Proactive Ethernet Congestion Control Based on Link utilization Estimation

Mahmoud Bahnasy*, Bochra Boughzala‡, Halima Elbiaze†, Brian Alleyne‡, Andre Beliveau‡§, Chakri Padala‡

*École de Technologie Supérieure, mahmoud-mohamed.bahnasy.1@ens.etsmtl.ca
†Université du Québec à Montréal, elbiaze.halima@uqam.ca
‡Ericsson, {bochra.boughzala,brian.alleyne,andre.beliveau,chakri.padala}@ericsson.com

***Abstract*—In the quest of providing a deterministic backplane-like behaviour a mechanism called ECCP (Ethernet Congestion Control & Prevention) was proposed where specialized and expensive devices were replaced by commodity Ethernet switches while still keeping the router properties, e.g. low-latency packet delivery, no packet loss within the fabric and fair share of the bandwidth. However, ECCP uses a self-induced congestion probing model which can cause queue length fluctuation in addition to the network bandwidth wastage due to the probes. Therefore, we propose enhancements to the algorithm used by ECCP to reduce probe packet overhead. In our solution NoP-ECCP (ECCP with no probes), we use a link utilization estimation technique instead of the estimation of the available bandwidth. The results obtained through simulations show that NoP-ECCP outperforms ECCP in terms of fairness, link utilization and queue length.**

***Keywords*—Congestion Control, Ethernet, Switch fabric.**

I. Introduction

Traditional routers are built in a monolithic structure where all the line cards are placed in a single chassis and they are connected together through the backplane. One of the main characteristics of a switch fabric in routers is that every packet that comes from an ingress line card into the fabric has to be delivered to the egress line card within a deterministic delay [1], [2]. In fact, the switch fabric requires guaranteed low latency packet delivery, fairness in bandwidth allocation, no packet loss and no head-of-line blocking behavior. Thus, specialized chips and proprietary protocols are applied in the switch fabric in order to maintain these tight properties in terms of bandwidth allocation and latency. These vendor-specific solutions are very expensive and they also have scalability limitations. In fact, with a rigid backplane it is not possible to expand the router and add more line cards unless the complete design of the entire system is reviewed.

In the other hand, Data Center Networks depict the field where flexible fabric is used to interconnect servers and network switches. Some enhancements, like QCN [3], [4], [5], are proposed by the IEEE Data Center Bridging (DCB) [6], [7] working group to introduce quality of service mechanisms in Data Center Networks. In addition, Data Center Networks use over-provisioning and they don't guarantee packet delivery at Layer 2 so they rely on the upper protocols like TCP to retransmit lost or dropped packets.

§ This author is no longer at Ericsson Research.

Our previous work [8] explored the possibility to deploy a router in a distributed fashion where the authors developed a new protocol called ECCP in order to replace the specialized switch by a standard Ethernet commodity switch. ECCP uses self-induced congestion probing model to estimate the available bandwidth. This model may generate probe trains of rates higher than the available bandwidth which can cause queue length fluctuation. In addition, ECCP probing model requires fixed inter-frame intervals and fixed frame size which prevent using data as probes. In this paper, we propose estimating link utilization instead of the available bandwidth and we implement a mechanism that combines ECCP with link utilization estimation.

The rest of the paper is organized as follows: Section II gives background to ECCP and identifies its limitations. Section III introduces the link utilization estimation technique. Section IV provides detailed description about our proposed solution NoP-ECCP. The performance evaluation of the proposed solution is presented in Section V. Finally, Section VI presents the conclusion and future work.

II. Ethernet Congestion Control and Prevention (ECCP)

ECCP is proposed as a congestion control mechanism for Ethernet networks. It is a distributed algorithm that can run either on end-hosts or the router's line cards without the need of switch participation. ECCP aims to keep the link utilization under the maximum link capacity by a certain percentage (Availability Threshold AvT). ECCP uses the estimation of the available bandwidth ($AvBw$) in order to limit the line card Sending Rate (R).

