Modeling and Understanding TCP's Fairness Problem in Data Center Networks

Shuli Zhang[1], Yan Zhang[1], Yifang Qin[1], Yanni Han[1], Song Ci[1,2]
[1]High Performance Network Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
[2]Department of CEEN, University of Nebraska-Lincoln, NE 68182, USA
Email:{zhangshl, zhangy, qinyf, hanyn, sci}@hpnl.ac.cn

***Abstract*—Due to the special topologies and communication pattern, in today's data center networks it is common that a large set of TCP flows and a small set of TCP flows get into different ingress ports of a switch and compete for a same egress port. However, in this case the throughput share of flows in the two sets will not be fair even though all flows have the same RTT. In this paper, we study this problem and find that TCP's fairness in data center networks is related with not only the network capacity but also the number of flows in the two sets. We propose a mathematical model of the average throughput ratio of the large set of flows to the small set of flows. This model can reveal the variation of TCP's fairness along with the change of network parameters (including buffer size, bandwidth, and propagation delay) as well as the number of flows in the two sets. We validate our model by comparing its numerical results with simulation results, finding that they match well.**

Keywords—TCP's fairness; data center networks; model; network capacity

I. INTRODUCTION

As the cost-effective infrastructure for cloud computing, today's data center networks (DCNs) are playing an increasingly important role for cloud services and applications. Compared with traditional wired or wireless networks, DCNs have some unique features, including multipath, small propagation delay, special communication patterns of many-to-one or many-to-many [1] and common use of low-cost commodity switches [2]. These features pose great challenges for TCP, which is originally designed for traditional network environments but occupies over 90% of the traffic in DCNs [2]. Recently, TCP's emerging problems in DCNs such as goodput collapse [3, 4], large latency [2, 5] and fairness problem [6] have attracted a lot of attentions.

TCP's fairness problem in DCNs was first reported in [6] as follows. When a large set of TCP flows and a small set of TCP flows arrive at different ingress ports of a switch and compete for a same egress port, the large set of flows can obtain significantly higher throughput on average than the small set, even though all flows have the same RTT. The author named this problem as TCP Outcast. In our study, we also focus on TCP's fairness problem in DCNs. We rebuild the simulations presented in [6] and find that TCP assuredly can't achieve good fairness between the two sets of TCP flows. However, we find some different results from those in [6]. First, when the number of flows in the two sets is small, the average throughput of the large set can be either higher or lower than that of the small set. Second, when the number of flows in the two sets is large, the average throughput of the large set will be much lower than that of the small set.

This work is supported in part by National Science Foundation of China (Grant No. 61303250 and 61303252).

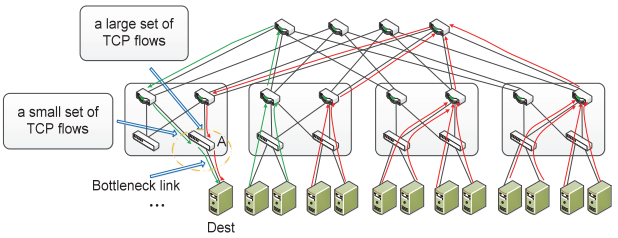

Fig. 1. An example of TCP's fairness problem scenario in data center networks

It is noteworthy that, since multi-rooted tree-based topologies [7] and many-to-one communication pattern (e.g., Map-Reduce [8]) widely exist in DCNs, it is common that a large set of TCP flows and a small set of TCP flows from two different ingress ports compete for a same egress port at the switch near the destination. For example, as shown in figure 1 which adopts a Fat-Tree topology, when 12 TCP sources synchronously send packets to the destination, 10 flows and 2 flows will compete at switch *A*.

In this paper, we first present our observations and analysis about TCP's fairness problem in DCNs. Results of our simulations indicate that TCP's fairness is related with not only the network capacity but also the number of flows in the large set and the small set. We explain TCP's fairness problem in DCNs by two reasons. One is the difference in the amount of bursty traffic between the large set of flows and the small set of flows. The other is that packets of different TCP flows within a set are not adequately mixed. According to our observations and analysis, we propose a mathematical model for TCP's fairness in DCNs, which is the main contribution of this paper. This model can reveal the variation of TCP's fairness along with the change of the network capacity and the number of flows in the two sets. We carry out comprehensive simulations and prove that our fairness model can match the simulation results well. To the best of our knowledge, there has not been any other work on modeling TCP's fairness problem in DCNs.

