Analysis of Backward Congestion Notification (BCN) for Ethernet In Datacenter Applications

Jinjing Jiang and Raj Jain
Department of Computer Science and Engineering
Washington University in Saint Louis
{jinjing, jain}@cse.wustl.edu

***Abstract*— IEEE 802.1 standards committee is working on a new specification for congestion notification in Ethernet networks. The goal of this work is to enable application of Ethernet in backend datacenter applications. Such applications typically use Fiber Channel and Infiniband due to their loss-free characteristics. A backward congestion notification (BCN) scheme has been proposed to avoid long delays and minimize loss in Ethernet networks. This paper presents an analysis of this scheme. We develop an analytical model to analyze the stability and the rate of convergence of the scheme. It is shown that BCN achieves proportional fairness and not max-min fairness. Simulation results are presented that validate the analytical results.**

I. Introduction

Recently, datacenter networks (DCNs) have attracted a lot of interest in the networking industry. DCNs are used to provide data storage where end stations are interconnected as clusters and bladed systems. DCNs require high throughput and low latency for efficient communications. Due to the large number of end stations configured in clusters, these networks are vulnerable to link congestion. For example, in a typical star topology running TCP traffic, the output link of the core switch could be congested frequently resulting in packet losses and timeouts that can severely jeopardize the overall system throughput. This is not tolerable for applications with huge amount of data exchange.

While significant amount of work has been done on congestion control in TCP/IP networks, Ethernet networks, even after 30 years of their invention run without congestion control in the datalink layer. This may be acceptable for elastic applications but not tolerable for datacenter applications. The packet loss rate in datacenter applications should be practically zero. This is why traditionally datacenters have used fiber channel and Infiniband networks that provide hop-by-hop flow control and sophisticated congestion control mechanisms to avoid packet losses.

datacenters cannot rely solely on TCP to take care of network congestion. There are many applications that use UDP and some don't even use IP, e.g., Veritas Cluster [1]. It is, therefore, important that Ethernet networks provide a congestion control mechanism in the datalink layer.

IEEE 802.1 standards committee has been discussing the possibility of and need for congestion control in datacenter networks for over a year now and is currently developing a project authorization request (PAR) to study congestion notification methods. In order to maintain a low latency in DCNs, queue lengths should be kept at a relative low level in order to avoid excessive queuing delays. Also, there should be no packet loss in the switches since a single packet loss can cause long timeout for TCP traffic and increase the latency significantly.

The paper is organized as follows. In Section II, the related work is summarized. In Section III, we describe the general system model and assumptions for DCNs. We discuss the Backward Congestion Notification (BCN) mechanism that has been presented at IEEE 802.1. We actually present an enhanced version of the mechanism since our analysis showed that the version as proposed does not work in certain situations. In Section V, an analytical model of BCN is presented, and several propositions are made on its performance. Section VI gives the simulation results to support our propositions. The paper ends with the conclusions and future directions. To aid the reader, a list of all symbols used in the paper is appended to the paper.

II. Related Work

In [8], McAlpine and Wadekar proposed the general architecture for congestion management in Ethernet Clusters, where link level control, layer 2 subnet control and end-to-end higher layer control protocols are discussed. Through simulations, they endeavor to find the appropriate set of congestion management methods that are compatible with IEEE802.1/802.3 standards. In [9], a simple switch-based ECN mechanism with a new source rate control mechanism using window limit was described. It achieves better fairness and throughput for both static and dynamic traffic. However, this scheme only works with TCP traffic.

The general BCN mechanism for DCNs was developed by Davide Bergasamo and his colleagues at Cisco and proposed in [1][2]. In this paper, we begin with this BCN mechanism. Our analysis and simulation show that there are a few problems in the original BCN mechanism. Therefore, a modification to BCN mechanism is proposed and the enhanced BCN is then analyzed in detail.

III. System Design and Assumptions

A. System Model

BCN is a rate-based closed-loop feedback control mechanism, shown in Fig. 1. It is assumed that the sources are

equipped with the rate regulators, which can be token-bucket traffic shaper. At the core switch, where congestions happens, congestion detector, which is an up-down counter, is integrated in the hardware to generate the feedback message.

Fig. 1. BCN System Model

As shown in Fig. 1, the core switch monitors the length of its output queue, and decides whether there is congestion. If congestion is detected, the switch signals the sources with a BCN message that contains information required for the source to adjust their rates. The source reacts to the received BCN and updates the rate of its regulator.

B. Design Goals

In designing BCN, the main goals are to maintain high throughput and minimize queuing delays. Meanwhile, multiple flows should share link capacity fairly. Also it should be stable and robust. In the following, each of these goals is explained in detail.

