



Standardization of Low-Latency TCP with Explicit Congestion Notification: A Survey

Junzhou Luo, Jiahui Jin, and Feng Shan • Southeast University, China

Explicit Congestion Notification (ECN) is an extension to TCP/IP that helps realize low latency in TCP. This article presents a thorough survey of IETF standardization efforts using ECN-based solutions, and also proposes suggestions and future directions on this topic.

Network latency for a packet is defined as the transmission duration from its sender to its receiver. It has a tremendous impact on Internet applications and services, such as Web search, social networks, and the emerging network-based applications in industry, economics, and healthcare. Network latency affects user experience, providers' revenue, and the quality of service; yet the Transmission Control Protocol (TCP), as the core networking protocol, has been optimized primarily for reliable packet delivery rather than timely packet delivery. Therefore, reducing network latency for TCP-based time-sensitive applications has been quite challenging.

The Industrial Internet (see www.iiconsortium.org) is an example that illustrates the challenge of low latency. The Industrial Internet is a highly cooperative global network, which connects machines, an environment, and humans together for smart manufacturing. As Figure 1 shows, control systems heavily rely on low-latency networks for supporting machine-machine communications, massive sensor data transfer, and real-time decision delivery, where latency below tens or hundreds of milliseconds is required by systems in order to support fast control.¹ With respect to most applications in the Industrial Internet, TCP has been the most widely adopted protocol for their enhanced reliability, while unexpectedly long latencies – which might be 10 times larger than the required latency – also might occur under certain scenarios.

The Internet Engineering Task Force (IETF) has made great strides on standards for low-latency networking in the last two decades. For example, the Real-Time Protocol (RTP) and Real-Time Streaming Protocol (RTSP) were proposed for multimedia applications (such as video and voice over IP), and the DiffServ protocol was designed for reducing TCP's latency by prioritizing TCP flows. However, these standards might not be applied readily to reduce the latencies for all TCP-based applications. On the one hand, the application-layer protocols, such as RTP, were designed for specific applications and could hardly be adopted to general scenarios. On the other hand, the protocols designed for the more general scenarios, such as DiffServ, sacrifice the latency of low-priority TCP flows and might not work well if most TCP flows have stringent latency requirements.

In recent years, the standards designed for low-latency TCP have garnered much attention from the networking research community. As recommended by the IETF, Explicit Congestion Notification (ECN)² – an extension to TCP/IP that will help realize low-latency TCP – should be widely deployed across the Internet. By applying ECN and its extensions, it's possible to (nearly) eliminate packet loss, shorten the buildup of queues in routers/switches, and design fine-grained TCP congestion control mechanisms. Such optimizations could greatly benefit TCP-based applications in terms of latency reduction and throughput improvement.

Related Internet RFCs

Recently, the IETF has proposed a series of Explicit Congestion Notification (ECN)-related RFCs and Internet drafts (I-Ds), and recommends widely deploying Active Queue Management (AQM) and ECN to improve the performance of today's Internet. In Table A, we list the latest RFCs and Internet drafts of AQM, ECN, and improved ECN, which were published in the last two years.

RFC/I-D for AQM. The IETF published RFC 7567 in 2015, which presents a strong recommendation for testing, standardization, and widespread deployment of AQM.¹ The IETF also published RFC 7928, which contains the guidelines to help quantify performance of AQM policies in terms of latency reduction, throughput maximization, and the tradeoff between these two.²

RFC/I-D for ECN. Because ECN is rarely deployed in the current Internet, the IETF recommends enabling ECN in the AQM routers and switches. A recent Internet draft lists the benefits of using ECN, which include not only the reductions in packet loss and latency, but also the improvement in throughput and reliability.³ The IETF community also shows the necessity and feasibility of extending the applications of ECN, and proposed the Internet drafts to enable ECN in TCP control packets⁴ and IP tunnel packets.⁵

RFC/I-D for Improved ECN. The IETF is working on novel transport mechanisms by improving ECN. The improved ECN redefines the responses to the reception of the ECN-Echo (ECE) flag for better congestion control,⁷ which allows senders to infer the congestion condition more accurately⁶ and reduce the congestion window more smoothly.^{8,9} Data Center TCP (DCTCP)⁹ and the Low-Latency, Low-Loss, and Scalable Throughput (L4S) framework,¹⁰ as the applications of the improved ECN, are being standardized by the IETF.

