Differentiated Ethernet Congestion Management for Prioritized Traffic

Shuo Fang, Chuan Heng Foh
School of Computer Engineering
Nanyang Technological University
Singapore

Khin Mi Mi Aung
Data Storage Institute
A*STAR
Singapore

***Abstract*—This paper proposes and studies a differentiated congestion control for Ethernet congestion management. Following the popular approach that uses a cooperation of an *Additive Increase and Multiplicative Decrease* (AIMD) based rate limiter and *Explicit Congestion Notification* (ECN) active queue management to combat the congestion in Ethernet, the proposal considers differentiated AIMD settings for rate limiters to achieve congestion control differentiation for traffic of different priorities. We illustrate that while the operation of AIMD and ECN are independent, by using different AIMD settings, we can achieve differentiated control of bandwidth utilization. We provide an analysis and its numerical results showing the effectiveness of this method. Our proposed method is also implemented in OMNET++ simulator with results showing the effectiveness of bandwidth ratio differentiation.**

***Index Terms*—Ethernet Congestion Management, Storage Area Networks, Computer Network Performance.**

I. Introduction

Fibre Channel over Ethernet (FCoE), a newly proposed standard by INCITS T11, aims to use Ethernet technology to carry Fibre Channel traffic [1]. In FCoE, Fibre Channel Frames are encapsulated in Ethernet to be transmitted using Ethernet technology. This allows a single technology in the data link to operate in a data center, and makes a significant step towards I/O consolidation among Local Area Networks (LANs) and Storage Area Networks (SANs). This consolidation offers a number of benefits, for example it reduces power consumption for I/O operation and related cooling, enables fewer points of management to control, and eliminates redundancy in the network architecture by reducing the number of server slots and switch ports.

Fundamentally, however, FC and Ethernet are designed to handle two different characteristics of traffic and have different considerations for traffic transportation. Precisely, FC technology is designed to achieve high speed lossless packet transportation that well suits computing clusters and SANs. Whereas Ethernet is designed as to carry best effort traffic. In the current design, FC uses credit-based flow control that results in strict admission control to combat traffic congestion, while Ethernet provides connectionless service with minimum control on the traffic flow and no control on the congestion. FCoE that puts FC traffic which requires lossless transportation onto Ethernet which only offers connectionless service will results in serious performance degradation due to the inadequate handling of FC traffic by Ethernet.

One potential solution to enable FCoE is the strengthening of congestion management in Ethernet to ensure lossless transportation of SAN traffic. Additionally, with FCoE, a mix of SAN and LAN traffic will appear in Ethernet. Owing to the different characteristics and importance of both types of traffic, differentiated handling of packet transportation should be considered. This gives rise to the need for traffic prioritization and service differentiation in FCoE. These two challenges are currently studied by two IEEE standard groups, where are the IEEE 802.1Qau standard group addressing the congestion control with layer-2 End-to-End congestion management protocol and the IEEE 802.1Q dealing with the traffic prioritization and service differentiation by defining a tag field to differentiated priorities.

In this paper, we propose a mechanism that deals with Ethernet congestion control with differentiated handling of different types of traffic. Our design considers combination of the *Additive Increase and Multiplicative Decrease* (AIMD) and *Active Queue Management* (AQM) to achieve differentiated congestion control in Ethernet. Both AIMD and AQM are mature technologies that have been used in TCP for Internet congestion control. AQM takes preemptive actions prior to a potential buffer overflow in a router queue. In the case of FCoE, *Explicit Congestion Notification* (ECN) that uses marking of AQM to notify network congestion without dropping packets is appropriate for lossless packet transportation.

AIMD and its throughput performance have been a target for study in the literature. An important study on AIMD for TCP congestion control is due to Padhye *et al.* [3], where a formulation to derive the mean window size from the packet loss rate under a certain AIMD setting is provided. Based on this result, bandwidth sharing behavior among several TCP connections of a symmetric setting [4], [5], [8] as well as an asymmetric setting [6], [7] are given. It has also been shown that with certain AIMD settings on several connections, different throughput levels can be achieved [9]. Therefore, AIMD can not only serve as a congestion control mechanism but also satisfy the prioritized traffic for congestion control differentiation requirements in FCoE.