The rest of this paper is organized as follows. Our observations and analysis on TCP's fairness problem in DCNs are presented in Section II. Then we deduce the model for TCP's fairness in DCNs in Section III. Section IV compares the numerical results of our fairness model with the simulation results under different scenarios. Finally, we conclude this paper in section V.

II. TCP'S FAIRNESS PROBLEM IN DATA CENTER NETWORKS

According to our study, the main reason for TCP's fairness problem in DCNs is that consecutive packet losses have different effects on the large set of TCP flows and the small set of TCP flows. Specifically, this difference is determined by the amount of bursty traffic and how the packets of different TCP flows within a set are mixed. First, since the number of flows in the large set is greater than that in the small set, the large set of flows can make more amount of bursty traffic at its ingress port, which will lead to more consecutive packet losses if a loss event occurs. Second, through our observations, we find that packets of flows in the small set are mixed relatively adequately, while packets of flows in the large set are mixed inadequately. So, when flows in the small set lose packets, the amount of packet losses will be relatively small and these dropped packets will be uniformly distributed in each TCP flow in the small set. Thus, all flows in the small set will almost simultaneously lose packets and get into the Fast Recovery (FR) stage. On the other hand, when flows in the large set lose packets, the amount of packet losses will be relatively large and these dropped packets will be distributed only in part of these flows. That is, some TCP flows may lose plenty of packets and unluckily experience timeout (TO). Table I shows the number of different loss events that occur within 5 seconds in one of our simulations (simulation topology will be introduced in Section III), where M is 2 and N is 12. From Table I we can see that, all loss events of flows in the small set cause FR, while some loss events of flows in the large set cause TO. That is, flows in the large set are more likely to experience TO than those in the small set.

TABLE I. NUMBER OF LOSS EVENTS WITHIN 5 SECONDS

Flow	Number of Loss Events	
	TO	***FR***
flow 1 in the small set	0	1816
flow 2 in the small set	0	1804
flow 1 in the large set	12	838
flow 2 in the large set	14	789
flow 3 in the large set	11	1013
flow 4 in the large set	13	903

As we know, in DCNs, since RTT is nearly one percent or thousandth of TCP's RTO_{min} (i.e. 200ms), TCP timeout can degrade throughput severely. However, although flows in the large set are more likely to experience TO than flows in the small set, it does not mean that the average throughput of the large set will be always lower than that of the small set. In fact, the difference between their average throughput is related with the network capacity, which is defined as the switch buffer size plus bandwidth-delay product (BDP). If the network capacity is limited, the average window size of all TCP flows will be relatively small. This means that the flows not experiencing TO in the large set cannot increase their window sizes high. So the negative effect of timeouts on some flows in the large set will be obvious, and the average throughput of flows in the large set will be lower than that of flows in the small set. On the contrary, if the network capacity is large, the average window size of all TCP flows will be relatively large. This means the following two points. First, the flows not experiencing TO in the large set can increase their window sizes to a high extent. Second, when consecutive packet losses occur, a larger window size will be less likely to trigger timeout than a smaller window size, so the events of TO will be fewer for all flows in the large set. Meanwhile, flows in the small set are still likely to get into the FR stage at each loss event. So, when the network capacity is large, the negative effect of timeouts on some flows in the large set will not be obvious, and the average throughput of flows in the large set can be higher than that of flows in the small set. Figure 2 shows the window evolutions of some TCP flows in the small set and the large set in our simulations when the network capacity changes, which can represent the above descriptions.

According to the above discussion, we know that the average throughput of the large set of flows can be either higher or lower than that of the small set of flows. However, this view is only tenable when the number of flows in the two sets is small. When the number of flows in the two sets is large, we observe that the average throughput of the large set will be much lower than that of the small set, even with plenty of network capacity. This is because the increment of the number of flows in the large set will make flows in the large set suffer TO events more frequently (details about this reason will be revealed in our modeling work in the next section). So the negative effect of TO events on the throughput of flows in the large set will be more significant. We observe that, increasing the network capacity may alleviate this problem, but is still hard to make the average throughput of the two sets close.

