Enhanced Forward Explicit Congestion Notification (E-FECN) Scheme for Datacenter Ethernet Networks

Chakchai So-In, Raj Jain, and Jinjing Jiang
Department of Computer Science and Engineering
Washington University in St. Louis
{cs5, jain, jinjing}@cse.wustl.edu

Keywords: Congestion Control, Congestion Notification, FECN, enhanced FECN, BCN, and DCN

Abstract

Ethernet is replacing the traditional storage networking technologies like Fiber Channel and Infiniband in Datacenters. The key feature of these traditional technologies that make them suitable for datacenter is their low-loss low-delay operation. Consequently IEEE 802.1 standards committee is developing new specification for congestion management for Ethernet in datacenter networks. Backward Congestion Notification (BCN) and Forward Explicit Congestion Notification (FECN) are two proposals. Each of the proposals has its own advantages and disadvantages. FECN outperforms BCN in fairness and response time while BCN is able to respond to sudden increases in load in less than a round trip time. In this paper, we propose an enhanced version of FECN that takes the best of both proposals and adds BCN if there is a sudden severe congestion. It is shown that E-FECN performs better than the previous proposals.

1. INTRODUCTION

Congestion control has traditionally been handled at the transport (TCP) layer [15]. TCP NewReno, TCP SACK, and TCP Vegas for wired networks, I-TCP, MTCP and Freeze TCP for wireless networks, TCP-F, ELFN, and ATCP for ad-hock wireless networks, E-TCP, STCP, FastTCP, and High Speed TCP for high-speed networks are examples of such proposals [12, 13, 14]. However, there is a significant amount of traffic that does not use TCP. The storage traffic in datacenter networks is one such example. Data link level algorithms are, therefore, being developed that control all traffic regardless of the transport (or even network) protocol.

Datacenter networks (DCNs) are used for data storage and file transfers. These applications require a high throughput and a low latency. The transmission speeds of links used in DCNs are 1-10 Gbps. To maintain a low latency and to avoid extensive large queueing delay, queue lengths should be kept low. Moreover, packet loss is unacceptable since every packet is critical and will need to be retransmitted resulting in unacceptable delays. Therefore, IEEE 802.1Qau group has been formed in IEEE 802.1 standards committee to develop new congestion control schemes specifically designed for datacenter networks.

Backward Congestion Notification (BCN) and Forward Explicit Rate Notification (FECN) are two of the proposals made in IEEE 802.1Qau group. BCN has many variations with optional features such as BCNMAX, BCN00, and BCN with drifting and over-sampling. The main issues of BCN are its unfairness and large oscillations in throughputs. FECN, on the other hand, is fair but has a slower start. In this paper, we propose Enhanced Forward Explicit Rate Notification (E-EFCN) scheme, which is basically FECN mechanism with BCN sent back to the source under severe congestion. We show that E-FECN can achieve perfect fairness like FECN and allows a fast start.

This paper is organized as follows. Related work is summarized in Section 2. BCN and FECN system model are also described in this section. We present comparisons of BCN and FECN in Section 3. Our proposed scheme E-FECN is discussed in Section 4. Section 5 shows the simulation results, and finally conclusions are presented in Section 6.

2. Related Work

BCN mechanism for datacenter networks was introduced by Davide Bergasamo and his colleagues at Cisco [1 and 2]. BCN is a rate-based closed-loop feedback control mechanism. As shown in Figure 1, rate regulators at the sources are used to adjust the rate of individual flows according to BCN messages received from switches. The switches are called congestion points (CPs) while the sources are the reaction points (RPs). The switches monitor their buffer utilization and send BCN messages back to the source based on the status and variation of the buffer queue. Two thresholds Qeq (equilibrium queue length) and Qsc (severe congestion queue length) are used to trigger BCN messages.

