# Analysing Convergence of Quantized Congestion Notification in Data Center Ethernet

Ran Shu, Jiao Zhang, Fengyuan Ren, Chuang Lin
Dept. of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
Tsinghua Nation Laboratory for Information Science and Technology, Beijing, 100084, China
Email: {shuran, zhangjiao08, renfy, chlin}@csnet1.cs.tsinghua.edu.cn

*Abstract*—**Enhancing Ethernet as the unified data center fabric to concurrently handle the traffic of Local Area Network (LAN), Storage Area Network (SAN), and High Performance Computing (HPC) has attracted much attention. Congestion management is one critical enhancement to fill the performance gap between traditional Ethernet and the unified data center fabric. Currently, Quantized Congestion Notification (QCN) has been approved as the standard congestion management mechanism. However, lots of work pointed out that QCN suffers from the problem of unfairness among different flows. In this paper, we found that QCN could achieve fairness, merely the convergence time to fairness is quite long. Thus, we build a convergence time model to investigate the reasons of the slow convergence process of QCN. The model indicates that the convergence time of QCN can be decreased if RPs have the same rate increase probability or the rate increase step becomes larger at steady state. We validate the precise of our model by comparing with experimental data on the NetFPGA platform. The results show that it well characterizes the convergence time to fairness of QCN. Based on the proposed model, the impact of QCN parameters, network parameters, and QCN variants on the convergence time is analysed. Finally, enlightened by the analysis, we proposed a mechanism, called QCN-T, which replaces the Byte Counter and Timer at sources with a single modified Timer, to reduce the convergence time of QCN.**

*Index Terms*—**Data Center Ethernet, Quantized Congestion Notification, Convergence, Modeling**

## I. Introduction

Recently, using a unified infrastructure to replace LAN, SAN and HPC networks in data centers has attracted much attention [1], [2]. Data Center Ethernet (DCE), also called Converged Enhanced Ethernet (CEE) or Data Center Bridging (DCB), is considered as the predominant choice for the unified infrastructure due to Ethernet's features of easy management, low cost and so on. SANs and HPC networks require lossless and low end-to-end delay. To satisfy these requirements, DCE needs to enhance the performance of traditional Ethernet. Congestion control in DCE, designed by the IEEE 802.1Qau work group, is one critical enhancement [2]. It aims to provide end-to-end congestion management for traffic without congestion control strategies above the link layer, such as Fibre Channel over Ethernet (FCoE), UDP. Also, it is expected to benefit protocols, such as TCP, whose congestion control mechanisms do not perform very well in data centers.

QCN protocol has been ratified to be the standard congestion management scheme for DCE in Mar. 2010 [2]. Many new switches have been designed to support the function of QCN, such as Cisco Nexus 700 [3], FocalPoint FM6000 [4].

978-1-4799-0913-1/14/$31.00 © 2014 IEEE

However, some recent work pointed out that flows could not obtain their fair share of bandwidth in QCN [5]–[7]. The unfairness of QCN will negatively impact the performance of services running over DCE. For example, the MapReduce programming model is widely employed by services in today's data centers [8], [9]. A large task will be partitioned into small jobs and assigned to different workers. The final completion time of the task is determined by the slowest worker. Therefore, if each worker could not get their fair share of bandwidth, the flow completion time of each worker will have large variance and thus the task will be lagged by the sluggish worker.

Some attempt has been made to explore the reasons for the unfairness of QCN. There are two main points. First, a Reaction Point (RP) decreases its rate upon receiving a negative feedback from a Congestion Point (CP). However, a CP transmits each feedback message to a *randomly* selected RP. The random feedback incurs unfairness among RPs. Second, the flows with higher rates have more opportunity to increase their rates [6]. RPs using both Byte Counter and Timer to control rate increase in QCN. If the Byte Counter of a RP shows that the RP has transmitted 150 KB data or the Timer has passed 15 milliseconds, the rate of the RP will increase. Generally the Byte Counter dominates the rate increase in DCE. Thus, the RPs with larger sending rate will increase their rates more quickly.

Based on the two kinds of reasons, some mechanisms are proposed to make QCN more fair. To avoid random feedback, AF-QCN [5] and FQCN [7] are proposed to fairly transmit feedback to each RP. However, these two mechanisms require each switch maintains the information of all the passing flows, which disobeys the design principle that QCN switch does not save information of flows. To avoid RPs with higher rates increase more quickly, the Byte Counter employed in QCN is modified to an adaptive Byte Counter [6]. However, it is difficult to determine a general value of the parameter used in the adaptive Byte Counter.

To substantially solve the problem of long convergence time in QCN at low cost, we need to thoroughly understand the radical reasons for the long convergence time in depth. In this paper, we firstly investigate the whole convergence process of QCN using experiments in a small testbed which is consisted of Dell Servers and NetFPGA. We found that QCN is actually fair. However, the convergence time to fairness is quite long, which possibly exceeds the investigation time in former work. Hence, many researchers stated QCN is not fair. Investigating the experimental data, we conclude that the convergence process of QCN could be partitioned into three stages. The first two stages determine the initial rate values of the third stage, and the third stage will lead to the long convergence time of QCN if the initial rates are not fair. The reasons for unfairness during the first two stages are straightforward. Thus, our model on the convergence time to fairness of QCN mainly characterizes the rate evolution during the third stage. The proposed model indicates that if the probability of increasing rates at RPs is irrespective of the current rate values of RPs or the rate increase value per time could be larger, then the convergence time of QCN can be reduced.

The proposed model is evaluated by comparing with the experimental data on the NetFPGA platform. The results show that the model well characterizes the convergence time to fairness of QCN. Furthermore, the impact of QCN parameters, network configuration and QCN variations on the convergence time of QCN is analyzed.

Enlightened by the experimental investigation and our model analysis, we conclude that using Timer can apparently reduce the convergence time of QCN. Therefore, we propose a mechanism, called QCN-T, which replaces the Byte Counter and Timer in the standard QCN with a single modified Timer to control the rate increase. The experimental results show that the proposed mechanism QCN-T could dramatically decrease the convergent time to fairness in QCN.

The remainder of this paper is organized as follows. Section II introduces the background. We investigate the rate evolution process of QCN through experiments on the NetFPGA platform in Section III. In Section IV, the convergence time model of QCN is described in detail. In Section V, the performance of our model is validated by comparing with experiment results on NetFPGA platform, and the impact of different factors on the convergence time of QCN is analyzed using our model. Finally, the paper is concluded in Section VII.

## II. Background

In this section, we will briefly describe the QCN mechanism, focusing on those parts that are relevant to our analysis. The whole description can be seen in [10] and [11]. QCN is composed of two parts.

- Switch or CP. CP samples packets and generates feedback frames according to the queue length information. Feedback frames are sent to the source of packets directly.
- Rate Limiter or RP. RP decreases its sending rate based on feedback, and probes for available bandwidth by self-increase.

