Absorbing Micro-burst Traffic by Enhancing Dynamic Threshold Policy of Data Center Switches

Danfeng Shan*†, Wanchun Jiang†§, Fengyuan Ren*†
*Tsinghua National Laboratory for Information Science and Technology, Beijing, China
†Department of Computer Science and Technology, Tsinghua University, Beijing, China
§School of Information Science and Engineering, Central South University, Changsha, China
Email:{dfshan, renfy}@csnet1.cs.tsinghua.edu.cn, jiangwc@csu.edu.cn

***Abstract*—In data center networks, micro-burst is a common traffic pattern and the packet dropping caused by it usually leads to serious performance degradation. Meanwhile, most of the current commodity switches employ on-chip shared memory, and the buffer management policies of them ensure fair sharing of memory among all ports. Among various polices, *Dynamic Threshold* (DT) is widely used by switch vendors. However, because DT needs to reserve a fraction of switch buffer, there is free buffer space while packets from micro-burst traffic are dropped. In this paper, we theoretically deduce the sufficient conditions for packet dropping caused by micro-burst traffic, and estimate the corresponding free buffer size. The results show that the free buffer size is very large when the number of overloaded ports is small. What's worse, to ensure fair sharing of memory among output ports, packets from micro-burst traffic may be dropped even when the traffic size is much smaller than the buffer size. In light of these results, we propose *Enhanced Dynamic Threshold* (EDT) policy, which can alleviate packet dropping caused by micro-burst traffic through fully utilizing the switch buffer and temporarily relaxing the fairness constraint. The simulation results show that EDT can absorb more micro-burst traffic than DT.**

***Index Terms*—dynamic threshold, switch buffer management, shared memory, micro-burst, packet dropping**

I. INTRODUCTION

Micro-burst is a common traffic pattern in modern data center networks, and has been brought into public attention recently [1]–[6]. Generally, it refers to bursty traffic with very small time-scale. It is usually generated by data center services and appears in the switch when packets from multiple concurrent flows are destined to the same output port. For example, in data centers deploying online services, the divide and conquer computing paradigm is widely used, thus large-scale concurrent flows may travel across networks. Micro-burst appears in a switch port when results are aggregated from multiple nodes [7], [8]. Packet dropping caused by micro-burst traffic is usually unacceptable, because micro-burst traffic is comprised of several delay-sensitive short flows, and the triggered timeouts always extend the flow completion time, which lowers the user experience and thus revenue [5], [8]–[10].

Packet dropping in a switch is directly related to its buffer architecture and buffer management policy. Today the majority of switches employ the on-chip shared memory to reduce latency by avoiding packet readings and writings to and from external memory. The on-chip packet buffer is dynamically shared across ports by statistical multiplexing [5], [11], [12]. However, shared memory switches might suffer the fairness problem that few output ports could occupy all of the shared buffer, starving other output ports. In order to overcome the problem, many buffer management policies were proposed to restrict the queue length on each output port.

Among various policies, *Dynamic Threshold* (DT) [13] has been widely used by switch vendors [12], [14]–[20]. In this policy, the queue length is restricted by a dynamic threshold shared by all output ports, which is proportional to the current amount of free buffer space. However, because DT needs to reserve a fraction of buffer so that the newly overloaded ports won't be starving, packets from micro-burst traffic may be dropped even when there is free buffer space in the switch.

In this paper, we theoretically deduce the sufficient conditions for packet dropping caused by micro-burst traffic and quantitively estimate the corresponding free buffer size in DT switches. The analysis results tell that the micro-burst traffic readily results in packet dropping. Besides, the free buffer size when packets are dropped is negatively correlated to the number of overloaded ports. Particularly, when the number of overloaded ports is small, the amount of wasted buffer would be especially large. If these buffer can be utilized by the overloaded ports, additional 50% - 100% micro-burst traffic can be absorbed. Further more, to ensure fair sharing of memory, the queue length of each overloaded port is restricted by the same threshold. As a result, when several ports are overloaded, packets from micro-burst traffic will be dropped even through the micro-burst traffic size is much smaller than the buffer size. However, it is of great importance to avoid packet dropping of micro-burst traffic in data center networks. On the other hand, when more buffer is temporarily allocated to the ports transmitting micro-bust traffic, there will be few effects on the fairness among ports transmitting long-lived flows, because the time-scale of micro-burst is quite short compared to the duration of a long-lived flow. Therefore, the fairness constraint of DT can be relaxed to absorb micro-burst traffic.

In light of these, we propose the *Enhanced Dynamic Threshold* (EDT) policy, which can absorb micro-burst traffic as much as possible through fully utilizing the buffer and temporarily relaxing the fairness constraint when micro-burst traffic arrives at a port. EDT has three advantages: (1). Buffer is fully used to absorb micro-burst traffic. (2). Buffer is fairly shared among output ports transmitting long-lived flows. (3). EDT is simple enough to be implemented in high-speed switches, as it is comprised by several counters and timers.

We evaluate DT and EDT on ns-2 platform [21]. The results show that in the worst case 50% of buffer remains unused when micro-burst traffic causes packet dropping in DT switches. In comparison, packets will not be dropped until there is no free buffer space in EDT switches. Moreover, although EDT temporarily relaxes the fairness constraint, buffer is fairly shared among output ports in the long run. Above all, in DT switches, only micro-burst traffic whose duration is no longer than 3ms can be absorbed. But in EDT switches, almost all micro-burst traffic can be absorbed when the traffic duration is shorter than 5ms.

The rest of the paper is organized as follows: In Section II, we introduce the DT policy, then the sufficient conditions for packet dropping caused by micro-burst traffic is deduced and the corresponding free buffer size is estimated. Section III describes the design of EDT . Evaluation is presented in Section IV. Finally, the paper concludes in Section V.

II. Analysis of Dynamic Threshold

A. DT Policy

Before analysis, we would briefly introduce the DT policy. DT is a threshold-based buffer management policy, in which the queue lengths of all ports are constrained by the same threshold. Packets are not allowed to enter into the queue whenever the queue length exceeds or equals to the threshold. The key idea of DT is that the threshold is proportional to the current amount of unused buffer space. More precisely, let $Q_i(t)$ be the queue length of port i at time t and B be the shared buffer size, then the threshold $T(t)$ can be given by

$$T(t) = \alpha \cdot \left(B - \sum_i Q_i(t) \right) \tag{1}$$

where α is a control parameter. DT reserves a fraction of buffer all the time such that other ports won't be starved.