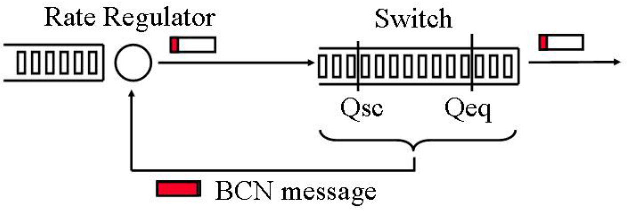

Figure 1: BCN System Model

Several variations of the basic BCN scheme have also been proposed, such as BCNMAX, BCN00, and BCN with drifting and over-sampling. The purpose of BCNMAX is to prevent the system from getting into severe congestion. When the queue length reaches $2\times Q_{eq}$, a BCNMAX message is sent back to the source. The source is expected to reduce the transmission rate substantially. This helps avoid severe congestion. Similarly, a BCN00 message from the switch causes the source to reduce to a very low rate. BCN was observed to have unfairness in the sense that once a source reduced its rate; it was more likely to stay low. To avoid this problem, BCN with drifting feature was introduced, which allowed sources to increase their rate randomly [4]. In order to keep the overhead low, it is necessary to keep the queue sampling rate low. However, during severe congestion, it is necessary to react fast and so an "over-sampling feature" was added to allow more frequent sampling during severe congestion. BCN is also known Ethernet Congestion Manager (ECM) [3].

It has been shown that BCN achieves only proportional fairness not max-min fairness [6]. Furthermore, BCN is slow in convergence to fair state and has large oscillations in throughput. FECN was proposed deal with the fairness and oscillation issues [7]. As shown in Figure 2, FECN is a close-loop explicit rate feedback control mechanism. The sources periodically send probe messages that pass through the switches and are returned back to the sources from the destination. On the forward path, switches reduce the rate field in the probe and when the probes return to the sources, they contain the exact rate that the flow should follow. Therefore, the sources change their rate to that in the probes. The switches advertise the same rate to all flows passing through the switch. This ensures that all flows are treated fairly.

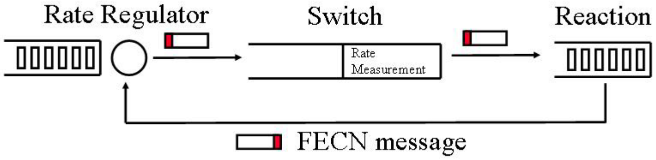

Figure 2: FECN System Model

Cyriel Minkenberg and Mitch Gusat introduced E^{2}CM [5] that adds the probing idea of FECN to ECM along with a few other enhancements. For example, E^{2}CM calculates per-flow loads at the source, uses source clock to determine forward latency, performs continuous probing, and accelerates rate recovery. Technically, E^{2}CM is much like ECM but with probing technique and also tuning up some parameters such as additive increase gain parameter and multiplicative decrease gain parameter.

3. Comparison of BCN and FECN

Both BCN and FECN are designed to resolve congestion in datacenter networks. Each of them has its own advantages and disadvantages. Table 1 shows some characteristics of BCN and FECN. For example, BCN messages can reduce the source rate quickly because the message is sent directly from the congestion point. On the other hand, with FECN, the sources need to wait for one round trip time in order for FECN probe message to come back. However FECN can reach the perfect fair state within a few round trip times because all sources get the same feedback. Unlike BCN, FECN does not have large oscillations in source throughput. The overhead of FECN is low and can be predicted because FECN message is sent every 1 ms and the message size is small with a payload of about 20 bytes.

From Table 1, it is seen that FECN outperforms BCN in several ways, but there are a few issues that still need to be considered, e.g., fast start, link disconnection, and the number of rate regulators. FECN was designed for congestion avoidance while BCN was designed for congestion control. In FECN, the sources start at a low rate and move to the equilibrium rates as successive probes return. Thus FECN requires each flow to begin with a rate regulator. In BCN, the sources start at full rate and come down if a BCN is received from a switch. Thus, in some cases, they may not need the rate regulator at all.

Secondly, if a link breaks, the FECN probes may not return and the source may continue to send at the current rate. BCN can issue a control message to the source in order to decrease or stop the source transmission. To deal with such situations, FECN has a timeout feature, which requires the sources to return to lower rate if the probes are not received. Nonetheless, there is some packet loss.

Finally, FECN, as originally proposed, requires as many regulators as the number of concurrent flows. This is because each flow starts in a regulated state. E^{2}CM, an enhanced ECM, limits the number of rate regulators to equal to a number of congested flows. A variation of FECN in which the number of rate regulators is equal to the number of congestion point has also been introduced [8]. An internal mapping mechanism is needed to be implemented in the Networks Interface Card (NIC). It turns out that this mapping scheme can be applied to both BCN and FECN.

