TSVWG
Internet-Draft
Intended status: Informational
Expires: April 25, 2020

R. Even
Huawei
R. Huang
Huawei Technologies Co., Ltd.

October 23, 2019

Data Center Fast Congestion Management
draft-even-iccrg-dc-fast-congestion-00

Abstract

Fast congestion control is discussed in academic papers as well as in the different standard bodies. There is no one proposal for providing a solution that will work for all use cases leading to multiple approaches. By congestion control we refer to an end to end solution and not only to the congestion control algorithm on the sender side. This document describes the current state of flow control and congestion for Data Centers and proposes future directions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Fast congestion control is discussed in academic papers as well as in the different standard bodies. There is no one proposal for providing a solution that will work for all use cases leading to multiple approaches. By congestion control we refer to an end to end solution and not only to the congestion control algorithm on the sender side.

The major use case that we are looking at is congestion control for Data Centers, a controlled environment[RFC8085]. With the emerging Distributed Storage, AI/HPC (High Performance Computing), Machine Learning, etc., modern datacenter applications demand high throughput(40Gbps and above) with ultra-low latency of less than 10 microsecond per hop from the network, with low CPU overhead. For the end to end the latency should be less than 50usec, this value is based on DCQCN [DCQCN] The high link speed (>40Gb/s) in Data Centers (DC) are making network transfers complete faster and in fewer RTTs. Network traffic in a data center is often a mix of short and long

flows, where the short flows require low latencies and the long flows require high throughputs.

On IP-routed datacenter networks, RDMA is deployed using RoCEv2 [RoCEv2] protocol or iWARP [RFC5040] RoCEv2 [RoCEv2] is a straightforward extension of the RoCE protocol that involves a simple modification of the RoCE packet format. RoCEv2 packets carry an IP header which allows traversal of IP L3 Routers and a UDP header that serves as a stateless encapsulation layer for the RDMA Transport Protocol Packets over IP. For Data Centers RDMA in ROCEv2 expect a lossless fabric and this is achieved using ECN and PFC. iWARP congestion control is based on TCP congestion control (DCTCP [RFC8257])

A good congestion control for data centers should provide low latency, fast convergence and high link utilization. Since multiple applications with different requirements may run on the DC network it is important to provide fairness between different applications that may use different congestion algorithms. An important issue from the user perspective is to achieve short Flow Completion Time (FCT).

This document investigates the current congestion control proposals, and discusses future data center congestion control directions which aims to achieve high performance and collaboration.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Abbreviations

RCM - RoCEv2 Congestion Management

PFC - Priority-based Flow Control

ECN - Explicit Congestion Notification

DCQCN - Data Center Quantized Congestion Notification

AI/HPC - Artificial Intelligence/High-Performance computing

ECMP - Equal-Cost Multipath

NIC - Network Interface Card

RED - Random early detection gateways for congestion avoidance

4. Alternative Congestion Management mechanisms

This section will describe alternative directions based on current work. Looking at the alternatives from the network perspective we can classify the alternatives as:

1. Based on estimation of network status: Traditional TCP, Timely.

2. Network provides limited information: DCQCN using only ECN, SCE and L4S

3. Network provides some information: HPCC.

4. Network provides proactive control: RCP (Rate Control Protocol)

Note that any research on congestion control that requires network participation will be irrelevant if we cannot find a viable deployment path where only part of the network devices support the proposed congestion control.

4.1. Mechanisms based on estimation of network status

Traditional mechanisms uses packet status as the congestion signal and feedback to the sender, e.g. loss or delay, which is based on the facts that packets will drop when a buffer is full and packets will be delayed when a queue is building up. It can simply be achieved by the interactions between the sender and the receiver, without the involvement of network. It works well on the internet for a very long time, especially for best effort applications that do not have specific performance requirements.

However, these mechanism are not optimized for some data center application because the convergence time and throughput are not good enough. Mainly because endpoints estimation of network status are not accurate enough, and these mechanisms lack further information to adjust the sender behaviors.

4.2. Network provides limited information

In these mechanisms, the network utilize the ECN field of IP header to provide some hints on network status. The following sections describe some typical proposals.