To understand the mechanism of DT, consider the following scenario. Assume that the switch buffer is empty and the k-th output port becomes overloaded at time $t = 0$, then $\sum_i Q_i(t) = Q_k(t)$ when $t = 0^+$. Let $\alpha = 2$, then $T(t) = 2 \cdot (B - Q_k(t))$. At time $t = 0$, $Q_k(0) = 0$ and $T(0) = 2B$, thus $Q_k(0) < T(0)$. Packets are allowed to enter into the buffer, and $Q_k(t)$ will increase until $Q_k(t) = T(t) = 2B/3$, as illustrated in Fig. 1. Once $T = Q_k$, the port is not allowed to occupy additional buffer and the queue length will not increase any longer. The reserved buffer size in this case is $B/3$.

B. Analysis

For the convenience of expression, we give the following names about the status of a switch output port.

1) **Overloaded and Underloaded State:** A port is in overloaded state if and only if the arriving rate of traffic to this port is larger than the port's transmitting rate. Otherwise, the port is in underloaded state. More precisely, let the arriving rate of traffic to the i-th output port be R_i. Let C denote the link capacity. Then port i is overloaded if and only if $R_i > C$.
2) **Steady State:** When a port is in the **overloaded** state, it reaches steady state if and only if its queue length is equal to the threshold and the queue length as well as the threshold will not change for a while. More precisely, port i reaches steady state at time t if and only if $T(t) = Q_i(t)$ and $T'(t) = Q'_i(t) = 0$.

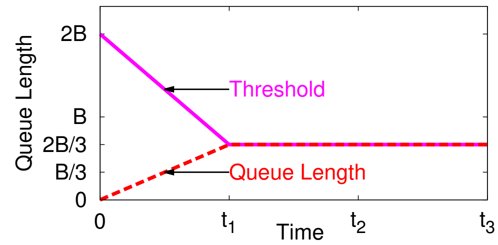


Fig. 1: Queue length and threshold evolutions

TABLE I: Notations

Not.	Description
R_i	the arriving rate of traffic to i-th port
C	link capacity
$Q_i(t)$	queue length of i-th port at time t
B	total buffer size
$T(t)$	threshold at time t
$F(t)$	free buffer size at time t
d_i	duration of flows in i-th port

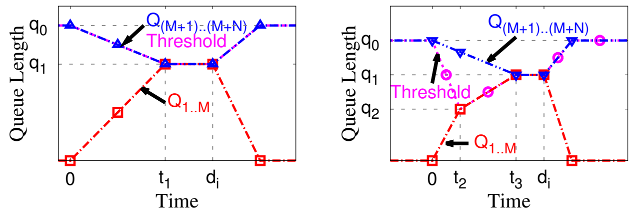
Consider a switch with P output ports and buffer size B. At time $t = 0$, the queues of port $1, \cdots$, port M are empty, and port $(M+1), \cdots$, port $(M+N)$ have reached their steady states. Port $1, \cdots$, port M begin to transmit micro-burst traffic and become overloaded at time $t = 0^+$. Let R_i be the arriving rate of micro-burst traffic to port i and d_i be the duration of micro-burst traffic in port i. The free buffer size at time t is denoted by $F(t)$. These notions are summarized in TABLE I for the sake of terseness.

In the rest of this section, we'll deduce the sufficient conditions for packet dropping caused by micro-burst traffic and estimate the corresponding free buffer size in a particular case in the beginning. Following the same way, we'll make the analysis in more general cases.

1). $R_i\,(i = 1, 2, \cdots, M)$ is constant and $R_1 = R_2 = \cdots = R_M = R$

The evolutions of queue lengths and threshold in this case have been analyzed in [13] in detail. However, for the convenience of explaining the following cases, we'll briefly show the analysis.

At time $t = 0^+$, as micro-burst traffic arrives at port $1, \cdots$, port M, the unused buffer will be occupied. Thus, the threshold will decrease, which makes $Q_{M+1}, \cdots, Q_{M+N}$ decrease. The maximum decreasing rate of queue length is C, when no packets are entering into the queue and packets in



(a) $R \leqslant C\left(1+\frac{1+\alpha N}{\alpha M}\right)$ (b) $R > C\left(1+\frac{1+\alpha N}{\alpha M}\right)$

Fig. 2: Evolutions of queue lengths and threshold

the queue are transmitted at a rate of C (the port transmitting rate). Therefore, there are two cases.

a). $R \leqslant C\left(1+\frac{1+\alpha N}{\alpha M}\right)$

In this case, $|T'(0^+)| \leqslant C$. Therefore, at time $t = 0^+$, $Q_{M+1}, \cdots, Q_{M+N}$ will decrease at the same rate as that of threshold, as is illustrated in Fig. 2a. Meanwhile, $Q_1, \cdots, Q_M$ will increase at a rate of $(R - C)$, until $Q_1, \cdots, Q_M$ hit the threshold at time $t = t_1$. According to [13], time t_1 is given by

$$t_1 = \frac{\alpha B}{[1+\alpha(M+N)](R-C)} \tag{2}$$

Then packets are dropped since port 1, $\cdots$, port M are not allowed to acquire additional buffer. Therefore, the sufficient condition for packet dropping in port i is

$$d_i \geqslant t_1 \tag{3}$$

According to [13],

$$T(t_1) = \frac{\alpha B}{1+\alpha(N+M)} \tag{4}$$

Therefore, the free buffer size while packets are dropped is

$$F(t_1) = \frac{T(t_1)}{\alpha} = \frac{B}{1+\alpha(N+M)} \tag{5}$$

b). $R > C\left(1+\frac{1+\alpha N}{\alpha M}\right)$

In this case, $|T'(0^+)| > C$. Therefore, at time $t = 0^+$, $Q_M, \cdots, Q_{M+N}$ will decrease at a rate of C, which is lower than the decreasing rate of threshold, as is illustrated in Fig. 2b. Meanwhile, $Q_1, \cdots, Q_M$ will increase at a rate of $(R - C)$, until $Q_1, \cdots, Q_M$ hit the threshold at time $t = t_2$. According to [13], time t_2 is given by

$$t_2 = \frac{\alpha B}{(1+\alpha N)[(1+\alpha M)(R-C)-\alpha NC]} \tag{6}$$

Then packets are dropped since the increasing rate of $Q_1, \cdots, Q_M$ is limited by DT. At the same time, $Q_{M+1}, \cdots, Q_{M+N}$ will keep decreasing. Thus, the threshold and $Q_1, \cdots, Q_M$ will increase at the same rate until all of the ports reach the steady state. In this case, the sufficient condition for packet dropping in port i is

$$d_i \geqslant t_2 \tag{7}$$

And according to [13],

$$T(t_2) = \frac{\alpha(R-C)B}{(1+\alpha N)[(1+\alpha M)(R-C)-\alpha NC]} \tag{8}$$

