Achieving Fast Convergence and High Efficiency using Differential Explicit Feedback in Data Center

Ning Jiang, Jiawei Huang, Sen Liu, Jinbin Hu, Jianxin Wang
School of Computer Science and Engineering
Central South University
Changsha, China
{csu_jn, jiaweihuang, sen.liu, jinbinhu, jxwang}@csu.edu.cn

***Abstract*—Since most flows are short-lived in data center networks, fast convergence becomes very important to help the short flows effectively utilize high bandwidth. Though current feedback-based transport control protocols (TCPs) provide fast convergence via fine-grained explicit congestion information from customized switches, they unavoidably incur large traffic overhead for widely existing small packets in data center applications, resulting in suboptimal network efficiency. To solve this issue, we propose a datacenter TCP based on differential feedbacks, called DECN, to achieve fast convergence without any traffic overhead. Specifically, DECN feeds rate difference between the target and current rate back to the source by using multiple consecutive packets. The experimental results of NS2 simulation and testbed implementation show that DECN achieves comparable fast convergence as XCP without incurring any extra feedback overhead. Compared with the state-of-the-art feedback-based TCPs, DECN reduces the flow completion time by up to 34.1% in typical data center applications.**

***Index Terms*—data center, differential rate, explicit feedback**

I. Introduction

In modern data center, the majority of flows are short-lived or small flows, which are usually less than 100KB with very short transmission time [1]. This traffic characteristic in data center highlights the significance of convergence speed of short-lived flow [2]. A small flow needs to quickly and fairly converge to its target rate to make full use of very large available bandwidth in data center networks [3]. How to achieve the fast convergence is essential for TCP design, particularly in the high speed data center networks [4], [5].

On the other hand, in many typical datacenter applications, the size of data packets is small. For example, the median packet size is far below 200Bytes in well-known applications, such as Web Server and Cache Lead [6]. Therefore, the design of datacenter TCP should consider how to avoid the degradation of transmission efficiency due to large overhead.

Although a serial of feedback-based congestion control protocols have been proposed to solve convergence problems, they usually face a dilemma between fast convergence and large overhead. The classical XCP [7] and RCP [8] use a large number of extra bits in the header of data packet to carry fine-grained network state to achieve fast convergence, however, leading to huge overhead and low transmission efficiency. Recently, HPCC [9] is proposed to obtain accurate link load information and control traffic precisely by leveraging intranetwork telemetry (INT). However, HPCC still requires standard INT feature in switches to carry the large state information (i.e., each INT Metadata header contains 8Bytes of feedback information). On the flip side, DCTCP [10] and VCP [11] use only one bit or a few bits in the packet header to notify congestion state. Although the overhead is well controlled by using coarse-grained feedback, it takes a long time to converge to the target rate due to less accurate congestion information.

We propose a datacenter TCP called DECN to achieve fast convergence without any extra traffic overhead. The key point of DECN is that, for each flow, the switch feeds back the rate difference between the target rate and current rate of a flow, rather than the target rate in the traditional designs. Our design rationality is twofold. When the new arrival flows increase their rates, their rate differences are smaller than the target rates. For the existing flows, since the number of concurrent flows is usually very large (i.e., > 500) in data center, their rate differences are also smaller than their target rates when they converge to the fair bandwidth sharing. Therefore, DECN is able to use a fewer number of bits to present the rate difference than the target rate. Moreover, to eliminate the extra feedback overhead, DECN utilizes multiple consecutive packets to carry the feedback information. Therefore, DECN achieves fast convergence via fine-grained congestion feedback, while keeping zero extra feedback overhead.