The combination of AIMD and AQM is currently considered as a strong candidate for Ethernet congestion control in FCoE. One recent proposal is due to Bergamasco by Cisco

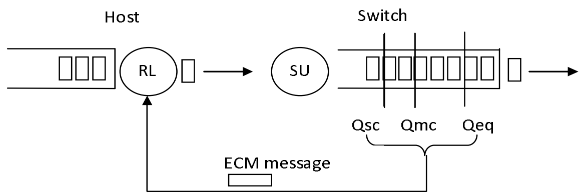

Fig. 1. System Model of ECM

where Ethernet Congestion Manager (ECM) is proposed [2]. This congestion control mechanism is based on AIMD and AQM operations to achieve traffic congestion control for lossless traffic. While current proposals reveal their congestion control mechanism, the lack of differentiation on prioritized traffic has made them inadequate for FCoE.

With a proper design, we introduce a congestion control mechanism that supports differentiated handling for prioritized traffic. Through analytical study, we investigate the potential of our proposed mechanism for the differentiated congestion control in Ethernet. The paper is organized as follows. Section II introduces Ethernet Congestion Management in Data Center operation. In Section III, we describe our proposed method on differentiated congestion control for Ethernet as a solution for converged priority and congestion management. We present an analysis for our proposed method with numerical result discussions given in Section IV. Based on the above results, Section V provides simulation results and verifies the effectiveness of the proposal. Finally, some conclusions are drawn in Section VI.

II. Ethernet Congestion Management

FC technology is the current popular solution used in SANs. FC technology offers lossless transportation between hosts and storages, and this lossless transportation helps SAN achieve high performance. Using FCoE immediately causes problem as Ethernet technology is currently incapable to deliver traffic with lossless demand. Designing a congestion control for a FCoE based SAN must ensure that Ethernet takes preemptive actions to prevent congestion and buffer overflow from occurring. This leads to the proposals of implementing a particular ECN at switches and rate limiters (RL) at hosts. This design allows excessive traffic to hold at the hosts, which pushes congestions from the core to the edges of networks. Precisely, when a frame is sampled from the sampling unit (SU), the system will check the buffer utilization. If it reaches a certain level, the switch as a congestion point (CP) notifies the source about the status, and the rate limiter of the source reduces its sending rate accordingly at the end host as a reaction point (RP), as shown in Fig. 1 [2]. In the following, we describes the current ECM proposal.

In ECM, a switch consists of a CP that samples incoming frames at a certain probability. Whenever a frame is sampled, the buffer utilization is also checked. This checking is to ensure that relevant ECN actions can be taken when traffic arrivals threaten buffer overflow. An adequate sampling rate should be sought to achieve fast enough reaction to congestion without excessive computational overheads.

The CP checks the buffer utilization against three thresholds, namely, Q_{eq}, Q_{mc} and Q_{sc}. When buffer utilization crosses any of these thresholds, corresponding ECM message is sent from the CP to the RP whose frame is being sampled. Based on the received ECM message, the RP adjusts its rate accordingly in order to ease the congestion and prevent a buffer overflow from occurring. Table I shows the relationship among thresholds, ECM messages and the reactions of RP.

Based on the three thresholds, a CP may stays in one of the three states. In each of these states, a corresponding ECM message is generated and sent to a RP whose frame is sampled. In the severe congestion situation where the buffer utilization is almost reaching its full capacity, the CP generates an ECM(0,0) message. In this case, the RP must halt its transmission by adjusting its rate to zero for a certain time period. In the mild congestion situation, where the buffer is moderately utilized, the CP generates an ECM-MAX message. It happens when the buffer utilization crosses Q_{mc} but stays below Q_{sc}. The CP will keep over-sampling from then on and the generated message causes the RP to reduce its rate with a maximum cut. Finally, in the equilibrium situation where the buffer utilization stays below Q_{mc} suggesting a low buffer overflow threat, the CP generates an ECM(Q_{off}, Q_{delta}) message. This message contains two parameters, where Q_{off} is the offset of the current buffer utilization with respect to the equilibrium threshold Q_{eq}, and Q_{delta} is the change in length of the queue since the last sampled frame. There are two cases in this situation. When the buffer utilization exceeds Q_{eq}, a positive value of Q_{off} is reported indicating the approaching of mild congestion. Otherwise, if a negative value of Q_{off} is reported indicating the ease of buffer utilization. Corresponding rate adjustments are performed at the RP based on ECM(Q_{off}, Q_{delta}).