Therefore, the free buffer size while packets begin to be dropped is

$$F(t_2) = \frac{T(t_2)}{\alpha} = \frac{(R-C)B}{(1+\alpha N)[(1+\alpha M)(R-C)-\alpha NC]} \tag{9}$$

Considering these two cases, we can summarize the sufficient conditions for packet dropping and free buffer size while the packets from micro-burst traffic are dropped into the following theorem.

Theorem 1. *When $R_1 = R_2 = \cdots = R_M = R$, the packets from micro-burst traffic will be dropped in port k ($k = 1, 2, \cdots, M$) if*

$$d_k \geqslant \begin{cases} \dfrac{\alpha B}{[1+\alpha(M+N)](R-C)}, \\ \quad \text{if } R \leqslant C\left(1+\frac{1+\alpha N}{\alpha M}\right) \\ \dfrac{\alpha B}{(1+\alpha N)[(1+\alpha M)(R-C)-\alpha NC]}, \\ \quad \text{if } R > C\left(1+\frac{1+\alpha N}{\alpha M}\right) \end{cases} \tag{10}$$

and the free buffer size while packets are dropped is

$$F = \begin{cases} \dfrac{B}{1+\alpha(M+N)}, \\ \quad \text{if } R \leqslant C\left(1+\frac{1+\alpha N}{\alpha M}\right) \\ \dfrac{(R-C)B}{(1+\alpha N)[(1+\alpha M)(R-C)-\alpha NC]}, \\ \quad \text{if } R > C\left(1+\frac{1+\alpha N}{\alpha M}\right) \end{cases} \tag{11}$$

Remarks:

When $R \leqslant C\left(1+\frac{1+\alpha N}{\alpha M}\right)$, equation (10) can be rewritten as

$$R \cdot d_k - C \cdot d_k \geqslant \frac{\alpha B}{1+\alpha(M+N)} \tag{12}$$

If the micro-burst traffic size (i.e., $R \cdot d_k$) is fixed, then the condition (12) can be easily satisfied for small d_k or larger R. This is why micro-burst traffic readily results in packet dropping.

Besides, when the packets are dropped, the free buffer size is negatively correlated to the number of overloaded ports (i.e., $M+N$). Particularly, when the number of overloaded ports is small, the free buffer size would be very large (e.g. $B/2$ if $M+N=1$ and $\alpha = 1$). DT reserves this fraction of memory for two reasons. Firstly, it provides a cushion for newly overloaded ports, so that these ports will not starve for memory. Secondly, because the threshold of DT is proportional to the amount of unused memory, the action that the reserved memory is occupied can be used to notify DT to change the threshold. However, the reserved buffer should be utilized when a port is transmitting micro-burst traffic. Because on the one hand, the time-scale of micro-burst traffic is quite short. Occupying reserved buffer will only last for relatively short time and is worthwhile since it contributes to absorbing the micro-burst traffic. On the other hand, DT can be simply implemented by using a shift register and a free buffer size counter if α is a power of two. The actions that a packet enters into and departs from the buffer can be used to inform DT of adjusting threshold instead.

Moreover, from Fig. 2a, we have the following observation. To ensure fair buffer sharing among overloaded ports, the packets from micro-burst traffic will be dropped after the queue lengths of newly overloaded ports reach the queue lengths of other ports. As a result, packets may be dropped even though the micro-burst traffic size is far smaller than the buffer size. However, avoiding packet dropping caused by micro-burst traffic is of great importance. In addition, it has few effects on the fairness among ports transmitting long-lived flows that more shared buffer is allocated to the ports transmitting micro-burst traffic, because the time-scale of micro-burst traffic is quite short compared to the durations of long-lived flows. Therefore, the fairness constraint of DT could be temporarily relaxed to absorb micro-burst traffic.

The similar insights can be obtained in the case $R > C\left(1+\frac{1+\alpha N}{\alpha M}\right)$.

2). $R_i\,(i=1,2,\cdots,M)$ is constant and $R_1 \geqslant R_2 \geqslant \cdots \geqslant R_M$

In this case, the sufficient conditions for packet dropping caused by micro-burst traffic and the corresponding free buffer size can be given by the following two theorems.

Theorem 2. *When $\sum_{i=1}^{M}(R_i - C) \leqslant \frac{(1+\alpha N)C}{\alpha}$, packets will be dropped in port k $(k=1,2,\cdots,M)$ if*

$$d_k \geqslant t_k \tag{13}$$

where

$$\begin{cases} t_k &= \dfrac{\alpha\left[F_{k-1} + \alpha F_{k-1}(N+k-1) + G_k t_{k-1}\right]}{(R_k - C)[1+\alpha(N+k-1)] + \alpha G_k} \\ F_k &= F_{k-1} - \dfrac{G_k(t_k - t_{k-1})}{1+\alpha(N+k-1)} \\ G_k &= \sum_{i=k}^{M}(R_i - C) \end{cases} \tag{14}$$

Time t_k denotes the first time when the queue length Q_k hits the threshold; $t_0 = 0$. And F_k denotes the free buffer size at time $t = t_k$. At $t = 0$, $F_0 = B/(1+\alpha N)$. Next, we'll use mathematical induction to proof this theorem.