ECCP's main components are depicted in Fig. 1. ECCP *probe sender* starts the control cycle by sending probe trains of N packets each. ECCP *probe sender* stamps each probe packet with its sending time. Probe rate μ is limited by the available bandwidth margin which is equal to the availability threshold percentage of the sending rate ($AvT \times R$). At the receiving line card, the *probe receiver* receives the probes, reads their information, adds receiving time, and then forwards this information to the *bandwidth estimator*. ECCP *bandwidth estimator* uses a modified version of BART (Bandwidth Available on Real Time) [9] to estimate $AvBw$ and then sends this information back to the paired line card. Once ECCP *rate controller* receives $AvBw$ information, it calculates a feedback value (F_b) and then it controls the sending rate (R) according

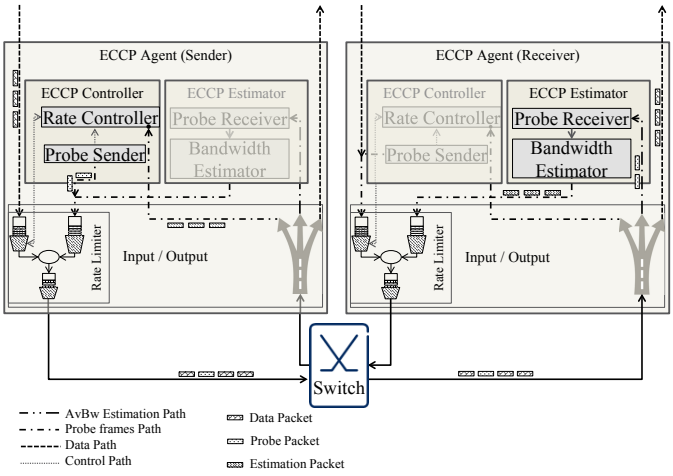


Fig. 1. ECCP's main Components

to the Additive Increase Multiplicative Decrease (AIMD) [10], [11] principle (Equation 1).

$$R \leftarrow \begin{cases} R(1 - G_d \times |F_b|) & if\ F_b < 0 \\ \frac{1}{2}(R + TR) & otherwise \end{cases} \quad (1)$$

Where G_d is a fixed value and is taken in a way that $G_d|F_{b_{max}}| = \frac{1}{2}$ and TR is the target rate which is equal to the last sending rate before congestion.

Similarly to QCN reaction point algorithm [5], ECCP rate increase process defines three phases: Fast Recovery (FR), Active Increase (AI) and Hyper-Active Increase (HAI).

ECCP determines the increasing phases based on a $ByteCounter$ and a $Timer$ (Fig. 2). The Fast Recovery phase defined by 5 cycles, where each cycle consists of 150 KBytes of data transmission. In case the host sending rate is very slow, the end of the cycle is triggered by the $Timer$ expiration (T = 3ms). After each cycle, the *rate controller* changes R according to equation 1 while keeping TR unchanged.

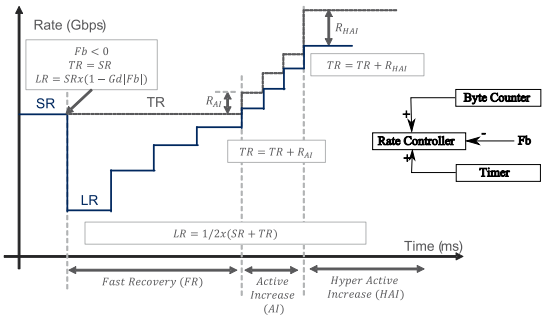


Fig. 2. ECCP rate control phases

If the $ByteCounter$ or the $Timer$ completes 5 cycles in FR phase without calculating negative feedback, the *rate controller* enters the Active Increase phase and increases TR by $R_{AI} = 5Mbps$. In this phase, the $ByteCounter$ counts 75 KBytes for each cycle, and the $Timer$ counts T/2. Finally, if both the $ByteCounter$ and the $Timer$ complete 5 cycles, the *rate controller* enters the Hyper-Active Increase phase and increases TR by $R_{HAI} = 50Mbps$ (Further details can be found in [8]).