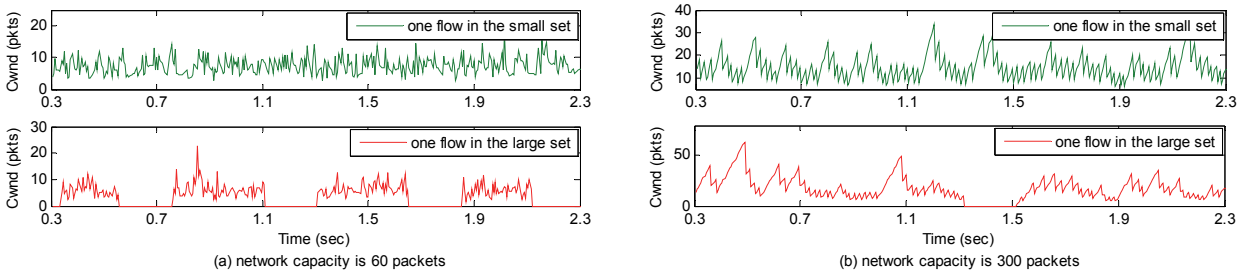

Fig. 2. The window evolutions of some TCP flows in the small set and the large set with different network capacity

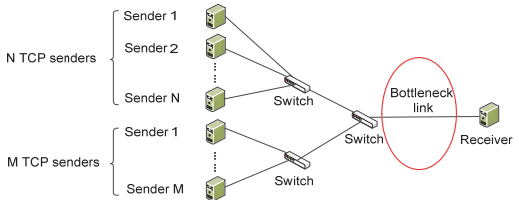

Fig. 3. The topology used in our model and simulations

III. Modeling TCP's Fairness in Data Center Networks

A. Assumptions and Notations

The topology used in our model and simulations is shown in figure 3. In this topology, lots of servers simultaneously use TCP connections to transmit data to a same receiver through a common bottleneck link. As in [9], the TCP version used in our study is NewReno, which is widely used in practice. All switches in the topology use DT queue management scheme. Also as in [9], we assume that the packets will be dropped only due to buffer overflow at the bottleneck link, rather than other causes such as random error, link failure, etc.

It is noteworthy that, in DCNs, TCP will show unfairness when a large set of TCP flows and a small set of TCP flows from two different ingress ports of a switch compete for a same egress port, no matter what the realistic topology is. So, figure 3 is an effective abstraction of the topologies in which TCP's fairness problem exists. Similar method of topology abstraction can be found in research works on TCP Incast problem in DCNs [3, 4, 9], where a simple fan-in topology is used to make simulations and investigations.

Table II summarizes the key notations used in our model. There are N TCP flows in the large set and M TCP flows in the small set respectively, where N is larger than M. We assume that the propagation delay between each sender and receiver, the bandwidth of each link, and the buffer size of each switch are all the same.

We use F_i (i=1, 2, …, N) and f_j (j=1 , 2, …, M) to represent the i-th TCP flow in the large set and the j-th TCP flow in the small set respectively. We assume that each flow in the small set shares the same window evolution. In specific, f_j will stay in the Congestion Avoid (CA) stage until a loss event occurs, and each loss event will make f_j enter into a new CA stage again after FR. So, we define the duration of a CA stage of f_j as a base cycle, where we regard the FR stage as part of a CA stage. We assume that there exists a steady state where the window evolution of f_j is the same in each base cycle (Figure 2 can prove this state approximately exists). So, in our model we can use W^f to denote the max window size of f_j in the steady state.

As we know, given the above assumptions, F_i may stay in one of three different states after each base cycle. First, it will keep on staying in the previous CA stage if it has no packet loss. Second, it will experience FR and start a new CA stage if it loses only a small number of packets. Third, it will be unlucky to enter into a TO stage if it loses a great number of packets. So, the window evolution of each flow in the large set will not be synchronous. However, approximately, we can assume that each flow in the large set shares the same P^{loss} and P^{TO}. Thus they will have the same trend of window evolution. We assume that, in the steady state of f_j, F_i will also be in a steady state. So, we can use a normalized curve to depict the window evolutions of flows in the large set, although their window evolutions are not synchronous. In our model, we use W^F to indicate the max window size of F_i in the steady state.