Proof:

a). Basis: Inequation (13) and equation (14) hold for port 1 (i.e., $k = 1$)

At $t = 0$, only port $(M+1)$, $\cdots$, port $(M+N)$ are overloaded and they have reached their steady states, therefore we have

$$\begin{cases} T(0) &= \alpha F_0 \\ F_0 &= B - \sum_{i=M+1}^{M+N} Q_i(0) \\ Q_i(0) &= T(0), \quad i = M+1, M+2, \cdots, M+N \end{cases} \tag{15}$$

Solving F_0 from (15), we get

$$F_0 = \frac{B}{1+\alpha N} \tag{16}$$

Port 1, $\cdots$, port M become overloaded at time $t = 0^+$; the traffic arriving rate in port i is R_i. Thus, at time $t = 0^+$, $Q_1, \cdots, Q_M$ will increase at a rate of $(R_i - C)$. As port 1, $\cdots$, port M occupy the free buffer, the free buffer size will decrease, which causes the decreasing of the threshold, and then $Q_{M+1}, \cdots, Q_{M+N}$ will exceed the threshold and decrease. Let D denote the decreasing rate of $Q_{M+1}, \cdots, Q_{M+N}$ $(D < 0)$, Then, at $t = 0^+$, the free buffer size will change as

$$F(t) = F_0 - G_1 \cdot t - ND \cdot t \tag{17}$$

Thus, the dynamic threshold will change as

$$T(t) = \alpha\left(F_0 - G_1 \cdot t - ND \cdot t\right) \tag{18}$$

Differentiating both sides of (18), we have

$$T'(t) = -\alpha G_1 - \alpha ND, \quad t = 0^+ \tag{19}$$

When $G_1 \leqslant \frac{(1+\alpha N)C}{\alpha}$, the decreasing rate of threshold at time $t = 0^+$ is no larger than C, namely,

$$T'(t) \geqslant -C \tag{20}$$

We can proof this by contradiction. The maximum decreasing rate of queue length is C. Thus, if $T'(t) < -C$, $Q_{M+1}, \cdots, Q_{M+N}$ will decrease at a rate of C. Meanwhile, since $G_1 \leqslant \frac{(1+\alpha N)C}{\alpha}$, we have

$$T'(t) \geqslant -C - \alpha N(C + D) \tag{21}$$

Substituting $D = -C$ into (21), we have $T'(t) \geqslant -C$, which contradicts with the previous hypothesis.

Inequation (20) means that the threshold will decrease at a rate lower than the port transmitting rate. Therefore, $Q_i\,(i = M+1, M+2, \cdots, M+N)$ will decrease at the same rate as that of threshold, namely, $D = T'(t)$. Combining (19), we have

$$D = T'(t) = -\frac{\alpha G_1}{1+\alpha N} \tag{22}$$

Substituting (22) into (18), we yield

$$T(t) = \alpha\left(F_0 - \frac{G_1}{1+\alpha N} \cdot t\right), \quad t = 0^+ \tag{23}$$

Equation (23) will hold until the queue length in port 1 hits the threshold at time $t = t_1$, then the packets in port 1 are dropped, namely,

$$T(t_1) = (R_1 - C) \cdot t_1 \tag{24}$$

Solving t_1 from (24), we get

$$t_1 = \frac{\alpha F_0(1+\alpha N)}{(R_1 - C)(1+\alpha N) + \alpha G_1} \tag{25}$$

Therefore, in port 1, packets are dropped if $d_1 \geqslant t_1$.

At time t_1, the free buffer size reduces to

$$F_1 = F_0 - \frac{G_1 t_1}{1+\alpha N} \tag{26}$$

Thus, inequation (13) and equation (14) hold for $k = 1$.

b). Inductive step:

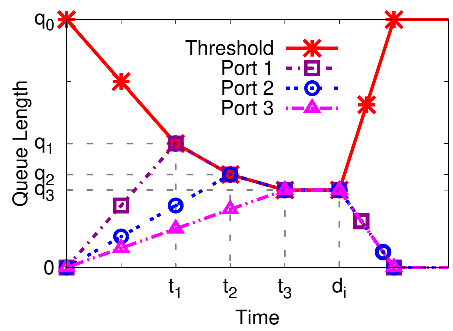


Fig. 3: The evolutions of queue lengths and threshold when $M = 3$ (Theorem 2)

We assume that inequation (13) and equation (14) hold for port i $(1 \leqslant i \leqslant M-1)$.

After the queue length of port i hits the threshold at time t_i, the evolutions of queue lengths and threshold are the same as those at time $t = 0^+$, except that the free buffer size is F_i, and there are $N_i = N + i$ output ports whose queue lengths decrease at the same rate as that of threshold. Equation (23) can be rewritten as

$$T(t) = \alpha \cdot \left[F_i - \frac{G_{i+1}}{1+\alpha N_i} \cdot (t - t_i)\right], \quad t = t_i^+ \tag{27}$$

Equation (27) holds until the queue length Q_{i+1} hits the threshold at $t = t_{i+1}$, namely, $T(t_{i+1}) = (R_{i+1} - C) \cdot t_{i+1}$. Then the packets in port $(i+1)$ are dropped. Solving t_{i+1}, we have

$$t_{i+1} = \frac{\alpha\left[F_i(1+\alpha N_i) + G_{i+1}t_i\right]}{(R_{i+1}-C)(1+\alpha N_i) + \alpha G_{i+1}} \tag{28}$$

Therefore, the packets in port $i+1$ will be dropped if $d_{i+1} \geqslant t_{i+1}$.

At time t_{i+1}, the free buffer size reduces to

$$F_{i+1} = F_i - \frac{G_{i+1}(t_{i+1} - t_i)}{1+\alpha N_i} \tag{29}$$

Thus, the inequation (13) and equation (14) hold for $k = i+1$.

In conclusion, the inequation (13) and equation (14) hold for $k = 1, 2, \cdots, M$.

The evolutions of queue lengths and threshold are illustrated in Fig. 3 ■

We also have the following theorem when $\sum_{i=1}^{M}(R_i - C)$ is larger than $\frac{(1+\alpha N)C}{\alpha}$:

Theorem 3. *When $\sum_{i=1}^{M}(R_i - C) > \frac{(1+\alpha N)C}{\alpha}$, packets in port k $(k = 1, 2, \cdots, L)$ will be dropped if*

$$d_k \geqslant t_k \tag{30}$$

where

$$\begin{cases} t_k = \dfrac{\alpha\left\{F_{k-1} + [G_k - (N+k-1)C]t_{k-1}\right\}}{\alpha[G_k - (N+k-1)C] + R_k - C}, \\ F_k = F_{k-1} - [G_k - (N+k-1)C](t_k - t_{k-1}), \\ G_k = \sum_{i=k}^{M}(R_i - C) \end{cases} \tag{31}$$

L is the largest k such that $G_k > \frac{(1+\alpha N_k)C}{\alpha}$ and $L \leqslant M$.