4.2.1. ECN and DCTCP

The Internet solutions use ECN [RFC3168] for marking the state of the queues in the network device, they may use some AQM mechanism (fq_coDel [RFC8290]], PIE [RFC8033]) in the network devices and a congestion algorithm (New Reno [RFC5681], Cubic [RFC8312] or DCTCP[RFC8257]) on the sender side to address the congestion in the network. Note that ECN is signaled earlier than packet drop but may cause earlier exit from TCP slow start.

One of the problem for TCP is that ECN is specified for TCP in such a way that only one feedback signal can be transmitted per Round-Trip Time (RTT). [I-D.ietf-tcpm-accurate-ecn] specifies an alternative feedback scheme that provides more accurate information that can be used by DCTCP and L4S.

Traditional TCP uses ECN signal to indicate congestion experienced instead of packet loss, however, it does not provide information about the degree of the congestion. DCTCP [RFC8257] is trying to solve this issue. It estimates the fraction of bytes that encounter congestion rather than simply detecting the congestion presence. DCTCP further scales its sending rates accordingly. DCTCP is widely implemented in current data center environments.

4.2.2. DCQCN

An enhancement to the congestion handling for ROCEv2 is the Congestion Control for Large-Scale RDMA Deployments [DCQCN] providing similar functionality to QCN [QCN] and DCTCP [RFC8257], it is implemented in some of the ROCEv2 NICs but is not part of the ROCEv2 specification. As such, vendors have their own implementations which make it difficult to interoperate with each other efficiently.

DCQCN tests are assuming that the Congestion Point is using RED-ECN for ECN marking and the RDMA CNP message is used by the Notification Point (the receiver) to report ECN Congestion Experienced (CE). DCQCN as presented includes parameters that should be set. It provides the parameters that were used during the specific tests using Mellanox NICs. One of the comments about DCQCN is that it is not simple to define the parameters in order to get an optimized solution. This solution is specific to ROCEv2 and addresses only the congestion control algorithm and is implemented in the NIC.

DCQCN notification is using CNP that only report that at least one packet with CE marking was received in the last 50usec; this is similar to TCP reporting. Other UDP based transports like RTP and QUIC provides information about how many packets marked with CE, ECT(0,1) were received.

4.2.3. SCE - Some Congestion Experienced

[I-D.morton-taht-tsvwg-sce] ECT(1) to be an early notification of congestion on ECT(0) marked packets, which can be used by AQM algorithms and transports as an earlier signal of congestion than CE ("Congestion Experienced").

The ECN specification say that the congestion algorithm should treat CE marks the same as a drop packets. Using ECT(1) to signal SCE permits middleboxes implementing AQM to signal incipient congestion, below the threshold required to justify setting CE. Existing [RFC3168] compliant receivers MUST transparently ignore this new signal with respect to congestion control, and both existing and SCE-aware middleboxes MAY convert SCE to CE in the same circumstances as for ECT, thus ensuring backwards compatibility with ECN [RFC3168] endpoints.

This solution is using ECT(1) which was defined in ECN [RFC3168] as a one bit Nonce but this use is obsoleted in RFC8311 and SCE is using it for the SCE mark. There may be other documents trying to use this bit for example L4S use it to signal L4S support. The SCE marking are done by the AQM algorithm (RED, CODEL) and are sent back to the sender by the transport so there may be a need to add support for conveying the SCE marking to the sender (QUIC for example already has support for reporting the count of ECT(0) and ECT(1) separately). This solution is simpler than HPCC but provide less information.

[I-D.heist-tsvwg-sce-one-and-two-flow-tests] presents one and two-flow test results for the SCE reference implementation. These tests are not intended to be a comprehensive real-world evaluation of SCE, but an illustration of SCE's influence on basic TCP metrics in a controlled environment. The goal of the one-flow tests is to analyze the impact of SCE on the TCP throughput and TCP RTT of single TCP flows across a range of simulated path bandwidths and RTTs. The tests were with RENO and DCCP. Even though using SCE gave in general better results there were significant under-utilization at low bandwidths (<10Mb/sec; <25Mb/sec) and a slight increase in TCP RTT for DCTCP-SCE at 100Mbit / 160ms and a slight increase in TCP RTT for SCE RENO at high BDPs. The document does not describe the congestion algorithm that was used for DCTCP-SCE or RENO-SCE and comment that further work need to be done to understand the reason for this behvior.