- *High Throughput*: Since the demand for data exchange in DCNs is extremely large compared with other networks, the first goal is to maximize the throughput. Actually, here the throughput should be goodput, i.e., retransmissions are harmful for DCNs. If there are no retransmissions, generally, maximization of the throughput is equivalent to maximizing the link utilization.
- *Low Queuing Delays*: The higher the throughput, the higher the link utilization. However, high link utilization can often lead to large queue lengths and long queuing delays. In order to keep the delays under some acceptable value and the high link utilization, we aim to control the queue length at a constant level at the core switch. Thus, the variation of the latency is minimized.
- *Fairness*: When there are multiple sources sharing a bottleneck, it is important that the capacity be fairly allocated. Fairness can be defined in many different ways. Two commonly used definitions are: max-min fairness and proportional fairness [6][5]. If there is only one congestion point in the network, max-min fairness and proportional fairness result in the same resource allocation. However, when there are multiple congestion points, proportional fairness can result in a very different resource allocation than the max-min fairness. Our analysis shows that BCN is approximately proportionally fair.
- *Stability*: The stability of a system depends on the control target. In BCN, the design goal is to limit the queue length in the buffer. Therefore, we generally regard the stability of queue lengths as the stability of the whole system. However, under some assumptions, we observe that even when there is small rate variation for individual flows, it can still be regarded as stable. This is presented in Section V. Another key issue for stability is that the system should converge to the stable state from any initial settings. For example, several flows in a network may be inactive initially. When these flows become active, the system should converge to a new stable state.
- *Robustness*: Convergence and stability in control systems is often accompanied with oscillations around the final goal. Robustness relates to the size of these oscillations [11]. For example, our simulations show that for some bursty traffic, whose burst period is approximately equal to the system settling time, BCN can cause large oscillations in queue length. However, the stochastic average of the queue length may still be around the target level.

Note that BCN is only an initial endeavor to solve the congestion management problem for DCNs. In this paper, we provide both analytical and simulation results to support our claim.

C. Assumptions

BCN messages are specially formatted packets sent through switches in DCNs. It is important that the messages follow a format that is acceptable to legacy switches that are BCN-unaware. As suggested in [3], the BCN message format should be VLAN-tagged to ensure the coexistence and interoperability between BCN-aware and BCN-unaware switches. Propagation delay in DCNs is generally small - of the order of a few tens or hundreds of microseconds. This is because the diameter of the network is only a few hundreds meters. Links are assumed to be of high capacity. Most links are either 1 Gigabit per second or 10 Gigabit per second. In our analysis, we assumed switch buffers are FIFO and output queued.

IV. BCN Mechanism

BCN works in three phases: Detection, Signaling and Reaction. In the following, each phase is explained.

A. Congestion Detection

For each link, the network manager sets a target buffer utilization level at equilibrium Q_{eq}. This is the desired number of packets that should be in queue. Two thresholds Q_{sc} and Q_{st} are also set by the manager to indicate tolerable congestion levels on the link. The switches simply count the number of arriving and departing packets, and sample the incoming packets with a probability P_m. When a packet is sampled, the switches determine the congestion level on the link and may send a BCN message to the source of the sampled packet if necessary. If the congestion is severe, the switch may send a "PAUSE" message.

B. Backward Signaling

BCN has three kinds of signals: PAUSE frames, BCN normal messages, and BCN STOP messages.

First, PAUSE frames are a hop-by-hop congestion control mechanism used in IEEE802.3x. When the queue length is larger than Q_{st}, the switch simply sends out PAUSE/OFF to ask all its uplink neighbors to stop dequeuing packets. In turn, since the output link is stopped, the buffers of the neighbors will fill up, and ultimately a PAUSE frame reaches the end stations. Then the end station stops and packets accumulate in

the internal buffer and the the source is eventually blocked. When the queue length becomes lower than some predefined level, another PAUSE/ON frame is sent out to enable the dequeue function.

BCN messages use 802.1Q tag format[2]. The key fields in BCN message are shown in Fig. 2.

SA	DA
EthernetType=BCN	CPID
e_i	C

Fig. 2. Key Fields of BCN message

Here *SA* is the address of the switch; *DA* is the source address of the sampled packets. *EthernetType* tells the switch and end stations whether it is a BCN message. *CPID* is the ID for the congestion point, which can be the MAC address of the switch interface. e_i is some information about the buffer that is fed back to the source, C is the capacity of congested link. If $e_i > 0$, we say, BCN message is positive. BCN message is negative if $e_i < 0$. Note that the capacity field is included in the BCN message only in the enhanced version of BCN. We show later that capacity is required for the sources to correctly adjust their rates. All results presented in this paper assume this enhanced version of BCN.

BCN STOP message is generated when severe congestion happens. The source getting the STOP message simply sets its regulator's rate to 0 for a random period, then recovers by setting the rate as $\frac{C}{100}$, where C is the capacity of the bottleneck link.

C. Source Reaction

When a normal BCN message reaches the end station, the end station uses e_i to calculate the new rate, and updates the settings for rate regulator. Once the source gets one BCN message, every packet going through the rate regulator is tagged. The rate regulator tag (RRT) also uses IEEE 802.1Q tag format. The key fields are shown in Fig. 3

SA	DA
EthernetType=RRT	CPID

Fig. 3. Key Fields of RRT

Note that BCN works in a defined BCN-aware region of the network. When the packets exit this region, the bridges remove the RRT tags.