TABLE I
Functionality of the Thresholds and Corresponding Messages

Threshold	Description	Message	Reaction
Q_{eq}	Equilibrium Threshold	ECM(Q_{off},Q_{delta})	The rate limiter adjusts its rate according to the two components of the ECM feedback.
Q_{mc}	Mild Congestion Threshold	ECM-Max	This message causes the rate decrement of a rate limiter.
Q_{sc}	Severe Congestion Threshold	ECM(0,0)	The rate limiter sets the rate to zero temporarily.

In ECM, each RL implements a variation of AIMD for its rate adjustment in the rate limiter. The role of AIMD in an RL is to regulate traffic flow from hosts to the network according to the congestion status of the network so that no excessive traffic can enter the network causing network congestion. The design of AIMD directly affect the utilization of networks. A rate control design being too conservative may cause low utilization of networks. On the other hand, a rate control design being too aggressive may cause over utilizing of network which results in network congestion and buffer overflow.

ECM employs an AIMD in a much smoother way. The rate limiter periodically increases its sending rate. If a feedback is detected, the feedback signal Fb is calculated using Q_{eq} and Q_{delta} by the formula (1).

$$Fb = -\left(Q_{off} + w \cdot Q_{delta}\right) \tag{1}$$

Based on the value of Fb, the rate adjust its rate according to the following rules, where

$$R \leftarrow \begin{cases} R + \min(Gi \cdot Fb \cdot Ru, \beta \cdot C), & Fb > 0 \\ R \cdot (1 - \min(Gi \cdot |Fb|, \alpha)), & Fb < 0 \end{cases}$$

and where both quantities of Gi and Gd are the increase gain and decrease gain respectively, Ru is the rate unit in the rate limiter, which is the granularity of the rate adjustment, and C is the capacity of the link draining the rate limiter. More details on the operation of the rate limiter and its design principle can be found in [2].

III. Differentiated Congestion Control

Given the wide application of AIMD rate controller and AQM in the Internet congestion control, it is expected that Ethernet congestion control for FCoE will follow the same design principle. However, due to the consolidation of FC and Ethernet, FCoE will face coexistence of the traditional LAN traffic and SAN traffic. Preparing for the handling of a mix of traffic with different characteristics, we argue the need for differentiated congestion control handling. In this paper, we propose using different AIMD parameter sets for the rate limiters to achieve congestion control differentiation, and show its performance feasibility for this design. In ECN, we simplify the design with two thresholds, namely T_{eq}, T_{sc}, to ensure lossless traffic transportation. The use of two thresholds instead of three can be viewed as that the setting of Q_{mc} coincides with that of Q_{sc}. Similar to Q_{eq}, T_{eq} is the equilibrium congestion point. After reaching this threshold, notification will be sent back to the source whose frame is sampled. After receiving this notification, the source decreases to a proportion of its current rate according to preset AIMD parameters. When the buffer utilization runs greater than T_{sc}, severe congestion happens and the source receiving this notification halts its transmission and resume the slow start operation as in ECM.

Our proposed mechanism is illustrated in Fig. 2. Although in general, our proposed mechanism can easily satisfy an arbitrary number of traffic priorities for different congestion control handling, in view of our application, we assume that in the practical operation, FCoE may primarily focus on two types of traffic, which are the traditionally LAN traffic and SAN traffic. Given two types of prioritized traffic, we need a design of two sets of AIMD parameters. In the following, we shall describe the AIMD operation related to our design.