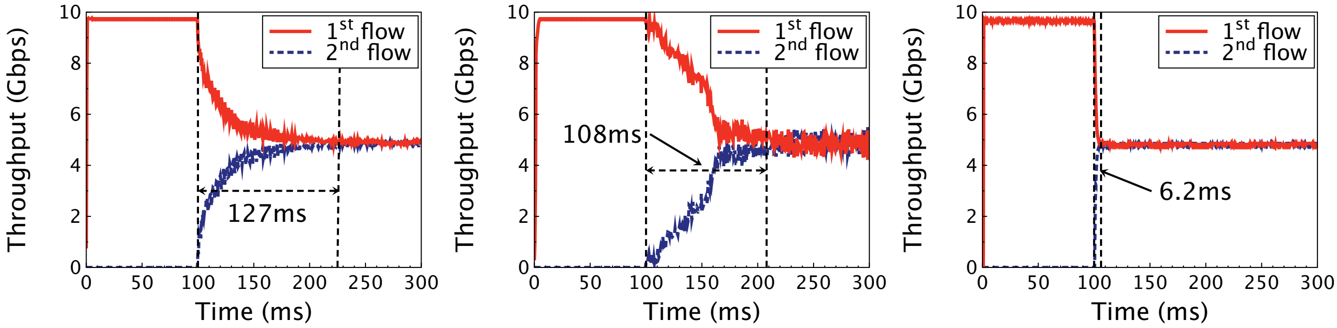
In short, the major contributions of this paper are:

- We conduct the extensive study to show that the congestion control with coarse-grained explicit feedback is hard to achieve fast convergence, and the schemes with fine-grained explicit feedback incur large overhead for the widely existing small packets in data center applications.
- We propose a datacenter TCP called DECN to achieve fast convergence and zero extra overhead by feeding the rate difference between the target and current rate with multiple consecutive packets.
- We evaluate DECN with NS2 simulation and testbed implementation. The results show that DECN achieves comparable convergence speed yet much smaller overhead than XCP. Compared with the state-of-the-art feedback-based TCPs, DECN reduces the flow completion time by up to 34.1% in well-known data center applications.

The rest of the paper is organized as follows. We present the design motivation of DECN in Section II. In Section

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.



(a) DCTCP (b) MLCP (c) XCP

Fig. 1: Convergence speed.

III, we describe the basic idea and present the details of DECN. We evaluate the performance of DECN with testbed implementation in Section IV and NS2 simulations in Section V. The related works are presented in Section VI. Finally, we make conclusion in Section VII.

II. Motivation

Many TCP variants have been proposed to use explicit feedback to achieve fast convergence for utilizing high link bandwidth. Table I shows the number of bits used by different protocols to carry congestion state. In this section, we investigate the performances of feedback-based TCPs in data center networks.

TABLE I: The number of bits used for congestion feedback

Protocol	The number of bits	Protocol	The number of bits
XCP [7]	128	VCP [11]	2
RCP [8]	96	UNO [14]	2
BMCC [13]	18	VCP-BE [15]	2
MLCP [12]	7	DCTCP [10]	1

A. Convergence Speed

As the representatives of feedback-based TCPs, XCP [7], MLCP [12] and DCTCP [10] employ 128 bits, 7 bits and 1 bit to carry congestion status information, respectively. Here, we conduct NS2 simulation to test the convergence speed of XCP, MLCP and DCTCP. The test uses a dumbbell topology, in which two senders send two flows to two receivers via a switch with buffer size of 200 packets. Each link is with 10Gbps bandwidth and 100μs latency. The 2nd flow starts at 100ms after the 1st flow has achieved full bandwidth. In the test, tp is set as 0.1ms for MLCP, and K is 65 for DCTCP. Here, tp is the period to computing the load factor, and K is the ECN marking threshold of instantaneous queue length.

As shown in Fig.1(a), DCTCP spends about 127ms to achieve fair bandwidth sharing between two flows. This convergence time is too long for small flows, which usually have less than 100KB size and finish transmission in a few RTTs. Fig.1(b) shows that, MLCP reduces its convergence time to about 108ms due to its larger feedback information than DCTCP. Fig.1(c) shows that XCP has the minimal convergence time as 6.2ms, because it uses much more bits to feed congestion state back than the other protocols. The above results show that, since the number of bits used to feed information back affects the accuracy of congestion status, the TCP protocol with more feedback information makes more precise congestion control decisions, thus achieving faster convergence.

B. Feedback Overhead

The production data center hosts diverse applications, which have different distributions of packet size. According to the data statistics in Facebook data center [16], we plot the distribution of packet size in four typical data center applications, including Web Server, Cache Lead, Cache Follower and Hadoop. As shown in Fig.2, except Hadoop, most packets in the typical applications are less than 400Bytes.