References

1. F. Baker and G. Fairhurst, *IETF Recommendations Regarding Active Queue Management*, IETF RFC 7567, July 2015; www.rfc-editor.org/rfc/rfc7567.txt.
2. N. Kuhn et al., *Characterization Guidelines for Active Queue Management (AQM)*, IETF RFC 7928, July 2016; www.rfc-editor.org/rfc/rfc7928.txt.
3. G. Fairhurst and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)," IETF Internet draft, work in progress, Nov. 2015.
4. M. Bagnulo and B. Briscoe, "Adding Explicit Congestion Notification (ECN) to TCP Control Packets," IETF Internet draft, work in progress, July 2016.
5. B. Briscoe, J. Kaippallimalil, and P. Thaler, "Guidelines for Adding Congestion Notification to Protocols That Encapsulate IP," IETF Internet draft, work in progress, July 2016.
6. M. Kuehlewind, R. Scheffenegger, and B. Briscoe, *Problem Statement and Requirements for Increased Accuracy in Explicit Congestion Notification (ECN) Feedback*, IETF RFC 7560, Aug. 2015; www.rfc-editor.org/rfc/rfc7560.txt.
7. N. Khademi et al., "Updating the Explicit Congestion Notification (ECN) Congestion Control Response," IETF Internet draft, work in progress, May 2016.
8. N. Khademi et al., "TCP Alternative Backoff with ECN (ABE)," IETF Internet draft, work in progress, May 2016.
9. S. Bensley et al., "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters," IETF Internet draft, work in progress, July 2016.
10. B. Briscoe, K. D. Schepper, and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Problem Statement," IETF Internet draft, work in progress, June 2016.
11. K.D. Schepper et al., "DualQ Coupled AQM for Low Latency, Low Loss, and Scalable Throughput," IETF Internet draft, work in progress, Mar. 2016.
12. K.D. Schepper, B. Briscoe, and I.J. Tsang, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay," IETF Internet draft, work in progress, Mar. 2016.

Table A. Explicit Congestion Notification (ECN)-related IETF RFCs and Internet drafts since 2015.

Topic	RFC and Internet drafts (since 2015)	Description	RFC category
Active Queue Management (AQM)	RFC 7567 ¹	IETF recommendations regarding AQM	Best current practice
	RFC 7928 ²	Characterization guidelines for AQM	Informational
ECN	draft-ietf-aqm-ecn-benefits ³	Benefits of ECN	Informational
	draft-bagnulo-tsvwg-generalized-ecn ⁴	ECN for TCP control packets	Informational
	draft-ietf-tsvwg-ecn-encap-guidelines ⁵	ECN for protocols that encapsulate IP	Best current practice
Improved ECN	RFC 7560 ⁶	Requirements of more accurate ECN feedbacks	Informational
	draft-khademi-tsvwg-ecn-response ⁷	Updates of ECN congestion control response	Standards track
	draft-khademi-alternativebackoff-ecn ⁸	TCP alternative backoff with ECN	Experimental
	draft-ietf-tcpm-dctcp ⁹	Data Center TCP	Informational
	draft-briscoe-tsvwg-aqm-tcpm-rmcat-l4s-problem ¹⁰	Problem statement of the Low-Latency, Low-Loss, and Scalable Throughput (L4S) framework	Informational
	draft-briscoe-aqm-dualq-coupled ¹¹	DualQ coupled AQM for L4S	Standards track
	draft-briscoe-tsvwg-ecn-l4s-id ¹²	Identifier of DCTCP-like flow for L4S	Experimental

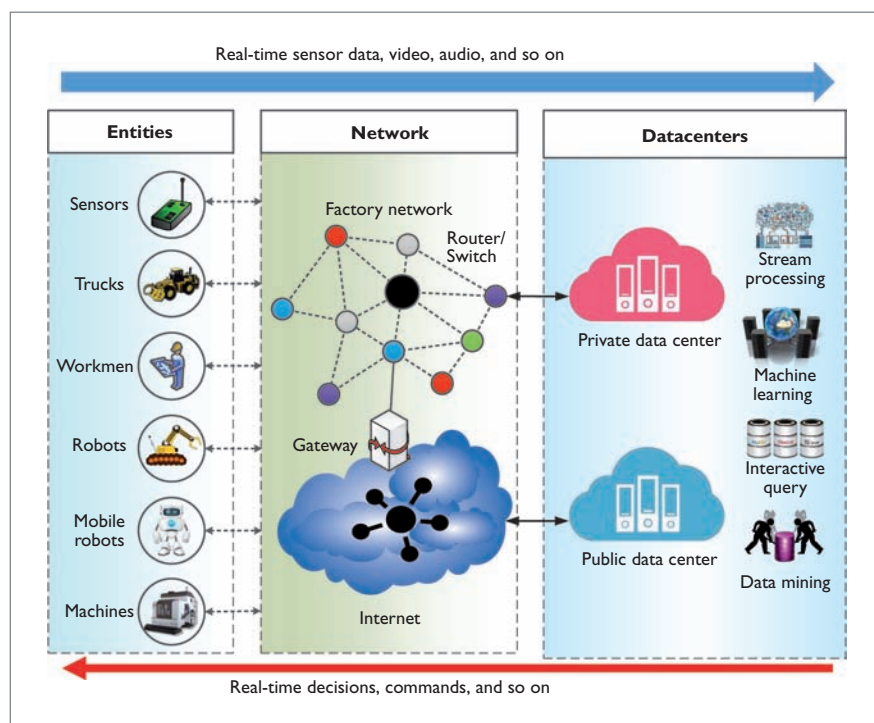


Figure 1. Time-sensitive applications in the Industrial Internet. The control systems require tens or hundreds of milliseconds of latency for fast control, but TCP's latency can be up to several seconds.