Each source operates an AIMD-based rate limiter. The rate limiter maintains a variable called *congestion window*. The value of congestion window is initialized to be one. This value

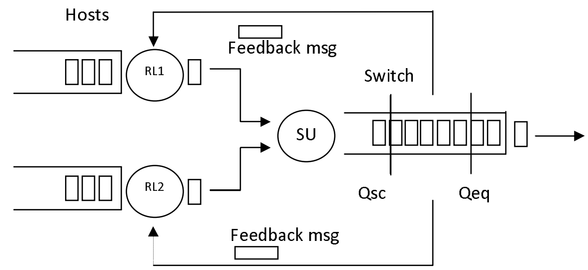

Fig. 2. System Model of Differentiated Congestion Control

increases linearly over a predefined constant time interval called a *slot*. Congestion window specifies the number of frames a source can transmit at the beginning of each slot. As the value of congestion window increases, the number of frames a source can emit into the network increases accordingly. This increase in frames into the network may cause network congestion. Each queue in the Ethernet switch executes the above mentioned ECN to monitor the buffer utilization and notify the source to perform rate cut if the buffer utilization exceeds T_{eq}.

When a notification of rate cut is received, the source immediately reduces the value of its congestion window by a certain percentage. This action directly reduces the number of frames the source can transmit to the network and eases the network congestion. After that, the congestion window continues to increase for every slot time until the next notification of rate cut appears. Let $x_i(t)$ be the load source i transmits to the network at time t, source i will adjust its load at the next slot by

$$x_i\left(t+1\right) = \begin{cases} a_i + x_i\left(t\right), & X\left(t\right) \le T_{eq} \\ b_i x_i\left(t\right), & \text{otherwise} \end{cases} \tag{2}$$

where $X(t) = \sum_i x_i(t)$ is the sum of the number of frames transmitted by all sources. Note that AIMD can be described by two parameters, namely a and b where $a > 0, 0 \le b < 1$. This gives rise to several strategies of congestion control. In our design, focusing on congestion control differentiation of SAN and LAN traffic, we restrict our study to two sets of AIMD parameters for SAN and LAN traffic sources respectively.

IV. Analysis and Numerical Results

We expect that the throughput ratio varies among different sets of parameters. It is important to know the influence on ratios exerted by different parameters. An analysis is studied considering flows with two types of traffic priorities utilizing a common switch. We set T_{sc} as an infinite point in this analysis, that means we focus on only one round of performance from each restart to each severe congestion. This is reasonable since the system will restart whenever T_{sc} is reached and the process of operation is similar in each round. Similarly as in [9], the evolutions of congestion windows can be considered as semi-Markov process, we define each *epoch* as a period beginning with a congestion window of bW [7]. With m

simultaneous flows, the evolutions are modeled into an m-dimensional process, which is $\{W_1(n), W_2(n), ..., W_m(n)\}$ where $0 \leq W_i \leq T_{eq}$ and $i = 0, 1, ..., m$. We use the same approach as in [9] for our analysis, adjust the model accordingly, and report important results in the following.

As in [9], $k(n)$ is defined as number of slots between n-th and $(n+1)$-th epochs, according to (2), congestion window of a particular flow, say flow i, at the next epoch, $W_i(n+1)$, can be either expressed as

$$W_i(n+1) = (k(n)-1)a_i + b_i(a_i + W_i(n)) \tag{3}$$

or as

$$W_i(n+1) = k(n)a_i + W_i(n) \tag{4}$$

and $k(n)$ can be easily solved by the fact that the sum of all the congestion windows at the next epoch must reach equilibrium threshold T_{eq} indicating a full utilization of the buffer, that is [9]

$$\begin{cases} \text{minimize } k(n) \\ \text{subject to } \sum_i W_i(n+1) \geq T_{eq} \end{cases} \tag{5}$$

where $k(n)$ is a nonzero positive integer.

To study the interaction of two flows with different priorities, we adopt the above-mentioned discrete congestion window and semi-Markov Chain to describe the evolution of the congestion windows.

