Combining ECN and RTT for Datacenter Transport

Gaoxiong Zeng
HKUST

Wei Bai
HKUST

Ge Chen
HKUST

Kai Chen
HKUST

Dongsu Han
KAIST

Yibo Zhu
Microsoft

ABSTRACT

Datacenter transports should provide low average and tail flow completion times (FCT) to achieve desired application performance. While most prior datacenter transports take either ECN or RTT as congestion signal, this paper makes a case that both signals are indispensable: ECN, as a per-hop signal, is more effective to prevent packet loss; while RTT, as an end-to-end signal, controls end-to-end queueing delay better. As persistent low flow completion times imply low queueing delay and near zero packet loss, we introduce EAR, a new datacenter transport that *hears* and reacts to both ECN and RTT. Our preliminary results show that: 1) compared to delay-based DCTCP, EAR achieves up to 91% lower packet losses and 93% fewer timeouts; 2) compared to ECN-based DCTCP, EAR reduces RTT by up to 32% for cross-rack traffic in a 4-level fattree. As a result, EAR delivers persistent low average and tail completion times under various scenarios in large scale simulations.

CCS CONCEPTS

• **Networks → Transport protocols**;

KEYWORDS

Datacenter Networks, Transport Protocol, Congestion Signal, ECN, RTT

ACM Reference format:
Gaoxiong Zeng, Wei Bai, Ge Chen, Kai Chen, Dongsu Han, and Yibo Zhu. 2017. Combining ECN and RTT for Datacenter Transport. In *Proceedings of APNet'17, Hong Kong, China, August 03-04, 2017,* 7 pages.
https://doi.org/10.1145/3106989.3107002

APNet'17, August 03-04, 2017, Hong Kong, China

ACM ISBN 978-1-4503-5244-4/17/08…$15.00
https://doi.org/10.1145/3106989.3107002

1 INTRODUCTION

With the prevalence of various web applications and services (e.g, web search, cloud computing, social networking, etc.), datacenters [1, 2, 25] have been built at an unforeseen rate and scale around the globe. A typical datacenter hosts a variety of applications with diverse network requirements. Some applications desire small predictable latency for small messages while others require large sustained throughput for bulk transfers. This imposes stringent requirements on the underlying network fabrics and protocols.

Traditional TCP algorithms (e.g., [10, 11, 15, 29]) designed for Internet typically adopt packet loss as the congestion signal. The loss-based solutions are likely to cause excessive packet losses and large queueing delay, thus failing to provide low latency [5]. Motivated by this observation, many new transport designs [5–8, 13, 18, 20, 21, 23, 27, 28, 31] have been proposed to achieve high throughput and low latency simultaneously in datacenter networks (DCNs). Realizing the limitation of packet loss, above proposals typically adopt new congestion signals, e.g., Explicit Congestion Notification (ECN) and Round-Trip Time (RTT), to provide low latency for small flows.

As the nature of congestion signal greatly governs the behavior of congestion control, a thorough study on congestion signals becomes essential for transport designs in DCNs. In this paper, we focus on congestion signals that are widely supported by commodity hardware. In particular, we focus on two congestion signals: ECN and RTT. ECN is well supported by today's commodity switches, and many ECN-based transports have been widely deployed in production datacenters [17, 26, 31]. In the meanwhile, recent work [18, 20] has shown that RTT can be accurately measured with advanced NIC hardware and used for DCN transport designs.

While most prior solutions adopt either ECN or RTT as congestion signal, in this paper, we show that both ECN and RTT signals are indispensable: ECN, as a per-hop signal, is more effective to prevent packet losses; whereas RTT, as an end-to-end signal, controls end-to-end queueing delay better. As persistent low flow completion times (FCT) imply low queueing delay and near zero packet loss, combining ECN and RTT signals becomes a natural choice. Therefore, we introduce EAR, a new datacenter transport that hears

and reacts to both ECN and RTT. While the full design of EAR is still ongoing, the preliminary ns2 [3] simulations show that: 1) compared to delay-based DCTCP, EAR achieves up to 91% lower packet losses and 93% fewer timeouts; 2) compared to ECN-based DCTCP, EAR reduces RTT by up to 32% for cross-rack traffic in a 4-level fattree. As a result, EAR delivers persistent low average and tail completion times under various scenarios in large scale simulations.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 motivates the need for combining ECN and RTT signals. Section 4 presents an initial design, called EAR, that leverages both ECN and RTT signals. Preliminary evaluation using ns2 simulations is also shown. Section 5 summarizes pros and cons of the two signals in general and discusses issues like multi-bottleneck fairness. Section 6 concludes the work and presents our potential future work direction.