*1) The CP Algorithm:* The goal of CP is to maintain the queue at a desired length $Q_{eq}$. CP samples incoming packets in a period whose duration is related to congestion. Normally the period is the duration of transmitting 150KB data. Let $Q$ denote the current queue length and $Q_{old}$ denote the queue length of last sampling. CP calculates $f_b$ as follows:

$$f_b = -(Q_{off} + w * Q_\delta) \tag{1}$$

where $Q_{off} = Q - Q_{eq}$, $Q_\delta = Q - Q_{old}$, $w$ is a constant weight value. It is set to be 2 in the baseline implementation. Thus, both of the buffer excess and the rate excess are captured. Negative $f_b$ means that there is congestion or congestion is going to happen. The value of $f_b$ is quantized to a 6 bits value $F_b$, and a feedback frame containing $F_b$ will be sent to the source of this sampled packet. If $f_b$ is positive, no feedback frame will be sent.

*2) The RP Algorithm:* RPs decrease their sending rates upon receiving negative feedback frames from CPs. Since there is no positive feedback from CPs to make RPs increase their rates, RPs have a self-increasing algorithm. Thus, rate decreases and rate increases are separated in RP. Let Current Rate ($CR$) denote the sending rate of RP and Target Rate ($TR$) denote the sending rate just before the arrival of the last feedback frame. TR is used to control rates more accurately.

**Rate decreases:** When a feedback frame is received, RPs update $TR$ and $CR$ as follows:

$$\begin{cases} TR = CR \\ CR = CR(1 - G_d F_b) \end{cases} \tag{2}$$

where $G_d$ is chosen so that $G_d F_{bmax} = 0.5$, i.e. $G_d$ is $1/128$ in the baseline implementation.

**Rate increases:** The rate incerase interval is controled by the cooperation of Byte Counter and Timer at RPs. Each cycle of Byte Counter is 150KB and Timer is 15ms in 1Gbps baseline implementation. Each cycle of Byte Counter or Timer leads to a rate increase operation. The rate increase of RPs has three phases:

1) *Fast Recovery (FR)*: After a rate decrease, both Byte Counter and Timer are reset. RP tries to get the lost rate back. At the end of each cycle, $TR$ remains unchanged while $CR$ is updated as follows:

$$CR = \frac{1}{2}(CR + TR) \tag{3}$$

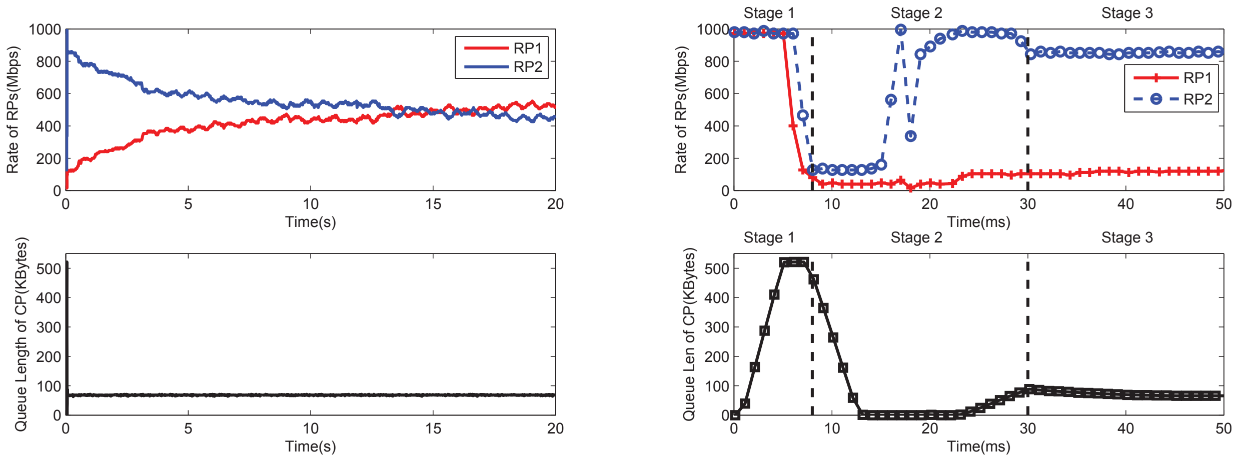
2) *Active Increase (AI)*: With either Byte Counter or Timer larger than 5, RP enters the AI phase to probe for extra bandwidth. The duration of each cycle is cut by half in FR i.e. 75 KB for Byte Counter and 7.5 ms for Timer for a more frequent probing. At the end of each cycle, $TR$ is added by a constant value while $CR$ is updated the same as in FR:

$$\begin{cases} TR = TR + R_{AI} \\ CR = \frac{1}{2}(CR + TR) \end{cases} \tag{4}$$

$R_{AI}$ is set to be 0.5 Mbps in the baseline implementation with 1Gbps link rate and 5 Mbps with 10 Gbps link rate.

3) *Hyper-Active Increase (HAI)*: With a longstanding absence of feedback, RP enters HAI phase. This happens if both of Byte Counter and Timer cycles are larger than 5. In this phase, TR increases more aggressively:

$$\begin{cases} TR = TR + iR_{HAI} \\ CR = \frac{1}{2}(CR + TR) \end{cases} \tag{5}$$



(a) 0-20 seconds.

(b) 1-50 milliseconds.

Fig. 1: Rate evolution of RPs and queue length variation of CP in QCN.

where $R_{HAI}$ is set to 10 times of $R_{AI}$ and i is the times of HAI after the last rate decrease.

There are some extra features in QCN providing better performance when the aggregated rate at CP suddenly drops, such as Extra Fast Recovery and Target Rate Reduction. Due to space limitation, we skip them in this paper. Those interested can refer to [10], [11] for more details.

## III. Experimental Investigation

In this section, we will investigate the rate evolution process of QCN through experiments on the NetFPGA platform [12].

In a dumbbell topology, 2 RPs transmit data to a receiver. The link bandwidth $C = 1$ Gbps, $Q_{eq} = 64$ KB, $Q_{max} = 512$ KB. The values of other parameters are the same as that in the standard QCN, that is, $w = 2, G_d = \frac{1}{128}$, $R_{AI} = 0.5$ Mbps. Since the default sending rate in Ethernet equals the link bandwidth, the initial sending rate of RPs is set to 1 Gbps. The experiment lasts for 20 seconds.

Figure 1(a) shows the variation of sending rates of RPs and queue length of CP during the whole evolution process. We can see that the queue length of CP becomes stable quickly. However, each RP reaches to its fair share of bandwidth after 10 seconds, which is quite long in data centers with many short traffic bursts. It is likely that the traffic of RPs changes before the fair state is reached. Thus, QCN might never reach fairness. The observation duration of existing work is smaller than the convergence time to fairness [5], [6], [13]–[15]. Therefore, it is stated that QCN is not fair. Actually, we can see that QCN is able to reach fairness. But the convergence time to fairness is so large that it is difficult to reach to fairness.

To observe what happens at the beginning of the experiment in more detail, the rates of RPs and queue length of CP during 0-50 ms is enlarged in Figure 1(b). We can see that the queue length of CP quickly increases until the buffer is overfilled, then it becomes empty quickly and at last gradually increases to the equilibrium point. The rates curves of two RPs are very different due to the random feedback during the rate decrease period.