The main advantages of using ECCP are that it has a proactive control on the sending rate and it doesn't wait for congestion to happen in order to react. Moreover, ECCP ensures fairness between all the nodes since they continue probing for the available bandwidth individually and they adapt to changes in network conditions. The simulation results of ECCP show that all the flows converge to a fair share of the link capacity (Fig. 8b).

ECCP is also scalable as the probe amount is independent of the number of hosts and flows and it is limited by the available bandwidth margin. Another advantage of ECCP is that it doesn't require switch modification, it is implemented either on the end-hosts or the router's line cards and can function with simple standard Ethernet switches. The main objective of ECCP is to build router using Ethernet commodity switches. Thus, in the rest of the paper we demonstrate how to implement ECCP agent within router's line cards.

A. Limitations of ECCP

ECCP is based on a self-induced congestion probing model that allows the probes to be generated in a rate higher than the available bandwidth. When the system approaches the network congestion, queue length fluctuation is observed. Such behavior impacts the packet latency and jitter. Also in order to limit the chance of hitting congestion, ECCP limits the link utilization to never reach the maximum link capacity which causes a waste of the network bandwidth. The probes generated by ECCP controller depicts also an amount of the bandwidth that is taken from the actual network traffic workload.

In addition to the wastage of the network bandwidth, the probe generation requires processing power and that introduces an overhead in terms of CPU usage.

B. Our contribution

In this paper, we propose a new variant of ECCP with the intent of addressing the limitations mentioned before. We explore a technique to estimate link utilization [12] and we use it to replace the available bandwidth estimation that is used in the original ECCP. We developed NoP-ECCP (No Probe ECCP) and we evaluated its performance through simulations. We also made the comparison of this new mechanism against the original ECCP and we concluded that NoP-ECCP achieves better performance in terms of queue length, network latency and link utilization than ECCP.

III. Link Utilization Estimation

As mentioned earlier, ECCP uses BART as the available bandwidth estimation technique which is based on self-induced congestion. The concept of self-induced congestion is simple as if the injected probe rate μ is less than or equal to $AvBw$, the arrival probe rate r matches the probe rate at the sender ($\mu/r = 1$). However, if μ exceeds $AvBw$, the probe packets are queued and the output probe delay is increased consequently reducing r ($\mu/r > 1$) (Fig. 3). Thus, this technique requires sending probes with rates higher than the available

bandwidth in order to estimate $AvBw$ which might put the investigated path into congestion.

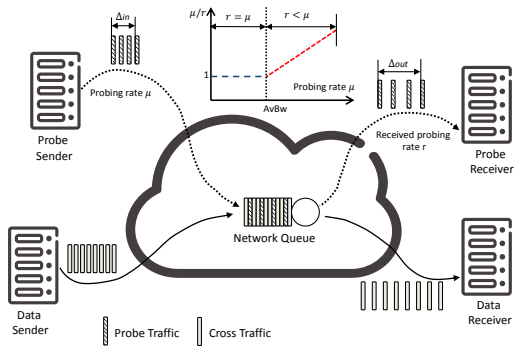


Fig. 3. Active probing scheme based on self-induced congestion

On the other hand, ECCP controls sending rate using a self-increase process, and rate decrease process based on negative feedback. Therefore, ECCP does not require an exact estimation of $AvBw$, it only requires a feedback when the link is close to congestion in order to trigger the rate decrease process. Thus, an indication of the increasing in link utilization would be enough for ECCP.

In this paper we propose estimating link utilization rather than estimating available bandwidth. We use a technique to estimate link utilization using low rate probe traffic [12], [13]. In this technique, the end-to-end network path is modeled as concatenated hops where each hop consists of an input queue and a transmission link. The utilization of the queue in a single-hop scenario is $U = 1 - \pi$. Where π is the probability that the queue is empty. By sending a low rate probe μ, the link utilization can be expressed as $U(r) = \min(1, U + \mu/C)$, where C is the capacity of the link. For multi-hop case, this equation can be approximated as the following first order equation (See [13]):

$$U(r) \approx \min(1, a\mu + b) \tag{2}$$

Where a and b are constants.