TABLE II. Key Notations in Our Model

Not.	Description
C	Link bandwidth, counted by packets per second
B	Switch buffer size, counted by packets
D	Propagation delay between each sender and receiver
T^O	Duration of a timeout stage, equals to RTO_{min}
N	The number of flows in the large set of TCP flows
M	The number of flows in the small set of TCP flows
W^F	Max window size of F_i in the steady state
W^f	Max window size of f_j in the steady state
U	The average total number of dropped packets of all flows in the large set at each loss event
X_i	The average number of packets from F_i which enter into the switch consecutively
x_i	The number of dropped packets of F_i when F_i loses packets
P^{loss}	Conditional probability that F_i loses packets at each loss event
P^{TO}	Conditional probability that F_i will enter into a TO stage when F_i loses packets
Y^F	Expected number of packets transmitted by F_i in a window evolution cycle of F_i
T^F	Expected duration of a window evolution cycle of F_i
Y^f	Expected number of packets transmitted by f_j in a window evolution cycle of f_j
T^f	Expected duration of a window evolution cycle of f_j
$R^{F,f}$	Average throughput ratio of the large set of TCP flows to the small set of TCP flows

The steady state assumption and related notations are shown in figure 4. As in [9], we ignore the unabiding slow start stage. From figure 4 we can see that, a window evolution cycle of F_i contains a TO stage and several consecutive CA stages in which the max window size of F_i is W^F, while a window evolution cycle of f_j is the duration of a CA stage (i.e., a base cycle) in which the max window size of f_j is W^f.

In our model, we define $R^{F,f}$, the average throughput ratio of the large set of TCP flows to the small set of TCP flows as TCP's fairness index in DCNs. Next, we'll model TCP's fairness in DCNs and deduce a mathematical model of $R^{F,f}$.

B. Modeling

1) RTT approximation: Without consideration of processing delay, RTT is composed of propagation delay and

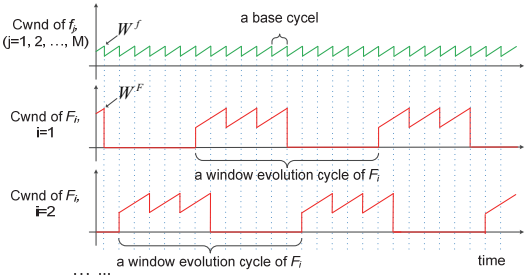

Fig. 4. The window evolutions of flows in the small set and the large set in an ideal steady state

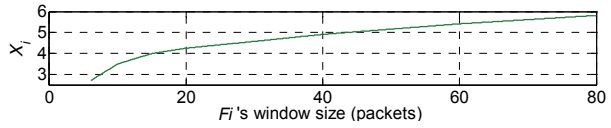

Fig. 5. The value of X_i when F_i's window size varies

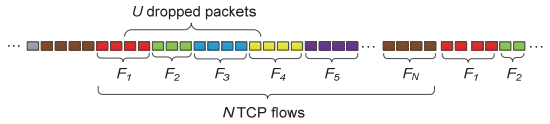

Fig. 6. Distribution of dropped packets in flows of the large set

queueing delay. We assume the average queueing delay for all packets approximates to $B/2C$. So RTT can be expressed as

$$RTT = D + \frac{B}{2C} \tag{1}$$

2) Calculating P^{loss} and P^{TO}: In our model, we use X_i to denote the average number of packets from F_i which enter into the switch consecutively. Through observation and analysis in our simulations, we find X_i has some deterministic numerical relationship with F_i's window size, as figure 5 shows. We can use empirical Eq. (2) to fit this relationship.