By observing Figure 1(a) and (b), we could classify the evolution process of QCN into three stages, which have different impacts on the convergence time of QCN.

1) RPs inject traffic at the rate of the link bandwidth. Thus, the queue length of CP dramatically increases. Then RPs will decrease their sending rates after receiving negative feedback frames until the queue length of CP becomes empty. During this stage, the rate decrease of different RPs varies because of the random feedback.
2) The rates of RPs increase until the bottleneck bandwidth is fully utilized, and the queue length of CP approximately reaches to the equilibrium point. During this stage, all the feedback is 0. The rate of RPs increases all the time. Since the rate decrease of RPs is different, the time that RPs spend in entering into the HAI phase is different. Once a flow enters into the HAI phase, it can obtain almost all the bandwidth in 2-3 ms. Thus, the rates of RPs become quite different after this stage.
3) The sending rates of different RPs gradually reache to fairness. The duration of the former two phases lasts only about 10-20 ms, while the duration of the third stage exceeds 10 seconds. Therefore, the third stage dominates the convergence time of QCN.

Next we will build a model to characterize the rate variation during the third stage, and deduce the convergence time of QCN. Besides, we will analyze the reasons for long convergence time of QCN and give some insights on how to reduce the convergence time.

## IV. Modeling

In this section, we will build a model on the convergence time of QCN by analyzing the rate variation at every sample interval. First, the main assumptions and notations will be listed, then the model on the convergence time of QCN will be described in detail.

TABLE I: Frequency of different feedback sampled by CP.

| $F_b$ | RP 1 | RP 2 |
|---|---|---|
| 0 | 1947 | 2832 |
| 1 | 794 | 1211 |
| $> 1$ | 30 | 40 |

TABLE II: Key notations in our model

| Not. | Description |
|---|---|
| $G_d$ | System parameter, $1/128$ in baseline implementation |
| $R_{AI}$ | Rate increase value of $TR$ in each avtive Increase period |
| $F_b$ | Average feedback value of feedback frame |
| $R_{dec}$ | Rate change of rate decrease |
| $R_{inc,j}$ | Rate change of rate increase step j |
| $p_{dec}$ | Probability of rate decrease in one sampling period |
| $p_{inc,j}$ | Probability of rate increase step j in one sampling period |
| $CR_0$ | Current Rate before rate decrease |
| $CR_j$ | Current Rate before rate increase step j |
| $TR_j$ | Target Rate before rate increase step j |
| $p_{[Fb>0]}$ | Probability of generate a feedback in one sampling period |

### A. Assumptions and Notations

**Assumptions.** To theoretically analyze the convergence time of QCN, five main assumptions are made based on the features of DCE and QCN. 1) Most of the topologies of data center networks are symmetric [16]–[18]. Thus, we assume that the sources are homogeneous, that is, they pass through paths with the same link bandwidth and round-trip time. 2) Since the feedback frames are directly sent from a CP to RPs and the propagation delay in DCE is quite small, the propagation delay of feedback frames is neglected. 3) The rate increase at RPs is controlled by a Byte Counter and a Timer. When the Byte Counter exceeds 150 KB or the Timer passes 15 milliseconds, the rate increases by a value. In networks with high speed, the Byte Counter plays a leading role. For example, in a DCE with link rate at 1 Gbps, the time of transmitting 150 KB is about 1.2 milliseconds, which is far less than 15 milliseconds. Therefore, in our model, we only consider the Byte Counter and ignore the Timer. Thus, HAI is absent in our model. 4) Since the queue length oscillation at CP is very small in steady state, we assume that the queue is neither empty nor overfilled. Therefore, the sending rate of each link at CPs keeps constant. Correspondingly, the time of transmitting 150 KB data does not change, which indicates that the sampling period at CPs is constant. 5) As the queue length is very stable in steady state, $F_b$ is quite small. We count the $F_b$ value of each sampling period using experiments. As shown in Table I, most of the feedback value is 0 or 1. Since only non-zero feedback will be sent to RP, we assume that all the $F_b$ value received by RPs is 1.

**Notations.** $R_{dec}$ represents the rate decrease value and $R_{inc,j}$ stands for the rate increase at step $j$. $p_{dec}$ and $p_{inc,j}$ is the probability of rate decrease and rate increase at step $j$ during one sampling period, respectively. The main notations are listed in Table II for the sake of terseness and clarity.

### B. Modeling of Convergence Time

There are two challenges in building the model on the convergence time of QCN. First, the asynchronous rate variation among different RPs. Since CP transmits each feedback to a random RP, the rate variation of RPs are not synchronous. The random feedback brings great challenges to the analysis of the rate variation of RPs. In our model, we choose the sampling period at CPs as the round of computing the rate variation since one feedback is generated during one sampling period and the sampling period does not change at the stable state. Second, the rate evolution at RPs are related to not only CR but also TR and the variation of CR and TR is coupled. If we could obtain the relationship between CR and TR at the previous round, we could eliminate TR and simplify the analysis.

We first get the expectation of rate variation during one sampling period at each RP, then iteratively derive the convergence time.

The rates of RPs decrease only upon receiving negative feedback frames. However, the rates of RPs increase in many situations, including FR and AI as stated in Section II. We number every rate increase period. The first five rate increase periods belong to FR phase, the following ones belong to AI phase. The expectation of the rate variation at a RP during one sampling interval is

$$\mathbb{E}(\Delta R) = R_{dec}p_{dec} + R_{inc,1}p_{inc,1} + R_{inc,2}p_{inc,2} \cdots \quad (6)$$

Next we will describe how to obtain the rate decrease value and probability as well as the rate increase value and probability.

**Rate decrease value $R_{dec}$.** Denote the sending rate of this RP as $R$. According to the QCN algorithm, we could obtain the rate decrease during one sample interval is

$$R_{dec} = -G_d F_b R \quad (7)$$

**Probability of rate decrease $p_{dec}$.** At a CP, the ratio of the sampled packets from a RP is proportional to the sending rate of this RP. Thus, in the stable state when the outgoing speed at CP is $C$ bps, the ratio of the sampled packets from a RP with sending rate $R$ is $\frac{R}{C}$. Since only nonzero feedback will trigger rate decrease at a RP, and the probability of generating nonzero feedback is $p_{[fb>0]}$, thus the probability that a RP receive a rate decrease feedback message is

$$p_{dec} = p_{[F_b>0]}\frac{R}{C} \quad (8)$$

In order to save space, we let $p = p_{[F_b>0]}\frac{R}{C}$ which means the probablity of receiving a feedback frame by this RP in a sampling period.

**Rate increase value $R_{inc,j}$.** Let $CR_0$ be the rate value before decrease. $CR_j$ and $TR_j$ represent the CR and TR values before the $j$-th rate increase, respectively. $CR_j$ is related to $CR_{j-1}$ and $TR_{j-1}$. We first obtain the relationship between $CR_j$ and $CR_{j-1}$ by eliminating the items that related to $TR_{j-1}$. Figure 2 depicts the rate evolution of CR and TR.