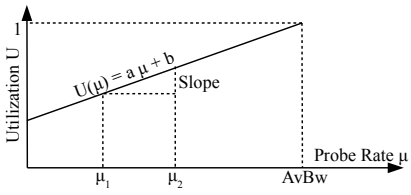


Fig. 4. Link Utilization Estimation

Equation 2 shows that the relation between the link utilization and the probe rate is linear (Fig. 4). By calculating a and b using two probe rates (Fig. 4), $AvBw$ can be estimated as the probe rate at the point where the link utilization is 1 (Equation 3).

$$AvBw = (1 - b)/a \tag{3}$$

The key to solve equations 2 and 3 is to estimate link utilization. [13] suggests sending a packet train and compute the fraction of packets that have experienced queuing delay along the path. By stamping the probe packets at the sender and the receiver, one can compute the minimum one-way delay of any set of packets $\{p_1, p_2, .., p_N\}$. This minimum delay corresponds to the packets that have not experienced any queuing delay. Thus, the packets that experience delay greater than the minimum delay are the fraction of packets that is queued. Hence the link utilization can be estimated using this equation:

$$\bar{U} = \frac{||d_i > min(D) \mid d_i \in D||}{||D||} \tag{4}$$

Where $D = \{d_1, d_2 ... d_N\}$ is the set of one-way delays experienced by packets $\{p_1, p_2, .., p_N\}$. In the next section, we depict how to use the estimated link utilization to calculate a feedback value which reflects the state of congestion and use this value to control line cards' sending rate.

IV. ECCP WITH NO PROBES (NOP-ECCP)

Unlike BART, the link utilization estimation technique, discussed in section III, does not require fixed inter-frame interval and does not need to inject probe with rate higher than $AvBw$. By eliminating these restrictions, we can use data frames as probes. Yet, data frames need to be time stamped, which is impossible at the Ethernet layer (There is no field to add time stamp). In order to overcome this issue, we propose keeping track of the sending time, Frame Check Sequence (FCS) and frame length of the last H frames in a Frame Information List (FIL) at the sender side (Fig. 5). NoP-ECCP uses FCS and frame length pair as frame identifier because the possibility of having repeated FCS and frame length within H frames is very rare [1]. In addition, ECCP is originally designed to be Ethernet protocol, consequently FCS will not be changed as long as the frame remains within the same Ethernet network. FIL length H is taken to be greater than the number of frames that can be sent while waiting for data information acknowledge $(H > (C * T/L))$, where L is the average frame size and T is the time between two acknowledges $(H > 10 * 10^9 * 0.5 * 10^{-3}/(1000 * 8) = 625)$. Fig. 5 depicts that NoP-ECCP does not use probe generator which reduces the required computational power.

At the receiver side, NoP-ECCP data sampler samples the received data based on a byte counter BC_r (Fig. 5). The sampling Byte counter BC_r is taken in the simulation to be equal 300 KB. Note that the sampling is based on byte counter instead of timer, which achieves fairness by generating more feedback messages for the high rate flows. Once this counter expires, NoP-ECCP receiver sends the receiving time, FCS and frame length of the last N frames encapsulated in an Ethernet frame to the sender line card (N is equal 32 in the simulation). At the sender side, when NoP-ECCP estimator receives this information, it searches in the FIL for the sending time of each frame based on FCS and frame length. Then, it uses a simplified version of the previously introduced link utilization estimation technique to calculate an estimation of the congestion (CE) as the percentage of packets that exceed the minimum delay (Equation 5).

[1]Other fields in upper layer, like ID in layer 3 or sequence number in layer 4, could be used as identifiers or alternatively source time stamp can be sent with the packet as meta-data, for example as a header extension at the IPv6 level, and the congestion estimation computation performed at the receiver instead of at the sender as described in the paper.