2 RELATED WORK

ECN-based transports in datacenter networks: ECN-based transports consist of two components: ECN-aware rate control at the end host and ECN marking at the switch. Since DCTCP [5] in 2010, the networking research community has made many efforts [5, 6, 23, 27, 31] on ECN-based transports in DCNs. HULL [6] trades a little bandwidth to achieve near zero switch buffering. D2TCP [27] and L2DCT [23] modify the window adjustment function of DCTCP to meet deadlines and minimize FCT, respectively. DCQCN [31], built on the top of QCN [24] and DCTCP, is to enable deployment of Remote Direct Memory Access (RDMA) in large, IP-routed DCNs.

Delay-based transports in datacenter networks: Delay-based transports use RTTs measured at the hosts for congestion control without touching switches in the network. Recent advances [20] in NIC hardware enable RTT measurements with microsecond-level accuracy, which is sufficient to estimate switch queueing in DCNs. Motivated by this observation, several delay-based transports [18, 20] have been proposed. TIMELY [20] is a rate-based protocol that uses RTT gradients for high throughput and low latency RDMA communications. By contrast, DX [18] is a window-based protocol with an additive increase and a multiplicative decrease that's proportional to the average queuing delay.

Transports based on other signals: Some DCN transport designs (e.g., D3 [28], PDQ [13], and FCP [12]) adjust the sending rate based on explicit congestion feedback from the switch. This line of works is difficult to deploy in production datacenters as they require non-trivial modifications to commodity switch hardware. Hence, they are beyond the discussion scope of this paper.

Other related work: Recently, Zhu et al. [32] have analyzed ECN-based DCQCN and delay-based TIMELY using fluid models and simulations. It suggests that ECN is better than delay as it can guarantee both fairness and a bounded delay. However, their analysis only assumes a single bottleneck. By contrast, our work tries to analyze ECN and delay signals in large multi-hop DCNs with multiple bottlenecks.

3 THE TALE OF TWO SIGNALS

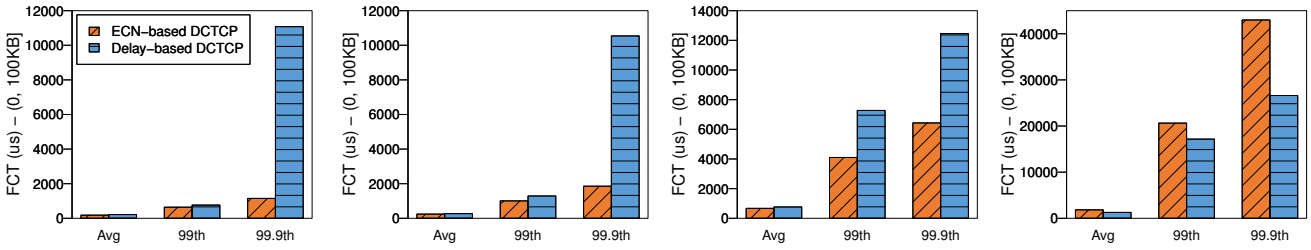
In this section, we begin by introducing the study methodology that we adopt to analyze and compare ECN and delay signals. Then, we explore and show the network performance under congestion control using either ECN or delay signals in a 4-level Fattree [4, 19]. After observing that neither ECN nor delay signal can achieve persistent low flow completion times, we dig into the underlying reasons, which consist of two aspects - end-to-end queueing control for low RTTs and per-hop queueing control for low packet loss and timeouts.

3.1 Delay-based DCTCP

For a generic congestion control system, there are basically three modules: (1) Congestion Measurement; (2) Congestion Control Intelligence; (3) Rate Enforcement. Thus, in order to compare two signals, instead of building two totally different congestion control systems, a better way should be using different signals only for the congestion measurement module while keeping the same congestion control intelligence and rate or congestion window (CWND) enforcement mechanism. In our work, we choose to use the well-studied control laws of DCTCP [5] as the underlying congestion control intelligence. And we adopt the prevalent CWND-based enforcement scheme.