D. Feedback and Rate Adaption Algorithm

The key measure of congestion on a link is e_i. This consists of a weighted sum of the instantaneous queue offset and the queue variation over the last sampling interval:

$$e_i = q_{\text{off}}(t) - W q_{\text{delta}}(t) = (Q_{\text{eq}} - q(t)) - W(q_a - q_d), \quad (1)$$

Here, W is the weight; $q_{\text{off}}(t)$ is the instantaneous queue offset defined as

$$q_{\text{off}}(t) = q(t) - Q_{\text{eq}}$$

q_{delta} is the queue variation over the last sampling interval and is defined as the difference in the number of packets that arrived q_a and the number of packets that were served q_d since the last sampling event.

Heuristically, various possible circumstances are as follows:

- If $q(t) < Q_{\text{eq}}$, $q_a \approx q_d$, then $e_i > 0$. In this case, the queue length is short; the sources can increase their rates.
- If $q(t) < Q_{\text{eq}}$, $q_a \gg q_d$, then $e_i < 0$. In this case, even though the queue length is small, it is increasing. The congestion is building up. The sources are asked to decrease their rates.
- If $q(t) > Q_{\text{eq}}$, $q_a \approx q_d$, then $e_i < 0$. The large queue indicates that the link is congested. The sources are asked to decrease their rates.
- If $q(t) > Q_{\text{eq}}$, $q_a \ll q_d$, then $e_i > 0$. Even though the queue is large at the moment, it is decreasing and so the sources are encouraged to increase their rates.

Note that $q_{\text{off}}(t)$ indicates the instantaneous load while $q_{\text{delta}}(t)$ indicates the rate of change of the load. The weighted sum is a rough prediction of the future load.

In BCN, the source adjusts its rate using a modified Additive Increase and Multiplicative Decrease (AIMD) algorithm. AIMD has been proven to be sufficient and necessary of efficiency and fairness under certain common conditions[4]. AIMD is implemented in BCN as follows:

$$r_i = \begin{cases} r_i + G_i e_i R_u & \text{if } e_i > 0; \\ r_i(1 + G_d e_i) & \text{if } e_i < 0. \end{cases} \quad (2)$$

Here, G_i is the additive increase gain parameter, R_u is the increase rate unit parameter and G_d is the multiplicative decrease gain parameter. In other words, the scheme has three parameters G_i, G_d, and R_u, which must be appropriately set by the network manager. The capacity field in the BCN feedback message is used by the sources to set the rate unit parameter R_u.

E. Feedback Conditions

The arriving packets are sampled with probability P_m and for each sampled packet the BCN messages are generated as follows:

- If the packet does not contain rate regulator tag
  - If $q(t) < Q_{\text{eq}}$, no BCN message is sent.
  - If $Q_{\text{eq}} < q(t) \leq Q_{\text{sc}}$, normal BCN message is sent.
  - If $q(t) > Q_{\text{sc}}$, BCN STOP message is sent.
- If the packet contains rate regulator tag
  - If $q(t) < Q_{\text{eq}}$ and *CPID* field matches with this switch's CPID, a positive BCN message is sent.
  - If $Q_{\text{eq}} < q(t) \leq Q_{\text{sc}}$, normal BCN message is sent.
  - If $q(t) > Q_{\text{sc}}$, BCN STOP message is sent.

V. Analytical Model of BCN

In this section, we analyze the performance of BCN analytically. Actually, BCN control mechanism can be separated into two parts: link control and source control.

At the link, assuming that the queue length $q(t)$ is differentiable, we have:

$$\frac{dq(t)}{dt} = \frac{1}{S}\left\{\sum_{i=1}^{n} r_i(t) - C_l\right\}, \quad (3)$$

where S is the size of packets in bits, n is the number of sources sharing the link, C_l is the link capacity, $r_i(t)$ is the rate or load of the ith source on this link.

The vector of rates of all sources is denoted as $\vec{r}(t) = [r_1(t), \ldots, r_N(t)]$. Here, N is the number of all sources in the network.

Based on eqn. (1), the BCN feedback parameter e_i generated by link l is:

$$e_i(t) = (Q_{\text{eq}} - q(t)) - W\frac{dq(t)}{dt}\left(\frac{S}{C_l \cdot P_m}\right). \quad (4)$$

Note that eqn. (4) is the continuous version of the discrete feedback in eqn (1). Each BCN message is sent back to the individual source of the sampled packet.

At the ith source, assume t_n is the time at nth rate update event. Then:

$$\begin{aligned} r_i(t_{n+1}) &= \underbrace{e_i(t_n)^\dagger[1 - |e_i(t_n)|G_d]r_i(t_n)}_{\text{Multiplicative Decrease}} \\ &+ \underbrace{(1 - e_i(t_n)^\dagger)[r_i(t_n) + G_i|e_i(t_n)|R_u]}_{\text{Additive Increase}}, \quad (5) \end{aligned}$$

where

$$e_i(t_n)^\dagger = \begin{cases} 1, \text{if } e_i(t_n) < 0, \\ 0, \text{otherwise.} \end{cases}$$

The above equation can be rewritten as:

$$\begin{aligned} r_i(t_{n+1}) - r_i(t_n) &= |e_i(t_n)|\{G_i R_u \\ &\quad - e_i(t_n)^\dagger(G_d r_i(t) + G_i R_u)\} \quad (6) \end{aligned}$$

Following the stochastic approximation mentioned in [5] and assuming absolute value $|e_i(t)|$ is independent of the sign of $e_i(t)$, the above discrete time equation can be approximated into an ordinary differential equation (ODE)[5][10]:

$$\frac{dr_i(t)}{dt} = \frac{E\{|e_i(t)|\}\{G_i R_u - b_i(t)(G_d r_i(t) + G_i R_u)\}}{\mu_i(t)}. \quad (7)$$

Here $b_i(t)$ is the expectation of $e_i(t_n)^\dagger$ and $\mu_i(t)$ is the expected update interval for the ith source.