The goal of the two-flow tests is to measure fairness between and among SCE and non-SCE TCP flows, through either a single queue or with fair queuing.

The initial results show that SCE enabled flows back off in the face of competition, whereas non-SCE flows fill the queue until a drop or CE mark occurs so fairness is not achieved. By changing the ramp by which SCE is marked and marking SCE when closer to drop or CE the fairness is better.

4.2.4. L4S - Low Latency, Low Loss, Scalable Throughput

There are three main components to the L4S architecture [I-D.ietf-tsvwg-l4s-arch]

1. Network: L4S traffic needs to be isolated from the queuing latency of Classic traffic. However, the two should be able to freely share a common pool of capacity. This is because there is no way to predict how many flows at any one time might use each service and capacity in access networks is too scarce to partition into two. The Dual Queue Coupled AQM [I-D.ietf-tsvwg-aqm-dualq-coupled] was developed as a minimal complexity solution to this problem. The two queues appear to be separated by a 'semi-permeable' membrane that partitions latency but not bandwidth. Per-flow queuing such as in [RFC8290] could be used but it partitions both latency and bandwidth between every end-to-end flow. So it is rather overkill, which brings disadvantages, not least that large number of queues are needed when two are sufficient.

2. Protocol: A host needs to distinguish L4S and Classic packets with an identifier so that the network can classify them into their separate treatments. [I-D.ietf-tsvwg-ecn-l4s-id] considers various alternative identifiers, and concludes that all alternatives involve compromises, but the ECT(1) and CE codepoints of the ECN field represent a workable solution.

3. Host: Scalable congestion controls already exist. They solve the scaling problem with TCP that was first pointed out in [RFC3649]. The one used most widely (in controlled environments) is Data Center TCP (DCTCP [RFC8257]). Although DCTCP as-is 'works' well over the public Internet, most implementations lack certain safety features that will be necessary once it is used outside controlled environments like data centers. A similar scalable congestion control will also need to be transplanted into protocols other than TCP (QUIC, SCTP, RTP/RTCP, RMCAT, etc.) Indeed, between the present document being drafted and published, the following scalable congestion controls were implemented: TCP Prague, QUIC Prague and an L4S variant of the RMCAT SCReAM controller [RFC8298].

Using Dual Queue provides better fairness between DCTCP and Reno/ Cubic . This is less relevant to Data Centers where the competing streams may use DCQN and DCTCP.

4.3. Network provides more information

The new-generation high-speed cloud network congestion control protocol HPCC (High Precision Congestion Control) [HPCC], aiming to achieve the ultimate performance and high stability of the high-speed cloud network at the same time. HPCC has been presented at ACM SIGCOMM 2019.

The key design choice of HPCC is to rely on switches to provide fine-grained load information, such as queue size and accumulated tx/rx traffic to compute precise flow rates. This has two major benefits: (i) HPCC can quickly converge to proper flow rates to highly utilize bandwidth while avoiding congestion; and (ii) HPCC can consistently maintain a close-to-zero queue for low latency.

HPCC is a sender-driven CC framework. Each packet a sender sends will be acknowledged by the receiver. During the propagation of the packet from the sender to the receiver, each switch along the path leverages the INT feature of its switching ASIC to insert some meta-data that reports the current load of the packet's egress port, including timestamp (ts), queue length (qLen), transmitted bytes (txBytes), and the link bandwidth capacity (B). When the receiver gets the packet, it copies all the meta-data recorded by the switches to the ACK message it sends back to the sender. The sender decides how to adjust its flow rate each time it receives an ACK with network load information.

Current IETF activity in IOAM [I-D.ietf-ippm-ioam-data] provides a standard mechanism for inserting metadata by the switches in the middle. IOAM can provides an optional method for sending the metadata feedback by the network to the endpoints on congestion status. But to using IOAM, the following points should be considered:

1. Is the current IOAM data fields sufficient for congestion control.

2. The encapsulation of IOAM in data center for congestion control.

3. The feedback format for sender driven congestion control.

The HPCC framework requires each node in the middle to add information about its state to the forward going packet until it reaches the receiver who will send the acknowledgment. We can think

of others modes like having the nodes in the middle updating the status information based on its available resources. This solution requires support for INT or IOAM, both protocols need to specify the packet format with the INT/IOAM extension. The HPCC document specify how to implement it for ROCEv2 while for IOAM there are some drafts in IPPM WG describing how to implement it for different transports and layer 2 packets.