$$X_i = 1.35 \times \ln(w_i^F) \tag{2}$$

where w_i^F is the window size of F_i. When F_i loses packets, the window size of F_i is W^F, namely, $w_i^F = W^F$. So we can obtain

$$X_i = 1.35 \times \ln(W^F) \tag{3}$$

We assume that f_j loses only one packet at each loss event. So the total number of dropped packets of all flows in the small set at each loss event will be M. As analysis in section II, the number of consecutive dropped packets from one ingress port is relevant to the amount of bursty traffic injected into this port. Moreover, the amount of bursty traffic is mainly determined by the number of TCP flows accessed in this port. Hence, we can obtain the average total number of dropped packets of all flows in the large set at each loss event as follows:

$$U \approx 1 \times M \times \frac{N}{M} = N \tag{4}$$

As shown in figure 6, when the large set of flows lose about U consecutive packets in total at each loss event, these consecutive dropped packets will be distributed in part of the flows in the large set. If F_i loses packets, the number of consecutive dropped packets of F_i can be from 1 to $\min\{U, X_i\}$. In general, U is greater than X_i, so x_i can be from 1 to X_i. From figure 6 we can see that, there are $U + X_i - 1$ different cases in which F_i may lose packets. Among these cases, there are two cases in which F_i will lose one packet, two cases in which F_i will lose two packets, …, two cases in which F_i will lose $X_i - 1$ packets , and $U - X_i + 1$ cases in which F_i will lose X_i packets. We assume that each case has the same probability to occur. So we can get the probability distribution function of x_i as

$$P(x_i = k) = \begin{cases} \dfrac{2}{U + X_i - 1} & k = 1, 2, ..., X_i - 1 \\ \dfrac{U - X_i + 1}{U + X_i - 1} & k = X_i \end{cases} \tag{5}$$

We use $\mathrm{E}[x_i]$ to present the average number of dropped packets of F_i when F_i loses packets. The average number of flows which lose packets at each loss event in the large set can be expressed as $U / \mathrm{E}[x_i]$. So the conditional probability of losing packets for F_i at each loss event is

$$P^{loss} = \frac{U / \mathrm{E}[x_i]}{N} = \frac{U}{\mathrm{E}[x_i] \times N} \tag{6}$$

For simplicity, we make the assumption that F_i will get into a TO stage if all of its X_i consecutive packets are lost. Thus,

$$P^{TO} = P(x_i = X_i) = \frac{U - X_i + 1}{U + X_i - 1} \tag{7}$$

Based on (4) and (7), we know that any increment of N will increase the value of P^{TO}. In our simulations we actually see that a bigger value of N will result in more TO events for flows in the large set, which verifies this point.

3) Influence of network capacity: We use w_i^F and w_j^f to denote the window size of F_i and f_j respectively. Generally, a loss event means that the network capacity can't hold all packets transmitted in the network at this moment. So, at each loss event, we can infer that

$$\sum_{j=1}^{M} w_j^f + \sum_{i=1}^{N} w_i^F \geq B + CD \tag{8}$$

We simplify this inequation to Eq. (9)

$$\sum_{j=1}^{M} w_j^f + \sum_{i=1}^{N} w_i^F = B + CD \tag{9}$$

According to the foregoing analysis, we know w_j^f equals to W^f. In a window evolution cycle of F_i, there are T^F / T^f loss events in total in the network. Among these loss events, there are $1/P^{TO}$ times where F_i will lose packets and its window size is W^F, T^O / T^f times where F_i will stay in the TO stage and its window size is 0, $(T^F / T^f - 1/P^{TO} - T^O / T^f)$ times where F_i will stay in the CA stage and its window size is between $W^F/2$ to W^F (Its expectation can be $3W^F/4$). So the average window size of F_i at each loss event can be calculated as

$$\mathrm{E}[w_i^F] = \frac{1}{T^F / T^f} \times \left[W^F \times \frac{1}{P^{TO}} + 0 \times \frac{T^O}{T^f} + \frac{3}{4} W^F \times \left(\frac{T^F}{T^f} - \frac{1}{P^{TO}} - \frac{T^O}{T^f} \right) \right] \tag{10}$$