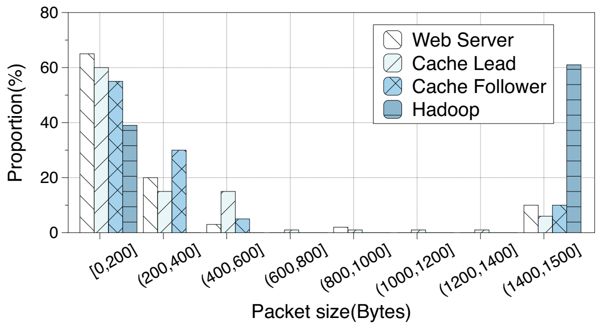


Fig. 2: Packet size distribution.

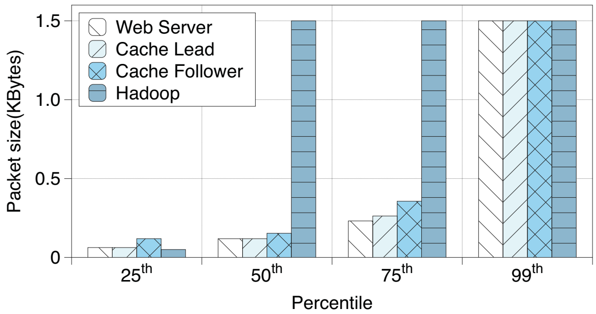


Fig. 3: Packet size distribution.

We further measure the 25^{th}, 50^{th}, 75^{th} and 99^{th} percentile packet size of four applications. Fig.3 shows that, for Web Server, Cache Lead, Cache Follower, the 50^{th} percentile packet size is less than 200Bytes. Only for Hadoop, more than 50% of packets make full use of the 1.5KB MTU.

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.

Finally, we calculate the ratio of feedback overhead to packet size across different applications. Fig.4 shows that, the feedback overhead is comparably large in XCP and RCP. For example, in Hadoop application, the feedback overheads of XCP and RCP are 32% and 23% for the 25^{th} percentile packet size, respectively, resulting in loss of network efficiency.

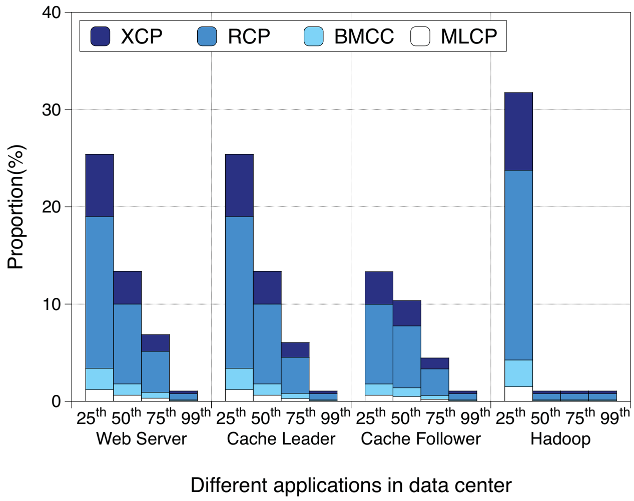


Fig. 4: Feedback overhead.

C. Conclusion

In summary, we make the following conclusions. (1) The existing feedback-based TCPs need large number of bits to carry the fine-grained network state to make accurate congestion control and thus achieve fast convergence. (2) In the data center applications dominated with small packets, large feedback overhead leads to low network efficiency. To provide fast convergence, we propose a datacenter TCP DECN based on differential feedbacks without any traffic overhead. We give the design details of DECN in the rest of this paper.

III. Design

In this section, we firstly describe the design overview of DECN. Then, we give the design details at the sender, switch and receiver sides.

A. Design Overview

In Fig.5, we plot the architecture of DECN, which includes three components as follows.

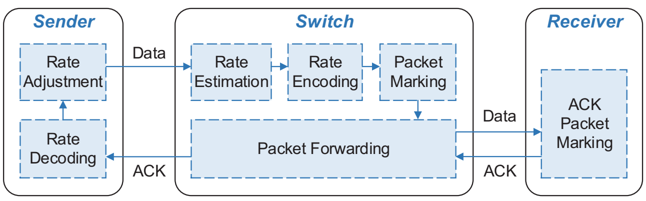


Fig. 5: DECN Architecture.