The last step is to transform different signals into uniform outputs of congestion measurement module. For the original ECN-based DCTCP transport, network switches mark ECN on packets when the instant queueing exceeds the preset threshold K. Then receivers feedback the ECN mark with ACKs. The senders count the ECNs in each window time and calculate an ECN fraction as the congestion measurement output. Interestingly, there are actually one-to-one mapping from the instant queueing on congested switches to end-to-end packet delay, measured as round-trip time (RTT) at endhosts, assuming that link capacity is uniform and there is only one congestion point. Therefore, for delay-based DCTCP, we can simply maintain a base RTT for each connection and use an end-to-end packet delay threshold T to act as the corresponding K to identify congested packet. To put it simple, ECN-based DCTCP take ACKs with ECN marks or experience:

$$queue > ecn\_threshold(K) \quad (1)$$



(a) Traffic within single rack (b) Traffic within 2-level pod (c) Traffic within 3-level pod (d) Traffic within 4-level fattree

Figure 1: Small flow completion times under traffic patterns with various network locality

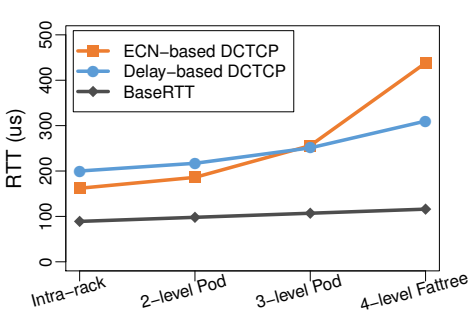
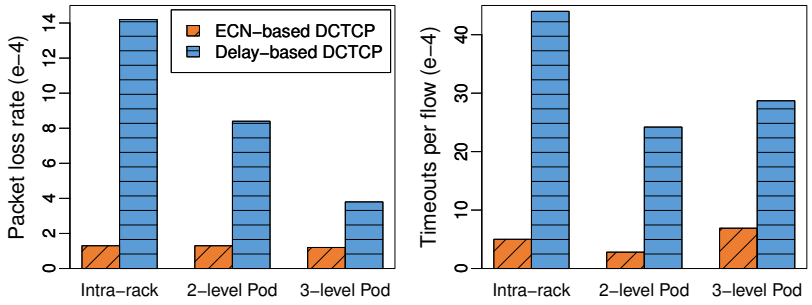


Figure 2: End-to-end queueing control



(a) Packet loss (b) Timeouts

Figure 3: Per-hop queueing control

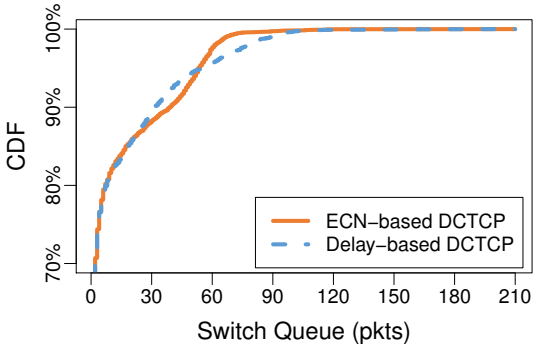


Figure 4: CDF of single switch queue

as congestion indicators while delay-based DCTCP take ACKs with:

$$rtt > base\_rtt + delay\_threshold(T) \quad (2)$$

as congestion indicators. With these indicators, delay-based DCTCP senders also count the congestion indicators in each window time and calculate a congestion indicator fraction as the congestion measurement output.

3.2 Delay-based vs. ECN-based

Our goal is to achieve persistent low latency - more specifically, low small flow completion time (FCT), for data center networks (DCNs). In our simulation, we measure the transport performance under traffic patterns with various network locality[1] in a 4-level Fattree. We set 4 cases: (1) 100% traffic cumulates on single rack; (2) 100% traffic cumulates on 2-level pod; (3) 100% traffic cumulates on 3-level pod; (4) traffic distributes evenly across the whole 4-level Fattree. Detailed simulation settings are described in Section 4.2.