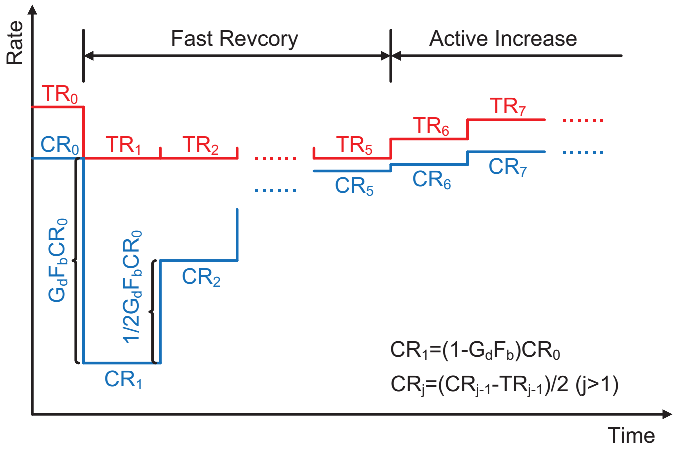


Fig. 2: Rate evolution of $CR$ and $TR$.

We will observe the rate variation rules and then decouple $CR$ and $TR$.

During the FR phase, since each rate increase is related to last rate decrease value, and the rate decrease value as well as the current rate value are related to the rate before decrease, $CR_0$, we could infer the next rate increase value according to current rate value. The relationship between the rate value before the $j$-th FR and $CR_0$ is

$$R = CR_j = (1 - \frac{1}{2^{j-1}} G_d F_b) CR_0 \tag{9}$$

The relationship between the rate increase value and $CR_0$ is

$$R_{inc,j} = \frac{1}{2^j} G_d F_b CR_0 \tag{10}$$

Combing eqs. (9) and (10), we could obtain that

$$R_{inc,j} = \frac{\frac{1}{2^j} G_d F_b R}{1 - \frac{1}{2^{j-1}} G_d F_b}, \quad j = 1, 2, ..., 5 \tag{11}$$

During the AI phase, the rate increase value is

$$R_{inc,j} = \frac{1}{2}(TR_j - R) \tag{12}$$

In which $TR$ increases by $R_{AI}$ every interval during the AI phase, thus

$$TR_j = CR_0 + (j-5) R_{AI} \tag{13}$$

$TR_j$ includes unknown variable $CR_0$. Since the rate increase value is not directly related to current rate value or $CR_0$, we first iteratively compute the relationship between $CR_j$ and $CR_0$, then we could express $CR_0$ using $R$. Current rate value can be expressed as

$$\begin{aligned} R =& CR_j \\ =& \frac{1}{2}(CR_{j-1} + TR_{j-1}) \\ & \vdots \\ =& \frac{1}{2^{j-6}} CR_6 + \sum_{i=1}^{j-6} \frac{1}{2^i} TR_{i-1} \\ =& \frac{1}{2^{j-6}} \left(1 - \frac{1}{2^5} G_d F_b\right) CR_0 + \\ & \sum_{i=1}^{j-6} \left(\frac{1}{2^i} CR_0 + \frac{j-i-5}{2^i} R_{AI}\right) \\ =& \left(1 - \frac{1}{2^{j-1}} G_d F_b\right) CR_0 + \left(j - 7 + \frac{1}{2^{j-6}}\right) R_{AI} \end{aligned} \tag{14}$$

Therefore, we could obtain the relationship between $CR_0$ and $R$

$$CR_0 = \frac{1}{1 - \frac{1}{2^{j-1}} G_d F_b} R - \frac{j - 7 + \frac{1}{2^{j-6}}}{1 - \frac{1}{2^{j-1}} G_d F_b} R_{AI} \tag{15}$$

Combing eqs. (15) and (12), we get that

$$\begin{aligned} R_{inc,j} =& \frac{\frac{1}{2^j} G_d F_b R}{1 - \frac{1}{2^{j-1}} G_d F_b} \\ & + \frac{1}{2} \left(j - 5 - \frac{j - 7 + \frac{1}{2^{j-6}}}{1 - \frac{1}{2^{j-1}} G_d F_b}\right) R_{AI}, \quad j = 6, 7, ... \end{aligned} \tag{16}$$

**The probability of rate increase.** First we summarize the conditions of rate increase. If the $j$-th rate increase occurs during current sampling period, then we can infer that the RP received a nonzero feedback message before the $j$ rounds of rate increase and no feedback message is received by the RP during the $j$ rounds of rate increase. The sending rate at a RP decreases by $G_b F_b CR_0$ during the rate decrease period. The summation of the rate increase during the FR phase is less than $G_b F_b CR_0$. The rate increases by about $R_{AI}$ per time during the AI phase. Since $G_d F_b$ equals 1/128 and $R_{AI}$ equals 0.5 Mbps, rate changes are far less than the sending rate. Therefore, the sending rate of each RP changes slightly during the stable state. When computing the probability of rate increase, we assume that the sending rate between the last rate decrease and current state is $R$, and $p_{[F_b > 0]}$ keeps almost the same as rate changes. Thus, the probability that a RP received a feedback message before the $j$ rounds of rate increase is $p$. Let $T_s$ denote the sampling period of CPs and $T_c$ denote the rate increase interval controled by Byte Counter. Since both the sample period and the rate increase interval during the FR phase is the time of transmitting 150 KB data, the sample time per rate increase interval is

$$= \frac{\frac{T_c}{R}}{\frac{T_s}{C}} = \frac{C}{R} \tag{17}$$

Then during the $j$ rounds of rate increase, there are $j\frac{C}{R}$ sample events. The probability that these samples do not lead to rate decrease at a RP is $(1-p)^{j\frac{C}{R}}$. Therefore, the probability that the $j$-th rate increase occurs during current sample interval is

$$p_{inc,j} = p\,(1-p)^{j\frac{C}{R}}, \quad j = 1, 2, ..., 5 \tag{18}$$

During the AI phase, the rate increase period decreases by half. Correspondingly, the sample interval decreases by half. Thus, we could get that the probability of rate increase during the AI phase is

$$p_{inc,j} = p\,(1-p)^{\frac{j+5}{2}\frac{C}{R}}, \quad j = 6, 7, ... \tag{19}$$

According to the rate variation values and the probability of rate decrease and rate increase, we could get the expected rate variation

$$\begin{aligned}&\mathbb{E}(\Delta R) = -G_dF_bRp+\\&\sum_{j=1}^{5}\frac{\frac{1}{2^j}G_dF_bR}{1-\frac{1}{2^{j-1}}G_dF_b}p(1-p)^{j\frac{C}{R}}+\\&\sum_{j=6}^{\infty}\left(\frac{\frac{1}{2^j}G_dF_bR}{1-\frac{1}{2^{j-1}}G_dF_b}+\frac{1}{2}\left(j-5-\frac{j-7+\frac{1}{2^{j-6}}}{1-\frac{1}{2^{j-1}}G_dF_b}\right)R_{AI}\right)\\&p(1-p)^{\frac{j+5}{2}\frac{C}{R}}\end{aligned}$$

