Differentiated Congestion Management of Data Traffic for Data Center Ethernet

Shuo Fang, *Student Member, IEEE,* Chuan Heng Foh, *Senior Member, IEEE,*
and Khin Mi Mi Aung, *Senior Member, IEEE*

***Abstract*—This paper aims at designing a congestion and priority solution 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 congestions 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 operations of AIMD and ECN are independent, by using different AIMD settings, we can achieve differentiated control of bandwidth utilization. We develop a control theoretic analytical model to study the effectiveness of our proposed method. Moreover, we implement our proposed method in OMNET++ simulator to conduct simulation experiments. Our analytical and simulation results both indicate 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 to carry best effort traffic. In the current design, FC uses credit-based flow control that results in strict admission control to prevent traffic congestion from happening, while Ethernet provides connectionless service with minimum control on the traffic flow and no control on the traffic congestion. FCoE that puts FC traffic which requires lossless transportation onto Ethernet which only offers connectionless service will result 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 the introduction of 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 several IEEE standard groups, which 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 which maps storage traffic to Priority three and IP traffic to Priority one. Besides, IEEE 802.1Qaz group works on a hardware efficient mechanism that separates LAN and SAN to support strict priority scheme [2].

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 *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]–[6] as well as an asymmetric setting [7], [8] are given. It has also been shown in our earlier work that with certain AIMD settings on several connections, different throughput levels can be achieved [9].

Manuscript received September 29, 2010; revised March 22, 2011 and July 5, 2011. The associate editor coordinating the review of this paper and approving it for publication was K. Van der Merwe.

S. Fang and C. H. Foh are with the School of Computer Engineering, Nanyang Technological University, 639798 Singapore (e-mail: fang0037@e.ntu.edu.sg).

K. M. M. Aung is with the Data Storage Institute, A*STAR, 117608 Singapore.

Digital Object Identifier 10.1109/TNSM.2011.110911.100076

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 has been proven to be a viable way for congestion management [10]–[12] and it is currently considered as a strong candidate for Ethernet congestion control in FCoE. One recent proposal is due to Bergamasco by Cisco where Ethernet Congestion Manager (ECM) is proposed [13]. This congestion control mechanism is based on AIMD and AQM operations to achieve traffic congestion control for lossless traffic. However, the lack of differentiation on prioritized traffic has made their congestion control mechanism inadequate for FCoE to deal with a mix of SAN and LAN traffic.

In this paper, we introduce a congestion control mechanism that supports differentiated handling for prioritized traffic. Through analytical and simulation studies, we demonstrate the potential of our proposed mechanism for the differentiated congestion control in Ethernet. The paper is organized as follows. In the next section, we provide related work on congestion control in data center networks. Section III provides an overview of Ethernet Congestion Management in data center operation. In Section IV, 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 result discussions in Section V. Based on the results, Section VI contains OMNET++ simulations for a real scenario and experiment tests to verify the effectiveness of our proposal. Finally, important conclusions and future work are given in Section VII.

II. Related Work

Traditionally, SANs based on FC technology use buffer-to-buffer credit-based flow control to avoid traffic congestion in the network. On a data link, the amount of traffic that a sender can transmit depends on the amount of credits allocated by the receiver. The receiver can prevent excessive incoming traffic by a proper credit allocation scheme, and hence traffic congestion can be avoided.

The introduction of FCoE has triggered demands for Ethernet to deal with traffic congestion. An immediate solution to enable equivalent credit-based flow control on the Ethernet is to employ PAUSE mechanism [2]. It is suggested that a proper implementation of PAUSE mechanism may enable lossless transportation within an Ethernet network. In PAUSE mechanism, a downstream port may issue a PAUSE frame to halt data transmission from the upstream port for a specified period of time to avoid traffic congestion. However, stopping transmission of an upstream port might lead to further congestion points in its predecessors, and continually, traffic congestions can spread to all upstream ports even for those that do not transmit excessively. In other words, rather than dealing with the source of the problem, this mechanism penalizes all sources when a traffic congestion event occurs. Additionally, due to the buffer-to-buffer control, the control of a congestion and the recovery from a congestion are done hop-by-hop which takes some time to converge. Moreover, the existing PAUSE mechanism does not offer differentiated control for different traffic types.

Priority-based Flow Control (PFC) [14] has extended PAUSE mechanism on per-priority basis, which enables a finer-grained flow control with coexistence of different types of traffic. The IEEE 802.1Q standard defines a tag containing three-bit priority field. According to the priority field, PFC maps traffic classes to different priorities and prevents traffic interference. However, apart from offering differentiated congestion control, PFC inherits all other shortcomings of PAUSE mechanism.

To avoid congestion from spreading across the entire network, Cisco has proposed Ethernet Congestion Management (ECM), previously called BCN and now known as QCN with significant improvements over BCN, as a Layer-2 end-to-end congestion notification protocol. This mechanism aims to hold excessive traffic at the edge of a network. With less number of traffic flows, the flows that caused congestion can be easily constrained and regulated. In ECM, a congested switch (congestion point) samples and sends feedbacks with its state towards the source of its congestion (reaction point). Upon receiving the feedback message, the reaction point controls its traffic volume from entering the network. By using an end-to-end congestion control, ECM directly deals with the source with excessive load rather than all sources. However, the key feature of differentiated control remains missing from ECM.

Recently, a cross-layer congestion control, named data center TCP or DCTCP in short, operating at Layer-4 for data center networks has been proposed [15]. DCTCP proposes marking of ECN based on Layer-2 buffer state and a new rate adjustment scheme replacing AIMD. While DCTCP operates at Layer-4, with the introduction of ECN based on Layer-2 buffer state, it is capable of reacting to a potential congestion before a congestion event occurs. However, no differentiated congestion control is introduced in DCTCP. Moreover, the relatively long host-to-host turnaround time at Layer-4 has created a lag in the control.

In view of the need for a differentiated congestion control for data center Ethernet, and given the potential performance advantages of Layer-2 end-to-end congestion control, we propose using Layer-2 end-to-end congestion control with different characteristics of rate limiters to achieve congestion control differentiation. We introduce the use of different AIMD parameters to characterize rate limiters for congestion control differentiation at Layer-2 and present a comprehensive study confirming the effectiveness of our solution. Specifically, we employ control theoretic approach to model our solution and provide formulations for the design of differentiated congestion control. We also perform extensive simulation experiments to show the achievement of differentiated congestion control and performance behavior of our solution.