Simulation result is shown in Figure 1. The observation is that neither ECN-based DCTCP nor delay-based DCTCP can achieve persistent low small flow completion times under various network locality traffic patterns: (1) When the network locality traffic dominates within single rack (Figure 1(a) for case (1)) or 2-level pod (Figure 1(b) for case (2)), delay-based DCTCP gets up to 10× and 5× larger 99.9th percentile small flow completion times respectively compared to ECN-based

[1]The term network locality [16] is used to refer to the fact that a task is scheduled on the machine or rack containing most of its input data. Cluster scheduling techniques like [14, 16, 22, 30] suggest that application traffic might cumulate on single rack or 2-level pod even though a 3-tier or 4-tier Clos network is deployed typically.

one; (2) When the network locality traffic reduces (no network locality in Figure 1(d) for case (4)), ECN-based DCTCP degrades dramatically and comes to up to 1.5× larger 99.9th percentile small flow completion times compared to delay-based one. Similar trend can slao be observed with regards to both average and 99th percentile small flow completion times.

3.3 ECN and Delay are Complementary

Based on our simulation results, we make two important observations that highlight the complementary nature of ECN and delay signals.

Claim 1: ECN-based DCTCP cannot well control end-to-end queueing, leading to rapidly increasing RTT as hop count goes up. To compare the end-to-end queueing control efficiency of both signals, we measure the RTTs achieved in the previous experiments and the result is shown in Figure 2. As we expected, both ECN-based and delay-based DCTCP get larger RTTs when network locality traffic reduces because flows tend to traverse more hops and thus potentially more congestion points and queueing. However, when we compare the RTT gradient of these two transports, ECN-based DCTCP has up to 2× larger RTT increasing rate compared with the delay-based one, which leads to worse RTTs and small flow completion times in case (4) shown in previous subsection.

Claim 2: Delay-based DCTCP cannot well control per-hop queueing, leading to severe packet loss and thus severe timeouts. To quantify the per-hop queueing control efficiency, we count the packet loss and timeouts in the previous experiments and the result is shown in Figure 3. We found that delay-based DCTCP can have up to 10× more packet loss and 8× more timeouts compared with ECN-based DCTCP when network locality traffic dominants in single racks. Even for case (3), where average RTT for delay-based DCTCP is 251μs slightly smaller than 256μs of ECN-based one, there are still up to 3× more packet loss and 4× more timeouts. This phenomenon can be well understood with the CDF of single switch queue shown in Figure 4. Even though average queueing for delay-based DCTCP is smaller, the tail is still much larger than ECN-based one, leading to worse packet loss and thus timeouts.

4 PRELIMINARY DESIGN AND EVALUATION

In this section, we first introduce a preliminary version of EAR that simply combines ECN and RTT, and further design considerations will be discussed in Section 6. Then, we evaluate the performance of EAR using ns2 simulations [3]. Given the full design of EAR is still ongoing, our goal here is to show the early promise of combining both signals.

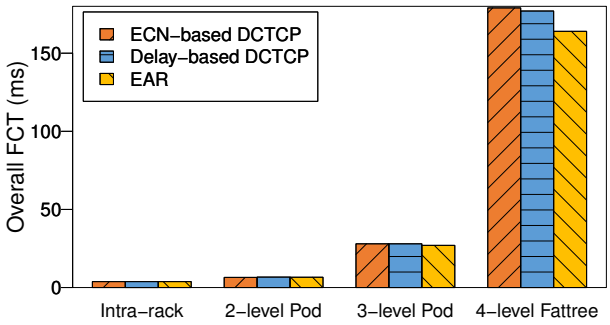


Figure 5: Overall average flow completion times

4.1 Preliminary Design with Combined Signals

From Section 3, we learn that both low packet loss and low RTTs are essential to achieve persistent low small flow completion times, and that neither ECN nor delay signal can maintain both low packet loss and low RTTs simultaneously. Therefore, intuitively, we propose EAR to combine ECN and RTT signals, with ECN signal controlling per-hop queueing and thus packet loss and timeouts, and with RTT signal controlling end-to-end queueing and thus bounded packet round trip times.