The denotations and initial values of t_k and F_k are the same as those in Theorem 2. Again, we use mathematical induction to proof this theorem.

Proof:

a). Basis: Inequation (30) and equation (31) hold for port 1 (i.e., $k = 1$)

In this case, the decreasing rate of threshold is larger than the port transmitting rate. Therefore, at time $t = 0^+$, Q_k $(k = M+1, M+2, \cdots, M+N)$ will decrease at a rate of C. Meanwhile, Q_k $(k = 1, 2, \cdots, M)$ will increase at a rate of $(R_k - C)$. Therefore, the threshold will change as

$$T(t) = \alpha\left[F_0 - (G_1 - NC) \cdot t\right] \tag{32}$$

where F_0 is given in (16).

Equation (32) holds until $t = t_1$ when Q_1 hits the threshold and the packets in port 1 are dropped, namely,

$$T(t_1) = (R_1 - C)t_1 \tag{33}$$

Solving t_1 from (33), we have

$$t_1 = \frac{\alpha F_0}{\alpha(G_1 - NC) + (R_1 - C)} \tag{34}$$

Thus, the packets in port 1 will be dropped if $d_1 \geqslant t_1$.

The free buffer size at time t_1 is given by

$$F_1 = F_0 - (G_1 - NC)t_1 \tag{35}$$

Thus, inequation (30) and equation (31) hold for $k = 1$

b). Inductive step:

We assume that inequation (30) and equation (31) hold for port i $(1 \leqslant i \leqslant L-1)$.

After Q_i hits the threshold at time t_i, the evolutions of queue lengths and threshold are the same as those at time $t = 0^+$, except that the free buffer size is F_i and there are $N_i = N + i$ output ports whose queue lengths decrease at the rate of C. Thus, equation (32) can be rewritten as

$$T(t) = \alpha \cdot \left[F_i - (G_{i+1} - N_i C) \cdot (t - t_i)\right], \quad t = t_i^+ \tag{36}$$

Equation (36) holds until the queue length Q_{i+1} hits the threshold at $t = t_{i+1}$, namely, $T(t_{i+1}) = (R_{i+1} - C) \cdot t_{i+1}$. Then the packets in port $(i+1)$ begin to be dropped. Solving t_{i+1}, we have

$$t_{i+1} = \frac{\alpha[F_i + (G_{i+1} - N_i C)t_i]}{\alpha(G_{i+1} - N_i C) + R_{i+1} - C} \tag{37}$$

Thus the packets in port $(i+1)$ will be dropped if $d_{i+1} \geqslant t_{i+1}$.

At time t_{i+1}, the free buffer size reduces to

$$F_{i+1} = F_i - (G_{i+1} - N_i C)(t_{i+1} - t_i) \tag{38}$$

Therefore, the inequation (30) and equation (31) hold for $k = i+1$.

In conclusion, the inequation (30) and equation (31) hold for $k = 1, 2, \cdots, L$. ■

3). $R_i (i = 1, 2, \cdots, M)$ varies with time

Following the same way, the sufficient conditions for packet dropping caused by micro-burst traffic and the corresponding free buffer size can be given in this case. But we leave out the analysis because of the limitations of space.

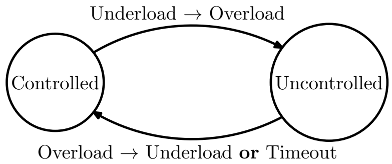


Fig. 4: State transition diagram of EDT in each port.

III. EDT Policy

Analysis results indicate that the switch buffer should be fully utilized and the fairness constraint of DT should be temporarily relaxed to absorb micro-burst traffic. Therefore, in this section, we propose *Enhanced Dynamic Threshold* (EDT) policy to avoid packet dropping caused by micro-burst traffic. The basic idea is presented next, followed by details of EDT.

A. Basic Idea

EDT allows an output port to aggressively occupy buffer in a relatively short interval when the port becomes overloaded. Specifically, for each port, EDT has two states: controlled state and uncontrolled state. In the controlled state, the port threshold is determined by DT. In the uncontrolled state, the port threshold is temporarily set to the buffer size. Fig. 4 depicts the state transition diagram of EDT in each port. At the beginning, EDT is in controlled state. It turns into uncontrolled state when bursty traffic arrives and the port becomes overloaded. If the port is transmitting micro-burst traffic, it will become underloaded after a very short time. Then EDT will return to controlled state. If the port is transmitting long-lived flows, EDT will return to controlled state after a specified period. The specified period is longer than the duration of most micro-burst traffic and much shorter than the durations of long-lived flows.

EDT has three advantages:

1) The output port can occupy every piece of available buffer when it becomes overloaded. Thus packets from micro-burst traffic are dropped only when it is inevitable.
2) Buffer could be fairly shared among output ports transmitting long-lived flows, because the period over which EDT stays in uncontrolled state is very short.
3) EDT is simple enough to be implemented in high-speed switches, as it only requires several additional timers and counters.

The main challenge of EDT is how to recognize that the output port becomes overloaded. From analysis, we observe that when the output port becomes overloaded, *its queue length is increasing and no packets are dropped* at the beginning, so we use this characteristic for recognition.

B. Details of EDT

Fig. 5 illustrates the circuit diagram of EDT added to each output port. Inputs of this diagram are enqueue signal, dequeue signal, and packet dropping signal generated by each logic output queue of the port. A pulse is generated on them whenever a packet is enqueued, dequeued, and dropped

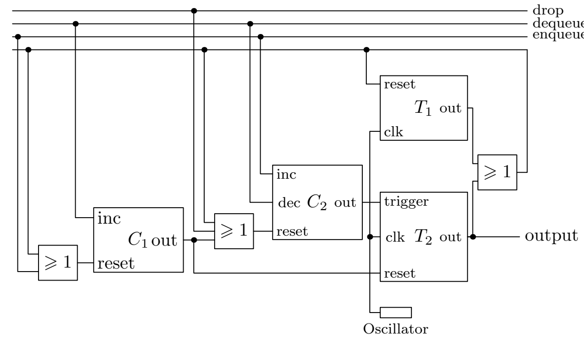


Fig. 5: Circuit diagram

respectively to and from the queue. The output in this diagram determines whether EDT is in uncontrolled state. T_1 and T_2 are countdown timers, and they begin to count down from their default values once they are enabled. C_1 and C_2 are counters, and they increase or decrease their values for every input pulse. Next, we'll show the designing details of these timers and counters one by one.

