig. 6. Responsiveness

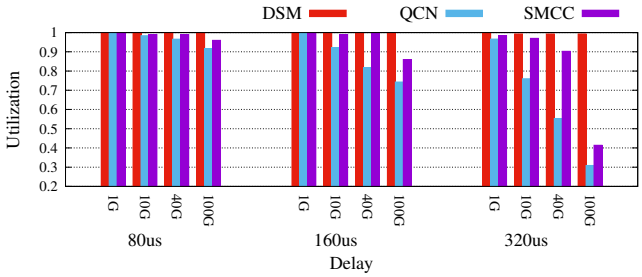
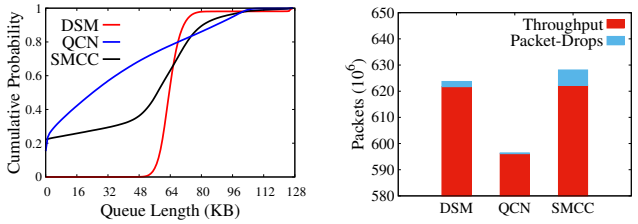


Fig. 7. Adaptability

Stability: To compare the stability of DSM, QCN and SMCC, we let $N = 5$ and source S_i start a flow to R_i at the initial time. The initial speed of each flow equals to the bottleneck capacity. *Fig.*5(a) shows the evolution of the queue length with the feedback delay increases in $10Gbps$ Ethernet. When the delay is small ($100\mu s$), all the schemes are able to keep the queue stable around the desired point $q_0 = 64KB$, although QCN and SMCC oscillate slightly. But when the delay increases to $300\mu s$, while DSM still keeps the queue stable, both QCN and SMCC become out of control and oscillate. However these oscillations are not too violent to underflows or overflows. Finally the delay increases to $500\mu s$, DSM works well. On the other hand, QCN begins to underflows and SMCC oscillates so seriously that it becomes unstable. We repeats the simulations when the link capacity increases to $100Gbps$. Results are shown in *Fig.*5(b). Obviously, DSM works well when the delay is as large as $160\mu s$, while the largest tolerable delay for QCN and SMCC are only tens of microseconds. Therefore, DSM is more stable than QCN and SMCC with feedback delays.

Responsiveness: To explore the responsiveness of DSM, QCN and SMCC, we let 3 sources start at the beginning and other two sources start one by one every $0.1s$ later. Each flow starts at a speed of the bottleneck capacity and two flows will finishes at $0.3s$ and $0.4s$, respectively. As illustrated in *Fig.*6, in face of new flows, DSM and QCN can drain out the switch buffer rapidly while SMCC keeps the switch buffer full for a long time. When a flow finishes, DSM and SMCC can converge fast to 100% utilization, but QCN always wastes a few utilization. Hence, DSM responds faster than QCN and SMCC.

Adaptability Choosing different bottleneck link capacities of $1Gbps$, $10Gbps$, $40Gbps$ and $100Gbps$ and different feedback delays of $80\mu s$, $160\mu s$, $320\mu s$, we let $N = 5$ and all sources start a flow at the initial time. The initial speed of each flow equals to the bottleneck capacity. We record the utilization of the bottleneck link as results in *Fig.*7. Consistent with the result of [19], the larger the bandwidth, the lower the utilization for QCN, which is the same for SMCC. Especially when the link capacity increases to $100Gbps$, the utilizations of QCN and SMCC both greatly degrade due to large delays. In contrast, DSM remains almost 100% utilization regardless of the delays and the bandwidth.



(a) Queue Length (b) Throughput and Packet-Drops

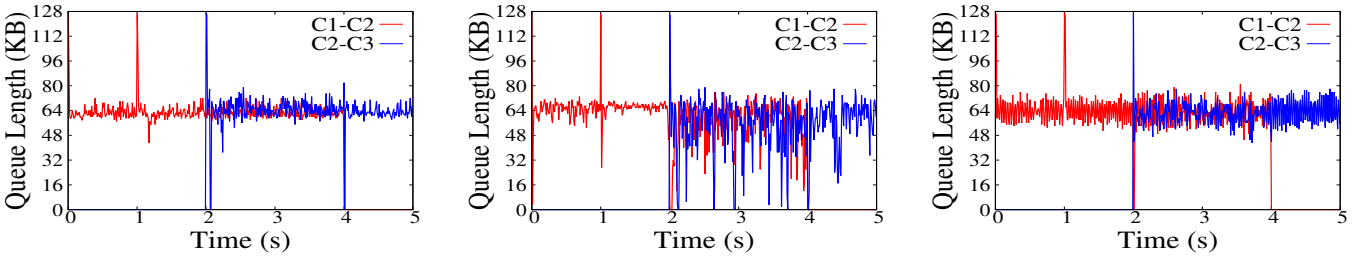
Fig. 8. Heterogeneous and Time-Varying Delays