Specifically, the EAR senders treat ACKs satisfying either Equation 1 or Equation 2 as congestion indicators. Then following the same way as the original ECN-based DCTCP, the senders sum up the congestion indicator counts in a window time and calculate a congestion indicator fraction as the congestion measurement output. The subsequent congestion control intelligence module adjusts congestion window size based on the well-studied control laws of DCTCP [5]. In particular, when an ACK is received and identified as non congestion indicator, the window size is increased as

$$cwnd = cwnd + 1/cwnd \tag{3}$$

When an ACK is received and identified as congestion indicator, the window size is reduced [2] as

$$cwnd = cwnd \times (1 - \alpha/2) \tag{4}$$

where α is the weighted average of the fraction of marked packets. Other features of TCP such as slow start, fast retransmission when getting 3 duplicate ACKs or fast recovery from packet lost are left unchanged.

4.2 Simulation Settings

Network Topology: We use a 4-tier 6-radix Fattree [4, 19] topology with an oversubscription ratio of 2:1 at the edge

[2]DCTCP cuts its window size at most once per window time [5].

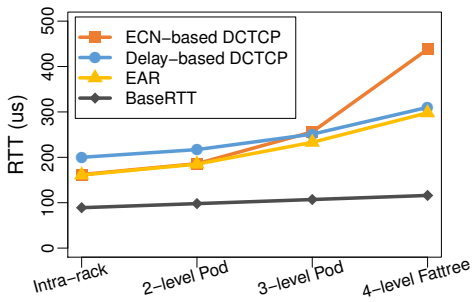
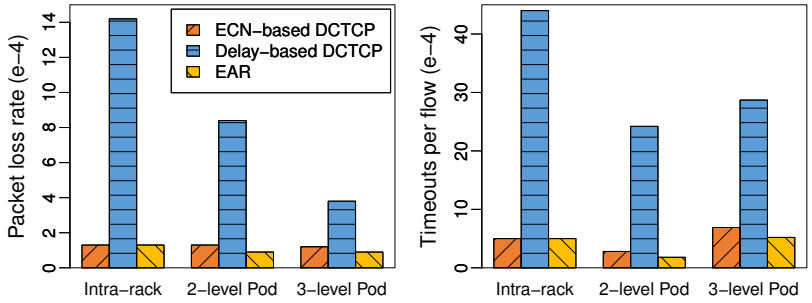
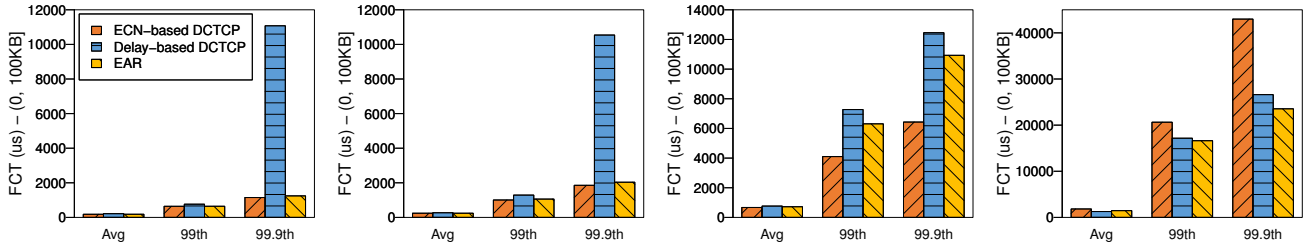


Figure 6: End-to-end queueing control



(a) Packet loss (b) Timeouts

Figure 7: Per-hop queueing control



(a) Traffic within single rack (b) Traffic within 2-level pod (c) Traffic within 3-level pod (d) Traffic within 4-level fattree

Figure 8: Small flow completion times under traffic patterns with various network locality

for all simulations throughout the paper. The topology interconnects 324 hosts through 3 layers of 54 intermediate switches and one layer of 27 core switches. All links are of 10Gbps. The end-to-end round-trip latency (in the absence of queueing) across the core switches (8 hops) is ~116μs, of which ~80μs is spent in the hosts.

Benchmark Workloads: We simulate empirical workloads modeled after traffic patterns that have been observed in a datacenter supporting web search [5]. Flows arrive according to a Poisson process and the source and destination of each flow are chosen according to different network locality (see Section 3.2) patterns.

Transports Compared: We compare ECN-based DCTCP, delay-based DCTCP and our new transport - EAR. ECN marking is based on instantaneous queue length with single threshold K. The delay signal we use is one-way signal measured in a similar way as DX [18]. The queueing delay threshold is set to T.