Based on (9) ~ (10), we can obtain

$$M \times W^f + \frac{N}{T^F / T^f} \times \left[\frac{W^F}{P^{TO}} + \frac{3W^F}{4} \times \left(\frac{T^F}{T^f} - \frac{1}{P^{TO}} - \frac{T^O}{T^f} \right) \right] = B + CD \tag{11}$$

4) Throughput ratio R^{Ff}: As we know, f_j's window size is increased by 1 in each RTT before some packets are dropped. In each base cycle, f_j's window size varies from $W^f/2$ to W^f. Counting the last RTT where f_j enters into the FR stage, a base cycle of f_j is composed of $(W^f - W^f/2 + 1 + 1)$ RTTs. The average duration of a base cycle for f_j is

$$T^f = (W^f / 2 + 2) \times RTT \tag{12}$$

We can get Y^f (The related derivation can be found in [10]).

$$Y^f = \frac{3}{8}(W^f)^2 + \frac{5}{4}W^f \tag{13}$$

Like T^f, we use T_{CA}^F to denote the duration of a CA stage of F_i.

$$T_{CA}^F = (W^F/2+2)\times RTT \tag{14}$$

Recall that the conditional probability that F_i loses packets at each loss event is P^{loss}. We can infer that, during a CA stage of F_i, f_j can experience about $1/P^{loss}$ CA stages. So we can get

$$T_{CA}^F = \frac{1}{P^{loss}}T^f \tag{15}$$

Based on (12), (14), (15), we can obtain the relationship between W^F and W^f as

$$W^f = P^{loss}\times W^F - 4(1-P^{loss}) \tag{16}$$

Recall that P^{TO} is the conditional probability of entering into a TO stage from a CA stage for F_i. We can infer that after about $1/P^{TO}$ CA stages, F_i will enter into a TO stage. This means that a window evolution cycle of F_i contains about $1/P^{TO}$ CA stages and a TO stage. So we can calculate the duration of a window evolution cycle of F_i as follows.

$$T^F = T_{CA}^F\times\frac{1}{P^{TO}} + T^O = (\frac{W^F}{2}+2)\times\frac{RTT}{P^{TO}} + T^O \tag{17}$$

In addition, we can get Y^F as

$$Y^F = \frac{1}{P^{TO}}\times\left[\frac{3}{8}(W^F)^2 + \frac{5}{4}W^F\right] \tag{18}$$

Based on (12), (13), (17), (18), we calculate the throughput ratio $R^{F,f}$ as follows:

$$R^{F,f} = \frac{Y^F/T^F}{Y^f/T^f} = \frac{\left[\frac{3}{8}(W^F)^2+\frac{5}{4}W^F\right]\times(\frac{W^f}{2}+2)\times RTT}{\left[\frac{3}{8}(W^f)^2+\frac{5}{4}W^f\right]\times\left[(\frac{W^F}{2}+2)\times RTT + P^{TO}\times T^O\right]} \tag{19}$$

Finally, we can combine (1), (3) ~ (7), (11), (16), (19) to get the throughput ratio $R^{F,f}$.

IV. Model Validation

In this section, we use simulations on ns-2 [11] to validate our model. We also discuss the effects of some parameters on TCP's fairness problem in DCNs. The topology used in our simulations has been shown in figure 3. We set the bandwidth of each link, the delay on each link, and the buffer size of each switch to be the same respectively. The queue manage mechanism adopted in the switches is DT. The RTO_{min} is set to be 200ms, which is the default value of TCP.

A. Variation of Network Capacity

Since the network capacity is the sum of buffer size and the product of bandwidth and delay, we need to investigate the impacts of all the three network parameters on TCP's fairness.

First, we carry out three groups of simulations, where we set the number of flows in the large set (i.e., N) to 12， and the number of flows in the small set (i.e., M) to 2. In each group of simulations, we change one of the three network parameters while keeping the other two parameters fixed. Figure 7-9 shows the variation of the throughput ratio $R^{F,f}$ of our proposed model and simulation results along with the change of buffer size, propagation delay, and bandwidth, respectively.

From figure 7-9, we can see that, as the network parameters change, our model can characterize the general trend of TCP's fairness well. That is, as the network capacity increases, the average throughput ratio of the large set of TCP flows to the small set of TCP flows grows from less than 1 to higher than 1. We can also see that, with different network capacity, the throughput ratio $R^{F,f}$ computed by our model is close to that of simulation results. This indicates that our model can match the simulation results well over a wide range of buffer size, propagation delay and bandwidth.