Table 1: A Comparison of BCN, FECN, and E-FECN

Parameters	BCN	FECN	E-FECN
Fairness	Unfair (better with drift)	Perfect	Perfect
Feedback Control	Backward	Forward	Forward with becon
Overhead	High (Unpredictable)	Low (predictable), 20bytes	Medium
Load Sensor	Queue based	Rate based	Rate + Queue based
Link Disconnection	Support	N/A	Support
Fast Start	Support	N/A	Support
Number of Rate Regulators	Variable (E2CM)	Fix (= number of source flows)	Variable

4. E-FECN: FECN with BCN00

In this section, we describe an enhancement to FECN that allows fast start and so reduces the number of rate regulators, and allows FECN to cope with link disconnection as well. This enhanced version of FECN is called E-FECN.

As shown in Figure 3, in E-FECN, in addition to the normal probing mechanism of FECN, the switches are allowed to send BCN messages under severe congestion. E-EFCN allows sources to start sending the data at full rate (Fast Start) without a rate regulator. If this results in congestion on any switch, the switch sends a BCN00 message that requires the source to reduce to a low initial rate.

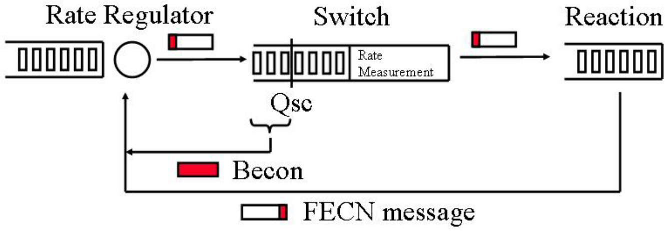

Figure 3: E-FECN System Model

The basic algorithm is as follows.

1. At the sources:
   a. All flows start sending at the full rate. Similar to BCN, the sources start at full rate and come down due to the congestion if either a BCN or FECN is received. Thus, in some cases, they may not need the rate regulator at all.
   b. Some congested flow rates are limited by the rate regulators.
   c. Flows are set to the proper rates once the source receives FECN control message, and it behaves the same as that in FECN mechanism.
2. At the switches:
   a. Almost all E-FECN operations behave the same operations as those in FECN.
   b. The switches also monitor their queue lengths.
   c. If the queue length becomes greater than the severe congestion threshold (Q_{sc}), a BCN00 message is sent back to source. The source then reduces the rate to R_{min} ($C/N0$).
   d. If a BCN00 message is sent to one source, to maintain the rate consistency, the switch also sets the advertised rate for all sources to R_{min}. This is the rate that is sent in FECN messages.

5. Performance Evaluations

In this section, we evaluate the performance of E-FECN compared to BCN and FECN in terms of fairness and convergence. Section 5.1 describes the simulation configuration and the results are discussed in Section 5.2.

5.1 Simulation Configuration

We used Networks Simulator Version 2 (NS2) [9] for our simulations. One sample simulation result is presented in this paper. The network configuration is shown in Figure 4 and networks parameters are listed in Table 2. There are four source nodes (SU1, SU2, SU3, and SU4) and only one destination node (DU). Each source node is linked to an edge switch (SW1, SW2, SW3, and SW4). All edge switches are linked to a single core switch (CS). Link propagation delays are 0.5 µs. Node processing delays are 1 µs. Link speeds are 10 Gbps. At all switches, a drop-tail queue mechanism is used if the buffers overflow. The switch output buffers can hold 100 packets of 1500 bytes each, i.e., the buffer size is 1,500×100 = 150,000 bytes). Ethernet's standard PAUSE mechanism is not used.

The traffic generation is at a constant bit rate (CBR) with UDP traffic over Ethernet. One CBR continuous flow is used per source node. The simulation duration is 100 ms. All four flows start at the 5 ms and two out of four ends at 80 ms. Other parameters are shown in Table 2. We used BCNMAX option with BCN, the maximum negative feedback is send back to the source at a threshold of $2{\times}Q_{eq}$, because it improves BCN and has been used by their developers in all recent simulations [3]. Together with over-sampling technique, BCNMAX results in a faster response to sudden and quick positive changes in queue length [2]. R_{min} is set to 500 Mbps since FECN recommends $N0 = 20$ for small topology [3] and in FECN with slow start, the initial rate is set to $C/N0$, which is 500 Mbps.