Parameter Settings: For ECN-based DCTCP, we set the ECN marking threshold K to the recommended value of 65pkts at 10Gbps. For delay-based DCTCP, the congestion indication threshold T should be larger than K so as to tolerate potential multiple bottlenecks. As shown in Figure 5, we recommend to set T=144μs (120pkts at 10Gbps), which maintains similar overall average flow completion times. Note that changing the threshold ±20pkts does not significantly affect the performance of delay-based DCTCP. For EAR, we currently set K=65pkts and T=120pkts. The optimal parameter settings depend on network conditions (e.g., topology, load), which will be further explored in our future work.

4.3 Preliminary Results

End-to-end Queueing Control: As shown in Figure 6, EAR controls end-to-end queueing better compared with both ECN- and delay-based DCTCP. When network locality traffic pattern dominates, ECN signal can control the queueing fairly well as there are only a few hops. When traffic spread evenly across the whole 4-tier Fattree, RTT signal kicks in to keep a bounded round trip time even when ECN does not sense any severe single hop congestion.

Per-hop Queueing Control: As shown in Figure 7, EAR controls per-hop queueing better compared with both ECN- and delay-based DCTCP. There are mainly two reasons. First of all, the average RTT or queueing level is smaller when traffic rate is controlled by two signals. Then, for single switch queueing, when it comes to a relatively large value

	Delay	ECN
End-to-end Queueing Control	✓	✗
Multiple Queue Scenario	✓	✗
Accuracy & Granularity [18]	✓	✗
Per-hop Queueing Control	✗	✓
Multi-bottleneck Fairness	✗	✓
Feedback Delay & Stability [32]	✗	✓

Table 1: Comparison between ECN and delay signals. In this paper, we only discuss End-to-end Queueing Control and Per-hop Queueing Control.

that potentially could lead to buffer overflow and packet loss, ECN signal kicks in to reduce the flow rate before delay signals indicate congestion.

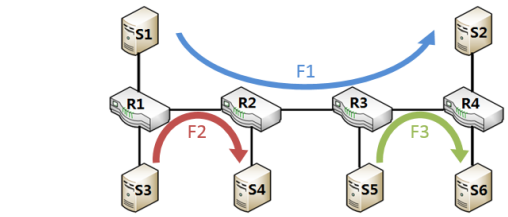
Small Flow Completion Times: As shown in Figure 8, in general, EAR keeps both average and tail small flow completion times low under various network locality traffic patterns. Notice that the compound DCTCP gets larger FCTs than ECN-based DCTCP under traffic cumulating within 3-level pod. We attribute this to the potential unfairness or starvation under multiple bottleneck scenarios, which is discussed in detail in Section 6.

5 DISCUSSION

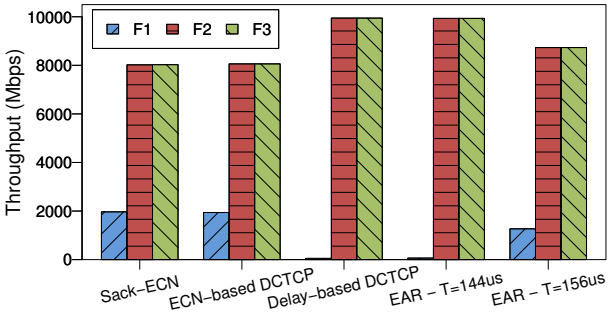
EAR is in its preliminary stage. In this paper, we mainly take advantage of delay signals for end-to-end queueing control and ECN signals for per-hop queueing control. However, there are other differences between these two signals that affect congestion control. Ideally, EAR should be designed to combine all the advantages of each signal. We summarize detailed pros and cons of the two signals in Table 1, and explain them briefly.

Multiple queue scenario is challenging for ECN marking as studied in MQ-ECN [9]. It shows that: 1) per-queue standard ECN suffers from high RTT when there are many active queues; 2) per-queue minimum ECN suffers from low throughput when only a few queues are active; 3) per-port or per service pool ECN suffers from unwanted inter-queue interaction, impairing the scheduling mechanism, e.g., WRR. However, we find that delay signal solves the previous problems naturally as it can directly reflect the queueing delay that each flow experiences (The result is not shown here due to space limitation). Thus, adding delay signal into ECN-based transport is benefitial. Which ECN marking scheme should be used with delay signal is still under exploration.