A. Stability via Lyapunov Method

Proposition 1: Without considering stochastic perturbations and time lags, the rate of each source converges to a unique rate that maximizes the link utilization. The system is stable.

Since the source reacts to every feedback it receives, the expected interval between successive BCN messages is $\frac{S}{r_i(t)P_m}$. The expected value of $e_i(t_n)^\dagger$ is simply the probability that a negative BCN message is generated. This probability is:

$$b_i(t) = \mu_i(t) r_i(t)\left[\sum_{l=1}^{L} p_l(f_l(t)) a_{li}\right] \quad (8)$$

Here $f_l(t) = \sum_{i=1}^{N} a_{li} r_i(t)$ is the total amount of traffic going through the lth link; a_{li} is the fraction of traffic from ith source that goes over lth link; This formulation allows for the possibility of splitting traffic over multiple parallel paths. If all traffic from a source goes over one path $a_{li} \in \{0, 1\}$. $p_l(f)$ is the probability that a negative BCN message is generated when the load is f. This probability is monotonically increasing and differentiable function of the load f [6]. Here we assume that there are L links in the network. So the model is not restricted to a single congestion point.

Following [5][7],

$$\sum_{l=1}^{L} p_l(f_l(t)) a_{li} = \frac{\partial}{\partial r_i(t)} \sum_{l=1}^{L} P_l(f_l) = \frac{\partial P(\vec{r}(t))}{\partial r_i(t)}, \quad (9)$$

Where P_l is a primitive of p_l. The ODE can now be written as:

$$\begin{aligned} \frac{dr_i(t)}{dt} &= E\{|e_i(t)|\} r_i(t)(G_d r_i(t) + G_i R_u) \\ &\quad \times \left\{\frac{G_i R_u P_m}{S(G_d r_i(t) + G_i R_u)} - \frac{\partial G(\vec{r}(t))}{\partial r_i(t)}\right\}. \quad (10) \end{aligned}$$

The Lyapunov function for the ODE is:

$$\begin{aligned} V(\vec{r}) &= \sum_{i=1}^{N} \int_0^{r_i} \frac{G_i R_u P_m}{S(G_d u_i + G_i R_u)} du_i - P(\vec{r}) \\ &= \frac{G_i R_u P_m}{S G_d} \log\left\{\frac{G_d}{G_i R_u} r_i + 1\right\} - P(\vec{r}) \quad (11) \end{aligned}$$

Hence we can write the ODE as:

$$\frac{dr_i(t)}{dt} = E\{|e_i(t)|\} r_i(t)(G_d r_i(t) + G_i R_u)\frac{\partial V(\vec{r})}{\partial r_i} \quad (12)$$

Since $V(\vec{r})$ is strictly concave, it can reach a unique maximum over any bounded region. Also we have:

$$\begin{aligned} \frac{d}{dt}V(\vec{r}) &= \frac{\partial V}{\partial r_i(t)}\frac{dr_i(t)}{dt} \quad (13) \\ &= E\{|e_i(t)|\} r_i(t)(G_d r_i(t) + G_i R_u)\left(\frac{\partial V(\vec{r})}{\partial r_i}\right)^2 \end{aligned}$$

Hence V increases along any solution, which converges towards the unique maximum of V. This shows that the rates $r_i(t)$ converge at equilibrium towards a set of values that maximize $V(\vec{r})$, i.e., the rate is stable. Therefore, by eqn. (3), the queue length is also stable. Since the maximization of $V(\vec{r})$ is constrained by the link capacity, it is obvious that the equilibrium maximizes the link utilization at the same time.

B. Fairness Analysis

Proposition 2: With multiple sources sharing a link, BCN is approximately proportionally fair.

From the Lyapunov function for the ODE, in the extreme case where the feedback expectation is close to Dirac delta function, the rates are distributed so as to maximize [5][6]

$$F_A(\vec{r}) = \sum_{i=1}^{N} \frac{G_i R_u P_m}{S G_d} \log\left\{\frac{G_d}{G_i R_u} r_i + 1\right\}, \quad (14)$$

subject to the constraints

$$\sum_{i=1}^{N} a_{li} r_i \leq C_l, \forall l \tag{15}$$

The presence of log term in eqn. (14) indicates that BCN provides proportional fairness, with a small positive bias given to small rates (generally, $1 \ll \frac{G_d}{G_i R_u} r_i$). For example, the weight given to r_i is close to $\log\left(\frac{G_d}{G_i R_u} r_i\right)$ for large r_i, which is larger than any rates that are smaller.