Following the similar method in [9], l is defined as the event of frame marking occurs when the current queue length exceeds threshold T_{eq}. According to our model, there are two possibilities, which are $l = 1$ denoting only when the switch samples a frame from flow 1, $l = 2$ denoting only the frames from flow 2 is sampled. The probability of a frame of a certain flow that is being sampled is proportional to the total arrivals. Given (3)-(4), the nonnull one-step transition probabilities[1] of the bi-dimensional semi-Markov Chain can be described by

$$\begin{aligned} &\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = q_1, \\ &\quad \hat{\omega}_1 = \lfloor (k_1(\omega_1, \omega_2) - 1) a_1 + b_1 (a_1 + \omega_1) \rfloor, \\ &\quad \hat{\omega}_2 = \lfloor k_1(\omega_1, \omega_2) a_2 + \omega_2 \rfloor, \end{aligned}$$

$$\begin{aligned} &\Pr\{\hat{\omega}_1, \hat{\omega}_2 | \omega_1, \omega_2\} = q_2, \\ &\quad \hat{\omega}_1 = \lfloor k_2(\omega_1, \omega_2) a_1 + \omega_1 \rfloor, \\ &\quad \hat{\omega}_2 = \lfloor (k_2(\omega_1, \omega_2) - 1) a_2 + b_2 (a_2 + \omega_2) \rfloor, \end{aligned}$$

where

$$q_1 = \frac{\omega_1}{\omega_1 + \omega_2}$$

$$q_2 = \frac{\omega_2}{\omega_1 + \omega_2}$$

and by (5), $k_l(\cdot)$ can be conservatively estimated as [9]

$$\begin{aligned} k_1(\omega_1, \omega_2) &= \left\lfloor \tfrac{T_{eq} - (b_1(a_1+\omega_1) - a_1) - \omega_2}{a_1 + a_2} \right\rfloor \\ k_2(\omega_1, \omega_2) &= \left\lfloor \tfrac{T_{eq} - \omega_1 - (b_2(a_2+\omega_2) - a_2)}{a_1 + a_2} \right\rfloor. \end{aligned}$$

[1]We adopt the short notations: $q_l = q_l(\omega_1, \omega_2)$, $\Pr\{\hat{\omega}_1, \hat{\omega}_2|\omega_1, \omega_2\} = \Pr\{W_1(n+1) = \hat{\omega}_1, W_2(n+1) = \hat{\omega}_2 | W_1(n) = \omega_1, W_2(n) = \omega_2\}$, and $k_l(\omega_1, \omega_2) = k_l(n|W_1(n) = \omega_1, W_2(n) = \omega_2)$.

Let $\pi_{u,v}$ be the stationary state probability distribution of the Markov Chain, to be precise, $\pi_{u,v} = \lim_{n\to\infty} \Pr\{W_1(n) = u, W_2(n) = v\}$. $\pi_{u,v}$ can be solved numerically by the following balance equation set [9]

$$\pi_{u,v} = \sum_{\omega_1=0}^{T_{eq}} \sum_{\omega_2=0}^{T_{eq}} \sum_{l=1}^{2} \left(\pi_{\omega_1,\omega_2} q_l \delta_{u, f_l(\omega_1,\omega_2)} \delta_{v, g_l(\omega_1,\omega_2)} \right)$$

where δ denotes Kronecker delta, $u, v \in \{0, 1, \ldots, T_{eq}\}$, with [9]

$$\begin{aligned} &\pi_{0,0} = 0, \\ &f_1(\omega_1, \omega_2) = \lfloor (k_1(\omega_1, \omega_2) - 1) a_1 + b_1 (a_1 + \omega_1) \rfloor, \\ &g_1(\omega_1, \omega_2) = \lfloor k_1(\omega_1, \omega_2) a_2 + \omega_2 \rfloor, \\ &f_2(\omega_1, \omega_2) = \lfloor k_2(\omega_1, \omega_2) a_1 + \omega_1 \rfloor, \\ &g_2(\omega_1, \omega_2) = \lfloor (k_2(\omega_1, \omega_2) - 1) a_2 + b_2 (a_2 + \omega_2) \rfloor, \end{aligned}$$