Second, to demonstrate that our model is not limited to the

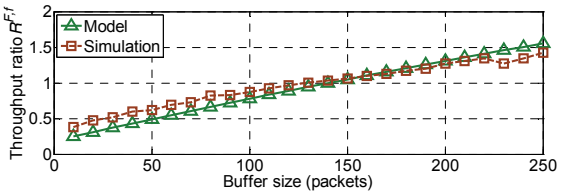

Fig. 7. Throughput ratio $R^{F,f}$ versus buffer size, where the propagation delay is 300 us and the bandwidth is 1 Gbps

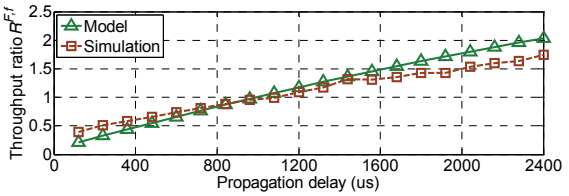

Fig. 8. Throughput ratio $R^{F,f}$ versus propagation delay, where the buffer size is 32 packets and the bandwidth is 1 Gbps

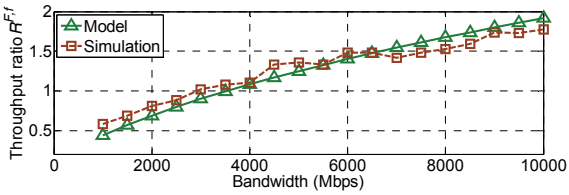

Fig. 9. Throughput ratio $R^{F,f}$ versus bandwidth, where the buffer size is 10 packets and the propagation delay is 500 us

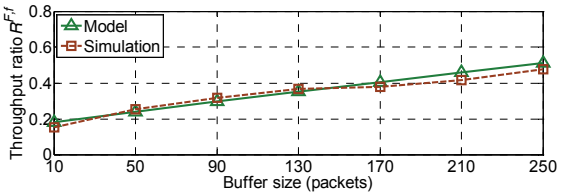

Fig. 10. Throughput ratio $R^{F,f}$ versus buffer size, where the BDP is 140 packets , M is 20 and N is 120

scenario in which the number of flows in the two sets is relatively small, we increase N from 12 to 120, and M from 2 to 20. Likewise, we set the propagation delay and bandwidth fixed while changing the buffer size. Figure 10 shows the variation of the throughput ratio $R^{F,f}$ of our proposed model and simulation results when the buffer size increases.

From figure 10, we observe that the throughput ratio $R^{F,f}$ computed by our model and simulation results have close growth curves. This indicates that, in the scenario where the number of flows in the two sets is relatively large, our model still works well. Moreover, we can also notice that in this scenario $R^{F,f}$ is always less than 1, even when the buffer size reaches 250 packets. In fact, according to equation (4) and (7) in our model, we have known that as N increases, the value of P^{TO} also increases. So, when N is relatively large, flows in the large set will suffer more TO events. This will make their average throughput much lower than the one of flows in the small set, even with plenty of network capacity. Our simulation traces also verify this point, which further indicates that our model is effective.

B. *Variation of Number of Flows*

In this subsection, we study the impacts of the number of flows in the large set and the small set on TCP's fairness. As shown in figure 11, we make three groups of simulations to compare our model with simulation results. In each group of simulations, we change M from 2 to 10 while keeping N to be a fixed multiple of M and the network capacity fixed. In each group of simulations, we set the multiple to 5, 10 and 15, and network capacity to 160, 110 and 70 packets, respectively.

From figure 11, we can see that the throughput ratio $R^{F,f}$ computed by our model is very close to that of simulation results. This indicates that our model can match the simulation results very well with different number of flows in the two sets. We can also see that in each group of simulations, the throughput ratio $R^{F,f}$ decreases as M and N increase. As we have pointed out in the former subsection, the increasing number of N will bring more TOs to each flow in the large set. So the value of $R^{F,f}$ decreases. Accordingly, we can see that, for a fixed value of M, the throughput ratio $R^{F,f}$ increases as the network capacity increases and N decreases, which is not beyond our expectation. In addition, we notice that in this figure the value of $R^{F,f}$ is below 1 in most cases. This is because of the same reason which we have mentioned for figure 10.