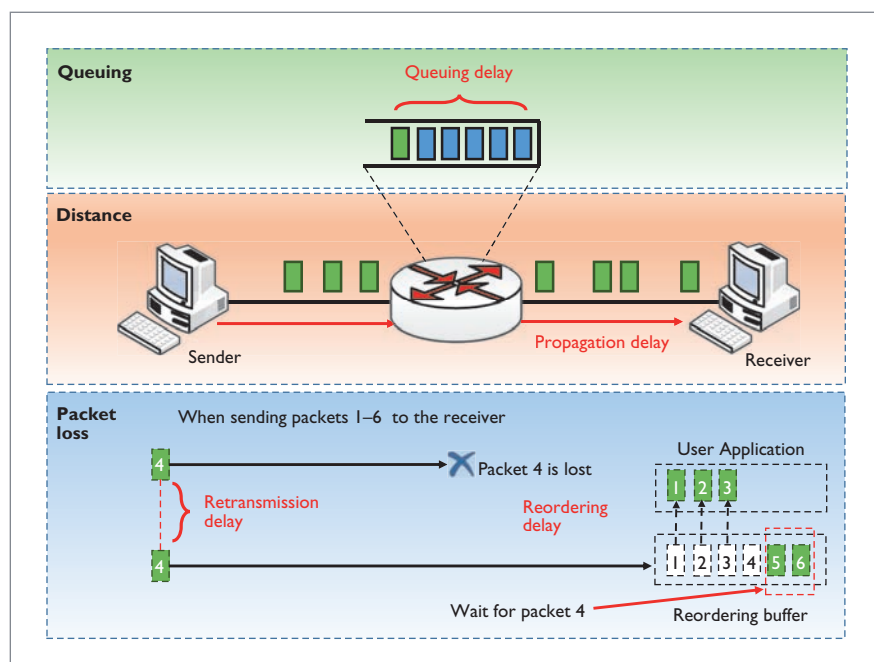


Figure 2. Primary causes of TCP's latency. The latency of a TCP packet mainly stems from three main sources: distance, queuing, and packet loss, including propagation delay, queuing delay, retransmission delay, and reordering delay.

With this in mind, here we present a short survey on ECN-based solutions standardized by the IETF for low-latency TCP. By analyzing the causes of TCP packets' latencies, we show how ECN and its extensions would significantly reduce such latencies. We also present the recent ECN-related RFCs and Internet drafts on this topic. In addition, we elaborate our suggestions to improve the ECN-based low-latency TCP.

Causes of TCP's Latency

The latency of a TCP packet stems from three main sources: distance, queuing, and packet loss (see Figure 2). We should note that the terms *delay* and *latency* will be used interchangeably in the following sections.

Distance

Distance causes propagation delay, which is the amount of time it takes for the bits to travel from the sender to the receiver over the transport medium. The typical inter-country propagation delay can be around 100 milliseconds. As it's only determined by distance and propagation speed, the propagation delay is stable, predictable, and inevitable, and can be reduced by shortening the distance between the sender and receiver.

Queuing

A router/switch causes queuing delay, which is the time a packet is waiting in the queue until it can be handled. Queuing is normal in a packet-switched network and is often necessary to absorb surges of bursty traffic and reshape the traffic patterns. However, it leads to unwanted delay in the meantime. The queuing delay in a path containing tens of routers/switches typically varies from a few milliseconds to hundreds of milliseconds.

Packet Loss

A packet loss can trigger the expiration of its associated retransmission timeout or activation of the fast-retransmission of TCP. Both would cause a delay, namely

retransmission delay, to recover the lost packet. According to the default value of retransmission timeout, the retransmission delay can be more than 200 hundred milliseconds.

Packet loss also causes reordering delay, which is the time TCP uses to enhance in-order delivery. When a packet is lost and then retransmitted, its subsequent packets may arrive at the receiver earlier, which breaks the order of packets. For in-order delivery, TCP requires these subsequent packets to stay in the system buffer until the delayed packet has been received.