T_2 is used for controlling the period over which EDT stays in uncontrolled state.[1] It begins to count down when its trigger pin receives a pulse signal, and stops when its value reaches 0. When T_2 is counting down, its output pin is set to 1 which signals EDT to set the threshold of this port to the buffer size. And when it reaches 0, its output pin is set to 0 to signal EDT to return to the controlled state. The default value of T_2 should be longer than the duration of most micro-burst traffic and much smaller than the durations of long-lived flows (e.g. 10ms).

C_2 is used for identifying that the output port becomes overloaded. It works when EDT is in the controlled state. It increases for each pulse on enqueue signal and decreases for each pulse on dequeue signal. Therefore, its value represents for the queue length increment. When it reaches its counting number, EDT will change into the uncontrolled state, and a pulse will be output to notify T_2 to start counting, then C_2 will restart counting from 0. The counting number influences the sensitivity of identifying overloaded state. On the one hand, if this value is too huge, T_2 will not be triggered until the packets from micro-burst traffic are dropped. On the other hand, if this value is too tiny, T_2 will be triggered frequently, which results in unfairness among output ports transmitting long-lived flows. Thus C_2 should obey the following three rules:

Rule 1: C_2 works only when the port becomes overloaded.
Rule 2: C_2 reaches its counting number before packets are dropped.
Rule 3: The counting number should be as large as possible on the premise of following Rule 2.

From Fig. 3, we notice that when a port becomes overloaded, its queue length is increasing and no packets are dropped at the beginning. Therefore, we let C_2 reset itself whenever a

[1] To simplify the implementation of the solution, we only use a timer here. However, this might be sub-optimal. We'll improve it in our future work.

packet is dropped to obey Rule 1. Let the counting number of C_2 be cn_2. Then cn_2 should satisfy the following inequality to obey Rule 2:

$$cn_2 \leqslant (R - C) \cdot t_1 \tag{39}$$

Meanwhile,

$$\begin{aligned} (R-C)\cdot t_1 &> \lim_{R\to\infty}\left[(R-C)\cdot t_1\right] \\ &= \frac{\alpha B}{(1+\alpha N)(1+\alpha M)} \\ &\geqslant \frac{4\alpha B}{(2+\alpha P)^2} \end{aligned} \tag{40}$$

where P is the number of switch ports. Thus cn_2 should satisfy inequality

$$cn_2 \leqslant \frac{4\alpha B}{(2+\alpha P)^2} \tag{41}$$

To obey Rule 3, we can set $cn_2 = \frac{4\alpha B}{(2+\alpha P)^2}$.

T_1 is used for making sure that T_2 is triggered only by bursty traffic. Because if the arriving rate of micro-burst traffic is too low, no packets will be dropped. Uncontrolling queue length in such scenario is unnecessary and may cause unexpected results. Therefore, it's essential to add bursty traffic detection to EDT. T_1 works as follows. When C_2 begins to increase, T_1 begins to count down from its default value as well. If the value of C_2 has not reached its counting number yet when T_1 reaches 0, a pulse is sent to C_2 to notify it to reset itself. If the value of C_2 reaches its counting number before T_1 reaches 0, T_1 is reset. In this way, T_2 is triggered only by bursty traffic. Unlike T_2, T_1 keeps working all the time. Its default value is given as follows. No packets are dropped when the arriving traffic duration (denoted by d) satisfies the following inequality:

$$d < t_1 = \frac{\alpha B}{[1+\alpha(M+N)]\,(R-C)} \tag{42}$$

Equation (42) can be rewritten as

$$R - C < \frac{\alpha B}{[1+\alpha(M+N)]\cdot d} \tag{43}$$

Meanwhile,

$$\frac{\alpha B}{[1+\alpha(M+N)]\cdot d} \geqslant \frac{\alpha B}{(1+\alpha P)\cdot d} \tag{44}$$

Thus, the packets will not be dropped if

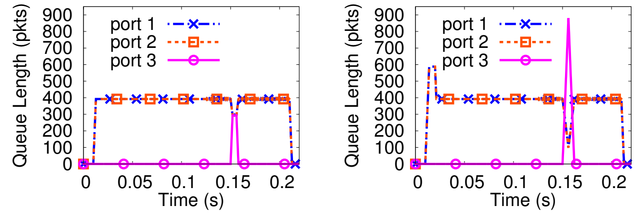
$$R - C < \frac{\alpha B}{(1+\alpha P)\cdot d} \tag{45}$$

If the period over which C_2 increases from 0 to cn_2 is denoted by t_{c2}, then packets will not be dropped if

$$t_{c2} > \frac{cn_2}{\alpha B/\left[(1+\alpha P)\cdot d\right]} = \frac{4(1+\alpha P)}{(2+\alpha P)^2}\cdot d \tag{46}$$

where d is longer than the duration of most micro-burst traffic. Thus the default value of T_1 should be set to $\frac{4(1+\alpha P)}{(2+\alpha P)^2}\cdot d$.

C_1 is used for identifying that the output port returns to the underloaded state. On the one hand, the queue length will not keep increasing all the time when the output port is overloaded, since the port is transmitting packets at the same time. Thus the shape of queue length evolution curve is like a sawtooth. On the other hand, if a few packets are dequeued without any new arrivals in the queue, EDT should be able to judge that the port is underloaded. Therefore we use C_1 to record the number of successive dequeued packets. Specifically, it increases when a packet leaves from the queue and resets itself when a packet enters into the queue in the buffer. When C_1 reaches its counting number, a pulse is sent to reset C_2. The counting number of C_1 is set depending on the network environment. It should usually be between 2 and 10.



(a) DT

(b) EDT

Fig. 6: Evolutions of queue lengths when $N = 2,\ M = 1$

IV. Simulation and Evaluation

In this section, we compare the performances of DT and EDT by simulations on ns-2 platform [21]. We use three metrics for evaluating:

- Buffer utilization when packets from micro-burst traffic are dropped
- The ability to absorb micro-burst traffic
- Fairness among output ports transmitting long-lived flows

We consider a 16-port 1Gbps switch with 1MB shared memory. When a port is overloaded, the arriving rate of traffic is 2Gbps. Packet size is fixed to 850B — the average packet size in data center networks [2]. Inferred from [13], we set α to 1 so that DT performs well. The counting number of C_1 is set to 3. The default value of T_2 is set to 10ms. According to the above guidelines about parameter settings, the counting number of C_2 is 14 and default value of T_1 is 2.1ms.