Accuracy and granularity problem of ECN signal is demonstrated in [18]. It claims that both instant and averaged ECN fraction cannot reflect the network congestion precisely, and that instant ECN fraction even suffers from



(a) Multi-bottleneck topology



(b) Throughput of three flows. F1 is the long flow.

Figure 9: Multi-bottleneck fairness issue

coarse granularity as it can represent only a limited number of congestion levels (no more than the window size). Delay signal can provide more accurate feedback and thus enable precise congestion window adjustment.

Multi-bottleneck fairness issue is illustrated in Figure 9. There are 3 persistent flows. Flow F1 shares bottleneck link R1-R2 with F2, and bottleneck link R3-R4 with F3. We find that even though transports using either ECN or delay signals cannot achieve max-min fair share of the network bandwidth, delay-based one performs much worse. In fact, the window size of F1 is reduced to no more than two packets for the delay-based transports, leading to severe starvation. Note that the relatively small ECN marking threshold helps to control the aggressive short-distance flows and thus alleviates the unfairness issue for EAR. However, for our large scale simulation under high load with dynamic traffic, there are potentially more than two bottlenecks and thus little improvement is observed. Systematic solution for this issue is still under exploration.

Feedback delay and stability issue of delay signal is presented in [32]. It concludes that ECN can use dequeue marking to decouple queueing delay with control feedback delay while delay signals cannot. And the large and fluctuating feedback delay can make the congestion control system less stable. Thus, more timely ECN signal can be used to avoid window adjustment triggered by outdated delay signal, especially under dynamic changing traffic workloads.

6 CONCLUSION AND FUTURE WORK

There is a vast literature on congestion control for both Internet and DCNs. From a congestion feedback's perspective, this paper focuses on ECN and RTT signals in DCN context. Through simulations under large DCN topologies, this paper shows that both ECN and RTT are indispensable to achieve low flow completion times. This motivates the new transport design - EAR, that hears and reacts to both signals. Evaluation results show that EAR can deliver more persistent low flow completion times under various scenarios.

For future work direction, we consider to extend EAR to inter-datacenter networks. For traffic between two datacenters, there are additional advantages of delay signal. Specifically, delay 1) guarantees bounded latency under large hop counts, in the presence of large propagation delay, (2) has better compatibility – not all devices in WAN support ECN marking. Meanwhile, the requirement of precise RTT measurement, *e.g.,* microsecond level for intra-datacenter, can be relaxed. However, using *only* delay has one more challenge than intra-datacenter – the heterogeneity of path segments. Take a path from DC *A* to WAN to DC *B* as an example. Because of large propagation and queueing delay in WAN, the protocol must set a large RTT threshold for inter-datacenter traffic. This makes transport more likely to drop packet inside DC *A* and *B*, since switches inside datacenters typically have shallow buffer and high link rates. Therefore, we need an ECN portion of the transport to control packet loss in DC *A* and *B*. We envision this can be incrementally deployed, because single administration domain (DC *A*, *B*, and the WAN between them) makes it easy to handle coexisting problem with other transports.

REFERENCES

[1] *Google Data Centers.* https://www.google.com/about/datacenters.

[2] *Microsoft Datacenter Infrastructure.* https://www.microsoft.com/en-us/cloud-platform/global-datacenters.

[3] *The Network Simulator NS-2.* http://www.isi.edu/nsnam/ns.

[4] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A scalable, commodity data center network architecture. In *SIGCOMM*.

[5] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data center tcp (dctcp). In *SIGCOMM*.

[6] Mohammad Alizadeh, Abdul Kabbani, Tom Edsall, Balaji Prabhakar, Amin Vahdat, and Masato Yasuda. 2012. Less is more: trading a little bandwidth for ultra-low latency in the data center. In *NSDI*.

[7] Mohammad Alizadeh, Shuang Yang, Milad Sharif, Sachin Katti, Nick McKeown, Balaji Prabhakar, and Scott Shenker. 2013. pfabric: Minimal near-optimal datacenter transport. In *SIGCOMM*.

[8] Wei Bai, Kai Chen, Hao Wang, Li Chen, Dongsu Han, and Chen Tian. 2015. Information-Agnostic Flow Scheduling for Commodity Data Centers. In *NSDI*.