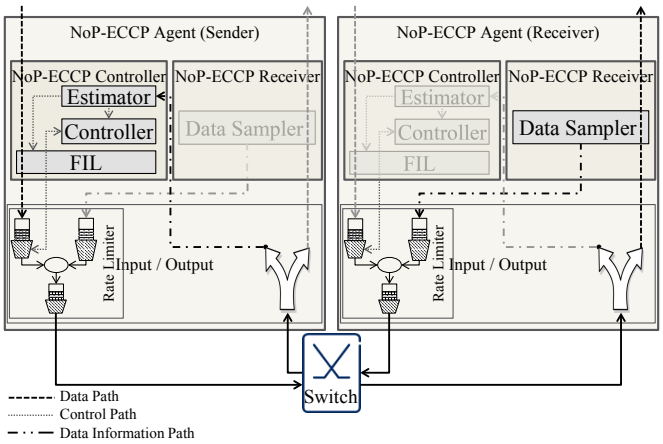


Fig. 5. NoP-ECCP Components

$$\bar{CE} = \frac{Count((d_i - min(D)) > 0) \mid d_i \in D}{Count(D)} \quad (5)$$

In order to reduce the effect of measurement error and system noise, we consider a certain delay threshold (D_{th}) before counting delayed packets (Equation 6).

$$\bar{CE} = \frac{Count((d_i - min(D)) > D_{th}) \mid d_i \in D}{Count(D)} \quad (6)$$

In NoP-ECCP data is used as probe, but data frame lengths is not fixed like probe frames. Hence, we normalize the frame delay to its length and then multiply it by the average frame size (which is equal 1000 Bytes in the simulation). Thus equation 6 becomes:

$$\bar{CE} = \frac{Count((\hat{d}_i - min(\hat{D})) > D_{th}) \mid d_i \in \hat{D}}{Count(\hat{D})} \quad (7)$$

Where $\hat{D} = \{\hat{d}_1 \hat{d}_2 ... \hat{d_N}\}$ is the set of normalized one-way delay, $\hat{d}_i = \frac{d_i}{l_i} \times 1000$, and l_i is the length of the i^{th} frame. After that, NoP-ECCP calculates a feedback value to indicate how close is the link utilization to 1 (Equation 8).

$$F_b = -K * (\bar{CE} + w * (\bar{CE} - \bar{CE_{old}})) \quad (8)$$

Where $\bar{CE_{old}}$ is the last calculated $\bar{CE}$ and K is constant and is taken to be equal 32 in order to keep F_b values of NoP-ECCP within the same range as the original ECCP.

Finally, it passes this calculated feedback value to the controller in order to execute either rate increase or rate decrease process (Equation 1) [2].

[2]NoP-ECCP use the same Rate control process as ECCP

V. PERFORMANCE EVALUATION

To evaluate the performance of the proposed mechanism, we have built a simulator for ECCP and NoP-ECCP using OMNEST [14]. The simulation parameters are given in table I. Two experiments are conducted in this simulation environment. The first simulation aims to evaluate the performance of NoP-ECCP and then compare the results with ECCP (Section V-A). The second experiment is presented in Section V-B to evaluate the performance and the scalability of the proposed mechanism on larger network.

TABLE I. SIMULATION PARAMETERS

Data Senders Parameters	
Frame size	Normal distribution $(avg = 1000, \sigma = 150)$
Min Frame size	200 Bytes
Max Frame size	1500 Bytes
ECCP Parameters (As shown in [8])	
Number of probe frames	$N = 32$
Minimum probe rate	50 Mbps
Maximum probe rate	$0.5 * AvT * SR$
Time between each train	1 ms
Strain threshold	ϵ_t = 0.003
Equilibrium available bandwidth ratio	$A_{eq} = 50\%$
Rate control timer	T = 3 ms
Rate control byte counter	$BC = 450$ KBytes
Kalman filter system noise	$Q = \begin{pmatrix} 0.00001 & 0.0 \\ 0.0 & 0.01 \end{pmatrix}$
Kalman filter measurement error	$P = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 100.0 \end{pmatrix}$
NoP-ECCP Parameters	
Number of probe frames	$N = 32$
Delay Threshold	$D_{th} = 10\mu s$
Rate control timer	T = 1 ms
Rate control byte counter	$BC = 450$ KBytes
Receiver byte counter	$BC_r = 300$ KBytes
Data history length	$H = 700$ frames

A. Experiment I - Evaluation and Comparison

In this experiment, we ran the simulation using the network topology depicted in Fig. 6. This topology is based on 3 line cards: line card 0, line card 1 and line card 2 connected respectively to port 0, 1 and 2 of an Ethernet switch. Also, three hosts connected to each line card to generate traffic interacting with the simulated router. All the links of the network topology between hosts, line cards and the switch have 10 Gbps capacity.