(1) **DECN switch:** The core design of DECN is rate encoding and packet marking. The DECN switch periodically estimates the rate difference between the current and target rate for each flow. If necessary, the DECN switch encodes and marks the differential rate on multiple consecutive data packets, which are called as feedback units.

(2) **DECN sender:** The DECN sender receives the ACKs and decodes the feedback rate carried in the packet header. Then the DECN sender adjusts the transmission rate as the sum of its current rate and the feedback rate.

(3) **DECN receiver:** The DECN receiver simply copies the header information of the marked data packet into the corresponding ACKs header and sends it back to the sender.

B. Rate Estimation

To achieve fast convergence, the DECN switch periodically calculates the differential rate between the current and target rate of each flow. Specifically, the switch takes a period of sampling time to measure the amount of arrival data packets in each flow. Moreover, for each flow, the switch also counts the bytes of packets in the buffer queue. Then, the switch gets the current rate *CR* as

$$CR = \frac{\sigma + q}{t_p}, \tag{1}$$

where t_p is the sampling time, σ is the input traffic during t_p, q is the persistent queue length in bytes during t_p. Taking the consideration of the RTT distribution in data center networks, we set t_p as 0.1ms.

To achieve the fair sharing for multiple flows, the target rate *TR* is calculated as

$$TR = \frac{C}{n}, \tag{2}$$

where n is the number of active flows and C is the link bandwidth. Thus, we can get the differential rate *DR* as

$$DR = TR - CR = \frac{C}{n} - \frac{\sigma + q}{t_p}. \tag{3}$$

C. Packet Marking

DECN employs three multiple consecutive data packets as a feedback unit to carry the differential rate, which is a 12-bit binary number with its 1st bit as the sign bit.

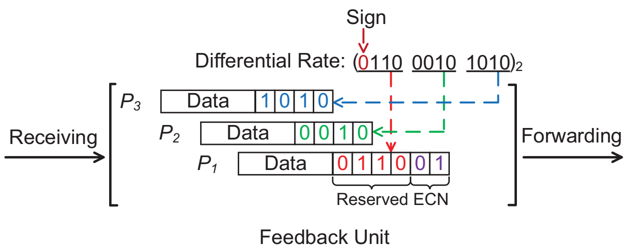


Fig. 6: Feedback Unit.

As shown in Fig.6, a feedback unit consists of three data packets. Four reserved bits in the header of each packet are marked. The differential rate with its unit as 1Mbps is successively written into the headers of the three packets. Therefore, the earlier arrival packet carries the higher four bits. Moreover, DECN marks the ECN bits of the 1st packet as $(01)_2$ to indicate the start of a feedback unit.

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.

In the end-to-end transmission, the sending rate of each flow should be determined by the most congested bottleneck link [13]. As shown in Fig.7, when receiving a data packet in flow f with its marked differential rate Δ_1, switch S_2 calculates the sum of current rate CR_2 of flow f and Δ_1. If CR_2+Δ_1 is less than the target rate TR_2 at S_2, the feedback information in packet header is not modified by S_2. When the packet arrives at S_3, however, if CR_3+Δ_1 is larger than TR_3, S_3 updates the feedback information to Δ_3. When receiving the data packets, the DECN receiver marks the corresponding ACK packets with Δ_3 and sends it back to the DECN sender. Finally, the DECN sender updates the sending rate R of flow f to R+Δ_3.

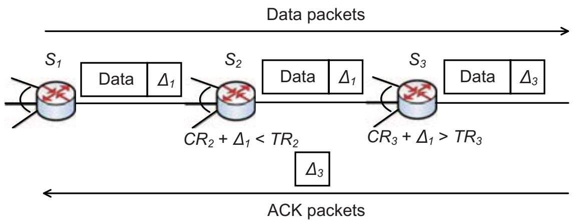


Fig. 7: The operation of DECN.

Algorithm 1 gives the details on packet marking at switches. Specifically, DECN switch firstly checks the ECN bits of arrival packets to distinguish the 1st packet of feedback unit. Then DECN switch compares the rate difference carried in each packet with its target rate difference. Finally, DECN switch updates the packet header with its rate difference information if necessary. Moreover, if DECN switch receives any out-of-order packets in a feedback unit, it will drop the current feedback unit to avoid making wrong marking decision.