III. 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. Enhancing existing FC technology with FCoE immediately causes problem as Ethernet technology is

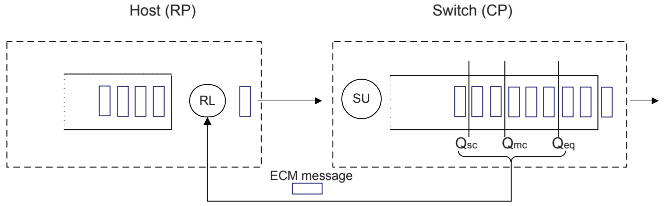


Fig. 1. System model of ECM.

currently incapable to deliver traffic with lossless demand. Designing a congestion control for an 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 at hosts. This design allows excessive traffic to be held at the hosts, which pushes congestions from the core to the edges of the network. Precisely, when the current buffer utilization reaches a certain level, the switch acting as the congestion point (CP) notifies the host acting as the reaction point (RP) about the status, and the rate limiter in the host reduces its sending rate accordingly as shown in Fig. 1 [13]. In the following, we describe 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 designed to ensure that relevant ECN actions can be taken when traffic arrivals threaten buffer overflow. The probability P of sampling an incoming frame is critical to ensure fast reaction to congestion without excessive computational overheads. In [13], the measure of sampling probability is described by sampling rate where a sampling is executed on every $E[L]/P$ received bytes, here average frame length is defined as $E[L]$.

The CP checks the buffer utilization against three thresholds, namely, Q_{eq}, Q_{mc} and Q_{sc}, where $Q_{eq} < Q_{mc} < Q_{sc}$. Based on the buffer utilization, corresponding ECM message is sent from the CP to the RP whose arriving frame is last sampled at the CP. Based on the received ECM message, the rate limiter adjusts the transmission rate of a source accordingly to let the buffer operate around the desired equilibrium level. Table I shows the relationship among thresholds, ECM messages and the reactions of a rate limiter.

Based on the three thresholds described in Table I, a CP is said to operate in three states, namely equilibrium, mild congestion and severe congestion. When the buffer of a CP is utilized below Q_{mc}, the CP is said to operate in the equilibrium state. In this state, an ECM(Q_{off}, Q_{delta}) message is used for notification. This message contains two parameters, where Q_{off} is the offset of the current buffer utilization with respect to 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, a negative value of Q_{off} is reported indicating the easing of buffer utilization after a rate cut. Corresponding rate adjustments are performed at the RP based on ECM(Q_{off}, Q_{delta}).

However, when the buffer utilization level of a CP crosses Q_{mc} but remains below Q_{sc}, the CP enters the mild congestion state. In this state, an ECM-Max message is used for notification. This message generally instructs the RP to reduce its transmission rate with a maximum cut. Finally, when the buffer utilization level of a CP crosses Q_{sc}, the CP reaches the severe congestion state. In this state, an ECM(0,0) message is used for the notification. Upon reception of this message, the RP must halt its transmission by adjusting its rate to zero for a certain predefined time period before resuming from the lowest transmission rate again.

In ECM, each rate limiter implements a variation of AIMD for its rate adjustment. The role of AIMD in a rate limiter 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 affects network utilization. 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 network over-utilization which results in network congestion and buffer overflow.

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

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

Based on the value of Fb, the rate limiter adjusts its rate, R, according to the rules

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

where 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, the quantities α and β are parameters to limit the upward and downward rate adjustments. More details on the operation of the rate limiter and its design principle can be found in [13].

IV. Differentiated Congestion Control

Given the wide application of AIMD rate controller and AQM in the Internet congestion control, this approach represents a potential candidate for congestion control in FCoE design. However, due to the consolidation of FC and Ethernet, FCoE will face coexistence of the traditional LAN traffic and SAN traffic [16]. 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. Our proposed mechanism is illustrated in Fig. 2. This model utilizes two components to regulate traffic, T_{eq} and T_{sc}. Queue management in the CP helps to detect a potential congestion, while rate limiters in the reaction point side contribute to the regulation of traffic flow to prevent congestions.

In general, our proposed mechanism can easily satisfy an arbitrary number of traffic priorities for different congestion control handling. Focusing on data center application, in this paper, we shall focus on congestion differentiation between LAN and SAN traffic.

TABLE I
FUNCTIONALITY OF THE THRESHOLDS AND CORRESPONDING MESSAGES

Label	Name	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 maximum rate decrement of a rate limiter.
Q_{sc}	Severe Congestion Threshold	ECM(0, 0)	The rate limiter sets the rate to zero temporarily.

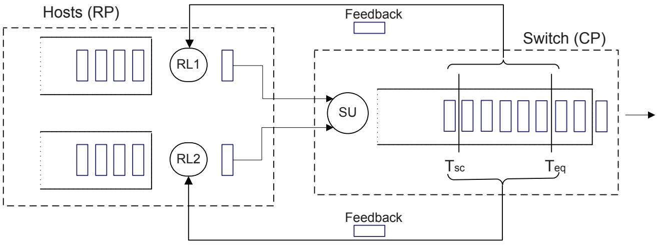


Fig. 2. System model of Differentiated Congestion Control.

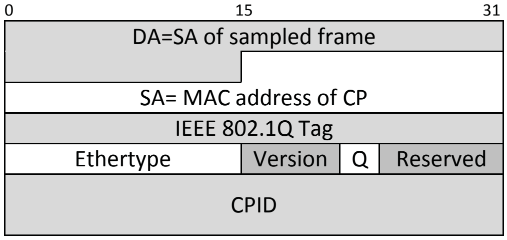


Fig. 3. Key fields of feedback message.

A. Queue Management in Congestion Point

A congestion point (CP) features a queue management which is mainly responsible for congestion detection, congestion notification and packet drop policy. For congestion detection, similar to ECM, we propose a sampling function to control the frequency of buffer inspection. In the sampling function, incoming frames are being sampled on a byte arrival basis. Whenever a frame is sampled, the CP also inspects the buffer status to detect congestion events. The CP samples a frame every fixed number of bytes received, and this fixed number of bytes is calculated by length specified $E[L]/p_s$ where $E[L]$ is the average frame length and p_s is a preset sampling probability.

The congestion notification is performed by a feedback function. When a frame is sampled and the buffer is inspected, the CP may notify the source whose frame is sampled to adjust its rate based on the buffer status. In our design, there are two thresholds, namely T_{eq} and T_{sc}, representing an equilibrium congestion level and a severe congestion level, respectively. During a buffer inspection, if the buffer utilization exceeds T_{eq}, a notification will be sent to the source whose frame is sampled. This notification instructs the rate limiter of the source to adjust its rate downward according to the AIMD parameters. If the buffer utilization exceeds T_{sc} indicating the appearance of severe congestion, a notification is sent instructing a source to halt a transmission for a predefined period of time, and the source shall resume its transmission at the lowest predefined rate.

Figure 3 shows the key fields of feedback frame. DA is the source address of the sampled frame, and SA is the address of the port in the switch that samples the frames. The message type can be identified as a notification, whether is to slow down or to stop the transmission, by the EtherType field. CPID is ID of the CP. The rest fields follow IEEE802.1Q standard, such as IEEE802.1Q Tag, Version, Q.

The CP also implements a packet drop policy to achieve differentiation in traffic congestion management when congestion persists. In our design, when the buffer utilization exceeds T_{sc}, all LAN traffic, being the low priority traffic, will be dropped until the buffer level returns below the equilibrium threshold, T_{eq}. This packet drop policy further protects the SAN traffic during severe congestion.