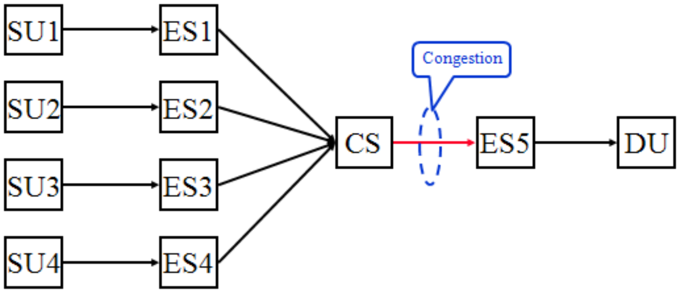

Figure 4: Network configuration

Table 2: BCN, FECN, and E-FECN parameters

Parameters	**BCNMAX**	**FECN**	**E-FECN**
Q_{eq} (equilibrium)	16 packets (24,000 bytes)	16 packets (24,000 bytes)	16 packets (24,000 bytes)
Q_{sc} (severe control)	80 packets (120,000 bytes)	80 packets (120,000 bytes)	80 packets (120,000 bytes)
Sampling Rate	2% (every 75,000 bytes)	N/A	N/A
Over-sampling ($Q>Q_{sc}$)	10% (every15,000 bytes)	N/A	N/A
W, R_u	W = 2, R_u = 1 Mbps	N/A	N/A
G_i, G_d	G_i = 0.533 , G_d = 0.0002667	N/A	N/A
a, b, c	N/A	a = 1.1, b=1.002, c = 0.1	a = 1.1, b=1.002, c = 0.1
Initial rate (Fast Start)	10 Gbps	10 Gbps (N0=1)	10 Gbps (N0=1)
Tagging frequency	N/A	Every 1ms	Every 1ms
R_{min} (E-FECN)	N/A	N/A	500 Mbps
Timeout	N/A	2 ms	2 ms