Heterogeneous and Time-Varying Delays: Subsequently, let the delays of links be randomly distributed and the latency of generating a feedback packet be time-varying, but ensure that all the feedback loop delays are in $[400\mu s, 800\mu s]$. We conduct simulations under the five sources dumbbell topology with the bottleneck link capacity of $10Gbps$. Repeating simulations 100 times and each simulation lasts $5s$, we can record the queue lengths in DSM, QCN and SMCC, and the cumulative distribution functions are drawn in *Fig.*8(a). As for DSM, the queue length is always held closer to the desired point $q_0 = 64KB$ and rarely underflows. In contrast, both QCN and SMCC suffer from a large probability of empty buffer such that link utilization is low. And the queue length CDFs with QCN and SMCC are not concentrated around the desired value $Q_0 = 64KB$, which indicates frequent large oscillations. Actually, DSM has higher throughput than QCN and less packet-drops than SMCC, as shown in *Fig.*8(b). Thus, DSM is much more suitable for heterogeneous and time-varying delays than QCN and SMCC.

C. Multiple Bottlenecks Scenario

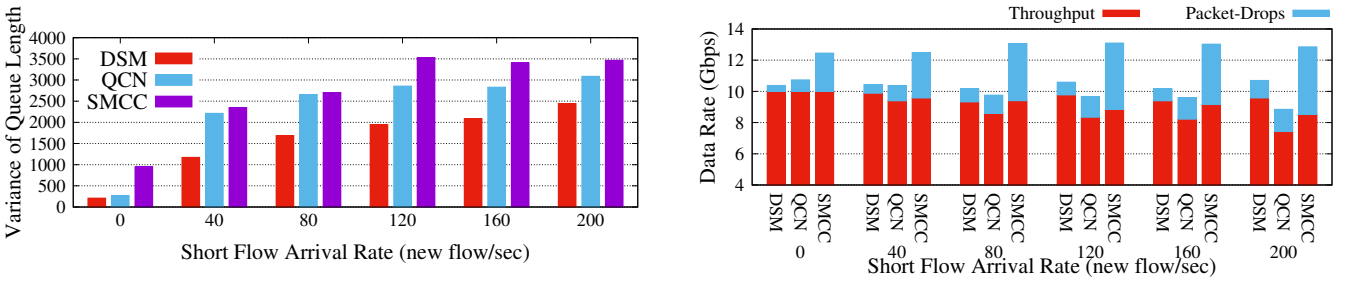
The parking-lot topology illustrated in *Fig.*4 is used to show the performance of DSM in multiple bottlenecks scenario. In this scenario, the capacity of each link is $10Gbps$ and the delay between two hops is $50\mu s$. Specially, we consider the signal traffic pattern of long-lived flows and the mixed traffic pattern reported in [23].

Long-lived Flows: Let S_1 start a flow F_1 at the initial time, S_2 start a flow F_2 at the initial time and finish it at $4s$, S_3 hold a flow F_3 during $[1s, 3s]$, and S4 start a flow F4 at $2s$. The initial speed of each flow is $10Gbps$. The delay and capacity of each link are $50\mu s$ and $10Gbps$, respectively.



(a) DSM (b) QCN (c) SMCC

Fig. 9. Long-lived flows in multiple bottlenecks scenario.



(a) Variance of Queue Length (b) Throughput

Fig. 10. Mixed flows in multiple bottlenecks scenario.

In this scenario, link $C1 \rightarrow C2$ is congested at the initial time, while link $C2 \rightarrow C3$ is surplus. Then a burst caused by F_3 occurs at $1s$ in link $C1 \rightarrow C2$, but link $C2 \rightarrow C3$ is not congested until F_4 starts at $2s$. With F_3 ends at $3s$, F_1 and F_2 share the link $C1 \rightarrow C2$ and compete link $C2 \rightarrow C3$ with F_4. As a result, link $C1 \rightarrow C2$ is surplus while $C2 \rightarrow C3$ is still congested, even after F_2 ends at $4s$. Predictably, link $C1 \rightarrow C2$ is congested during $[0s, 3s]$ and link $C2 \rightarrow C3$ is congested during $[2s, 5s]$. Queue length evolutions with different schemes in this scenario are drawn in *Fig.*9. Obviously, although all the schemes are responsive to the arrival and degrade of flows, DSM achieves more stable queues than QCN and SMCC, regardless of bottleneck shifting.