At the sender side, the DECN sender adjusts its sending rate every time all ACKs of a feedback unit are received. Furthermore, DECN sender marks the ECN bits on the 1st packet in feedback unit, which is generated once in every t_p time. Since the feedback units are successive, DECN sender is able to capture the real-time feedback information and adjust sending rate quickly.

Algorithm 1: Packet Marking

Initialization:
  $t_p \leftarrow 100\mu s$; //Sampling time
  $\sigma \leftarrow 0$; //Amount of traffic received during t_p
  $C \leftarrow$ Link bandwidth;
  $n \leftarrow 0$; // Number of flows
  $\Delta \leftarrow 0$; // Marked value in packet header
  $q \leftarrow 0$; // Persistent queue length per flow during t_p

On receiving packet P_i from flow f:
begin
  if $(P_i.ECN == (01)_2)$ **then**
    $k = 0$;
    $Flag = 0$;
    $SD = (DR >> 11)?1{:}(-1)$; //Sign of DR
    $SU = (P_{i+k}.\Delta >>3)?1{:}(-1)$; //Sign of $P_{i+k}.\Delta$
    while $(k < 3)$ **do**
      $DR' = (DR << 4k) >> (8-4k)$;
      if $(Flag == 1)$ **then**
        $P_{i+k}.\Delta = DR'$; //Mark packet P_{i+k}
      else
        if $((k{=}{=}0\ \&\&\ P_{i+k}.\Delta{>}DR')\ ||\ (k{>}0\ \&\&\ P_{i+k}.\Delta{\times}SU{>}DR'{\times}SD))$ **then**
          $P_{i+k}.\Delta = DR'$; //Mark packet P_{i+k}
          $Flag = 1$;
        else if $((k{=}{=}0\ \&\&\ P_{i+k}.\Delta{<}DR')\ ||\ (k{>}0\ \&\&\ P_{i+k}.\Delta{\times}SU{<}DR'{\times}SD))$ **then**
          break;
      k++;
      Receive the next packet from flow f;
      if *experience out-of-order* **then**
        return; //Drop current Feedback Unit

When the timer expires after t_p:
begin
  update σ, n and q;
  $CR \leftarrow (\sigma + q)\ /\ t_p$; //CR is a 12-bit binary number
  $TR \leftarrow C\ /\ n$; //TR is a 12-bit binary number
  $DR \leftarrow TR - CR$; //Differential rate
  reset the timer;

IV. Testbed Evaluation

In this section, we use a real testbed to evaluate the performance of DECN. The testbed consists of 4 servers, each of which has a Pentium(R) E6600 3.06GHz Dual-Core CPU, a 4GB DDR3 memory and an Intel Corporation 82580 Gigabit Network Interface Card. The operating system is CentOS 7.5 with Linux kernel 3.10.0-862.el7.x86. The link capacity and buffer size are 1Gbps and 64 packets, respectively. We compare DECN with XCP, VCP and DCTCP in different scenarios. The ECN marking threshold k is set to 20 packets in DCTCP as recommended in [10].

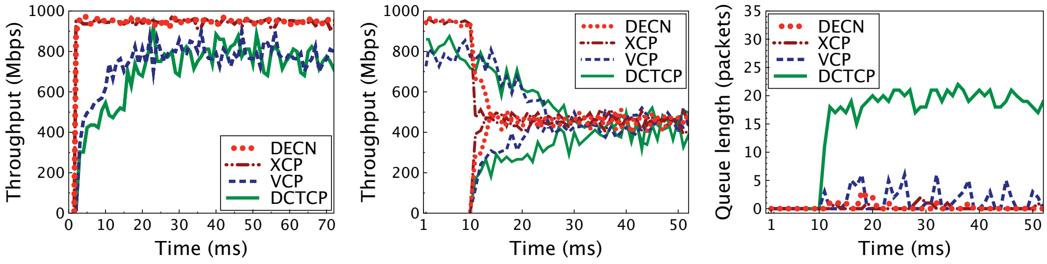
A. Utilization

We firstly conduct an experiment with only one flow to measure the link utilization. As shown in Fig.8(a), the flows in VCP and DCTCP are hard to achieve full utilization within a short time since VCP and DCTCP increase their congestion windows slowly according to congestion load. The DECN flow saturates the bottleneck link as fast as XCP since the sender can adjust its transmission rate based on the explicit differential feedback between the current and target rate from the marked packets. As a result, DECN achieves around 95% link utilization, about 15% higher than VCP and DCTCP.