Unlike the propagation delay, the latencies caused by queuing and packet loss can be unexpectedly long, and might be tens or hundreds of times larger than the propagation delay, especially when the network is congested. Nevertheless, the latter latencies can be (nearly) avoided by reducing the possibility of queuing and packet loss.

Explicit Congestion Notification

Explicit Congestion Notification (ECN; RFC 3168)² is one of the feasible solutions to reduce the latency of TCP. The conventional TCP/IP networks usually have high latency, since they notify network congestion by dropping packets, where packet loss is inevitable. ECN is an extension to TCP/IP, which allows end-to-end notification of network congestion without dropping packets.

The ECN-capable networks notify network congestion by setting marks in the TCP/IP headers of packets. Figure 3 shows the definition of the ECN field in the IP and TCP headers. In the IP header, two bits of the DiffServ field are used to encode four code points for three situations: Non-ECN-Capable Transport, or Non-ECT; ECN-Capable Transport, or ECT(0) and ECT(1); and Congestion Encountered (CE). In the TCP header, two one-bit ECN flags are defined. The bits, namely ECN-Echo (ECE) and Congestion Window Reduced (CWR), are used to echo back the con-

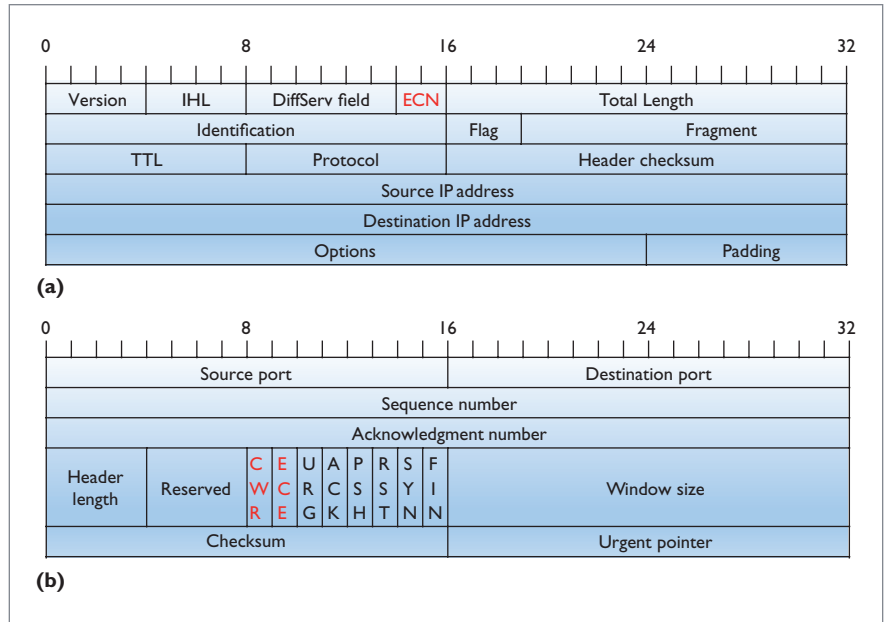


Figure 3. Definitions of Explicit Congestion Notification (ECN) in IP and TCP headers. (a) ECN in IP header and (b) ECN in TCP header. IHL = Internet Header Length, and TTL = Time to Live.

gestion indication and to acknowledge that the congestion-indication echoes are received, respectively.

The ECN-capable networks use Active Queue Management (AQM, RFC 2309)³ to decide whether and when to set the ECN marks. By working together with AQM, the ECN-capable networks can largely reduce the possibility of queuing and packet loss, so the TCP latency is reduced accordingly. Next, we show how to achieve low latency by using ECN and AQM in detail.

Achieving Low Latency by ECN

We first briefly introduce AQM, and then show how to achieve low latency by using ECN. AQM provides a general framework to signal the congestion information to TCP before the buffer of the router or switch is actually full. RFC 2309 describes a specific AQM algorithm, Random Early Detection (RED), and recommends using it as the default AQM algorithm in routers/switches. Specifically, when RED is deployed, the routers and switches will set thresholds and randomly drop

(or mark) the packets in the queue as a signal of congestion for TCP when the queue length exceeds the thresholds, so the sender can slow down the sending rate before the congestion becomes more severe. As a result, AQM adeptly controls the queue length.

Combining with AQM, ECN allows TCP/IP networks to avoid congestion without incurring packet loss. Instead of dropping packets, the router/switch changes the codepoint in the IP header to CE when congestion is detected. Upon receipt of a packet marked with CE, the receiver sets the ECE flag in the following ACKs to signal the congestion. When the sender receives an ACK with an ECE flag, it knows that a congestion is occurring on the path, and then slows down the packet-sending rate to avoid further congestion. After the sending rate is adjusted, the sender sets the CWR flags in the sent packets to notify the receiver to clear the ECE flags. This procedure eliminates the time for recovering lost packets, so in effect, it largely reduces the latency of TCP packet.