Since the system is approximately proportionally fair, the link capacity is generally not equally allocated among the flows sharing the link as would be the case with max-min fairness. In those configurations in which there is only one congestion point, proportional fairness is equivalent to max-min fairness. An example of topology with multiple congestion point is the so called "parking lot" configuration described later in Section VI.

C. Rate of Convergence

Proposition 3: Without considering time lags and stochastic perturbations, the rate of convergence for BCN generally deceases with G_i, G_d, and R_u.

Recall the ODE (7) for the system, following [6], let $r_i(t) = R_i + r_i^{1/2} s_i(t)$, where R_i is the rate at equilibrium. Denote $h_i(t) = \sum_{l=1}^{L} p_l(f_l(t)) a_{li}$. Linearizing eqn. (7) about R_i, we have:

$$\begin{aligned} \frac{ds_i(t)}{dt} &= E\{|e_i(t)|\}\frac{1}{R_i}\{\frac{1}{S} G_i R_u P_m \\ &\quad -(2G_d R_i + G_i R_u) h(t) \\ &\quad -R_i(G_d R_i + G_i R_u) h'(t)\} s_i(t) \end{aligned} \tag{16}$$

Generally, the rate of convergence depends on $E\{|e_i(t)|\}$, P_m, G_i, G_d and R_u. Also it depends on $h(t)$ and $h'(t)$. More precisely, define

$$\begin{aligned} \Delta(t) &= \frac{1}{S} G_i R_u P_m - (2G_d R_i + G_i R_u) h_i(t) \\ &\quad -R_i(G_d R_i + G_i R_u) h_i'(t) \end{aligned} \tag{17}$$

which dominates the convergence rate of the system. With several lines of algebra, it can be shown that $\Delta(t)$ is negative, decreases with G_i, G_d and R_u, and increases with P_m. Hence, to increase the rate of convergence, we need to decrease G_i, G_d and R_u, or increase P_m.

Note that eqn. (7) is a simplified model for the system, without considering the stochastic perturbations and time lags. In order to formulate this model, we assume that the expected BCN message interval is equal to $\frac{S}{r_i(t) P_m}$. Hence increasing P_m corresponds to the decreasing of time lag or feedback delay. Generally in feedback control theory, smaller time lag results in a more stable system. In [6], it is shown that time lags have severe impact on the system stability, i.e., increasing the rate of convergence will compromise the stability. Based on this result, the sampling probability P_m and the initial rate for the rate limiters should be carefully selected to avoid large delays in reacting to BCN messages. Another problem is that a large P_m may cause excessive signaling overhead on the reverse link.

VI. Simulation Results

In this section, we provide extensive simulation results that support the propositions in Section V and provide further information on performance of BCN.

A. Simulation Configuration

We used Network Simulator V2 (NS2)[12] for our simulations. Unless noted otherwise, all simulations presented in this paper use the following parameters and configurations. Link propagation delays are 0.5 μs, which are typical for optical fiber lengths of 100 m. Link speeds are 10 Gbps. The switch output buffer size is 100 packets. Drop-tail mechanism is used when buffers overflow. For TCP traffic, TCP Reno with Selective Acknowledgements (SACK) is used. The maximum timeout for TCP is tuned to 1 ms to enable fast recovery from segment losses. All packets are 1500 bytes. For TCP Reno, the maximum receive window is set to 44 packets, which is approximately equivalent to a window of 64 kB. The workload consists of FTP applications. The BCN parameters are: $Q_{\text{sc}} = 80, Q_{\text{eq}} = 16, W = 2, G_i = 4, G_d = 0.0124, R_u = 4$ Mbps.

B. Performance Improvement and Stability

As claimed before, BCN can protect fragile sources with lower rates. We use the 6-source topology shown in Fig. 4, where source SR_1 and SR_2 are reference sources with relatively low rates, whose sinks are DR_1 and DR_2, respectively. These sources have one connection which periodically sends out 10 kB data and then idle for several microseconds and transmit again. CS is the core switch, where the congestion can happen. ES_1, ..., ES_6 are edge switches, and ST_1, ..., ST_4 are TCP sources with bulk (infinite) traffic, whose sink is DT. Each of these sources has 10 continuous connections simultaneously.

Fig. 4. The 6-source topology

The simulation results are presented in Tables I and II. The throghput is expressed in both transactions per second (tps) and gigabit per second (Gbps). Here each 10 kB transfer is designated as one transaction.

From Table I, it is seen that BCN significantly improves the throughput and delay of reference source 1, which suffers the congestion. Reference source 2 is not congested and hence its performance is not affected by BCN. Table II shows that the congested link (between CS and ES_5) is almost fully utilized. Reference source SR_2 affects the throughput of bulk TCP traffic from source ST_1 resulting in a large variance in the throughputs of various bulk traffic sources. BCN reduces this variance significantly. Thus, BCN seems to allocate the congested link capacity fairly among the four bulk sources.