B. Rate Limiters in the Reaction Point

A reaction point (RP) on the host side implements rate limiters to regulate traffic flows. A rate limiter regulates the traffic flow by controlling the transmission rate using an AIMD based rate adjustment operation, and a CP provides necessary feedbacks for the AIMD operation to perform rate adjustment. We propose using different sets of AIMD parameters to achieve differentiated congestion handling of different types of traffic. In the following, we shall describe the AIMD operation related to our design.

Each source consists of an AIMD-based rate limiter. Similar to the mechanism employed in TCP, the rate limiter maintains a variable called *congestion window* to regulate the transmission rate. The value of congestion window is initialized to be one. This value 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 transmit into the network increases accordingly. This increase in frames into the network may cause network congestion. At the CP, each queue in the Ethernet switch executes the above mentioned ECN to monitor the buffer utilization and notify the source to regulate its transmission rate by adjusting the congestion window.

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

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

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, which are a and b where $a > 0, 0 \leq b < 1$. This gives rise to several strategies of congestion control. In the real implementation, a parameter called *step of a* is also used, as the unit of transmission rate, denoted as u. The real increase rate comes from the window size multiplying the step, namely $a \cdot u$.

V. Analysis

The selection of parameters for rate limiter design is critical to ensure that the targeted differentiated service is achieved while maintaining a high performance of operation. In our previous work in [17], we demonstrated AIMD operation for service differentiation specified by bandwidth differentiation ratio between two classes of traffic types. We introduced a semi-Markov process to model the AIMD operation where the model computes bandwidth differentiation ratios with various AIMD settings. The use of semi-Markov process, however, is limited to the study of a small number of traffic types due to scalability of the model.

In this paper, by taking a different approach, we introduce a control theoretic analytical model for the study of our proposed differentiated congestion control mechanism. This model enables the study of bandwidth differentiation ratios for an arbitrary number of traffic types and the description of marking probabilities for system stability analysis.

A. The Control Theoretic Model [18]

In [18], Hollot *et al.* propose a control theoretic analytical model for the performance study of TCP congestion control operation. Focusing on the equilibrium state of the system, a set of differential equations is constructed to describe the AIMD operation of TCP. Following the same approach, we model our proposed differentiated congestion control solution as a control system described in Fig. 4. As our system includes multiple rate limiters with different AIMD parameter settings connecting to a common switch, we extend the model presented in [18] from a single AIMD type to multiple AIMD types resulting several AIMD flow control blocks connecting to a single bottleneck as shown in Fig. 4. With this extension, the formulation in (1) given in [18] is modified accordingly from a single equation to a set of simultaneous equations as

$$\dot{W}_i\left(t\right) = \frac{a_i}{R(t)} - \frac{(1-b_i)\cdot W(t)W(t-R(t))}{R(t-R(t))} \cdot p_i \cdot \left(t - R\left(t\right)\right) \tag{3}$$

and

$$\dot{q}\left(t\right) = \sum_{i=1}^{n} \frac{W(t)}{R(t)} N_i\left(t\right) - C \tag{4}$$

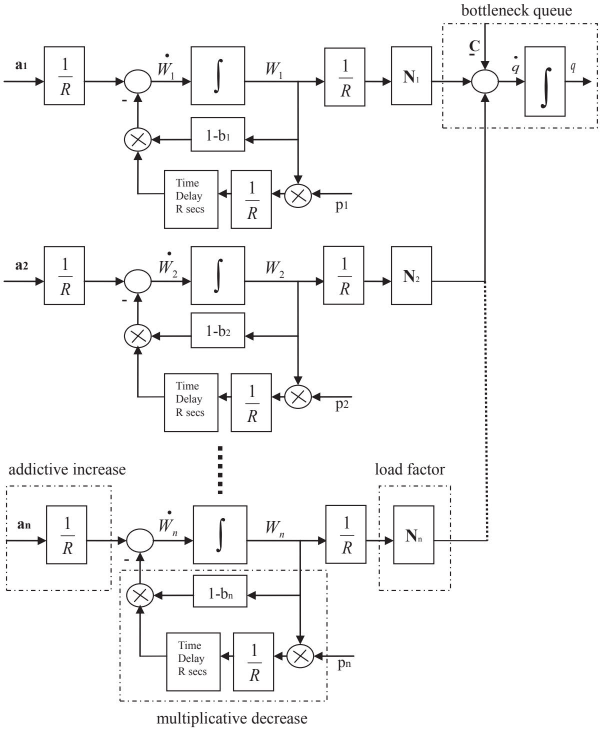


Fig. 4. Block-diagram of Differentiated Congestion Control flow-control model.

where a_i, b_i describe the AIMD parameters as defined in Section III, $\dot{x}$ denotes the time-derivative of x, and

$W_i \doteq$ expected window size of link i (frames);
$q \doteq$ expected queue length (frames);
$R \doteq$ round-trip time (secs);
$C \doteq$ network capacity (frames/sec);
$N_i \doteq$ load factor of link type i (links);
$p_i \doteq$ marking probability of link type i.

Similarly to TCP that employs AIMD congestion control, our differentiated congestion control solution also uses AIMD congestion control achieving stability in operation. As a result, the developed control theoretic model gives a steady-state solution, and the solution can be obtained by setting $\dot{W}_i = 0$ [18], [19]. In the steady-state operation, applying $\dot{W}_i = 0$ for all i, we have

$$\hat{W}_i^2 = \frac{a_i}{(1-b_i)p_i} \tag{5}$$

where $\hat{W}_i$ denotes the steady-state solution of $W_i(t)$.

B. Application of Control Theoretic Model for Multiple Traffic Flows and Types

In our design, the quantities of marking probabilities p_i have the same value. Consequently with j types of AIMD settings each with N_j flows, we obtain

$$\hat{W}_1 : \hat{W}_2 : \cdots : \hat{W}_j = \sqrt{\tfrac{a_1}{1-b_1}} : \sqrt{\tfrac{a_2}{1-b_2}} : \cdots : \sqrt{\tfrac{a_j}{1-b_j}} \tag{6}$$

which gives relationship between AIMD parameter settings and bandwidth utilization ratio. This relationship provides an AIMD parameter design guideline to achieve bandwidth differentiation among all j types of traffic.

Besides, in the steady-state, the aggregated utilization is bounded by C, thus we have $\sum_i \hat{W}_i \leq C$. With (5), we yield

$$\sum_{i=1}^{j} N_i \sqrt{\frac{a_i}{p_s(1-b_i)}} \leq C \tag{7}$$

where p_s is the common marking probability for all traffic types.

With a targeted bandwidth utilization ratio, (6) provides solutions for AIMD settings, or a_i and b_i for all i. Using the obtained a_i and b_i values and (7), we can find a marking probability for the system such that the aggregated send rate of all rate limiters matches the link utilization by simply solving $\sum_i \hat{W}_i = C$.