Mixed Flows: Since a large number of flows in data centers are short flows, it is important to investigate the impact of such dynamic flows on congestion control schemes. We assume that each source holds 5 long-lived flows during the simulation and generates short flows according to a Poisson process. The transfer size for short flows is derived from [23], and the average transfer size is $10Mbit$. We change the generation rate of short flows from 0 to 200 per second, such that 0% to 20% of the bottleneck bandwidth is accounted by the shot flows. *Fig.*10(a) shows the average and standard deviation of the queue length at the two bottlenecks. Obviously, the heavier the short flows, the more violent the oscillation of the queue length. However, DSM is more robust than QCN and SMCC in face of the burst of short flows. As a result, DSM can simultaneously achieve a high utilization and loss less packets, which is verified in *Fig.*10(b).

VI. Conclusion

In this work, we attempt to mitigate the negative impact of heterogeneous and time-varying delays on the congestion control schemes in CE. We build a dynamic model for CE with the standard congestion control framework and obtain a new congestion detector from the model. Leveraging the new congestion detector, we design the Delay-tolerant Sliding Mode (DSM) congestion control scheme. The computing complexity of DSM is only $O(1)$. DSM is excellently stable with delays, always responsive to the changes of available bandwidth, and adaptive for a large range of link speeds and topologies. Simulation results verify these great properties of DSM and show that DSM outperforms both QCN and SMCC in various scenarios, including burst arrival and degrade of flows, high speed link rate, heterogeneous and time-varying delays, multiple bottlenecks topology, and real traffic pattern. Commonly, while both QCN and SMCC encounter frequent under-flow and over-flow accidents, DSM can always achieve 100% network utilization and less than 5% packet drops.

VII. Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by National High-Tech Research and Development Plan of China (863 Plan) under Grant No.2015AA020101, National Natural Science Foundation of China (NSFC) under No.61502539, Suzhou-Tsinghua Special Project for Leading Innovation, and China Postdoctoral Science Foundation under Grant No.2015M582344 and No.2016T90761.

REFERENCES

[1] "Ieee 802.1 data center bridging task group," http://www.ieee802.org/1/pages/dcbridges.html.

[2] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, "Detail: reducing the flow completion time tail in datacenter networks," in *Proc. of SIGCOMM*, 2012.

[3] D. Crisan, R. Birke, G. Cressier, C. Minkenberg, and M. Gusat, "Got loss? get zovn!" in *Proc. of SIGCOMM*, 2013.

[4] M. Kaminsky and D. G. Andersen, "Using rdma efficiently for key-value services," in *Proc. of SIGCOMM*, 2014.

[5] "Ieee 802.1: 802.1qau - congestion notification," http://www.ieee802.org/1/pages/802.1au.html.

[6] "Data center ethernet congestion management: Backward congestion notification," http://www.ieee802.org/3/ar/public/0505/bergamasco_1_0505.pdf.

[7] J. Jiang, R. Jain, and C. So-In, "An explicit rate control framework for lossless ethernet operation," in *Proc. of ICC*, 2008.

[8] M. Gusat, C. Minkenberg, and R. Luijten, "Extended ethernet congestion management (e2cm): Per path ecm-a hybrid proposal," *IBM Research GmbH, Zurich*, 2007.

[9] "Qcn: Quantized congestion notification," http://www.ieee802.org/1/files/public/docs2007/au_prabhakar_qcn_overview_geneva.pdf.

[10] "Unified fabric: Cisco's innovation for data center networks," http://www.cisco.com.

[11] "Qfx3500 switch, datasheet," http://www.juniper.net.

[12] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion control for large-scale rdma deployments," in *Proc. of SIGCOMM*, 2015.

[13] D. Abts and B. Felderman, "A guided tour through data-center networking," *ACM Queue*, 2012.

[14] K. W. Fendick, M. A. Rodrigues, and A. Weiss, "Analysis of a rate-based control strategy with delayed feedback," in *Proc. of SIGCOMM*, 1992.

[15] S. Floyd, "Metrics for the evaluation of congestion control mechanisms," 2008.

[16] Y. Zhang, S. R. Kang, and D. Loguinov, "Delayed stability and performance of distributed congestion control," in *Proc. of SIGCOMM*, 2004.

[17] F. Kelly, G. Raina, and T. Voice, "Stability and fairness of explicit congestion control with small buffers," in *Proc. of SIGCOMM*, 2008.

[18] M. Alizadeh, A. Kabbani, B. Atikoglu, and B. Prabhakar, "Stability analysis of qcn: the averaging principle," in *Proc. of SIGMETRICS*, 2011.

[19] W. Jiang, F. Ren, and C. Lin, "Phase plane analysis of quantized congestion notification for data center ethernet," *IEEE/ACM Transactions on Networking*, 2015.

[20] U. Itkis, *Control systems of variable structure*, 1976.

[21] W. Jiang, F. Ren, R. Shu, and C. Lin, "Sliding mode congestion control for data center ethernet networks," in *Proc. of INFOCOM*, 2012.