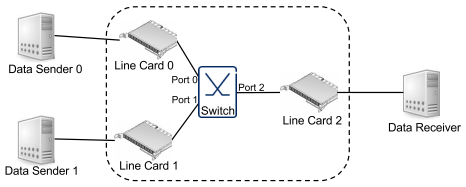


Fig. 6. ECCP simulation setup

Data senders are continuously transmitting frames towards data receiver at different rates (See Fig. 7), potentially creating congestion at port 2 of the switch. The simulations are driven to measure the line cards' sending rates, the congested link

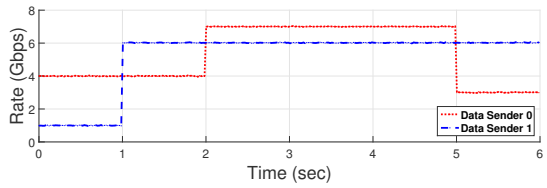
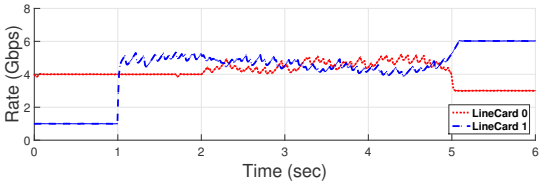
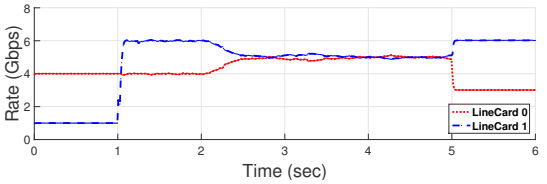


Fig. 7. Data Senders' Desired Data Rates (DRs)



(a) Line Cards' Sending Rates (ECCP)



(b) Line Cards' Sending Rates (NoP-ECCP)

Fig. 8. Simulation Results (Sending Rates)

cross traffic, and the congested port queue length. Figures 8a and 8b depict the resulted sending rate at the output ports of the line cards for ECCP and NoP-ECCP respectively. The figures show that NoP-ECCP has a better performance in terms of rate stability and fairness. One can notice that at $1 < t < 2$, NoP-ECCP does not throttle the sending rates because NoP-ECCP does not use any probe traffic which maximizes link utilization and does not cause congestion in this period (Fig. 8b).

Fig. 9 depicts cross traffic rates. It shows that the cross traffic controlled by ECCP within the congested period (t = 2 sec : t = 5 sec) never reaches the maximum link capacity. This limitation meets the restriction of ECCP to keeping the cross traffic around $(100\% - AvT * A_{eq})$ (95%) of the maximum link capacity. While in NoP-ECCP, this restriction is no longer needed and the cross traffic is equal to the maximum link capacity while having lower queue length (Fig. 10). Thus NoP-ECCP outperforms ECCP in terms of link utilization.

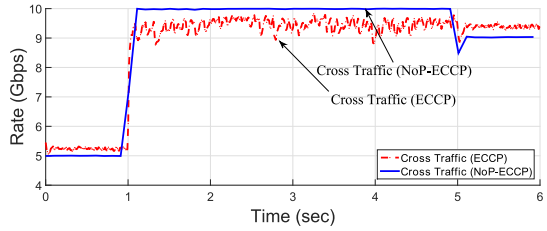


Fig. 9. Congested Link Cross Traffic

Fig. 10 shows the resulted queue length at the output queue of the congested port (port 2 in the switch). It shows that NoP-ECCP outperforms ECCP in keeping the queue length close to zero and eliminating the queue fluctuation. Thus data experience minimum latency within NoP-ECCP system boundaries.