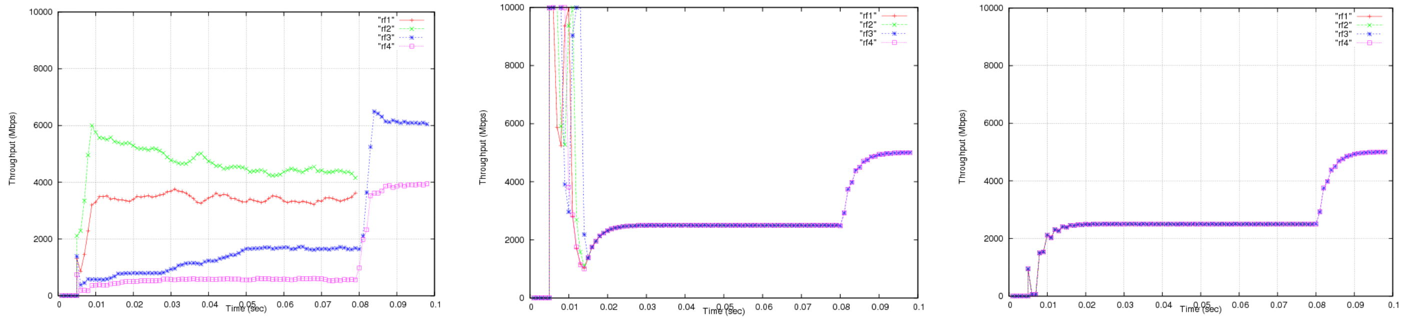

Figure 5: Source rates

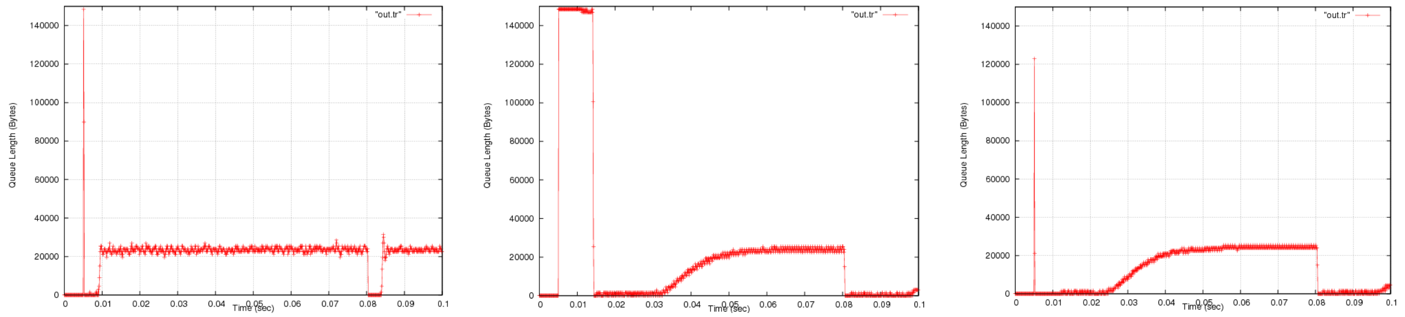

Figure 6: Queue length at the core switch

\

5.2 Simulation Results

Figure 5 shows source rates for the schemes. Left most graph is for BCN, middle graph is for FECN and the right most graph is for E-FECN. Notice that the four sources get very different rates with BCN. This shows the unfairness of BCN. The lowest rate source gets to increase at 80 ms when two of the four flows stop. Even after that the remaining two sources have different rates. With FECN, the four curves (for four sources) are on the top of each other and so it is fair but there are large transients in the beginning before steady state is achieved. With E-FECN, the transients are eliminated and fairness is maintained. Thus, we get both fast convergence and fairness. Note that the convergence time for fair and efficient throughput for FECN is around 20 ms, while for E-FECN, it is only 15 ms.

Figure 6 shows queue length at the core switch for the three schemes. Note that all three schemes can stabilize the queue at the desired Q_{eq} (24,000 bytes). BCN has a few packet losses at the beginning (spike). FECN has high queue for around 10 ms. E-FECN has no packet losses at all.

6. Conclusions and Future Work

BCN and FECN are two proposed schemes for congestion notification in datacenter Ethernet networks under IEEE 802.1Qau group. There are several variations of BCN and there are issues of parameter selections. BCN with over-sampling, drifting, BCNMAX seems to be the best among BCN variations. Therefore, this variation of BCN is used for comparison with FECN and E-FECN in this paper. Although, we present only one simulation result, it clearly shows the strengths and weaknesses of the three schemes. It is obvious that BCN is unfair; FECN is fair but needs to start at a low rate, which means that each flow needs a rate regulator.

In this paper, we proposed E-EFCN, an enhancement to FECN that also uses a backward congestion notification BCN00 message to limit the source rate under severe congestion. With this feature, E-FECN maintains the perfect fairness of FECN and also allows sources to start at high rates. This reduces the number of rate regulators to the same number as in BCN mechanism.

7. References

[1] Davide Bergamasco, “Datacenter Ethernet Congestion Management: Backward Congestion Notification,” IEEE 802.1 Meeting, May 2005.

[2] Davide Bergamasco and Rong Pan, “Backward Congestion Notification Version 2.0,” IEEE 802.1 Meeting, September 2005.

[3] Davide Bergamasco, “Ethernet Congestion Manager,” private communications, March 2007.

[4] Bruce Kwan and Jing Ding, “BCN Calibration Simulation with Global Pause and Drift,” IEEE 802.1 Meeting, October 2006.

[5] Cyriel Minkenberg and Mitch Gusat, “E^2CM updates,” IEEE 802.1 Meeting, May 2007.

[6] Jinjing Jiang, Raj Jain, and Manoj K. Wadekar, “Analysis of Backward Congestion Notification (BCN) for Ethernet In Datacenter Applications,” IEEE INFOCOM 2007, pp 2456-2460, May 2007.

[7] Jinjing Jiang, Raj Jain, and Chakchai So-In, “Congestion Management for Ethernet In Datacenter Application Using Forward Explicit Rate Notification,” WUSTL technical report, 2007.

[8] Jinjing Jiang, Raj Jain, and Chakchai So-In, “An Explicit Rate Control Framework for lossfree Ethernet Operation,” Accepted to appear in ICC 2008.

[9] NS2, Networks Simulator, 1991. Available at http://www.isi.edu/nsnam/ns/

[10] Manoj Wadekar, “CN-SIM: Topologies and Workloads,” IEEE 802.1 Meeting, September 2006.

[11] Van Jacobson and Michael J. Karels, “Congestion Avoidance and Control,” ACM SIGCOMM 1988, pp 314-329, May 1988.

[12] Joerg Widmer, Robert Denda, and Martin Mauve, “A Survey on TCP-Friendly Congestion Control,” IEEE Network, vol.15, no.3, pp.28-37, May 2001.

[13] Eric He, Pascale Vicat-Blanc, and Michael Welzl, “A Survey of Transport Protocols other than “Standard” TCP,” Global Grid Forum, Data Transport Research Group, April 2005.

[14] Frank.Kelly, “The mathematics of traffic in networks,” in the Princeton Companion to Mathematics,” Princeton University Press.

[15] Yi Lu, Rong Pan, Balaji Prabhakar, Davide Bergamasco, Valentina Alaria, and Andrea Baldini, “Congestion control in networks with no congestion drops,” September 2006.