Observing the expression of $E(\Delta R)$, we found that the summation of items could be rewritten as geometric progression if $\frac{1}{2^j}G_dF_bR$ could be approximately treat. Since $G_dF_b = 1/128$, $\frac{1}{2^{j-1}}G_dF_b \ll 1$. We could obtain that

$$1 - \frac{1}{2^{j-1}}G_dF_b \approx 1 \tag{20}$$

The rate increase value during the FR phase is about

$$R_{inc,j} = \frac{1}{2^j}G_dF_bR, \quad j = 1, 2, ..., 5 \tag{21}$$

Similarly, the rate increase value during the AI phase is approximate

$$R_{inc,j} = \frac{1}{2^j}G_dF_bR + \left(1 - \frac{1}{2^{j-5}}\right)R_{AI}, \quad j = 6, 7, ... \tag{22}$$

Thus, the simplified expected rate variation is

$$\begin{aligned}\mathbb{E}(\Delta R) =& -G_dF_bRp+\\&\sum_{j=1}^{5}\frac{1}{2^j}G_dF_bRp(1-p)^{j\frac{C}{R}}+\\&\sum_{j=6}^{\infty}\frac{1}{2^j}G_dF_bRp(1-p)^{\frac{j+5}{2}\frac{C}{R}}+\\&\sum_{j=6}^{\infty}\left(1-\frac{1}{2^{j-5}}\right)R_{AI}p(1-p)^{\frac{j+5}{2}\frac{C}{R}}\end{aligned} \tag{23}$$

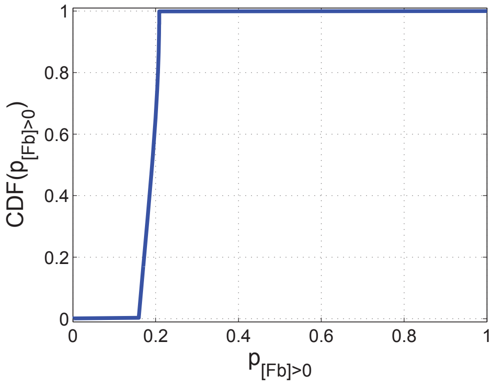


Fig. 3: $p_{[F_b>0]}$ variation.

Through geometric series summation, we obtain that

$$\begin{aligned}\mathbb{E}(\Delta R) =& -G_dF_bRp+\\&\frac{1}{2}G_dF_bRp(1-p)^{\frac{C}{R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{R}}}-\\&\frac{1}{64}G_dF_bRp(1-p)^{\frac{6C}{R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{R}}}+\\&\frac{1}{64}G_dF_bRp(1-p)^{\frac{11C}{2R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{2R}}}+\\&R_{AI}p(1-p)^{\frac{11C}{2R}}\frac{1}{1-(1-p)^{\frac{C}{2R}}}-\\&\frac{1}{2}R_{AI}p(1-p)^{\frac{11C}{2R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{2R}}}\end{aligned} \tag{24}$$

Since $0 < p < 1$ and $p_{[F_b>0]} \approx 0.2$ according to computation, we get that $0 < \frac{1}{2}(1-p)^{\frac{C}{R}} < 1$, $0 < \frac{1}{2}(1-p)^{\frac{C}{2R}} < 1$, and they are in the same order of granularity. The third and fourth items in the above expression is far less than the second item. Thus, the third and fourth items could be ignored. Finally, the expected rate variation is

$$\begin{aligned}\mathbb{E}(\Delta R) =& -G_dF_bRp+\\&\frac{1}{2}G_dF_bRp(1-p)^{\frac{C}{R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{R}}}+\\&R_{AI}p(1-p)^{\frac{11C}{2R}}\frac{1}{1-(1-p)^{\frac{C}{2R}}}-\\&\frac{1}{2}R_{AI}p(1-p)^{\frac{11C}{2R}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{C}{2R}}}\end{aligned} \tag{25}$$

In the final expression of the expected rate variation, there is still an unknown variable, $p_{[Fb>0]}$, whose value changes with the network congestion status. In the stable state, the queue length varies slightly, but is related to the sending rate of RPs. Since the queue length is close to the equilibrium point during most time of the stable state, we could assume that the summation of rate variations of all the RPs is zero, that is $\mathbb{E}(\sum \Delta R_i) = 0$. Besides, in the stable state, the summation of rates of all the RPs $\sum R_i = C$. Therefore, for any $n$ variables, $R_1, R_2, ..., R_n$ that satisfy $\sum R_i = C$, we could get

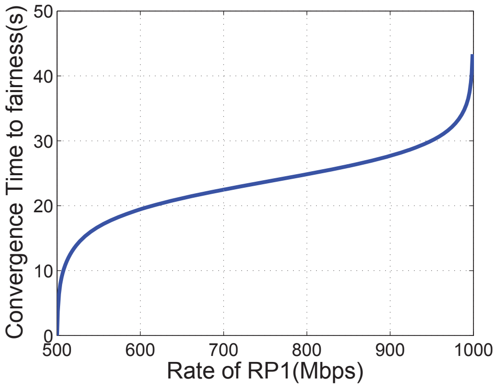


Fig. 4: Convergence time with different initial sending rates.

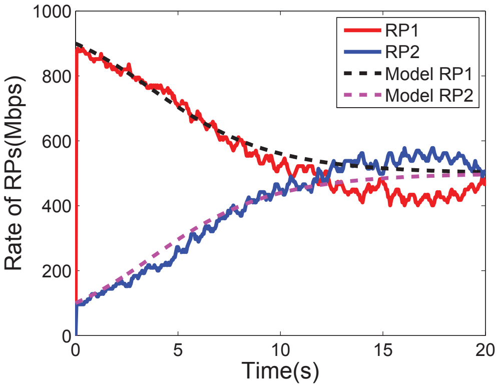


Fig. 5: Rate evolutions with the initial rate $(R_1, R_2) = (900, 100)$ Mpbs.

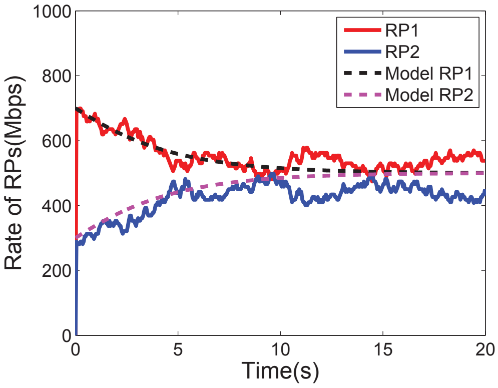


Fig. 6: Rate evolutions with the initial rate $(R_1, R_2) = (700, 300)$ Mpbs.

the probability of nonzero feedback, $p_{[F_b>0]}$, then we could get $\mathbb{E}(\Delta R_i)$ and thus infer the convergence time. Since it is difficult to get the analytical solution of $p_{[F_b>0]}$, we will use Matlab to compute the numerical result.