The queue length for the congested link is shown in Fig. 5. Notice that the queue length oscillates closely around Q_{eq}. If we define stable state as a maximum queue variation of

	Reference Source 1			Reference Source 2		
CM	Throughput(Tps)	Throughput(Gbps)	Latency(μs)	Throughput(Tps)	Throughput(Gbps)	Latency(μs)
None	556	0.06	1780.78	16634	1.44	59.11
BCN	6686	0.58	133.51	16624	1.44	59.16

TABLE I

PERFORMANCE OF REFERENCE SOURCES WITH AND WITHOUT BCN

CM	Average Throughput	Standard Deviation/Average (%)	Link Utilization (%)
None	2.49	16.84	99.9
BCN	2.35	0.73	99.9

TABLE II

PERFORMANCE OF BULK TRAFFIC WITH AND WITHOUT BCN

± 4 packets, i.e., a queue length in the range [12, 20], we find from the trace file that the system reaches the stable state within approximately 4 ms.

Fig. 5. Queue length with BCN for the 6-source topology

C. Optimal Parameter Setting

In this part, we present some *preliminary* results on parameter selection for BCN. We use the symmetric topology shown in Fig. 6. In this configuration, there are four bulk TCP sources with a common sink DT. The link between core switch CS and the edge switch ES_5 is the bottleneck. We simulated this topology with a congested link capacity set to 1 Gbps and then repeat it for a capacity of 10 Gbps.

Fig. 6. The symmetric topology

Four different combinations of parameters additive increase rate unit R_u and sampling probability P_m were analyzed using a 2^2 full factorial design [13]. The queue length variation for 1 Gbps and 10 Gbps bottleneck cases are presented in Fig. 7 and Fig. 8, respectively. From Fig. 7, we see that for 1 Gbps case, the best performance is obtained for R_u of 0.8 Mbps and P_m of 0.1. For 10 Gbps case, the best performance is obtained for R_u of 8 Mbps and P_m of 0.05. Notice that the parameter values depend upon the bottleneck capacity. In particular, the rate unit optimal R_u appears to be a fixed fraction of the bottleneck capacity. In the BCN as proposed, the sources do not know the bottleneck capacity and so cannot set the parameters correctly. We, therefore, recommend an extension to BCN in which the BCN messages include the bottleneck capacity. Thus, the enhanced BCN allows the sources to set the parameter R_u accordingly. All the results presented in this paper from here on use this enhanced version of BCN.

These figures also show that larger P_m can restrain the magnitude of the oscillations. However, this results in more frequent BCN messages thereby resulting in higher signaling overhead. So the sampling probability has to be set based on maximum acceptable overhead and the link capacity. On lower rate links, the sampling probability has to be higher to avoid unnecessary delays in feedback.

D. Fairness

With the analytical model, we showed that BCN is proportionally fair. To validate this, we simulated a parking lot topology shown in Fig. 9. Sources ST_1 through ST_4 communicate with a common destination DT_0, ST_5 communicates with DT_1, and ST_6 communicates with DT_2. Note that there are two bottlenecks. The link between switch SW_1 and SW_2 is shared by five sources SR_1 through SR_5. The link between Switch SW_2 and SW_3 is shared by SR_1 through SR_4 and SR_6. We assume that both links have capacity C. The max-min fair allocation for this configuration is obtained by maximizing the following two functions:

$$f_1(\vec{r}) = \min\{r_1, r_2, r_3, r_4, r_5\}$$

subject to

$$r_1 + r_2 + r_3 + r_4 + r_5 \leq C$$

and

$$f_2(\vec{r}) = \min\{r_1, r_2, r_3, r_4, r_6\}$$

subject to

$$r_1 + r_2 + r_3 + r_4 + r_6 \leq C$$

This results in the following max-min fair allocation:

$$r_1 = r_2 = r_3 = r_4 = r_5 = r_6 = C/5$$

Note that in this case, $r_5/r_1 = 1$.

Max-Min fairness assumes that the utility is a linear function of rates, i.e., users find 10 Mbps twice as useful as 5 Mbps. Actually, this may not be true in some cases. For example, a user with zero rate will find even a small additional allocation of 1 kbps very valuable, while a user with 10 Mbps will find little value in an additional allocation of 1 kbps. One possibility is to assume that the utility is a logarithm function of allocated rate. If this were the case, then the fair allocation will be one that maximizes the sum of log of user rates. This leads to the *proportional fair* allocation.

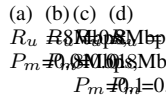

Fig. 7. Performance of BCN for the symmetric topology with 1 Gbps bottleneck link

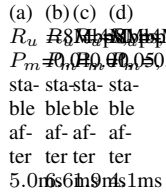

Fig. 8. Performance of BCN for the symmetric topology with 10 Gbps bottleneck link

Fig. 9. Parking Lot Topology

Fig. 10. A simple asymmetric topology

Fig. 12. Topology for Mixed Traffic

For the parking lot configuration, proportional fair allocation is obtained by maximizing the following two functions:

$$f_3(\vec{r}) = \log(r_1) + \log(r_2) + \log(r_3) + \log(r_4) + \log(r_5)$$

subject to

$$r_1 + r_2 + r_3 + r_4 + r_5 \leq C$$

and

$$f_4(\vec{r}) = \log(r_1) + \log(r_2) + \log(r_3) + \log(r_4) + \log(r_6)$$

subject to

$$r_1 + r_2 + r_3 + r_4 + r_6 \leq C$$

This results in the following proportional fair allocation:

$$r_1 = r_2 = r_3 = r_4 = C/6, r_5 = r_6 = C/3$$

Note that in this case, $r_5/r_1 = 2$.