The conclusion from the trials done were that HPCC can be a next-generation CC for high-speed networks to achieve ultra-low latency, high bandwidth, and stability simultaneously. HPCC achieves fast convergence, small queues, and fairness by leveraging precise load information from INT.

Similar mechanism is defined in Quick Start for TCP and IP[RFC4782]. There is a difference with the starting rate. While HPCC starts at maximum line speed [RFC4782] starts at a rate as specified in the Quick-Start request message. The Quick Start is specified for TCP, if other transport (UDP) is used there is a need to specify how the receiver send the Quick-Start response message.

4.4. Network provides proactive control

The typical algorithm in this category is RCP (Rate Control Protocol) [RCP]. In the basic RCP algorithm , a router maintains a single rate, R(t), for every link. The router "stamps" R(t) on every passing packet (unless it already carries a slower value). The receiver sends the value back to the sender, thus informing it about the slowest (or bottleneck) rate along the path. In this way, the sender quickly finds out the rate it should be using (without the need for Slow-Start). The router updates R(t) approximately once per roundtrip time, and strives to emulate Processor Sharing among flows. The biggest plus of RCP is the short flow completion times under a wide range of network and traffic characteristics.

The downside of RCP is that RCP involves the routers in congestion control, so it needs help from the infrastructure. Although they are simple, it does have per-packet computations. Another downside is that although the RCP algorithm strives to keep the buffer occupancy low most times, there are no guarantees of buffers not overflowing or of a zero packet loss.

5. Summary and Proposal

Congestion control is all about how to utilize the network resource in a better and reasonably way under different network conditions. Senders are the reaction points that consume network resource, and network nodes are the congestion points. Ideally, reaction points

should react as soon as possible when network statuses change. To achieve that, there are two directions:

5.1. Reflect the network status more accurately

In order to provide more information than just ECN CE marking there is a need to standardize a mechanism for the network device to provide such information and for the receiver to send more information to the sender. The network device should not insert any new fields to the IP packet but should be able to modify the value of fields in the packets sent from the data sender.

The network device will update the metadata in the forward going packet to provide more information than a single CE mark or SCE like solution.

The receiver will analyze the metadata and report back to the sender. Different from the Internet, data center network can benefit more from having more accurate information to achieve better congestion control. And this means network and hosts must collaborate together to achieve it.

Issues to be addressed:

- How to add the metadata to the forward stream (IOAM is a valid option since we are interested in a single DC domain). The encapsulations for both IPv4 and IPv6 should be considered.

- Negotiation of the capabilities of different nodes.

- The format of the network information feedback to the sender in the case of sender-driven mechanisms.

- The semantics of the message (notification or proactive)

- Investigation of the extra load on the network device for adding the metadata.

5.2. Notify the reaction point as soon as possible.

In this direction, it is worth to investigate if it's possible for the middle nodes to notify the sender directly (like IOAM Postcards) on network conditions, but such a method is challenging in terms of addressing security issues and the first concern will be that this can serve as a tool for DOS attack. But other ways, for example, carry the information in the reverse traffic would be an alternative as long as reverse traffic exists.

Issues to be addressed:

- How to deal with multiple congestion points?
- How to identify support by the sender and receiver for this mode and support legacy systems (same as previous mode).
- How to authenticate the validity of the data.
- Hardware implications

6. Security Considerations

TBD

7. IANA Considerations

No IANA action

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[CongestionManagment] "Understanding RoCEv2 Congestion Management", 12 2018, <https://community.mellanox.com/s/article/understanding-rocev2-congestion-management>.

[DCQCN] Zhu, Y., Eran, H., Firestone, D., Guo, C., Lipshteyn, M., Liron, Y., Padhye, J., Raindel, S., Yahia, M. H., and M. Zhang, "Congestion control for large-scale RDMA deployments. In ACM SIGCOMM Computer Communication Review, Vol. 45. ACM, 523-536.", 8 2015, <https://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p523.pdf>.