However, the ECN-based transport mechanisms defined in RFC 3168 can't

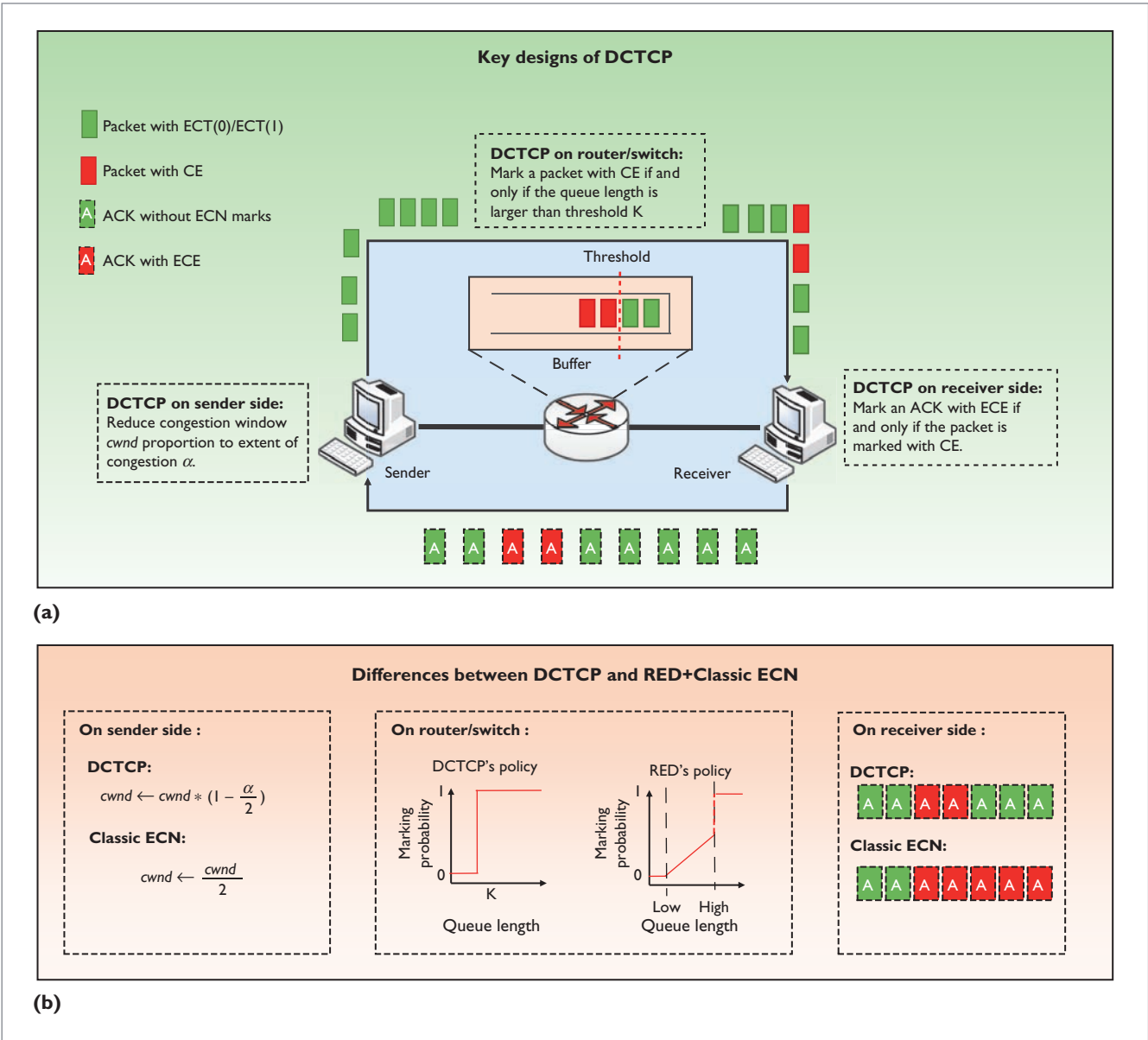


Figure 4. Overview of Data Center TCP (DCTCP). (a) Key designs of DCTCP and (b) key differences between DCTCP and the classic ECN scheme. CE = Congestion Encountered and ECT = ECN-Capable Transport.

achieve low latency and high throughput simultaneously. This is because ECN must work together with an AQM algorithm, but the default AQM algorithm – that is, RED – is sensitive to its parameters and is hard to configure. If the parameters are set incorrectly, the throughput would decrease significantly. Moreover, no matter how good the AQM, the big sawtooth rate of TCP will either cause queuing delay to vary or cause the link

to be underutilized. For these purposes, standardizing novel transport mechanisms is necessary to address this problem.