In our system, the marking probability directly governs the frequency of sampling of the queue status and incoming packets. In other words, the marking probability affects the timeliness of the detection and reaction on a congestion event, and the choice of this setting is critical to ensure the effectiveness and stability of the congestion control. If the marking probability is set too high, the system may overreact to a congestion event by issuing excessive feedback messages. Conversely, if the marking probability is set too low, the system may be slow in reacting to a congestion event making the congestion recovery difficult. Consequently, an optimal setting of marking probability not only achieves full utilization of the link, but also ensures the adequate reaction of a congestion event so as to achieve stable control.

In our design, the congestion point sends no feedback when the queue length is less than the equilibrium threshold T_{eq} as the buffer is underutilized. In our model, this is equivalent to setting the marking probability to zero. When the queue length crosses T_{eq} but not the severe congestion threshold T_{sc}, a constant marking probability p_s applies. In this case, each incoming packet is sampled along with the queue length with a constant probability of p_s. When the queue length crosses T_{sc}, the congestion point sends feedback to the source to halt its transmission immediately. In our model, we set marking probability to one to describe the halt of a transmission. This handling is adequate as when the marking probability is set to one, all transmitted packets will trigger a feedback instructing a downward rate adjustment. A series of these feedbacks will quickly bring the congestion window to zero which is equivalent to halting a transmission. The above discussion can be presented in the following as

$$p = \begin{cases} 0, & q < T_{eq} \\ p_s, & T_{eq} \leq q < T_{sc} \\ 1, & q \geq T_{sc} \end{cases} . \tag{8}$$

Since the objective of the design is to maintain the queue length $q(t)$ to operate in the range $[T_{eq}, T_{sc}]$ at any time t, we shall focus on the performance under the condition where $T_{eq} < q(t) < T_{sc}$. In our design, $q(t)$ is forced to return back into the range $[T_{eq}, T_{sc}]$ when it falls outside the range. Specifically, when $q(t) < T_{eq}$, each rate limiter increases its congestion window to bring $q(t)$ up above T_{eq}. Likewise, when $q(t) > T_{sc}$, a collection of rate limiters is notified to halt their transmissions to quickly bring $q(t)$ back below T_{sc}.

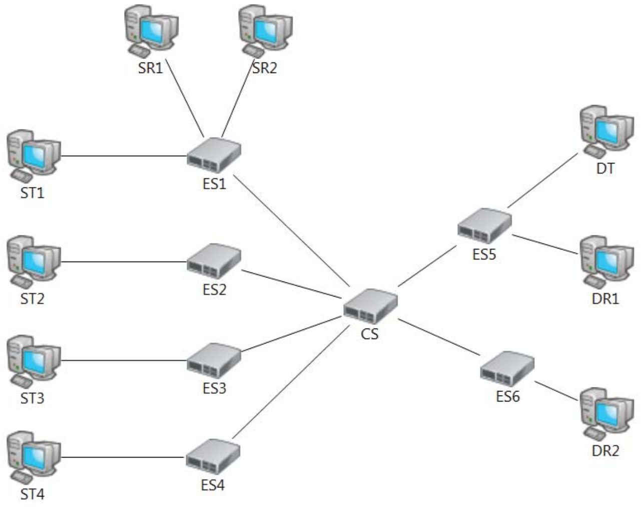


Fig. 5. Network topology for OMNET++ simulation study.

VI. Experiments and Result Discussions

Performance studies of our proposed differentiated congestion control mechanism are conducted. We perform a number of experiments to demonstrate the effectiveness and the performance benefits of our proposed mechanism. In all the presented experiments, we also present simulation results to validate our numerical analysis. Our simulation is implemented by OMNET++ 4.0 simulator [20]. In our rate limiters, the parameter u describing the step of a is set to 40Mbps. We also set the initial send rate to 40Mbps.

In our experiments, we follow the setup given in [21] and test the performance of our proposed mechanism. Figure 5 depicts the network topology used in our experiments. The data rate for all links is 1Gbps. In our scenario, the two sources SR_1 and SR_2 generate frames to the two destinations DR_1 and DR_2 at a certain rate. The two sources carry different types of traffic, where SR_1 carries the SAN and SR_2 carries the LAN traffic. Their transmission rates are regulated by rate limiters with SR_1 implementing AIMD(a_1, b_1) and SR_2 implementing AIMD(a_2, b_2)[1].

In Experiments 2-4, we include background traffic in the experiments. We use the nodes from ST_1 to ST_4 to generate CBR background traffic destined to the node DT through end switches ES_1 to ES_4. The rate of each CBR traffic is set to 4.8Mbps. As they are background traffic, they do not implement a rate limiter.

Given that all traffic flows share the common core switch CS, it represents the bottleneck of the network. The bottleneck may be due to lack of bandwidth capacity or limited switch processing capacity. In any case, this causes the buildup of buffer in the CS and activates the congestion control.

We apply heavy load into the network to test the performance of the congestion control in the bottleneck CS with the processing rate of 20,000 frames per second. For the frame length of 1500 bytes, this processing rate is equivalent to 240Mbps, which represents the bottleneck of the network. In

[1] We use the notation of AIMD(a, b) to describe a rate limiter implementing AIMD with parameters a and b.

the core switch, the thresholds T_{eq} and T_{sc} are set to 50,000 and 100,000 bytes respectively.

To give a comprehensive illustration of the effectiveness and the performance benefits of our proposed mechanism, we have conducted five experiments. We shall report the experiments and results in the following subsections.

A. Experiment 1: Performance Comparison with Various Schemes

In the first experiment, we compare our solution with normal Ethernet, ECM and DCTCP. The objective of this experiment is to investigate the effectiveness and system stability of our congestion control solution.

We compare five cases where in the first three cases, we consider standard Ethernet without congestion control, ECM and DCTCP respectively. To compare the performance with these schemes without the capability of differentiated congestion control, in the fourth case called DCC1, we reduce our solution to an undifferentiated congestion control by simply using a single AIMD parameter set in the system. In this setup, AIMD parameters in both sources are set to AIMD$(1, 0.5)$. Finally, in the fifth case called DCC2, we consider a scenario requiring a differentiated congestion control with coexistence of SAN and LAN traffic, where AIMD$(1.5, 0.8)$ is set for SAN and AIMD$(1, 0.5)$ is set for LAN. In order to focus on the influence of the various congestion control schemes on the performance, we do not include background traffic in this experiment.