B. Convergence and queue length

We evaluate the convergence of DECN in a two-to-one communication scenario, where each of 2 senders sends a flow to a single receiver via a switch. Fig.8(b) and (c) show the convergence speed and queue length, respectively. At the beginning, the link bandwidth is fully utilized by one flow. At

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.

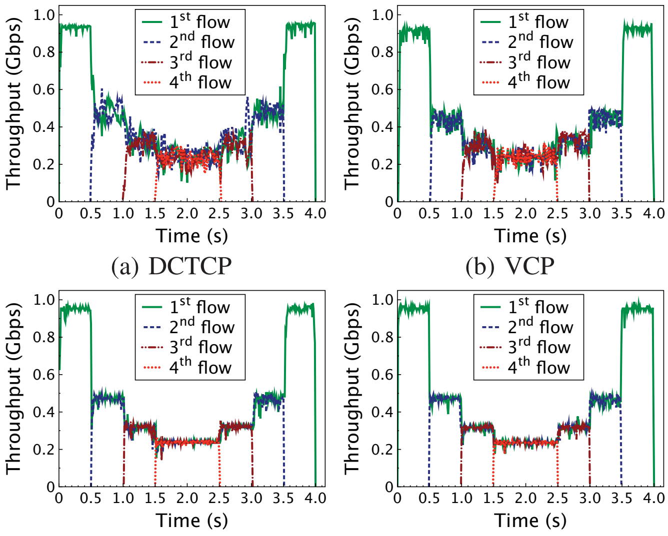


(a) Utilization (b) Convergence (c) Queue length

Fig. 8: Utilization, convergence and queue length.

10ms, another flow is generated to compete for the bottleneck link. As shown in Fig.8(b), VCP and DCTCP suffer from slow convergence speed to the fair sharing. The 2nd flow spends about 15ms and 25ms to achieve the fair rate in VCP and DCTCP, respectively. Compared to VCP and DCTCP, two flows of DECN converge to fair throughput quickly like XCP, because the senders conduct precise rate adjustment once receiving enough marked packets carrying the rate feedback.

Fig.8(c) shows the instantaneous queue lengths under four transport protocols. The instantaneous queue lengths change dynamically under VCP and DCTCP, since they adjust the congestion window according to limited feedback information of congestion state. For XCP and DECN, even when a new flow is introduced to compete bandwidth at the bottleneck link, the instantaneous queue length is still almost zero.



(a) DCTCP (b) VCP (c) XCP (d) DECN

Fig. 9: Fairness.

C. Fairness

We evaluate the fairness performance of DECN. In this test, four flows from independent senders are sent one after another one to a single receiver via a switch. Fig.9 shows the flow throughputs of DCTCP, VCP, XCP and DECN. Fig.9(a) and (b) show that, DCTCP and VCP take a longer time to achieve fairness when a new flow is injected into the bottleneck link, because they adjust the congestion window imprecisely by only leveraging one or two ECN bits for congestion feedback. As shown in Fig.9(c) and (d), XCP and DECN achieve much better per-flow fairness performances than DCTCP and VCP.