A. Deterministic Scenario

In deterministic scenario, N output ports are overloaded and have reached their steady states. Meanwhile, M output ports begin to transmit micro-burst traffic and become overloaded.

Firstly, to show how EDT works, we set the duration of micro-burst traffic to 6ms and let $N = 2$, $M = 1$. The queue length evolution of each output port is shown in Fig. 6. Port 3 begins to transmit micro-burst traffic at $t = 0.15$s and finishes transmission 6ms later. In DT switches, packets in port 3 are dropped immediately after the arriving of micro-burst traffic. In comparison, in EDT switches, port 3 can take over as much buffer as possible at the beginning and other ports will make way for it temporarily. When port 1 and port

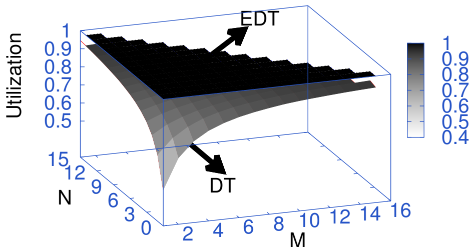


Fig. 7: Buffer utilization for different Ns and Ms when packets from micro-burst traffic are dropped

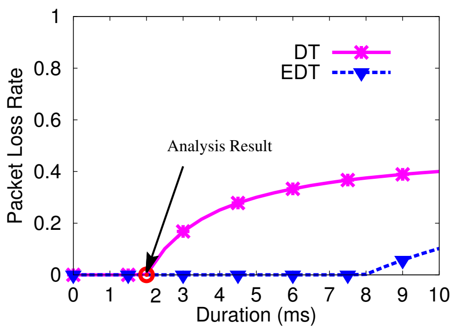


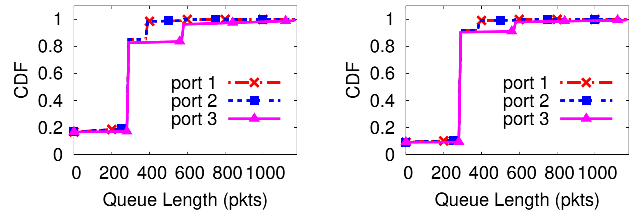
Fig. 8: Packet loss rate of micro-burst traffic as a function of its duration when $N = 2$ and $M = 1$

2 become overloaded, they can also take over as much buffer as possible at the beginning. However, after 10 milliseconds, timeout happens and their queue lengths are restricted. This period is very short compared with the duration of a long-lived flow, which is usually in the order of seconds.

The buffer utilization for different Ns and Ms when packets from micro-burst traffic are dropped is shown in Fig. 7. In DT switches, the utilization decreases as the number of overloaded ports decreases. In the worst case, the utilization is only 50.0%. Compared to it, in EDT switches, the utilization is 100% for all Ns and Ms, which implies that packets are dropped only when it is inevitable.

Fig. 8 illustrates the packet loss rate of micro-burst traffic as a function of its duration when $N = 2$ and $M = 1$. Apparently, the condition given by theorems in Section II agrees with the simulation result. Moreover, in DT switches, packet dropping caused by micro-burst traffic happens when the micro-burst traffic duration reaches 2ms. While in EDT switches packet dropping won't happen until the duration is longer than 8ms. Note that when the duration is 2ms, the traffic size is 2ms $\times$ 2Gbps $=$ 0.5MB and it only needs 0.25MB switch memory, while packets are dropped in DT switches with 1MB buffer in this scenario. On the other hand, when the duration is 8ms, the traffic size is 8ms $\times$ 2Gbps $=$ 2MB and it needs 1MB switch memory. Packet dropping is inevitable in this scenario.

Finally, we evaluate the fairness among output ports transmitting long-lived flows. The unfairness happens when an output port transmitting long-lived flows becomes overloaded, because the port can occupy much more buffer than other ports at that time. Therefore, we consider a scenario that a port becomes overloaded while other ports have reached their steady states. All of them are transmitting long-lived flows when they are overloaded, and the traffic arriving rate in each port is 2Gbps. The CDF of queue length in each port is shown in Fig. 9, where port 3 corresponds to the newly overloaded port. Apparently, EDT is fair when the durations of long-lived flows are 500ms and much fairer when the durations reach 1s.



(a) duration: 500ms (b) duration: 1000ms

Fig. 9: Queue length CDFs with different durations of long-lived flows

B. Stochastic Scenario

Next, we evaluate DT and EDT in a stochastic scenario. In this scenario, there are two kinds of traffic in each output port: background traffic and micro-burst traffic. We use Possion model to simulate background traffic and use exponential On/Off model to simulate micro-burst traffic. In exponential On/Off model, packets are generated at a constant rate of 2Gbps during "on" periods. Both "on" and "off" intervals follow exponential distribution. The average "on" and "off" period is set to 3ms and 191ms, respectively. The average arriving rate of background traffic is 0.33Gbps, so that utilization of each output port is 50% and the total background traffic size is 2 times that of micro-burst traffic.

Firstly, we evaluate DT and EDT by whether the buffer is fully utilized when packets from micro-burst traffic are dropped. Fig. 10 illustrates the average buffer utilization for different micro-burst durations. In DT switches, buffer is fully utilized only when the micro-burst traffic duration is shorter than 2ms. The buffer utilization is only 51% when the duration is longer than 3ms. In comparison, in EDT switches, buffer is fully utilized for almost all micro-burst traffic.