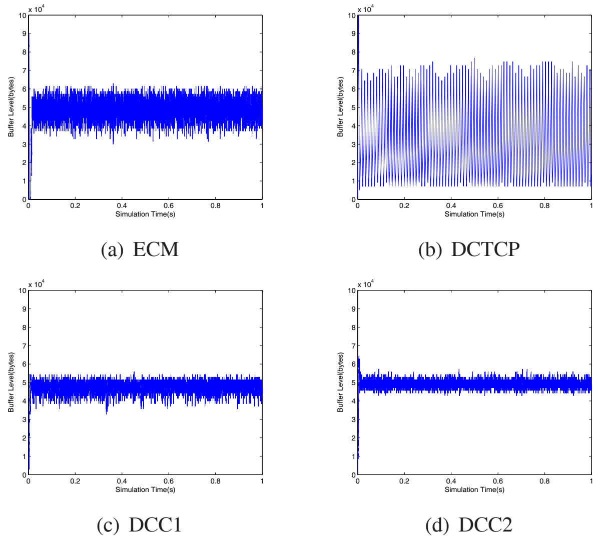
The parameter setting for this experiment is shown in Table II. The detailed description of the parameters for various schemes can be found in [13], [15]. We show the throughput of SR_1 and SR_2 separately and the total throughput of the two sources, in Megabyte (MB), in Table III. In the first case, the pure Ethernet switch performs its best effort to forward data frames. We can see that there is no differentiation in the bandwidth utilization between the two sources. Moreover, since we set a data frame processing time of 0.05ms for the core switch, the switch represents the congestion point in the network. With no control on the source, excessive frames generated by the two sources run freely to the core switch causing serious congestion and resulting high drop rate. In this test, almost half of the arriving traffic is discarded due to buffer overflow.

Whereas in the other four cases, with application of respective congestion control solutions, the core switch constantly regulates the buffer utilization through the rate adjustment at the sources, even with excessive generation of data frames from the two sources, the sources manage to hold the frames back from transmitting to the core switch. This traffic regulation successfully avoids frame drop in the core switch, which is indicated by zero frame loss reported in our result. In other words, all the solutions are effective in managing traffic congestion, and our solution can further provide differentiated congestion control.

Next we investigate the system stability for the cases with congestion control operations. We focus on the buffer level fluctuation for this study as a fast reaction to a buffer change allows stability of buffer level and avoids potential buffer overflow. From the results given in Fig. 6, we see that our solution has relatively low buffer level fluctuation compared to that of ECM and DCTCP. This result indicates that our solution has relatively high reaction in controlling traffic congestion. DCTCP records the highest buffer level fluctuation among all, this is mainly because DCTCP sources rely on TCP ACK to indicate congestion where the turnaround time between two end hosts is generally longer. Besides, the new rate adjustment scheme introduced in DCTCP has slower reaction even compared to TCP Reno which also contributes to the high buffer level fluctuation [22].

TABLE III
THROUGHPUT (MB) COMPARISON AMONG FIVE SCHEMES

CM	Ethernet	ECM	DCTCP	DCC1	DCC2
SR_1	14.9	14.9	14.6	15.0	20.1
SR_2	14.9	14.7	14.6	14.8	9.8
Total Throughput	29.8	29.6	29.2	29.8	29.9
Dropped at CS	22.2	0	0	0	0



(a) ECM

(b) DCTCP

(c) DCC1

(d) DCC2

Fig. 6. Buffer level evolution in the core switch (CS) with various schemes.

B. Experiment 2: Protections with Presence of Misbehaved Hosts

In the second experiment, we consider the situation where LAN sources do not regulate their rates based on the notification instructions given by the CP. They simply ignore the rate cut notifications and keep increasing their transmission rates. This action immediately causes unfairness in the bandwidth sharing at the CP making the congestion control meaningless.

In this experiment, we set the rate limiters of LAN sources such that they do not react to the downward rate adjustment notification. In other words, we use AIMD$(a, 1.0)$ for the LAN sources. Other parameters used in this simulation are listed in Table IV. We show the results in Fig. 7 and Table. V. Figure 7 shows the buffer level in the core switch. Although oscillations exist, the buffer level never goes over T_{sc}. In this simulation, only SAN sources respond to feedback notifications and LAN sources do not cut its rate, this results in over punishment of SR_1 and the ineffective traffic regulation. Instead of reaching a

TABLE II
PARAMETER SETTING IN EXPERIMENT 1

Parameters	Ethernet	ECM	DCTCP	DCC1	DCC2
Q_{eq} / T_{eq}	N/A	50,000B	50,000B	50,000B	50,000B
Q_{mc}	N/A	80,000B	N/A	N/A	N/A
Q_{sc} / T_{sc}	N/A	100,000B	N/A	100,000B	100,000B
Buffer Size	300,000B	300,000B	300,000B	300,000B	300,000B
W, Ru	N/A	W=2 R_u=50Mbps	N/A	N/A	N/A
G_i, G_d	N/A	G_i=0.3 G_d=0.6	N/A	N/A	N/A
AIMD(a_1, b_1) AIMD(a_2, b_2)	N/A	N/A	N/A	AIMD$(1, 0.5)$ AIMD$(1, 0.5)$	AIMD$(1.5, 0.8)$ AIMD$(1, 0.5)$
Initial Rate	N/A	1Gbps	N/A	40Mbps	40Mbps
Initial R_{min}	N/A	10Mbps	N/A	N/A	N/A
Initial T_{max}	N/A	1ms	N/A	N/A	N/A
g	N/A	N/A	0.02	N/A	N/A

TABLE IV
PARAMETER SETTING IN EXPERIMENT 2

	Parameters	Value
SR_1	Traffic Generation Rate	240Mbps
	Rate Limiter Setting	AIMD$(1.5, 0.8)$
SR_2	Traffic Generation Rate	240Mbps
	Rate Limiter Setting	AIMD$(1.0, 1.0)$

TABLE V
NUMBER OF SENT AND DROPPED FRAMES IN EXPERIMENT 2

Name	Value (frames)
Sent from RL_1	5008
Sent from RL_2	4951
Received at DR_1	5008
Received at DR_2	4371
Dropped at CS	580

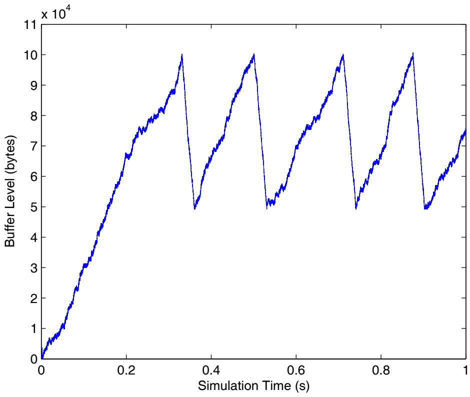


Fig. 7. Buffer level evolution in the core switch (CS) when excessive traffic arrives to the network from SR_2.

steady state around T_{eq}, the buffer level climbs up even higher due to the excessive LAN traffic. However, at simulation time of 0.33s when the buffer level hits the severe threshold T_{sc}, frame drop mechanism for LAN traffic is triggered to ensure lossless transmission of SAN traffic. We see the buffer level quickly reduces until it reaches the equilibrium threshold T_{eq} again at 0.36s, and the frame drop of LAN traffic is turned off again. From this point, the core switch handles the two sources equally and repeats the cycle of previous operations.

As can be seen from Table V that the number of frames sent by SR_1 is similar to that of the frames received by DR_1 and the difference between the number of frames sent by SR_2 and that of the frames received by DR_2 matches the dropped frames at the core switch. The results confirm that the mechanism of dropping LAN frames to protect SAN traffic when severe congestion happens is effective.