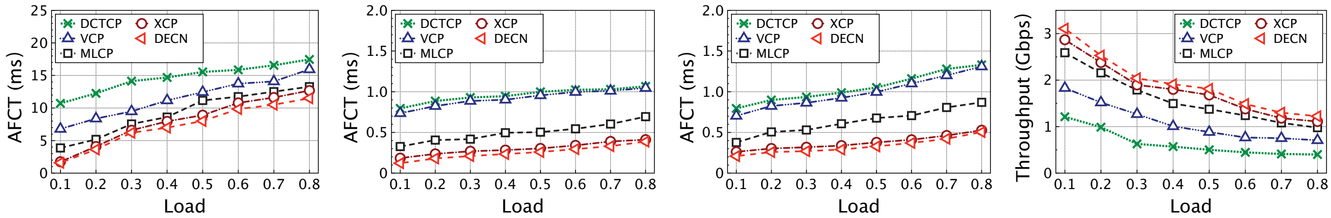
V. Simulation Evaluation

In this section, we conduct large-scale NS2 simulations to test DECN performance under a typical datacenter application, namely Cache Follower [16]. Specifically, in Cache Follower about 45% of flows are less than 10KB, 15% of flows are between 10KB and 100KB, 18% of flows are between 100KB and 1000KB, and the others are larger than 1000KB. The median packet size of this workload is less than 200Bytes, and nearly half of the packets are about 100Bytes.

We use a leaf-spine topology that consists of 8 core switches, 8 ToR switches and 400 hosts connected by 10Gbps links. The over-subscription ratio is about 6:1 at the leaf switch layer. The switch buffer size and the round-trip propagation delay are 128 packets and 100 μs, respectively. The flows are generated between the random pair of hosts. We change the traffic load from 0.1 to 0.8. We measure the flow completion time (FCT) and throughput of short flows (<100KB) and long flows ($\geq$ 100KB) under DECN, XCP, MLCP, VCP and DCTCP protocols.

Fig.10(a) shows the average FCTs of all flows under the Cache Follower workload. We observe that DECN improves AFCT significantly compared with DCTCP, VCP and MLCP across all loads. Specifically, DECN reduces AFCT by 34.1%, 27.6%, 13.4% and 6.6% at 0.8 load over DCTCP, VCP, MLCP and XCP, respectively. Fig.10(b) and (c) show the AFCT and 99^{th} percentile FCT of short flows, respectively. These test results demonstrate the advantage of DECN by using differential explicit feedback. DCTCP and VCP adjust their sending rates based on the imprecise congestion feedback information, resulting in performance degradation of FCT. Since the feedback overhead reduces the network efficiency, XCP also gets suboptimal performance. Compared with the other protocols, DECN adjusts the sending rate with zero extra feedback overhead, achieving lower FCT especially for short flows. Since long flows are throughput-sensitive, we test throughput of long flows. Fig.10(d) shows the throughputs of long flows with varying load. DECN also performs better than the other protocols. Specifically, DECN improves throughput of long flows by 196.3%, 67.9%, 16.1% and 7.0% at 0.8 workload over DCTCP, VCP, MLCP and XCP, respectively. Overall, the test results show that DECN effectively improves

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.



(a) AFCT of all flows
(b) AFCT of short flows
(c) 99^{th} percentile FCT of short flows
(d) Throughput of long flows

Fig. 10: Cache Follower application.

the convergence speed with zero extra traffic overhead, therefore providing better application performance.

VI. Related Work

In recent decades, many TCPs based on the explicit congestion feedback are proposed to achieve high efficiency in high speed network and data center networks.

The classical TCP + RED/ECN [17] uses the ECN fields to notify congestion information to the senders, achieving low loss rate and small queue length. However, when the sender receives the ECN-marked ACK packet, the congestion window is directly halved, possibly resulting in insufficient link utilization. To obtain fast convergence, XCP [7] and RCP [8] carry congestion information via a large number of extra bits, but resulting in huge overhead. VCP [11] leverages two ECN bits to feed congestion information back. Since two bits can only represent four different states, the inaccurate feedback leads to slow convergence. MLCP [12] delivers load and RTT information by 4 bits and 3bits, respectively. MPCP [18] employs the packet chain to carry network state. Though avoiding extra feedback overhead, MPCP still uses the so-called load factor to present different congestion levels, resulting in inaccuracy of rate feedback and slow convergence speed. BMCC [13] utilizes the Adaptive Deterministic Packet Marking (ADPM) and two ECN bits to obtain congestion estimation with 16 bits resolution.

In data center networks, DCTCP [10] notifies the congestion state by only one bit in the packet header. Similarly, D^2TCP [19] also experiences slow convergence due to only one-bit feedback. Recently, HPCC [9] utilizes intranetwork telemetry (INT) to obtain accurate load information and precisely control traffic. However, HPCC incurs large implementation overhead such as 8Bytes in each INT Metadata header.