Achieving Ultra-Low Latency and High Throughput by Improved ECN

In recent years, transport mechanisms with lower latency and higher throughput have attracted the network community's attention. These

mechanisms are critical for the future Internet, where network traffic across multiple data centers for real-time data analytics is expected to constitute a major portion of Internet traffic.

To meet such demand, ECN has been extended and improved, which provides opportunities for designing novel TCP mechanisms to support the future Internet. Some of the novel TCP mechanisms were originally

designed and evaluated for applications deployed inside data centers. Because of their effectiveness, the IETF community standardizes these novel TCP mechanisms and extends them to support scenarios outside data centers.

Inside Data Centers

Data Center TCP (DCTCP) is one representative of the state-of-the-art congestion control mechanisms based on the improved ECN. It was originally published in 2010,⁴ and is being standardized by the IETF TCP Maintenance and Minor Extensions (TCPM) working group.⁵ DCTCP accurately infers network congestion by using the information presented in the sequence of ECN flags, and reacts to congestion in proportion to the extent of congestion. Figure 4 shows the key designs of DCTCP, which contain three main components on the sides of the switch, receiver, and sender, respectively.

For a switch. The AQM algorithm of DCTCP is extremely simple, and there's only one parameter to be set: the queue length threshold K . Whenever a packet arrives at the queue, its CE flag is set if the current queue length exceeds K ; otherwise, the CE flag is unset. This scheme is a simplified variant of the RED algorithm and thus it's already widely supported by modern switches. As opposed to RED, whose parameters are hard to be set correctly, the parameter of DCTCP's AQM algorithm are configured easily.

For a receiver. When a packet with the CE flag arrives at the receiver, the receiver acknowledges this packet by setting the ECE flag in the corresponding ACK, whereas for packets without CE flags, the ECE flags won't be set in the ACKs. This is different from the actions defined by RFC 3168, which requires the receiver to keep setting ECE flags in all subsequent ACKs until the packets with CWR flags have been received.

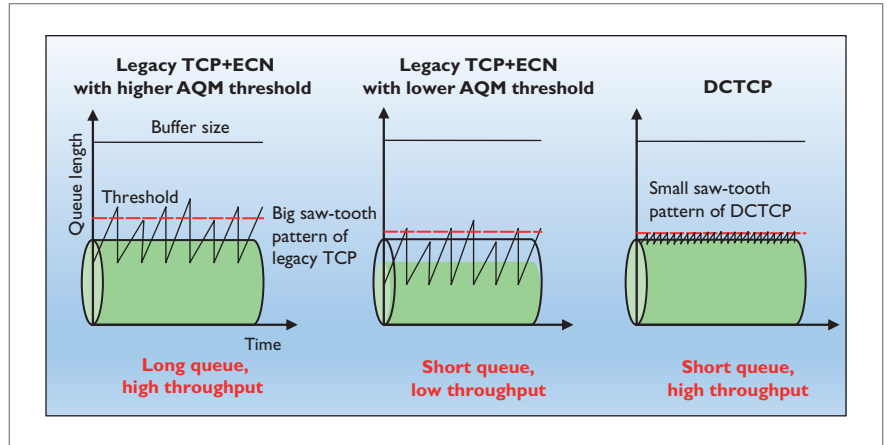


Figure 5. Comparing DCTCP with legacy TCP in terms of queue length and throughput.⁶ DCTCP achieves a higher throughput than the legacy TCP when they're using the same Active Queue Management (AQM) thresholds.

For a sender. After having received all the ACKs of the last congestion window, the sender computes the portion of the ACKs that have ECE flags and uses the portion to estimate the extent of congestion (denoted by α). Then, DCTCP updates the congestion window ($cwnd$) by the updating function: $cwnd = cwnd * (1 - \alpha/2)$.

Different from the legacy TCP that halves the congestion window on receipt of ECE flags, DCTCP sets the reduction of $cwnd$ proportional to α , which is smoother. In heavily congested networks, the value of α increases to 1 and the reduction of $cwnd$ is the same as the legacy TCP; while in slightly congested networks, the value of α is close to zero and $cwnd$ is reduced slightly.

DCTCP is more effective than the legacy TCP in terms of reducing queue length and improving throughput. The legacy TCP has a big saw-tooth pattern, where the sending rate is halved periodically due to the signals of congestion. To achieve high throughput, it requires the threshold of AQM to be large enough. As a result, the legacy TCP will either cause queuing delay to vary or cause the link to be underutilized. Different from the legacy TCP, DCTCP has a smaller saw-tooth pattern because the sending rate only decreases slightly

when the incipient congestion occurs. A small AQM threshold is enough for DCTCP to achieve high throughput, so DCTCP is able to achieve ultra-low latency. Figure 5 compares DCTCP with the legacy TCP in terms of queue length and throughput, where DCTCP achieves short queue length and high throughput simultaneously.