Enabling buffer to be fully utilized could make more micro-burst traffic absorbed. Fig. 11 illustrates the ratio of lossless micro-burst traffic for different micro-burst traffic durations. In DT switches, none of micro-burst traffic can be transmitted without packet dropping when its duration is longer than 3ms. Compared with it, in EDT switches, over 95% of micro-burst traffic can be absorbed when the duration is shorter than 5ms. However, for micro-bursts whose durations are shorter than 2ms, EDT performs a little worse than DT. This is because when micro-burst appears in multiple ports simultaneously, the

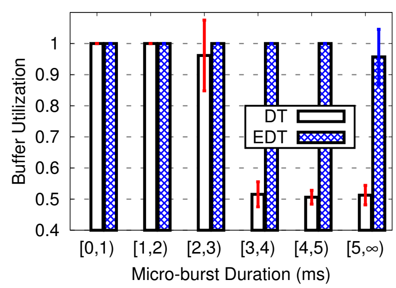


Fig. 10: Average buffer utilization when packets from micro-burst traffic are dropped for different micro-burst traffic durations

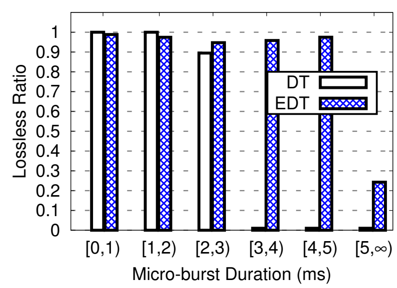


Fig. 11: The ratio of lossless micro-burst traffic for different micro-burst traffic durations.

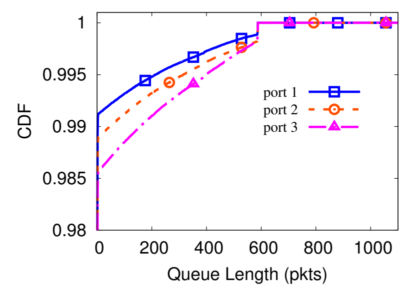


Fig. 12: Queue length CDFs

earlier appearing one will benefit more. However, this special case rarely happens (less than 2% in this scenario), because the duration of micro-burst traffic is very short.

Finally, we evaluate the fairness among switch ports. We select 3 ports and illustrate their queue length CDFs in Fig. 12, which implies that the queue lengths have similar distributions. Therefore, fairness among these ports is well promised. The queue length CDFs of other ports are similar.

V. Conclusion

Micro-burst is a common traffic pattern in data center networks. Packet dropping caused by micro-burst is usually unacceptable, and thus needs to be avoided. However, we find that packets from micro-burst traffic are dropped even though there is free buffer space in DT switches. We theoretically deduce the sufficient conditions for packet dropping caused by micro-burst traffic and estimate the corresponding free buffer size. The results show that the free buffer size is negatively correlated to the number of overloaded ports. And in order to ensure fair sharing of switch buffer among all ports, packets are dropped even when the micro-burst traffic size is far smaller than the buffer size. We propose EDT policy guided by the conclusions obtained from theoretical analysis. EDT can absorb micro-burst traffic as much as possible by fully utilizing the buffer and temporarily relaxing the fairness constraint.

VI. Acknowledgement

The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by National Basic Research Program of China (973 Program) under Grant No. 2012CB315803 and National Natural Science Foundation of China (NSFC) under Grant No. 61225011.

References

[1] "myths about "microbursts"," White Paper, Arista. [Online]. Available: http://www.arista.com/assets/data/pdf/7148sx-ixnetwork-microburst.pdf

[2] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM CCR*, vol. 40, no. 1, pp. 92–99, Jan. 2010.

[3] F. Uyeda, L. Foschini, F. Baker, S. Suri, and G. Varghese, "Efficiently measuring bandwidth at all time scales," in *Proceedings of USENIX NSDI*, 2011.

[4] V. Jeyakumar, M. Alizadeh, Y. Geng, C. Kim, and D. Mazières, "Millions of little minions: Using packets for low latency network programming and visibility," in *Proc. ACM SIGCOMM 2014*, 2014, pp. 3–14.

[5] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *Proc. ACM SIGCOMM*, 2010, pp. 63–74.

[6] D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: Incorporating time-varying network reservations in data centers," in *Proc. ACM SIGCOMM*, 2012, pp. 199–210.

[7] D. Meisner, C. Sadler, L. Barroso, W. Weber, and T. Wenisch, "Power management of online data-intensive services," in *Proc. ACM/IEEE ISCA*, 2011, pp. 319–330.

[8] B. Vamanan, J. Hasan, and T. Vijaykumar, "Deadline-aware datacenter tcp (d2tcp)," in *Proc. ACM SIGCOMM*, 2012, pp. 115–126.

[9] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, "Better never than late: Meeting deadlines in datacenter networks," in *Proc. ACM SIGCOMM*, 2011, pp. 50–61.

[10] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, "Detail: Reducing the flow completion time tail in datacenter networks," in *Proc. ACM SIGCOMM*, 2012, pp. 139–150.

[11] U. Cummings, P. P. Andrew Lines, and R. Southworth, "Shared-memory switch fabric architecture," U.S. Patent 7 814 280, Oct. 12, 2010.

[12] "Broadcom smart-buffer technology in data center switches for cost-effective performance scaling of cloud applications," White Paper, Broadcom, Apr. 2012. [Online]. Available: https://www.broadcom.com/collateral/etp/SBT-ETP100.pdf

[13] A. Choudhury and E. Hahne, "Dynamic queue length thresholds for shared-memory packet switches," *Networking, IEEE/ACM Transactions on*, vol. 6, no. 2, pp. 130–140, Apr 1998.

[14] "Congestion management and buffering in data center networks," White Paper, Extreme Networks, Dec. 2013. [Online]. Available: http://learn.extremenetworks.com/rs/extreme/images/Congestion-Management-and-Buffering-wp.pdf

[15] A. V. Bechtolsheim and D. R. Cheriton, "Per-flow dynamic buffer management," U.S. Patent 6 515 963, Feb. 4, 2003.

[16] D. R. Cheriton, "Approximated per-flow rate limiting," U.S. Patent 6 515 963, Apr. 20, 2004.

[17] S. Gai, T. Edsall, D. Bergamasco, D. Dutt, and F. Bonomi, "Network device architecture for consolidating input/output and reducing latency," U.S. Patent 7 830 793, Nov. 9, 2010.

[18] D. Bergamasco, A. Baldini, V. Alaria, F. Bonomi, and R. Pan, "Methods and devices for backward congestion notification," U.S. Patent 7 961 621, Jun. 14, 2011.

[19] C. DeSanti, S. Gai, and A. Baldini, "Fibre channel over ethernet," U.S. Patent 8 238 347, Aug. 7, 2012.

[20] V. Alaria, D. Bergamasco, M. Caramello, and C. Piglione, "Flexible and hierarchical dynamic buffer allocation," U.S. Patent 8 149 710, Apr. 3, 2012.

[21] "ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/