[22] "Qcn pseudo code v2.3," http://www.ieee802.org/1/files/public/docs2009/au-rong-qcn-serial-hai-v23.pdf.

[23] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *Proc. of SIGCOMM*, 2010.

APPENDIX

A. Proof of Proposition 1

Proof. When $\delta(k) < 0$ and $\widehat{Q_f}(k) < 0$ (i.e. $StateA$), inequality (20) is equivalent to

$$-a_1(\omega+1)T\widehat{Q_f}(k)-[b_1(\omega+1)T-1]\widehat{Q_v}(k)+(m+\omega+1)T\xi(k-m) > 0$$

Therefore, when $\delta(k) < 0$, $\widehat{Q_f}(k) < 0$ and

$$\begin{cases} a_1 > \left(1+\frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_f}(k)}\right| \\ b_1 \le \frac{1}{(\omega+1)T} \end{cases}, \tag{32}$$

inequality (20) holds.

Similarly, when $\delta(k) > 0$ and $\widehat{Q_f}(k) < 0$ (i.e. $StateB$), inequality (20) is equivalent to

$$-a_2(\omega+1)T\widehat{Q_f}(k)-[b_2(\omega+1)T-1]\widehat{Q_v}(k)+(m+\omega+1)T\xi(k-m) < 0$$

Thus, when $\delta(k) > 0$, $\widehat{Q_f}(k) < 0$ and

$$\begin{cases} a_2 \le 0 \\ b_2 > \frac{1}{(\omega+1)T} + \left(1+\frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_v}(k)}\right| \end{cases}, \tag{33}$$

inequality (20) is true.

With the same method, we can found that once the coefficients a_1, b_1, a_2, b_2 satisfy inequalities (32) and (33), the sliding mode motion sufficient condition (20) is fulfilled. On the other hand, because all the coefficients are nonnegative, there must be $a_2 = 0$ according to inequality (33). Moreover, for simplicity, we also set $b_1 = 0$ to satisfy inequality (32). Subsequently, we will draw the trajectory of system under feedback control u (21) in $StateA$ and $StateB$.

In $StateA$, defining $X(k) \triangleq \begin{bmatrix} \widehat{Q_f}(k) \\ \widehat{Q_v}(k) \end{bmatrix}$ and substituting the congestion control rule u (21) into (17), we have

$$X(k+1) = AX(k) + B\xi(k-m) \tag{34}$$

where

$$A = \begin{bmatrix} 1-a_1T & 0 \\ -a_1T & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} (m+1)T \\ T \end{bmatrix}$$

Note that when $0 < a_1 < \frac{2}{T}$, $A = G^{-1}RG$, where G is a nonsingular matrix and R is a rotation matrix with a clockwise angle of $\arccos\left(\frac{2-a_1T}{2}\right)$, i.e.

$$G = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \sqrt{\frac{1}{a_1T}-\frac{1}{4}} \end{bmatrix}$$

$$R = \begin{bmatrix} \frac{2-a_1T}{2} & \frac{\sqrt{4a_1T-a_1^2T^2}}{2} \\ -\frac{\sqrt{4a_1T-a_1^2T^2}}{2} & \frac{2-a_1T}{2} \end{bmatrix}$$

Hence, the trajectory of system under congestion control rule u (21) follows a clockwise ellipse in $StateA$, as drawn in *Fig.*2.

In $StateB$, let the disturbance $\xi(k-m) \equiv 0$, there is

$$\frac{\widehat{Q_v}(k+1)-\widehat{Q_v}(k)}{\widehat{Q_f}(k+1)-\widehat{Q_f}(k)} = \frac{-b_2T\widehat{Q_v}(k)}{(1-b_2T)\widehat{Q_v}(k)} = -\frac{b_2T}{1-bT} \tag{35}$$

Thus, when $0 < b_2 < \frac{1}{T}$, the trajectory of system under feedback control rule u (21) runs toward the boundary following the straight lines of slope $-\frac{b_2T}{1-b_2T}$ in $State\ B$, as well as drawn in *Fig.*2.

Combine (32)(33) and the above analysis of the trajectory, when coefficients a_1, b_1, a_2, b_2 satisfy

$$\begin{cases} \left(1+\frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_f}(k)}\right| < a_1 < \frac{2}{T} \\ b_1 = 0 \\ a_2 = 0 \\ \frac{1}{(\omega+1)T} + \left(1+\frac{m}{\omega+1}\right)\left|\frac{\xi(k-m)}{\widehat{Q_v}(k)}\right| < b_2 < \frac{1}{T} \end{cases}$$

the system will "slide" along the boundary line $\delta = 0$ under congestion control rule u (21).

□