The dimension of the initial values is $N-1$. The corresponding dimension of the solution is also $N-1$. To explore the convergence feature, we consider $N=2$.

Let the link bandwidth $C = 1$ Gbps. Other parameters follow the definitions in the QCN mechanism, $G_d = 1/128$ and $R_{AI} = 0.5$ Mbps. Besides, according to the assumptions, we have $F_b = 1$. Let $R_1$ denote the sending rate of the RP with higher initial rate. The CDF of $p_{[F_b>0]}$ with different $R_1$ is shown in Figure 3. We can see that the sending rate poses little impact on $p_{[F_b>0]}$, which indicates that the assumptions made to compute the rate increase value is reasonable.

The expected rate variation could be computed based on the value of $p_{[F_b>0]}$. According to the $\mathbb{E}(\Delta R_i)$ values, we could obtain the relationship between the convergence time and the initial rate values as shown in Figure 4.

## V. Validation and Analysis

### A. Experimental Validation

In this subsection, the accuracy of our model is evaluated through experiments on the NetFPGA platform.

The convergence process of QCN around the equilibrium point is usually quite slow, we care more about the convergence process to the equilibrium point. Therefore, we use rate evolution curves to show the convergence process. We set the initial sending rate for each flow and compute the rate evolution process. Observing Figure 4, we found that $R_1$ converges fast from 600 Mbps to 900 Mbps. Thus, we select two initial rates to represent different cases, $(R_1, R_2) = (900, 100)$ Mbps, and $(R_1, R_2) = (700, 300)$ Mbps. Two experiments are conducted.

Figure 5 draws the experimental rate variations and the model results with the initial rates $(R_1, R_2) = (900, 100)$ Mbps. We can see that our model well characterizes the convergence of QCN. However, the model is not quite accurate around the equilibrium point. This is because our model computes the expected rate variation. However, there are many stochastic factors around the equilibrium point. Besides, the figure shows that two RPs converge to their fair share of bandwidth in about 12 seconds, which is too large compared with the small round trip time of tens of microseconds. Therefore, QCN is indeed not fair in small time scale.

Figure 6 plots the experimental rate variations and the model results with the initial rates $(R_1, R_2) = (700, 300)$ Mbps. Our model well matches the experimental data. Compared with the results shown in Figure 5, the convergence time is much smaller.

### B. Analysis

In this subsection, we will analyze the impact of QCN parameters, mechanism modification, network parameters on the convergence time of QCN based on our model. By investigating the experimental data of QCN, we found that the initial rates of RPs are generally $(900, 100)$ Mpbs at the beginning of the third stage (as shown in Figure 1). Therefore, we choose $(R_1, R_2) = (900, 100)$ Mbps as the initial rates in all the analysis. Besides, we define $\varepsilon = \frac{\min_{i=1}^{N} x_i(t)}{\max_{i=1}^{N} x_i(t)}$ and draw the line of $\varepsilon = 0.9$ in the following figures to clearly see when the sending rates are close to fairness with different parameters

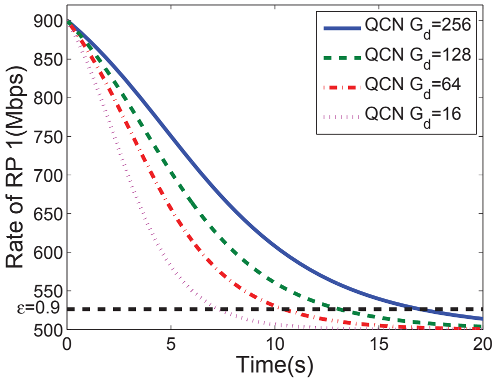


Fig. 7: Rate evolution with different $G_d$.

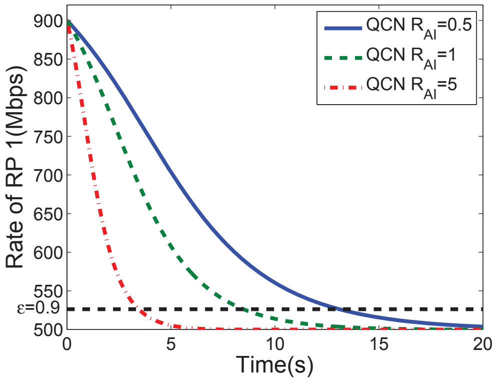


Fig. 8: Rate evolution with different $R_{AI}$

or mechanisms.

*1) QCN Parameters:* Related parameters in QCN will be analyzed together.

**Impact of $G_d$ and $F_b$.** We choose $G_d = 1/16$, $G_d = 1/64$, $G_d = 1/128$ (standard) and $G_d = 1/256$ and get the rate evolution curves as shown in Figure 7. As $G_d$ increases, the convergence time decreases. But the impact of $G_d$ is slight. Even when $G_d = 1/16$, the convergence time only decreases by half. Note that our analysis only applies to the stable state. If $G_d$ is changed, the performance of QCN will be greatly affected at other stages since the value of $G_d$ determines the range of rate decrease.

Then we consider the impact of $F_b$. If the queue length oscillates dramatically in the stable state, the assumption that $F_b$ is either 0 or 1 is not reasonable. We could not compute the probability for every possible $F_b$ value. Thus we only consider the average value of $F_b$. As the queue length oscillation grows, the average of the nonzero feedback increases. Due to the relationship between $G_d$ and $F_b$, increasing $F_b$ will lead to similar results as increasing $G_d$. Therefore, more dramatic queue length oscillation will lead to smaller convergence time.

**Impact of $R_{AI}$ and $T_{AI}$.** Let $R_{AI} = 0.5$ Mbps, $R_{AI} = 1$ Mbps and $R_{AI} = 5$ Mbps and the rate evolution curves are depicted in Figure 8. We can see that as $R_{AI}$ increases, the convergence time dramatically decreases, that is, if the rate increase step during the AI phase becomes larger, then QCN can converge to fairness more quickly.

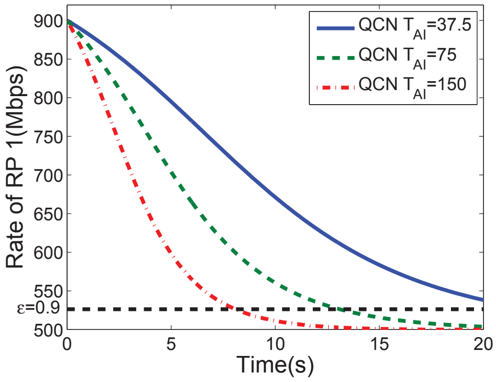


Fig. 9: Rate evolution with different $T_{AI}$.

Let $T_{AI}$ denote the sample interval during the AI phase. We choose several typical values, $T_{AI} = 37.5$ KB, $T_{AI} = 75$ KB and $T_{AI} = 150$ KB. The rate evolution results are depicted in Figure 9. The impact of $T_{AI}$ is opposite to that of $R_{AI}$.