and

$$\sum_{u=0}^{T_{eq}} \sum_{v=0}^{T_{eq}} \pi_{u,v} = 1.$$

Define $W_{i|l,\omega_1,\omega_2}$ to be the congestion window conditioned upon event l at state $\{\omega_1, \omega_2\}$, based on the above discussion, we obtain [9]

$$W_{i|l,\omega_1,\omega_2} = \begin{cases} (a_i + \omega_i) + \frac{a_i(k_l(\omega_1,\omega_2)-1)}{2}, \\ \qquad \text{if } (i=1, l=2) \text{ or } (i=2, l=1) \\ b_i(a_i + \omega_i) + \frac{a_i(k_l(\omega_1,\omega_2)-1)}{2}, \\ \qquad \text{otherwise} \end{cases}.$$

Using the above results, the mean congestion window for flow i, denoted $\overline{W}_i$, can be computed by [9]

$$\overline{W}_i = \sum_{\omega_1=0}^{T_{eq}} \sum_{\omega_2=0}^{T_{eq}} \sum_{l=1}^{2} \left(\pi_{\omega_1,\omega_2} q_l W_{i|l,\omega_1,\omega_2} \right). \tag{6}$$

Using the above developed formulas, we compute the numerical results focusing on differentiation capability of AIMD settings for congestion control. In particular, we wish to investigate the throughput utilization between two flows of different priorities given a particular pair of AIMD settings for the two flows. These AIMD settings can be useful in the network design for maintaining timely transmission of SAN traffic in the presence of LAN traffic.

We use the notation of AIMD(a, b) for the rate limiter implementing AIMD with parameters a and b. In our numerical test, the equilibrium threshold is fixed at 20. We fix the LAN source to AIMD(1,0.5), and vary the AIMD parameters of SAN source. Since SAN traffic has a higher priority, we use more aggressive settings than AIMD(1,0.5) for the SAN rate limiter. That is, we use AIMD(a, b) where $a > 1, b > 0.5$. In Fig. 3, we show the throughput ratio of SAN traffic over LAN traffic. As can be seen, varying the parameters a and b can provides differentiation in bandwidth utilization. We see that as the values of a and b increase for SAN traffic, the throughput ratio increases. This shows that when a switch experiences congestion, it penalizes LAN traffic with AIMD(1,0.5) more than that of SAN traffic with AIMD(a, b) where $a > 1, b > 0.5$. This allows SAN traffic to enjoy higher throughput.

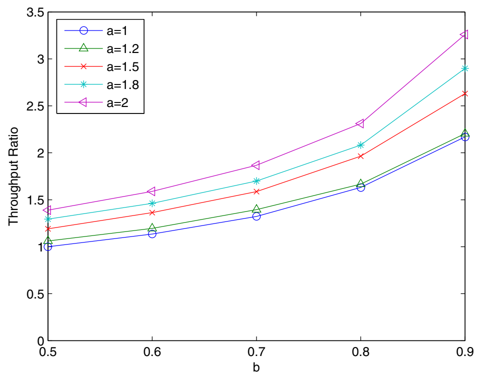

Fig. 3. Throughput ratio of SAN traffic with AIMD(a,b) over LAN traffic with AIMD(1,0.5)

V. Simulation Results

We follow the setup given in [10] and test the performance of our proposed mechanism. Figure 4 shows the screenshot of the network topology in Omnet++ 4.0 simulator. Link capacity is set to 1Gbps. In the scenario, SR_1 and SR_2 send frames to DR_1 and DR_2 respectively at the speed of 40Mbps and the average frame length is 250 bytes uniformly distributed between 200 and 300 bytes. Rate limiters are equipped within the two end hosts. ST_1 to ST_4 are CBR sources sending traffic destined to DT through end switches ES_1 to ES_4 to produce some background traffic. Their sending rate is set to 800kbps each and there will not be any rate limiters since they are simulating traffic outside data center.