The average throughputs in Gbps for all the sources are shown in Table III.

The simulation results show that r_5/r_1 is approximately 2. Thus, this supports the proposition that BCN is proportionally fair.

In another simulation, using the same topology, we set source ST_6 inactive for the first second. Therefore, in the first second, the first 5 sources are supposed to have equal rates. After one seconds, different levels of rates will be assigned. The average throughput in the last 2 seconds are shown in Table IV. Simple calculations show that it is still approximately proportional fair for all sources. Note that the link utilization is about 93%.

E. Asymmetric Topology and Multiple Congestion Points

In the previous experiments, we assumed that all the links have the same capacity. Now we relax this assumption and simulate the scenario with different capacity bottlenecks. This asymmetric topology is shown in Fig. 10. The link between edge switch ES_5 and DT_2 is 1 Gbps while all other links are 10 Gbps. Sources ST_1 and ST_2 communicate with DT_1 while ST_3 and ST_4 communicate with DT_2. Note that there are two congestion points. Both links ES_5 to DT_1 and ES_5 to DT_2 are congested.

The average throughput in Gbps for various sources is shown in the Table V.

Note that both congested links are highly utilized. The queue lengths are shown in Fig. 11.

Note that there are relatively large oscillations at SW_2. These are due to the congestion propagation between the two congested links, which leads to the perturbations over each other.

Remark on the Oscillations: Recall the derivation on stability and convergence from eqn. (6), the condition[10] that the $r_i(t)$ converges to some rate without oscillation depends on

$$|e_i(t)| \to 0, \text{ as } t \to \infty.$$

Actually, in the simulation for one source scenario, $e_i(t)$ is always non-zero. For single source simulation, $e_i(t) \approx 1$, while in four sources scenario, $e_i(t)$ is much larger, sometimes $e_i = \pm 6$, or $\pm$ 7, which introduces large oscillations in the source rates.

Furthermore, the measurement of the queue lengths does not show exactly how individual sources affect the congestion state of the link. So this error prone measurement, which can be modeled as stochastic perturbation, pushes the source rate away from the convergence point.

F. TCP and UDP Mixed Traffic

In this part, we use the topology shown in Fig. 12. SU_1, ..., SU_4 are the UDP constant bit rate (CBR) sources with rates of 5 Gbps each. These sources communicate with DU. ST_1, ..., ST_4 are TCP sources that communicate with DT.

The average throughput in Gbps for TCP and UDP sources are shown in Table VI.

r_1	r_2	r_3	r_4	r_5	r_6
1.46	1.45	1.54	1.73	3.08	3.02

TABLE III

AVERAGE THROUGHPUT FOR PARKING LOT TOPOLOGY

Time	r_1	r_2	r_3	r_4	r_5	r_6
$1s$	1.81	1.82	1.83	2.23	2.25	0
$2s$	1.68	1.72	1.83	1.64	3.07	3.12
$3s$	1.66	1.79	1.61	1.77	3.12	3.16

TABLE IV

AVERAGE THROUGHPUT FOR PARKING LOT TOPOLOGY: ST_6 IS ACTIVE AFTER 1 SECOND

ST_1	ST_2	ST_3	ST_4
4.34	4.04	0.51	0.49

TABLE V

THROUGHPUT FOR ASYMMETRIC TOPOLOGY

(a) Queue at SW_1 (b) Queue at SW_2

Fig. 11. Queue length in asymmetric topology

ST_1	ST_2	ST_3	ST_4	SU_1	SU_2	SU_3	SU_4
1.23	1.22	1.18	1.01	1.33	1.37	1.31	1.35

TABLE VI

AVERAGE THROUGHPUT FOR MIX TCP AND UDP TRAFFIC

Fig. 13. Queue length for mixed traffic

Note that the average throughput for TCP sources is 1.16 Gbps and for UDP sources it is 1.34 Gbps. These rates are very close. UDP flows have a slightly higher throughput than TCP because TCP flows automatically adjust their rates to a rate below that set by the rate regulators. This is the the peak rate that TCP sources can achieve. The queue lengths are shown in Fig. 13.

Note that UDP traffic does not seem to have significant impact the performance of BCN. Due to the rate regulators at the sources, UDP can also be regarded as elastic flows. Thus, BCN does not discriminate UDP traffic from TCP traffic. This is clearly an advantage over the transport and network layer congestion control mechanisms that reduce only TCP traffic.

G. Bursty Traffic

Using Pareto traffic, we show BCN is robust with bursty traffic. In this experiment, we still use the topology shown in Fig. 12 for mixed traffic. We simulate two scenarios: one with a burst period of 1 ms, the other with 100 ms. The results are shown in Fig. 14. There are very large oscillations when the burst period is 1 ms, which is comparable to the system settling time of 4 ms. With such short burst periods, it is hard to keep the system into a stable state. However, the average queue length is still around Q_{eq}. This shows that BCN is robust to bursty traffic.