*2) Mechanism Modification:* We will investigate the impact of some modified QCN mechanisms on the convergence time in this subsection.

**QCN-TIMER.** If RP only uses the Timer to trigger rate variation. In different phases, the rate variation value is the same as that using Byte Counter. But the probability of rate increase is different since the duration of one rate increase period changes. The $j$-th rate increase during the FR phase at a RP occurs only if none of the $j\frac{C}{R}$ times of feedback is transmitted to the RP. If only Timer is used, the $\frac{T_c}{R}$ is changed to $T_t$ in eq. (17) where $T_t$ is the rate increase interval controled by Timer. The number of feedback messages that are not received is $j\frac{CT_t}{T_s}$. During the AI phase, the number becomes $\frac{j+5}{2}\frac{CT_t}{T_s}$. Thus, the expected rate variation is

$$\begin{aligned}\mathbb{E}(\Delta R) = &- G_dF_bRp + \\ &\frac{1}{2}G_dF_bRp(1-p)^{\frac{CT_t}{T_s}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{CT_t}{T_s}}} + \\ &R_{AI}p(1-p)^{\frac{11CT_t}{2T_s}}\frac{1}{1-(1-p)^{\frac{CT_t}{2T_s}}} - \\ &\frac{1}{2}R_{AI}p(1-p)^{\frac{11CT_t}{2T_s}}\frac{1}{1-\frac{1}{2}(1-p)^{\frac{CT_t}{2T_s}}}\end{aligned} \tag{26}$$

Since Timer in QCN is only an assisted mechanism. The period is relatively long. Next we will investigate the impact of Timer with not only the standard value but also with smaller ones. If $T_t = 15$ milliseconds, $\sum\mathbb{E}(\Delta R_i) = 0$ does not always have a solution, that is, the so slow rate increase interval could not guarantee the stability of QCN. Thus, we change the standard $T_t$ to 10 milliseconds, which is the referred value before the QCN mechanism is determined [10]. Also, we let $T_t = 2.4$ milliseconds, which leads to similar

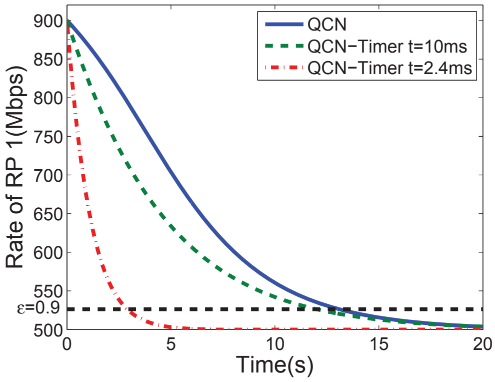


Fig. 10: Rate evolution with different Timer periods.

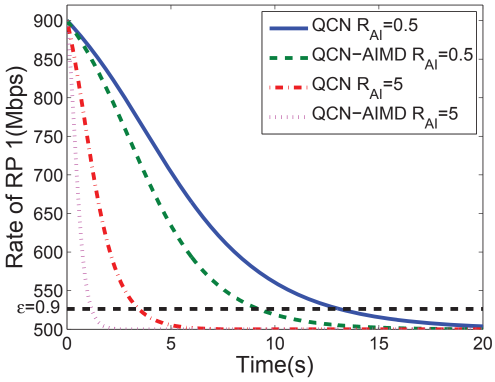


Fig. 11: Rate evolution of QCN-AIMD.

rate increase interval as the Byte Counter. Thus, we could compare the impact of Timer and Byte Counter fairly. The rate evolution curves with different $T_t$ are drew in Figure 10. We can see that Timer improves the convergence speed of QCN. When $T_t = 2.4$ milliseconds, the convergence time of QCN is reduced by several times. Even the standard Timer parameter also reduces the convergence time of QCN.

**QCN-AIMD.** We change the rate increase mechanism of QCN to AIMD, which is called QCN-AIMD. Besides, the FR period is skipped. A RP enters into the AI phase after rate increase period. The rate increase mechanism is

$$CR = CR + R_{AI} \tag{27}$$

The parameters of QCN-AIMD are set according to the definitions in [19]. The rate increase period is also the duration of sending 150 KB data. By eliminating the items related to the FR phase and change the probability of rate increase during the AI phase, we could obtain the expected rate variation

$$\mathbb{E}(\Delta R) = -G_d F_b R p + R_{AI} p (1-p)^{\frac{C}{R}} \frac{1}{1-(1-p)^{\frac{C}{R}}} \tag{28}$$

Let $R_{AI} = 0.5$ Mbps and $R_{AI} = 5$ Mbps, the rate evolution are depicted in Figure 11. For the same $R_{AI}$, the AIMD mechanism could speed up the convergence process. But the impact is slight compared with the impact of increasing $R_{AI}$.

*3) Network Parameters:* In this subsection, we will consider the impact of network parameters on the fairness of QCN.

**Link bandwidth.** We first explore the impact of link bandwidth. In our model, all the rate variations are triggered by the Byte Counter. Given a determined byte threshold, the convergence time will reduce to be $1/m$ when the link bandwidth is amplified to be $m$ times of the original value.

**Round trip time.** If we consider the round trip time, then the feedback message will suffer some latency. In our model, the feedback latency will neither affect the rate variation nor affect the probability of the rate variation. However, as the round trip delay increases, the queue length will become more oscillated. If the queue length still satisfies the fourth assumption, the convergence time will slightly decrease due to larger feedback values.

## VI. Speeding Up Convergence

Mechanisms could be designed to reduce the convergence time of QCN based on the features of the three phases. The first two phases determine the initial rate values of the third phase. Making flows more fair at the start of the third phase or reducing the convergence time during the third phase can both speed up the convergence to fairness of QCN.

During the first phase, the random feedback leads to the unfairness of different flows. If a mechanism is designed to mitigate the unfairness of QCN during this phase, then the feedback should be fairly send to each RP. However, this method either increases the complexity of CP or increase the control overhead of frames, which is not practical.

During the second phase, rate variations are only decided by RPs. Since the HAI phase is entered when the Byte Counter passes 5 cycles as well as the Timer is triggered for 5 times. When $T_c > R \times T_t$, that is, $R < 80$ Mbps, whether a RP will enter into the HAI phase only depends on the Byte Counter. The experiment results indicate that it is likely to happen that the rate of a RP, $R$, is smaller than 80 Mpbs at the start of the second phase. Therefore, the Byte Counter will dominate whether a RP could enter into the HAI phase. Obviously, the RPs with larger rates will enter into the HAI phase much earlier, and then quickly take up almost all the remaining bandwidth. To avoid unfairness during this phase, the conditions of entering the HAI phase and the frequency of increasing rates during the HAI phase should be independent of the sending rates of RPs. Using Timer to control rate increase can satisfy these requirements.

During the third phase, according to the analysis in Section V-B, increasing $R_{AI}$ and using QCN-Timer could dramatically decrease the convergence time in QCN.