We create a scenario such that congestions occurs in the core switch CS congestion, when heavy traffic arrives. CS has T_{eq} of 50,000 and T_{sc} of 100,000 bytes separately. SR_1 and SR_2 are two sources with different priorities, namely SAN and LAN traffic. The throughput ratio of them is set to 2 in the network design. To achieve this ratio requirement, parameters of $a_1 = 1.5, b_1 = 0.8$ and $a_2 = 1, b_2 = 0.5$ are chosen according to our numerical results presented in Fig. 3. Specifically, LAN uses the typical AIMD(1,0.5) while SAN has a higher priority with AIMD(1.5,0.8) in simulation. In fact, the numerical result for this setting of parameters is 1.96.

Differentiated congestion control mechanism is operating in the core switch CS, which pushes traffic back to the end hosts SR_1 and SR_2. In this case, the total amount of bytes sent by the two sources increase with varied ratios, shown in Fig. 5. The simulator records a ratio between the two source of 2.04, which meets our target setting.

VI. Conclusions

In this paper, we proposed a differentiated congestion control for Ethernet congestion management. Our proposed method considers using different AIMD settings for the rate limiter operations. We analyzed its effectiveness by studying the potential of throughput ratio. This is achieved by penalizing more on the LAN traffic than the SAN traffic when a congestion event about to occur. We performed omnet++ simulations and it reports consistent results with that of the numerical analysis.

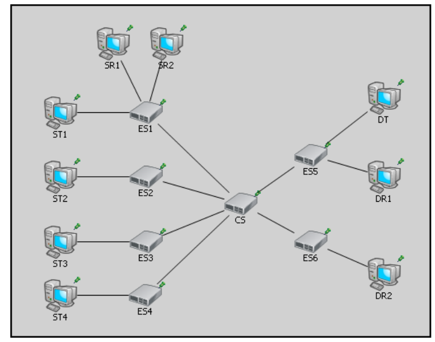

Fig. 4. Network topology for Omnet++ simulation study

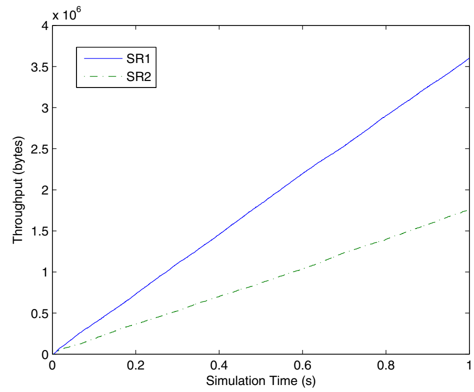

Fig. 5. Total bytes sent by the sources

References

[1] Fibre Channel Industry Association, "Fibre Channel over Ethernet: An Introduction," 2007.

[2] Davide Bergamasco, "Data Center Ethernet Congestion Manager: Backward Congestion Notification," *IEEE 802.1 Meeting*, May 2005.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. ACM SIGCOMM*, 1998.

[4] F. Baccelli and D. Hong, "The AIMD Model for TCP Sessions Sharing a Common Router," in *39th Annual Allerton Conf. on Communication, Control and Computing*, October 2001.

[5] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou, "Selfish Behavior and stability of the Internet: A Game Theoretic Analysis of TCP," in *Proc. ACM SIGCOMM*, August 2002.

[6] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," in *Proc. ICNP*, 2000.

[7] S. Floyd, M. Handley, and J. Padhye (May 2000), "A Comparison of Equation-Based and AIMD Congestion Control," *ACIRI Technical Report*.

[8] L. Massoulie and J. Roberts, "Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet," *IEEE Transactions on Networking*, vol. 10, no. 3, June 2002, pp. 320-328.

[9] Cheng Peng Fu, Chuan Heng Foh, Chiew Tong Lau, Zhihong Man, and Bu Sung Lee, "Semi-Markov Modeling for Bandwidth Sharing of TCP Connections with Asymmetric AIMD Congestion Control," in *IEEE GLOBALCOMM*, 2007.

[10] Jinjing Jiang and Raj Jain, "Analysis of Backward Cognestion Notification (BCN) for Ethernet in Datacenter Applications," in *IEEE INFOCOM*, 2007