[HPCC] Li, Y., Miao, R., Liur, H. H., Zhuang, Y., Feng, F., Tang, L., Cao, Z., Zhang, M., Kelly, F., Alizadeh, M., and M. Yu, "HPCC: High Precision Congestion Control", 8 2019, <https://liyuliang001.github.io/publications/hpcc.pdf>.

[I-D.heist-tsvwg-sce-one-and-two-flow-tests] Heist, P., Grimes, R., and J. Morton, "Some Congestion Experienced One and Two-Flow Tests", draft-heist-tsvwg-sce-one-and-two-flow-tests-00 (work in progress), July 2019.

[I-D.herbert-ipv4-eh] Herbert, T., "IPv4 Extension Headers and Flow Label", draft-herbert-ipv4-eh-01 (work in progress), May 2019.

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-07 (work in progress), September 2019.

[I-D.ietf-quic-transport] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-23 (work in progress), September 2019.

[I-D.ietf-tcpm-accurate-ecn] Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-09 (work in progress), July 2019.

[I-D.ietf-tsvwg-aqm-dualq-coupled] Schepper, K., Briscoe, B., and G. White, "DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S)", draft-ietf-tsvwg-aqm-dualq-coupled-10 (work in progress), July 2019.

[I-D.ietf-tsvwg-ecn-l4s-id] Schepper, K. and B. Briscoe, "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay (L4S)", draft-ietf-tsvwg-ecn-l4s-id-07 (work in progress), July 2019.

[I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K., Bagnulo, M., and G. White, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", draft-ietf-tsvwg-l4s-arch-04 (work in progress), July 2019.

[I-D.morton-taht-tsvwg-sce]
Morton, J. and D. Taht, "The Some Congestion Experienced ECN Codepoint", draft-morton-taht-tsvwg-sce-00 (work in progress), March 2019.

[IEEE.802.1QBB_2011]
IEEE, "IEEE Standard for Local and metropolitan area networks--Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks--Amendment 17: Priority-based Flow Control", IEEE 802.1Qbb-2011, DOI 10.1109/ieeestd.2011.6032693, September 2011, <http://ieeexplore.ieee.org/servlet/opac?punumber=6032691>.

[QCN] Alizadeh, M., Atikoglu, B., Kabbani, A., Lakshmikantha, A., Pan, R., Prabhakar, B., and M. Seaman, "Data Center Transport Mechanisms:Congestion Control Theory and IEEE Standardization", 9 2008, <https://web.stanford.edu/~balaji/papers/QCN.pdf>.

[RCP] Dukkipati, N., "RATE CONTROL PROTOCOL (RCP): CONGESTION CONTROL TO MAKE FLOWS COMPLETE QUICKLY", 10 2007, <http://yuba.stanford.edu/~nanditad/thesis-NanditaD.pdf>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", RFC 3649, DOI 10.17487/RFC3649, December 2003, <https://www.rfc-editor.org/info/rfc3649>.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <https://www.rfc-editor.org/info/rfc4782>.

[RFC5040] Recio, R., Metzler, B., Culley, P., Hilland, J., and D. Garcia, "A Remote Direct Memory Access Protocol Specification", RFC 5040, DOI 10.17487/RFC5040, October 2007, <https://www.rfc-editor.org/info/rfc5040>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <https://www.rfc-editor.org/info/rfc6679>.

[RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <https://www.rfc-editor.org/info/rfc8033>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <https://www.rfc-editor.org/info/rfc8290>.

[RFC8298] Johansson, I. and Z. Sarker, "Self-Clocked Rate Adaptation for Multimedia", RFC 8298, DOI 10.17487/RFC8298, December 2017, <https://www.rfc-editor.org/info/rfc8298>.

[RFC8312] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", RFC 8312, DOI 10.17487/RFC8312, February 2018, <https://www.rfc-editor.org/info/rfc8312>.

[RoCEv2] "Infiniband Trade Association. Supplement to InfiniBand architecture specification volume 1 release 1.2.2 annex A17: RoCEv2 (IP routable RoCE).", <https://cw.infinibandta.org/document/dl/7781>.

Authors' Addresses

Roni Even
Huawei

Email: roni.even@huawei.com

Rachel Huang
Huawei Technologies Co., Ltd.

Email: rachel.huang@huawei.com