From the above results, we prove that our model can reveal TCP's fairness problem well with different network capacity and number of flows in the large set and the small set.

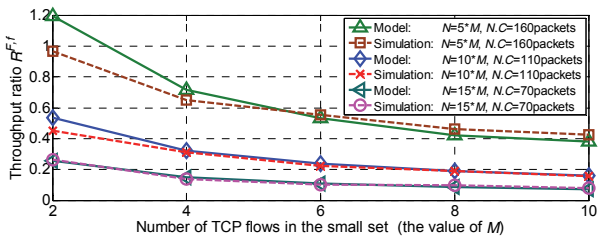

Fig. 11. Throughput ratio $R^{F,f}$ versus the number of TCP flows in the small set

Specifically, we can draw the following three conclusions about TCP's fairness problem in DCNs. First, when N and M are small, the average throughput ratio of the large set to the small set will grow from less than 1 to higher than 1 as the network capacity increases. Second, when N and M are large, the average throughput ratio of the large set to the small set would be always less than 1, but increasing the network capacity can alleviate this problem. Third, when N and M are large, for a fixed value of M, increasing N will make the fairness problem even worse.

V. CONCLUSION

In this paper we studied TCP's fairness problem in data center networks. TCP's fairness problem exists in the case where a large set of TCP flows and a small set of TCP flows get into different ingress ports of a switch and compete for a same egress port. And this case is common in today's DCNs due to their special topologies and communication pattern. Through comprehensive simulations and analysis, we found that TCP's fairness in DCNs is related with not only the network capacity but also the number of flows in the large set and the small set. When the number of flows in the two sets is small, the average throughput ratio of the large set to the small set will grow from less than 1 to higher than 1 as the network capacity increases. When the number of flows in the two sets is large, the average throughput of the large set will be much lower than that of the small set, even with plenty of network capacity. Our major contribution of this paper was to propose a mathematical model of the average throughput ratio of the large set to the small set. This model can reveal the variation of TCP's fairness along with the change of network capacity and the number of flows in the two sets. We validated our model by comparing its numerical results with simulation results. Results of our model match the simulation results well in a wide range of network parameters and the number of flows in the two sets.

REFERENCES

[1] Zhang Jiao, Fengyuan Ren, and Chuang Lin, "Survey on transport control in data center networks," *Network, IEEE* 27.4 (2013).

[2] M. Alizadeh et al., "Data Center TCP (DCTCP)," in *SIGCOMM*, 2010.

[3] Zhang, Jiao, et al., "Taming TCP incast throughput collapse in data center networks," in *ICNP*, 2013.

[4] Wu, Haitao, et al., "ICTCP: incast congestion control for TCP in data-center networks," *TON* 21.2 (2013): 345-358.

[5] Munir, Ali, Ihsan Ayyub Qazi, and Saad Bin Qaisar, "On achieving low latency in data centers," in *ICC,* 2013.

[6] Prakash, Pawan, et al., "The TCP Outcast Problem: Exposing Unfairness in Data Center Networks," in *NSDI*. 2012.

[7] Al-Fares, et al., "A scalable, commodity data center network architecture," *ACM SIGCOMM Computer Communication Review*. Vol. 38. No. 4. ACM, 2008.

[8] Dean, et al., "MapReduce: simplified data processing on large clusters," *Communications of the ACM*. 51.1 (2008): 107-113.

[9] Ren Fengyuan, Chuang Lin, and Li Tang, "Modeling and Solving TCP Incast Problem in Data Center Networks," *IEEE Transactions on Parallel and Distributed Systems* (2014): 1.

[10] Padhye, Jitendra, et al., "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Computer Communication Review*. Vol. 28. No. 4. ACM, 1998.

[11] S. McCanne, D. Floyd. ns Network Simulator. http://www.isi.edu/nsnam/ns/.