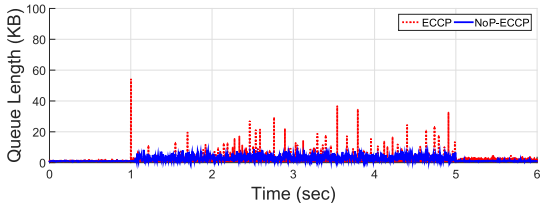


Fig. 10. Congested Port Queue Length

B. Experiment II - Evaluation in a Larger Network

In order to test the scalability of NoP-ECCP, we ran the simulation using a dumbbell topology of 8 hosts on each side (Fig. 11). Furthermore, a comparison between NoP-ECCP and ECCP is presented while ECCP is modified to use probe information to estimate link utilization (as in section III). In this simulation, data senders send data with rates shown in Fig. 12 causing congestion between t = 0.5 sec and t = 3.5 sec.

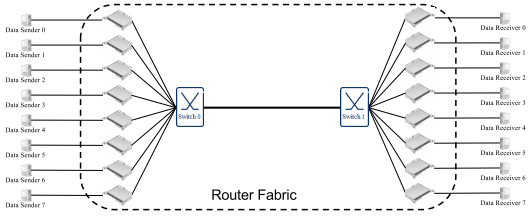


Fig. 11. ECCP Simulation Network

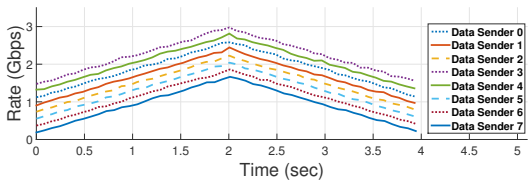


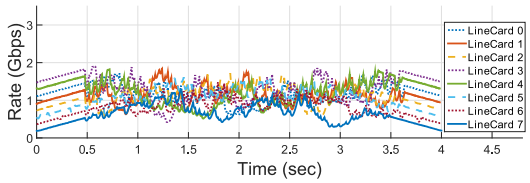
Fig. 12. Data Senders' Desired Data Rates DRs

Figures 13a and 13b show the sending rate at the output ports of the line cards for ECCP and NoP-ECCP respectively. The figures depict that both mechanisms succeeded in controlling hosts' sending rate. In addition, it shows that NoP-ECCP provides better fairness and stable performance comparing to ECCP.

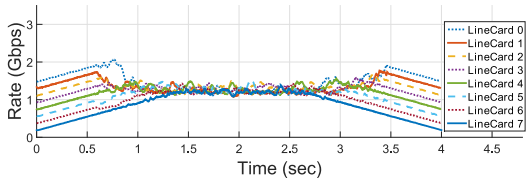
Fig. 14 depicts the resulted cross traffic of the congested link. It shows that ECCP never reaches the maximum link capacity (which is its original objective to keep a percentage of link capacity available) while NoP-ECCP reaches the maximum link utilization by eliminating the probes. Fig. 15 shows the resulted queue length at the output queue of the congested port. It shows that our proposed mechanism succeeds in keeping the queue length close to zero and therefore data experience the minimum delay within NoP-ECCP boundaries.

VI. Conclusion

In this paper, we present several enhancements to ECCP namely (1) estimating link utilization rather than available



(a) Line Cards' Sending Rates (ECCP)



(b) Line Cards' Sending Rates (NoP-ECCP)

Fig. 13. Simulation Results (Sending Rates)