Compared with the above explicit feedback protocols, DEC-N employs multiple packets to feed back the rate difference, which is usually smaller than the target rate. Therefore, DECN achieves fast convergence in rate adjustment with zero extra feedback overhead.

VII. Conclusion

This paper proposes a data center TCP called DECN to achieve fast convergence without any traffic overhead. DECN feeds back the rate difference between the target and current rate with multiple consecutive packets. The test results of NS2 simulation and testbed implementation show that DECN achieves high efficiency, fast convergence, low queueing delay and good per-flow fairness.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61872387, 61872403) and CERNET Innovation Project (NGII20170107).

References

[1] T. Benson, A. Akella, and D. A. Maltz. Network traffic characteristics of data centers in the wild, in Proc. ACM IMC, 2010.
[2] M. Mellia, I. Stoica, and H. Zhang. TCP model for short lived flows, IEEE Communication Letters, vol. 6, no. 2, pp. 85-87, February. 2002.
[3] T. Zhang, J. Huang, J. Wang, J. Chen, Y. Pan, and G. Min. Designing Fast and Friendly TCP to Fit High Speed Data Center Networks, in Proc. IEEE ICDCS, 2018.
[4] I. Cho, K. Jang, and D. Han. Credit-Scheduled Delay-Bounded Congestion Control for Datacenters, in Proc. ACM SIGCOMM, 2017.
[5] M. Handley, C. Raiciu, and A. W. Moore. Re-architecting datacenter networks and stacks for low latency and high performance, in Proc. ACM SIGCOMM, 2017.
[6] A. Roy, H. Zeng, J. Bagga, G. Porter, and A.C. Snoeren. Inside the social network's (datacenter) network, ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 123-137, August. 2015.
[7] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks, ACM SIGCOMM Computer Communication Review, vol. 32, no. 4, pp. 89-102, August. 2002.
[8] N. Dukkipati, M. Kobayashi, R. Zhang-Shen, and N. McKeown. Processor sharing flows in the internet, in Proc. IEEE IWQoS, 2005.
[9] Y. Li, R. Miao, H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, K. Frank, A. Mohammad, and M. Yu. HPCC: High Precision Congestion Control, in Proc. ACM SIGCOMM, 2019.
[10] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data Center TCP (DCTCP), in Proc. ACM SIGCOMM, 2010.
[11] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman. One more bit is enough, in Proc. ACM SIGCOMM, 2005.
[12] I.A. Qazi, and Y. Znati. On the design of load factor based congestion control protocols for next-generation networks, in Proc. IEEE INFOCOM, 2008.
[13] I.A. Qazi, T. Znati, and L.L.H. Andrew. Congestion control using efficient explicit feedback, in Proc. IEEE INFOCOM, 2009.
[14] N. Vasic, S. Kuntimaddi, and D. Kostic. One bit is enough: a framework for deploying explicit feedback congestion control protocols, in Proc. IEEE COMSNETS, 2009.
[15] J. Wang, P. Dong, J. Chen, J. Huang, S. Zhang, and W. Wang. Adaptive Explicit Congestion Control based on Bandwidth Estimation for High Bandwidth-Delay Product Networks, Computer Communications, vol. 36, no. 10-11, pp. 1235-1244, June. 2013.
[16] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the Social Networks (Datacenter) Network, in Proc. ACM SIGCOMM, 2015.
[17] K.K. Ramakrishnan, S. Floyd, D. Black. The addition of explicit congestion notification (ecn) to ip, RFC3168, https://tools.ietf.org/html/rfc3168, September. 2001.
[18] Li. X, and Yousefi'zadeh. H. MPCP: multi packet congestion-control protocol, ACM SIGCOMM Computer Communication Review, vol. 39, no. 5, pp. 5-11, October. 2009.
[19] B. Vamanan, J. Hasan, TN. Vijaykumar. Deadline aware datacenter tcp(D^2TCP), in Proc. ACM SIGCOMM, 2012.

Authorized licensed use limited to: Central South University. Downloaded on February 08,2022 at 06:36:39 UTC from IEEE Xplore. Restrictions apply.