Outside Data Centers

When deploying DCTCP outside data centers, it's important to ensure that DCTCP is safe for the public environment. For example, when DCTCP flows and legacy TCP flows coexist, the flow-rate control of DCTCP can be much more aggressive than that of the legacy TCP because DCTCP doesn't halve the sending rate on congestion. This could result in a scenario where the DCTCP flows use up all the available bandwidth and starve the legacy TCP flows. To address this issue, the IETF community is currently working on a framework, namely the Low-Latency, Low-Loss, and Scalable Throughput (L4S) Internet service, to support a set of DCTCP-like congestion control mechanisms outside data centers.

The core of the L4S framework is the DualQ Coupled AQM algorithm (*DualQ* for short).⁷ In general,

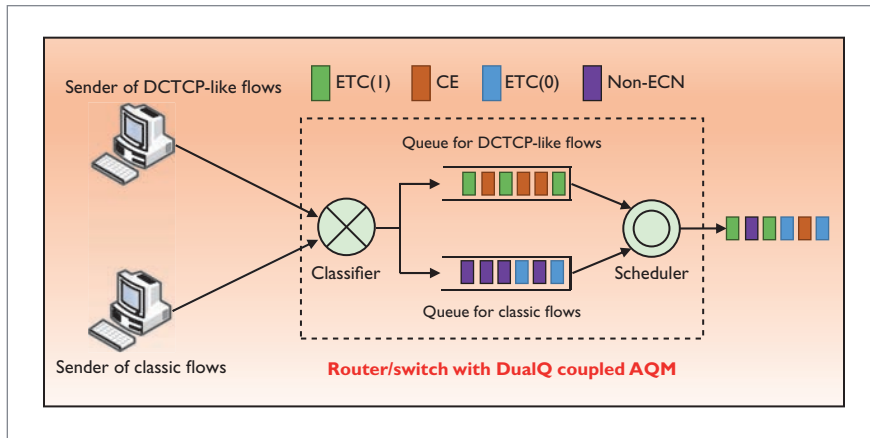


Figure 6. The DualQ Coupled AQM of the Low-Latency, Low-Loss, and Scalable Throughput (L4S) framework. The DCTCP-like flows and the classic flows are put in different queues by the classifier and are coordinated by the scheduler.

DualQ arranges incoming flows into two separate queues – that is, a queue for DCTCP-like flows and a queue for classic flows (for example, TCP Reno/Cubic) that are managed by the DCTCP AQM algorithm and the traditional AQM algorithms (such as RED), respectively.

As Figure 6 shows, the design of DualQ contains two main components: a classifier and a scheduler. The classifier puts a packet into the queue for DCTCP flows if its ECT(1) or CE flag is marked,⁸ while all other packets are put into the queue for classic flows. ECT(1) is recommended by L4S to be used as an identifier for DCTCP-like packets, and ECT(0) is (already) recommended by RFC 3168 to be used as an identifier for the classic ECN used by the legacy TCP. The scheduler grants higher priority to the queue for DCTCP-like flows to ensure that the DCTCP-like flows have low latency. In order to make a DCTCP-like flow run at roughly the same rate as a classic flow, the scheduler decides the dropping probability of the queue for classic flows to be proportional to the square of the marking probability of the queue for DCTCP-like flows. As the marking probability of DCTCP-like flow is much higher than the dropping prob-

ability of classic flow, the DCTCP-like flows would use the available bandwidth less aggressively, so that the fairness between the flows could be enhanced.

Our Suggestions

Low-latency TCP based on ECN is an effective way to support time-sensitive applications. For timesensitive applications, we suggest using low-latency TCP instead of UDP in public networks. Public networks, as with the Industrial Internet, can be heterogeneous and dynamic, consisting of wired or wireless devices that communicate through local-area or wide-area networks. If the applications are deployed in wide-area or wireless networks where packet loss isn't uncommon, tuning and optimizing the UDP-based applications for high throughput, high reliability, and low latency might be very difficult. In such scenarios, the low-latency TCP based on ECN is an easier and more reliable solution.

We also suggest that low-latency TCP should take wireless networks into account. For example, robots in the Industrial Internet move frequently and communicate with other robots or machines through wireless networks. For fast control, robots

need low-latency wireless networks. However, DCTCP might not work well in wireless networks, as packet loss might occur. If ACKs are lost in wireless networks, the DCTCP congestion control mechanism might be invalid because DCTCP's performance is highly affected by the accuracy of the congestion estimation inferred from the sequence of ACKs' ECE flags. For these purposes, we recommend the extensions of DCTCP be optimized for wireless networks.