VII. CONCLUSION AND FUTURE STEPS

In this paper, we have presented both the analytical and simulation results for BCN mechanism for DCNs. It is shown that BCN ensures high throughput and low latency. We also showed that BCN achieves proportional fairness and not max-min fairness. The difference is visible only when there are multiple congestion points in the networks. Furthermore, BCN is effective in controlling UDP traffic along with TCP traffic. For bursty traffic, the performance is acceptable and robust even when the traffic burst period is very short in the granularity of several milliseconds. In this paper, we have also described the general relationships between BCN parameters and system stability. However, it is difficult to find the optimal parameter settings from the limited set of topologies. Our future work will focus on the simulations to find some critical criterion for parameter selection. The other problem for BCN is the ubiquitous oscillations, especially for the source rates.

(a) Average burst period: 100ms (b) Average burst period: 100ms

Fig. 14. Queue length for burst traffic

As discussed in the paper, although BCN effectively controls queue variation, the source rates may still oscillate. We need to develop extensions that will reduce these oscillations.

VIII. Acknowledgements

Authors would like to thank Davide Bergamasco of Cisco for clearly explaining the BCN mechanism and helping answer questions during this analysis and encouraging us to analyze the scheme.

References

[1] D. Bergamasco, "Data Center Ethernet Congestion Management: Backward Congestion Notification," *IEEE 802.1 Meeting*, May 2005.

[2] D. Bergamasco and R. Pan, "Backward Congestion Notification Version 2.0," *IEEE 802.1 Meeting*, September 2005.

[3] M. Seaman, "Congestion Notification," *Notes for IEEE 802.1 Congestion Management Interim Meeting*, January 2006.

[4] D.M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comp Networks and ISDN System*, 1989.

[5] Jean-Yves Le Boudec, "Rate adaptation, congestion control and fairness: a tutorial," *Ecole Polytechnique Federale de Lausanne(EPFL)*, Chapter 1, November 22, 2005.

[6] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability" *Journal of Operational Research Society*, 49, 1998.

[7] S. Golestani and S. Bhattacharyya, "End-to-end congestion control for the internet: A global optimization framework," *Proc of ICNP*, October 1998.

[8] G. Mcalpine, M. Wadeker et. al., "An architechture for congestion management in Ethernet clusters," IEEE International Parallel and Distributed Processing Symposium, 2005.

[9] J.R. Santos, Yoshio Turner, G. Janakiraman, "End to end congestion control for Infiniband," *IEEE INFOCOM*, 2003.

[10] L. Ljung, "Analysis of Recursive Stochastic Algorithms," *IEEE Transactions on Automatic Control*, Vol. AC22, No.4, August 1977.

[11] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal and B. Vandalore, "The ERICA switch algorithm for ABR traffic management in ATM networks," *IEEE/ACM Transactions on Networking*, Vol. 8, No. 1, February 2000.

[12] NS2, "Network Simulator," available at http://www.isi.edu/nsnam/ns/.

[13] R. Jain, "The Art of Computer Systems Performance Analysis," Wiley, 1991.

IX. List of Symbols

Symbol	Meaning
a_{li}	Fraction of traffic from the ith source over the lth link
$b_i(t)$	Expected value of $e_i(t)^\dagger$
C_l	Capacity of lth link
$\Delta(t)$	Rate of convergence indicator
$e_i(t)$	Feedback to ith source at time t
$f_l(t)$	Total traffic over lth link
G_i	Additive rate increase gain parameter
G_d	Multiplicative decrease gain parameter
$h_i(t)$	$\sum_{l=1}^{L} p_l(t)a_{li}(t)$ = Probability of ith source getting a negative BCN message
$h_i'(t)$	$dh(t)/dt$ = Derivative of $h_i(t)$
i	Source index
l	Link index
L	Number of links in the network
n	Number of sources sharing the bottleneck
N	Number of sources in the network
P_m	Sampling probability of packets
$p_l(t)$	Probability of generating a negative BCN message from lth link at time t
$P_l(t)$	Cumulative probability of generating a negative BCN message from lth link at time t = Primitive of $p_l(x)$ = $\int_{x=0}^{t} g(x)dx$
$P(t)$	Cumulative probability of generating a negative BCN message from all links at time t = $\sum_{l=1}^{L} P_l(t)$
$q(t)$	Queue length at the bottleneck at time t
Q_{sc}	Queue threshold for severe congestion
Q_{eq}	Queue threshold for equilibrium
Q_{st}	Queue threshold for stop (pause) signal
$q_{\text{off}}(t)$	Queue offset = $Q_{eq} - q(t)$
q_a	Number of packets arriving in the last sampling interval
q_d	Number of packets served in the last sampling interval
$q_{\text{delta}}(t)$	Queue difference in the last sampling interval = $q_a - q_d$
R_u	Additive increase rate unit parameter
$r_i(t)$	Rate of ith source at time t
r_i	Same as $r_i(t)$
R_i	Rate of ith source at equilibrium
$s_i(t)$	Rate variation around equilibrium = $r_i(t) - R_i$
S	Size of packets
t	Time
t_n	Time at nth rate update for the ith source
$\mu_i(t)$	Expected time between rate updates
$V(\vec{x})$	Liapunov function of vector $\vec{x}$
W	Relative weight of queue offset and queue difference