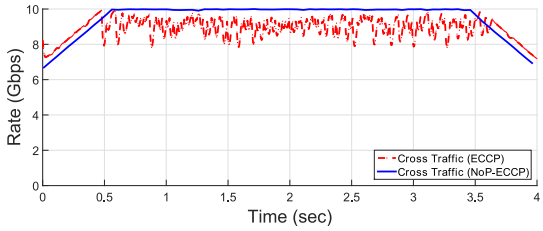


Fig. 14. Congested Link Cross Traffic

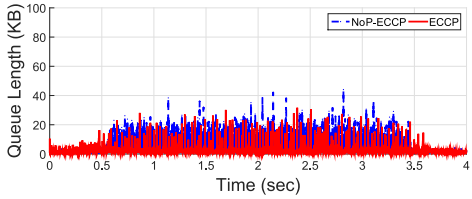


Fig. 15. Congested Port Queue Length

bandwidth and (2) using data as probes (No probes) NoP-ECCP. In these enhancements, we use link utilization estimation technique based on calculating the number of packets that experience minimum delay. Then, we propose NoP-ECCP which exploits the link utilization information to estimate an indication of congestion and uses it to control hosts' sending rate. Link utilization estimation technique removes the restriction of having a fixed inter-packet interval that exists in ECCP. Therefore, we were able to use data traffic for implementing and performing the computations required by the NoP-ECCP algorithm. We have implemented NoP-ECCP in OMNEST simulator and we have executed several experiments to evaluate the performance of our proposed solution. Through the simulation results, we have shown that NoP-ECCP has a reduced queue fluctuation and a better link utilization compared to ECCP. In addition, NoP-ECCP reduces the complexity of the algorithm implementation.

For future work, evaluating NoP-ECCP scalability in larger and various network topologies will be performed within the simulator. We also aim to implement NoP-ECCP in real lab environment using DPDK (Data Plane Development Kit) [15]. In addition, a mathematical analysis of NoP-ECCP model with the object to prove its stability is ongoing.

ACKNOWLEDGMENT

This work is supported by Ericsson Research, the Fonds de recherche du Quebec - Nature et technologies (FRQNT) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] A. Bachmutsky, *System design for telecommunication gateways*. John Wiley & Sons, 2011.

[2] D. Medhi, *Network routing: algorithms, protocols, and architectures*. Morgan Kaufmann, 2010.

[3] "IEEE standard for local and metropolitan area networks– virtual bridged local area networks amendment 13: Congestion notification," *IEEE Std 802.1Qau-2010 (Amendment to IEEE Std 802.1Q-2005)*, pp. c1–119, April 2010.

[4] I. 802.1. (2011) 802.1qau - congestion notification. [Online]. Available: http://www.ieee802.org/1/pages/802.1au.html

[5] R. Pan, B. Prabhakar, and A. Laxmikantha, "Qcn: Quantized congestion notification," *IEEE802*, vol. 1, 2007.

[6] I. 802.1. (2013) The Data Center Bridging (DCB) task group (tg). [Online]. Available: http://www.ieee802.org/1/pages/dcbridges.html

[7] I. Network. (2010) Data Center Bridging (DCB) Congestion Notification (802.1qau). [Online]. Available: http://blog.ipspace.net/2010/11/data-center-bridging-dcb-congestion.html

[8] M. Bahnasy, A. Beliveau, B. Alleyne, B. Boughzala, C. Padala, K. Idoudi, and H. Elbiaze, "Using ethernet commodity switches to build a switch fabric in routers," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*. IEEE, 2015, pp. 1–8.

[9] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-E. Mangs, B. Melander, and M. Bjorkman, "Real-time measurement of end-to-end available bandwidth using kalman filtering," in *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*. IEEE, 2006, pp. 73–84.

[10] J.-Y. Le Boudec, "Rate adaptation, congestion control and fairness: A tutorial," *Web page, November*, 2005.

[11] M. Vojnovic, J.-Y. Le Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2000, pp. 1303–1312.

[12] A. Cabellos-Aparicio, F. J. Garcia, and J. Domingo-Pascual, "A novel available bandwidth estimation and tracking algorithm," in *Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE*. IEEE, 2008, pp. 87–94.

[13] M. Neginhal, K. Harfoush, and H. Perros, "Measuring bandwidth signatures of network paths," in *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. Springer, 2007, pp. 1072–1083.

[14] O. Community. (2014) OMNEST - High-Performance Simulation. [Online]. Available: http://http://www.omnest.com/

[15] Intel. (2014) Dpdk: Data plane development kit. [Online]. Available: http://dpdk.org/