[9] Wei Bai, Li Chen, Kai Chen, and Haitao Wu. 2016. Enabling ecn in multi-service multi-queue data centers. In *NSDI*.

[10] Sally Floyd. 2003. HighSpeed TCP for large congestion windows. In *RFC 3649*.

[11] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. In *SIGOPS*.

[12] Dongsu Han, Robert Grandl, Aditya Akella, and Srinivasan Seshan. 2013. FCP: A Flexible Transport Framework for Accommodating Diversity. In *SIGCOMM*.

[13] Chi-Yao Hong, Matthew Caesar, and P Godfrey. 2012. Finishing flows quickly with preemptive scheduling. In *SIGCOMM*.

[14] Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar, and Andrew Goldberg. 2009. Quincy: fair scheduling for distributed computing clusters. In *SOSP*.

[15] Van Jacobson. 1988. Congestion avoidance and control. In *SIGCOMM*.

[16] Virajith Jalaparti, Peter Bodik, Ishai Menache, Sriram Rao, Konstantin Makarychev, and Matthew Caesar. 2015. Network-aware scheduling for data-parallel jobs: Plan when you can. In *SIGCOMM*.

[17] Glenn Judd. 2015. Attaining the Promise and Avoiding the Pitfalls of TCP in the Datacenter. In *NSDI*.

[18] Changhyun Lee, Chunjong Park, Keon Jang, Sue Moon, and Dongsu Han. 2016. DX: Latency-Based Congestion Control for Datacenters. In *ToN*.

[19] Vincent Liu, Daniel Halperin, Arvind Krishnamurthy, and Thomas E Anderson. 2013. F10: A Fault-Tolerant Engineered Network. In *NSDI*.

[20] Radhika Mittal, Nandita Dukkipati, Emily Blem, Hassan Wassel, Monia Ghobadi, Amin Vahdat, Yaogong Wang, David Wetherall, David Zats, and others. 2015. TIMELY: RTT-based Congestion Control for the Datacenter. In *SIGCOMM*.

[21] Ali Munir, Ghufran Baig, Syed M Irteza, Ihsan A Qazi, Alex X Liu, and Fahad R Dogar. 2014. Friends, not foes: synthesizing existing transport strategies for data center networks. In *SIGCOMM*.

[22] Ali Munir, Ting He, Ramya Raghavendra, Franck Le, and Alex X Liu. 2016. Network Scheduling Aware Task Placement in Datacenters. In *CoNEXT*.

[23] Ali Munir, Ihsan A Qazi, Zartash A Uzmi, Aisha Mushtaq, Saad N Ismail, M Safdar Iqbal, and Basma Khan. 2013. Minimizing flow completion times in data centers. In *INFOCOM*.

[24] Rong Pan, Balaji Prabhakar, and Ashvin Laxmikantha. 2007. QCN: Quantized congestion notification. In *IEEE802*.

[25] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C Snoeren. 2015. Inside the social network's (datacenter) network. In *SIGCOMM*.

[26] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, and others. 2015. Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network. In *SIGCOMM*.

[27] Balajee Vamanan, Jahangir Hasan, and TN Vijaykumar. 2012. Deadline-aware datacenter tcp (d2tcp). In *SIGCOMM*.

[28] Christo Wilson, Hitesh Ballani, Thomas Karagiannis, and Ant Rowtron. 2011. Better never than late: Meeting deadlines in datacenter networks. In *SIGCOMM*.

[29] Lisong Xu, Khaled Harfoush, and Injong Rhee. 2004. Binary increase congestion control (BIC) for fast long-distance networks. In *INFOCOM*.

[30] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. 2010. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In *EuroSys*.

[31] Yibo Zhu, Haggai Eran, Daniel Firestone, Chuanxiong Guo, Marina Lipshteyn, Yehonatan Liron, Jitendra Padhye, Shachar Raindel, Mohamad Haj Yahia, and Ming Zhang. 2015. Congestion control for large-scale RDMA deployments. In *SIGCOMM*.

[32] Yibo Zhu, Monia Ghobadi, Vishal Misra, and Jitendra Padhye. 2016. ECN or Delay: Lessons Learnt from Analysis of DCQCN and TIMELY. In *CoNEXT*.