C. Experiment 3: Bandwidth Utilization Differentiation

In the third experiment, we analyze the bandwidth differentiation feature of our proposed system, using the previous developed analytical model in Section V. In addition, the analysis results are compared with simulation results from OMNET++. This study tests the effectiveness of bandwidth differentiation and provides guideline for the implementation of our solution. We illustrate the bandwidth utilization differentiation by showing the bandwidth differentiation ratio of SAN traffic to that of LAN traffic.

In this study, the AIMD setting for the SAN source is fixed and that of LAN source is varied over a range that is less aggressive than of the SAN source. We report the bandwidth differentiation ratio of SAN traffic over LAN traffic with different parameter settings in Fig. 8. As can be seen, varying the parameters a and b of LAN traffic with AIMD(a, b) can provide differentiation in bandwidth utilization. We see that as the values of a and b decrease, the bandwidth differentiation ratio of SAN to LAN traffic increases. We observe that when a switch experiences congestion, it penalizes LAN traffic with AIMD(a, b) where $a < 1, b < 0.5$ more than that of SAN traffic with AIMD(1,0.5). We also observe the potential of AIMD congestion control differentiation where SAN traffic can utilize as high as 3.5 times that of LAN traffic in the network. Under some extreme configurations such as AIMD(0.2,0.2), our analytical results give good agreement to that of the simulation.

To further illustrate the bandwidth differentiation ratio, we show analytical results in Fig. 9, where a fixed setting of AIMD(1,0.5) is used for the LAN source while varying AIMD parameter setting for the SAN source. Since SAN traffic has

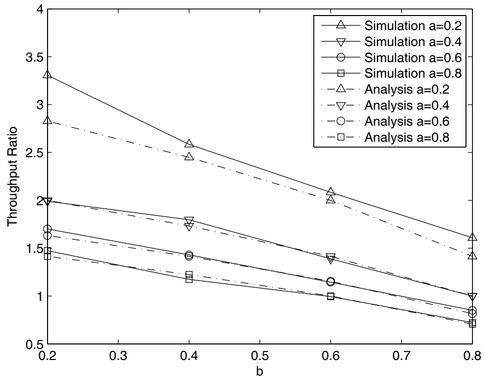


Fig. 8. Comparison between analytical and simulation results of bandwidth differentiation ratio of SAN traffic with AIMD(1,0.5) to LAN traffic with AIMD(a,b).

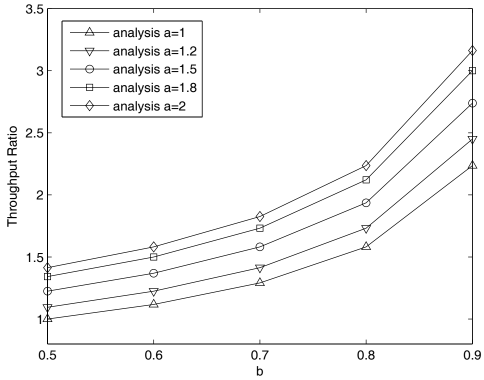


Fig. 9. Bandwidth differentiation ratio of SAN traffic with AIMD(a,b) to LAN traffic with AIMD(1,0.5) obtained from control theoretic analysis.

a higher priority, a more aggressive setting than AIMD(1,0.5) for the SAN source is used. Specifically, we use AIMD(a, b) where $a > 1, b > 0.5$. In the following, we demonstrate the design of AIMD settings for a particular desirable ratio with two examples. The first example considers a desired bandwidth differentiation ratio of 2:1, that is if we wish the SAN source to utilize two times higher bandwidth than that of the LAN source. Based on the analytical results given in Fig. 9, we may choose AIMD(1.5,0.8) and AIMD(1,0.5) for the SAN and LAN sources respectively. In the second example, we consider a desirable bandwidth differentiation ratio of 1.6:1. According to Fig. 9, we may use AIMD(1,0.8) for the SAN source and AIMD(1,0.5) for the LAN source. To confirm that our selections give accurate desirable ratios, we perform simulation measuring the throughput with results showing in Table VI. We see that the desirable bandwidth differentiation ratios are indeed achieved.

D. Experiment 4: Setting of Marking Probabilities

We now study the sensitivity of marking probability on the overall throughput performance at the CP. In our considered system, the core switch marks incoming traffic from different sources with the same probability. As (6) suggests, varying the value of marking probability does not affect the bandwidth differentiation ratio. Therefore, we can select an optimized probability without compromise on bandwidth differentiation ratio.

TABLE VI
BANDWIDTH DIFFERENTIATION PERFORMANCE FOR ANALYSIS AND SIMULATION RESULTS

Case 1	**Case 2**
$a_1 = 1.5, b_1 = 0.8$ $a_2 = 1, b_2 = 0.5$	$a_1 = 1, b_1 = 0.8$ $a_2 = 1, b_2 = 0.5$
Ratio (analysis) = 1 : 1.96 Ratio (simulation) = 1 : 2.04	Ratio (analysis) = 1 : 1.58 Ratio (simulation) = 1 : 1.61

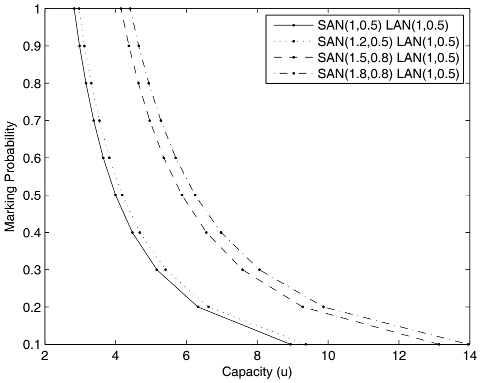


Fig. 10. The relationship between marking probability and the throughput at the core switch.

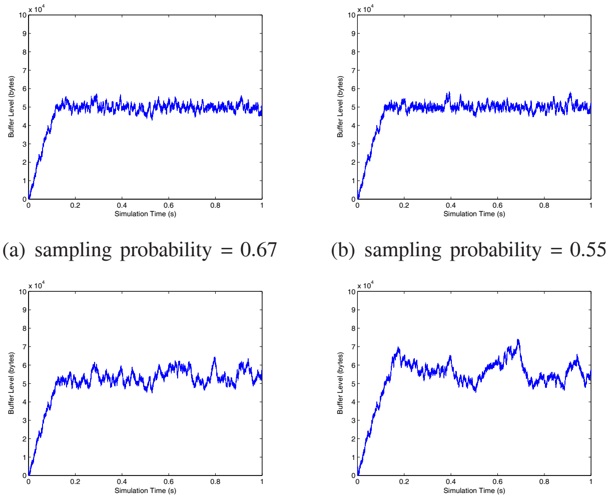
As shown in (7), we have derived the relationship between the marking probabilities and the network capacities. Based on this relationship formula, the best match marking probability can be computed given a certain network capacity. Figure 10 shows the capacities of the core switch and their relevant marking probabilities under four different AIMD settings. Specifically, to each given capacity, after acquiring a set of AIMD parameters that satisfies ratio requirements from Fig. 9, Fig. 10 offers a marking probability demanded by the model accordingly. Although setting a higher marking probability can ensure a stable buffer level around equilibrium threshold as well, this is unnecessary due to the more overhead it may introduce.