Based on the above analysis, we can see that using Timer to trigger rate change will reduce the convergence time of QCN. Thus, we proposed a QCN variation, called QCN-T, which gets rid of the Byte Counter at RPs and only keeps the Timer in the standard QCN. The Timer period in QCN-T is changed to $t$ milliseconds. The threshold of the FR and AI phases does not change. The threshold of entering the HAI phase is changed from 5 cycles to $\frac{15\ \mathbf{ms}\times 5}{t}$ cycles. This is because the Timer

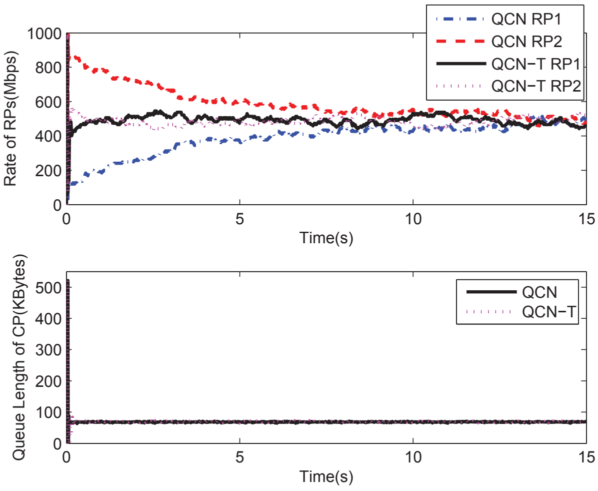


Fig. 12: Convergence time of QCN and QCN-T.

period in the standard QCN is 15 milliseconds. In a network with link capacity $C = 1$ Gbps, the Timer period in QCN-T, $t$, could be set to 2.4 milliseconds. This is because the sampling interval at CP is the duration of transmitting 150 KB data. Thus, if there are two RPs, the fair share bandwidth of each RP is 500 Mbps, then the each RP requires to spend $\frac{150 \times 8}{500} = 2.4$ milliseconds. If there are more RPs, then rates at RPs will vary more frequently, which does not affect the fairness of QCN-T.

We implemented the proposed QCN-T mechanism in our testbed and conducted experiments in the same scenario as that in Section III. Figure 12 shows the results of QCN and QCN-T, we can see that QCN-T significantly decreases the convergence time of QCN. Besides, the queue length is still stable in QCN-T.

## VII. Conclusion

The congestion management mechanism, QCN, plays a critical role in DCE which enhances the performance of traditional Ethernet to be a unified data center fabric. However, it is pointed out that QCN suffers the problem of unfairness. Through investigating experimental data on the NetFPGA platform, we found that QCN can achieve fairness, but the convergence time to fairness is quite large. To thoroughly understand why the convergence of QCN is slow, a model on the convergence time of QCN is built. The proposed model shows that the convergence time of QCN can be decreased if RPs have the same probability of rate increase or the rate increase step becomes larger. By comparing with experimental data, the proposed model is proved to well characterize the convergence time of QCN. Also, the impact of different QCN and network parameters on the convergence time of QCN is analyzed. At last, a mechanism, called QCN-T, is proposed to speed up the convergence of QCN.

## Acknowledgment

The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by National Basic Research Program of China (973 Program) under Grant No. 2014CB347800 and 2012CB315803, and National Natural Science Foundation of ChinaNSFCunder Grant No. 61225011

## References

[1] Cisco, "Unified Fabric: Cisco's Innovation for Data Center Networks," in *http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns783/white_paper_c11-462422.pdf*, (San Jose, CA, USA), 2004.

[2] "IEEE 802.1 Data Center Bridging Task Group," in *http://www.ieee802.org/1/pages/dcbridges.html*.

[3] B. Chia, "Data Center Technology Update," in *http://www.cisco.com/web/SG/learning/dc_partner/files/Nexus_Update.pdf*, 2010.

[4] "Fulcrum Announces 1 Billion Packet per Second 10G/40G Ethernet Switch Chips for Efficient Scaling of Virtualized Data Center Networks," in *http://www.businesswire.com/news/home/20101101006001/en/Fulcrum-Announces-1-Billion-Packet-10G40G-Ethernet*, Nov. 2010.

[5] A. Kabbani, M. Alizadeh, M. Yasuda, R. Pan, and B. Prabhakar, "AF-QCN: Approximate Fairness with Quantized Congestion Notification for Multi-Tenanted Data Centers," in *High Performance Interconnects (HOTI), 2010 IEEE 18th Annual Symposium on*, pp. 58–65, IEEE, 2010.

[6] Y. Hayashi, H. Itsumi, and M. Yamamoto, "Improving Fairness of Quantized Congestion Notification for Data Center Ethernet Networks," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pp. 20–25, IEEE, 2011.

[7] Y. Zhang and N. Ansari, "On Mitigating TCP Incast in Data Center Networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 51–55, IEEE, 2011.

[8] J. Dean, S. Ghemawat, and G. Inc, "MapReduce: Simplified Data Processing on Large Clusters," in *USENIX OSDI*, 2004.

[9] M. Alizadeh, A. Greenberg, D. A. Maltz, and J. Padhye, "Data Center TCP ( DCTCP )," in *ACM SIGCOMM*, pp. 63–74, 2010.

[10] M. Alizadeh, B. Atikoglu, A. Kabbani, A. Lakshmikantha, R. Pan, B. Prabhakar, and M. Seaman, "Data Center Transport Mechanisms: Congestion Control Theory and IEEE Standardization," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pp. 1270–1277, IEEE, 2008.

[11] R. Pan, "QCN Pseudo Code Version 2.1," in *http://www.ieee802.org/1/files/public/docs2008/au-pan-qcn-serial-hai-2-1-0408.zip*.

[12] "NetFPGA Project," in *http://netfpga.org/*.

[13] P. Devkota and A. N. Reddy, "Performance of Quantized Congestion Notification in TCP Incast Scenarios of Data Centers," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, pp. 235–243, IEEE, 2010.

[14] A. S. Anghel, R. Birke, D. Crisan, and M. Gusat, "Cross-Layer Flow and Congestion Control for Datacenter Networks," in *Proceedings of the 3rd Workshop on Data Center-Converged and Virtual Ethernet Switching*, pp. 44–62, ITCP, 2011.

[15] Y. Zhang and N. Ansari, "Fair Quantized Congestion Notification in Data Center Networks," *IEEE TRANSACTIONS ON COMMUNICATIONS*, vol. 61, no. 11, 2013.

[16] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," in *ACM SIGCOMM*, pp. 63–74, 2008.

[17] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A Scalable and Flexible Data Center Network," in *ACM SIGCOMM*, 2009.

[18] C. Guo, H. Wu, K. Tan, L. Shiy, Y. Zhang, and S. Luz, "Dcell : A Scalable and Fault-Tolerant Network Structure for Data Centers," in *ACM SIGCOMM*, pp. 75–86, 2008.

[19] M. Alizadeh, A. Kabbani, B. Atikoglu, and B. Prabhakar, "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.