We've discussed the IETF's efforts on standardizing low-latency TCP with ECN (see also the related sidebar). These efforts include recommendations on widely deploying AQM and ECN, and standardizing novel TCP congestion control mechanisms. We hope to see these novel low-latency TCP mechanisms successfully applied to the public Internet, which would largely benefit the Industrial Internet and other time-sensitive Internet applications. □

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grants 61632008, 61320106007, 61572129, 61502097, 61370207, 61532013, and 61602112; the International S&T Cooperation Program of China 2015DFA10490; the Jiangsu research prospective joint research project under grant BY2013073-01; the Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201; the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under grant 93K-9; and partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization and the Collaborative Innovation Center of Wireless Communications Technology.

References

1. K. Pister et al., *Industrial Routing Requirements in Low-Power and Lossy Networks*, IETF RFC 5673, Oct. 2009; www.rfc-editor.org/rfc/rfc5673.txt.

2. K. Ramakrishnan, S. Floyd, and D. Black, *The Addition of Explicit Congestion Notification (ECN) to IP*, IETF RFC 3168, Sept. 2001; www.rfc-editor.org/rfc/rfc3168.txt.
3. B. Braden et al., *Recommendations on Queue Management and Congestion Avoidance in the Internet*, IETF RFC 2309, Apr. 1998; www.rfc-editor.org/rfc/rfc2309.txt.
4. M. Alizadeh et al., "Data Center TCP (DCTCP)," *Proc. ACM Sigcomm 2010 Conf.*, 2010, pp. 63–74.
5. S. Bensley et al., "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters," IETF Internet draft, work in progress, July 2016.
6. K. De Schepper et al., "Data Center to the Home": Ultra-Low Latency for All, tech. report, RITE project, June 2015; www.simula.no/file/dctthpreprintpdf/download.
7. K.D. Schepper et al., "DualQ Coupled AQM for Low Latency, Low Loss, and Scalable Throughput," IETF Internet draft, work in progress, Mar. 2016.
8. K.D. Schepper, B. Briscoe, and I.J. Tsang, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay," IETF Internet draft, work in progress, Mar. 2016.

Junzhou Luo is a full professor in the School of Computer Science and Engineering, Southeast University, Nanjing, China. Luo has a PhD in computer networking from Southeast University. His research interests include next-generation networks, protocol engineering, network security and management, cloud computing, and wireless LAN. He's a member of IEEE and ACM, and co-chair of the IEEE Systems, Man, and Cybernetics (SMC) Technical Committee on Computer-Supported Cooperative Work in Design. Contact him at jluo@seu.edu.cn.

Jiahui Jin is an assistant professor in the School of Computer Science and Engineering, Southeast University. His current research focuses on large-scale data processing, data center networking, and parallel task scheduling. Jin has a PhD in computer sci-

ence from Southeast University. Contact him at jjin@seu.edu.cn.

Feng Shan is an assistant professor in the School of Computer Science and Engineering, Southeast University. His research interests include energy harvesting, wireless power transfer, and algorithm design and analysis. Shan has a PhD in computer science from Southeast University. Contact him at shanfeng@seu.edu.cn.

myCS

Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>.

Recognizing Excellence in High Performance Computing

Nominations are Solicited for the

SEYMOUR CRAY, SIDNEY FERNBACH, & KEN KENNEDY AWARDS

SEYMOUR CRAY COMPUTER ENGINEERING AWARD

Established in late 1997 in memory of Seymour Cray, the Seymour Cray Award is awarded to recognize innovative contributions to high performance computing systems that best exemplify the creative spirit demonstrated by Seymour Cray. The award consists of a crystal memento and honorarium of US\$10,000. This award requires 3 endorsements.



Deadline: 1 July 2017

All nomination details available at <http://awards.computer.org>



SIDNEY FERNBACH MEMORIAL AWARD

Established in 1992 by the Board of Governors of the IEEE Computer Society. It honors the memory of the late Dr. Sidney Fernbach, one of the pioneers on the development and application of high performance computers for the solution of large computational problems. The award, which consists of a certificate and a US\$2,000 honorarium, is presented annually to an individual for "an outstanding contribution in the application of high performance computers using innovative approaches." This award requires 3 endorsements.

ACM/IEEE-CS KEN KENNEDY AWARD

Established in memory of Ken Kennedy, the founder of Rice University's nationally ranked computer science program and one of the world's foremost experts on high-performance computing. A certificate and US\$5,000 honorarium are awarded jointly by the ACM and the IEEE Computer Society for outstanding contributions to programmability or productivity in high performance computing together with significant community service or mentoring contributions. This award requires 2 endorsements.



IEEE computer society