Now with experiment settings given by Table VII, we further test the impact on the buffer level with different marking probabilities specified by the term of sampling rates. Similar to the earlier experiments, this scenario considers two SRs sending with a source rate of 240Mbps each, and four STs sending background traffic with a source rate of 4.8Mbps each. We consider a frame length of 1,500 bytes with the processing rate of 20,000 frames per second. Likewise, the processing rate represents the bottleneck of the network capacity where the CS can only deliver 240Mbps of traffic. Excluding the background traffic, the total available network capacity for SRs is 220.8Mbps. With $u = 40$Mbps, this gives $C = 5.52$ in the unit of u.

Using our developed formula given in (7), with $C = 5.52$, an adequate setting of marking probability that maintains high overall link utilization can be determined to be 0.56, and the corresponding sampling rate should be set to 2700 bytes.

TABLE VII
PARAMETER SETTING IN EXPERIMENT 4

Parameters	Value
Sampling Rate	2200,2700,3600,4300bytes
SR_1 Rate Limiter	AIMD(1.5,0.8)
SR_2 Rate Limiter	AIMD(1.0,0.5)



(a) sampling probability = 0.67 (b) sampling probability = 0.55

(c) sampling probability = 0.42 (d) sampling probability = 0.35

Fig. 11. Buffer level in the core switch with different sampling rates.

For performance comparison, we test with the sampling rates of 2200, 2700, 3600, 4300 bytes, which correspond to the sampling probabilities of 0.67, 0.55, 0.42, 0.35, respectively. The simulation results are shown in Fig. 11.

Based on the results presented in Fig. 11, it is observed that the marking probabilities of 0.67, 0.55 can maintain a stable buffer level, as shown in Fig. 11(a) and Fig. 11(b). The comparison between these two figures also shows that sampling more frequently does not significantly improve buffer stability when a stable buffer have achieved. At the same time, increase in sampling rate also results in more overhead in the system. On the other hand, as the sampling probability decreases, the buffer level oscillates in a larger range as shown in Figs. 11(c), 11(d). Though a lower sampling probability incurs less overhead, it leads to system instability and jeopardizes traffic with high priorities. Therefore, sampling rate of 2700 bytes, with corresponding sampling probability of 0.55, is a recommendation setting for the given experiment as we expect.

E. Experiment 5: A Case Study for Multiple Types of Traffic

Finally, we present a case study for multiple traffic types. The considered network topology is the same as that shown in Fig. 5. In this case study, we provide specific roles for each device as shown in Fig. 12. Our scenario deals with uploading of files from clients and a total of three types of traffic are involved.

Client1 and Client2 first connect to ApplicationServer1 sending upload requests of LAN traffic type, and this traffic is light. On reception of any request, ApplicationServer1 connects to MetadataServer for data index, and the MetadataServer replies with index to identify and track data storage locations. Traffic load between ApplicationServer1 and MetadataServer can be high, especially with the arrival of multiple simultaneous requests. After index being obtained, Client1 and Client2 connect to the storage directly for file writing, which is SAN traffic. Concurrently, we consider that ApplicationServer2 and ApplicationServer3 also generate index request traffic but for other arbitrary applications, the index requests between the three Application Servers and Metadata Server are considered as LAN traffic. At the same time, HPCServer1 communicates with HPCServer2 with Inter Process Computing (IPC) traffic.

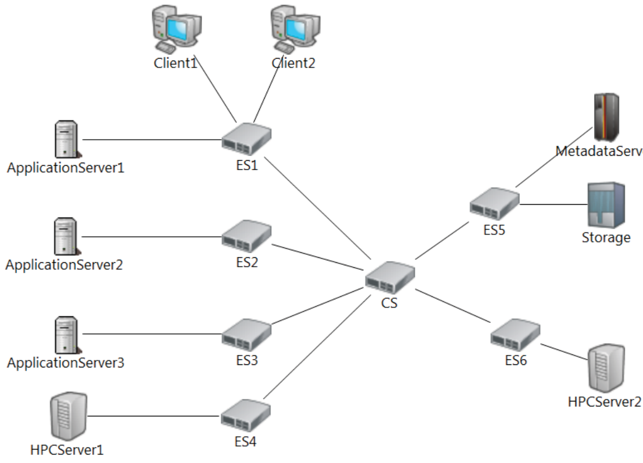


Fig. 12. Illustration of a case study for multiple sources transmission.

In the aforementioned scenario, we select three traffic flows in the purpose of ratio comparison. The first flow comes from ApplicationServer1 to MetadataServer, denoted as *LAN* traffic below. Since TCP takes control of congestion handling, we assigned this LAN flow with the lowest priority. Traffic sending from Client1 to Storage acts as the second flow with higher priority, namely *SAN* traffic. Finally, *IPC* traffic between HPCServer1 and HPCServer2 is the third flow with the highest priority. We adopt a target bandwidth utilization ratio of 1:1.4:1.8 for $\hat{W}_1 : \hat{W}_2 : \hat{W}_3$ where $\hat{W}_1$, $\hat{W}_2$ and $\hat{W}_3$ describe bandwidth utilization for LAN, SAN and IPC traffic.

Following the guideline provided in Experiment 3, we calculate parameters for AIMD and configure them accordingly in the simulations. Thus the targeted ratio can be achieved. Note that there is more than one solution of AIMD parameter settings that can fulfill the targeted ratio, we apply the following method to search for the appropriate AIMD parameter sets. Specifically, we first consider $a_1 = 1, b_1 = 0.5$ for the AIMD setting of the LAN traffic which is also serving as a reference setting to other types of traffic. Given (6), recursively, we find a particular $a_{i+1} \geq a_i$ and a particular $b_{i+1} \geq b_i$ such that $W_i : W_{i+1} = r_i : r_{i+1}$. We also apply rounding whenever appropriate on the AIMD parameters for the ease of practical operation.

Regarding the simulation settings, the processing capacity is kept the same as the previous setting. For simplicity, we set the main communication parties with average traffic generation rate of 240Mbps each, which includes index requests between three Application Servers and Metadata Server, interprocess computing between the two HPC Servers, and file upload-

TABLE VIII
AIMD PARAMETERS AND ACHIEVED RATIOS FOR CASE STUDY.

AIMD Parameters	Achieved Ratios ($\hat{W}_1 : \hat{W}_2 : \hat{W}_3$)
$a_1 = 1.0, b_1 = 0.5$	
$a_2 = 1.2, b_2 = 0.7$	$1 : 1.46 : 1.78$
$a_3 = 1.9, b_3 = 0.7$	
$a_1 = 1.0, b_1 = 0.5$	
$a_2 = 1.6, b_2 = 0.6$	$1 : 1.44 : 1.77$
$a_3 = 1.6, b_3 = 0.75$	
$a_2 = 1.0, b_2 = 0.5$	
$a_2 = 1.6, b_2 = 0.6$	$1 : 1.44 : 1.76$
$a_3 = 1.9, b_3 = 0.7$	

ing traffic from clients. Service requests from the clients to ApplicationServer1 are also simulated, but the traffic is comparably small. However, we believe the results of this work are also applicable if they generate traffic at different rates.

We illustrate the achieved bandwidth utilization ratios in Table VIII for various settings of AIMD parameters. With an ideal mechanism, the targeted ratio of 1:1.4:1.8 should be achieved with high precision. As can be seen from Table VIII, a certain level of deviations in the ratios is reported for the three settings. This is because our solution is based on stochastic mechanism where the sampling is random. Nevertheless, the deviations are considered minor for all these settings. This case study involving in three types of traffic again confirms the effectiveness of our design.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a differentiated congestion control for Ethernet congestion management. Our proposed method considered using different AIMD settings for the rate limiter operations and took protection of high priority traffic into account to regulate malicious parties. We analyzed the effectiveness of the solution by studying the potential of the bandwidth differentiation ratio with results showing that this method has a potential to allow the SAN traffic utilizing as high as 3.5 times that of LAN traffic throughput. At the same time, guidelines for parameter settings have been constructed with requirements on bandwidth differentiation ratio. With developed relationship between network capacity and sampling probability, a suggested sampling probability can also be computed given source parameters and ratio requirements, thus a higher system performance with low overhead can be achieved.

In the future, we plan to perform further investigations on the compatibility of our solution with other higher layer protocols in the same protocol stack. An immediate study is the interaction between our solution and TCP as it is expected that SAN traffic will be dominated by TCP packets. We have performed some simulation tests and our preliminary results show that the bandwidth is generally governed by the rate limiter at Layer-2 as the rate limiter at Layer-2 reacts faster than that at Layer-4. This suggests that the targeted differentiated congestion control remains effective when the network carries TCP traffic. A comprehensive study of such issues shall be addressed in our future work.

REFERENCES

[1] "Fibre Channel over Ethernet in the data center: an introduction." Available: http://www.fibrechannel.org, Fibre Channel Industry Association, 2007.

[2] G. Silvano and D. Claudio, "I/O consolidation in the data center: A complete guide to data center Ethernet & Fibre Channel over Ethernet," 2009.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. ACM SIGCOMM*, vol. 28, no. 4, 1998, pp. 303–314.

[4] F. Baccelli and D. Hong, "The AIMD model for TCP sessions sharing a common router," in *Proc. 2001 Annual Allerton Conf. Commun., Control Comput.*, pp. 900–911.

[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*, vol. 32, no. 4, 2002, pp. 117-130.

[6] G. Hasegawa, K. Kurata, and M. Murata, "Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet," in *Proc. 2000 International Conf. Netw. Protocols*, pp. 177–186.

[7] Y. R. Yang and S. S. Lam, "General AIMD congestion control," in *Proc. 2000 International Conf. Netw. Protocols*, pp. 187–198.

[8] S. Floyd, M. Handley, and J. Padhye, "A comparison of equation-based and AIMD congestion control." Available: http://www.aciri.org/floyd/papers.html, 2000.

[9] C. P. Fu, C. H. Foh, C. T. Lau, Z. Man, and B. S. Lee, "Semi-Markov modeling for bandwidth sharing of TCP connections with asymmetric AIMD congestion control," in *Proc. 2007 IEEE Global Telecommun. Conf.*, pp. 2014–2018.

[10] Y. Ariba, F. Gouaisbaut, and Y. Labit, "Feedback control for router management and TCP/IP network stability," *IEEE Trans. Netw. Service Manag.*, vol. 6, no. 4, pp. 255–266, 2009.

[11] F. Paganini, Z. Wang, S. H. Low, and J. C. Doyle, "A new TCP/AQM for stable operation in fast networks," in *Proc. 2003 IEEE INFOCOM*, pp. 96–105.

[12] V. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, vol. 30, no. 4, 2000, pp. 151–160.

[13] D. Bergamasco, "Data center ethernet congestion management: Backward congestion notification," in IEEE 802.1 Meeting, 2005.

[14] "IEEE. 802.1Qbb - Priority-based Flow Control." Available: http://www.ieee802.org/1/pages/802.1bb.html, IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Amendment2008c, 2008.

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

[16] "Converging SAN and LAN Infrastructure with Fibre Channel over Ethernet." Available: http://download.intel.com/support/network/sb/ciscointelfcoewhitepaper.pdf, Cisco and Intel technical report.

[17] S. Fang, C. H. Foh, and K. M. M. Aung, "Differentiated Ethernet congestion management for prioritized traffic," in *Proc. 2010 IEEE International Conf. Commun.*

[18] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoretic analysis of RED," in *Proc. 2001 IEEE INFOCOM*, vol. 3, pp. 1510–1519.

[19] Y. Lu, R. Pan, B. Prabhakar, D. Bergamasco, V. Alaria, and A. Baldini, "Congestion control in networks with no congestion drops," in *Proc. 2006 Annual Allerton Conf. Commun., Control, Comput.*, pp. 891–898.

[20] "OMNeT++." Available: http://www.omnetpp.org.

[21] J. Jiang and R. Jain, "Analysis of backward congestion notification (BCN) for Ethernet in datacenter applications," in *Proc. 2007 IEEE INFOCOM*, pp. 2456–2460.

[22] M. Alizadeh, A. Javanmard, and B. Prabhakar, "Analysis of DCTCP: stability, convergence, and fairness," in *Proc. 2011 ACM SIGMETRICS*.

Shuo Fang received her B.Eng. degree on Information Security from Beijing University of Technology, China, in 2008. She is currently a Ph.D candidate in School of Computer Engineering, Nanyang Technological University, Singapore. Her research interests are in the areas of congestion control and architecture design for data center networks.

Chuan Heng Foh received his Ph.D. degree from the University of Melbourne, Melbourne, Australia, in 2002. From July 2002 to December 2002, he was a Lecturer at Monash University. In December 2002, he joined the School of Computer Engineering, Nanyang Technological University, Singapore as an Assistant Professor. He is serving on the editorial board of the *International Journal of Communications Systems*. His research interests include protocol design and performance analysis of wireless local area and mesh networks, sensor networks, and storage area networks.

Khin Mi Mi Aung received the B.S degree from Yangon University, Myanmar, in 1995, the M.S degree from University of Computer Studies, Yangon, Myanmar, in 1999, and the Ph.D. degree in Computer Engineering from Korean Aerospace University in 2006. She is currently a Research Scientist with The Agency for Science, Technology and Research (A*STAR), Data Storage Institute, Singapore. Her research interests include computer network protocols, storage network and data center technologies.