Increasing Robustness of XCP (eXplicit Control Protocol) for Dynamic Traffic

Yusuke Sakumoto, Hiroyuki Ohsaki, and Makoto Imase
Graduate School of Information Science and Technology
Osaka University, Suita, Osaka 565-0871, Japan
{y-sakumt,oosaki,imase}@ist.osaka-u.ac.jp

***Abstract*—XCP (eXplicit Control Protocol) has been proposed as an efficient transport protocol for a wide-area and high-speed network. XCP is a transport-layer protocol that performs congestion control using explicit feedback from routers. In the literature, many simulation-based performance studies of XCP has been performed. However, the effect of traffic dynamics on the XCP performance has not been investigated. In this paper, through simulation experiments, we first show that XCP has the following problems: (1) utilization of the bottleneck link is lowered due to XCP traffic dynamics, and (2) in environment where non-XCP traffic and XCP traffic coexist, control of XCP becomes unstable. We then propose XCP-IR (XCP with Increased Robustness) that operates efficiently even for dynamic traffic. Through simulation experiments, we show that XCP-IR operates efficiently even for dynamic traffic.**

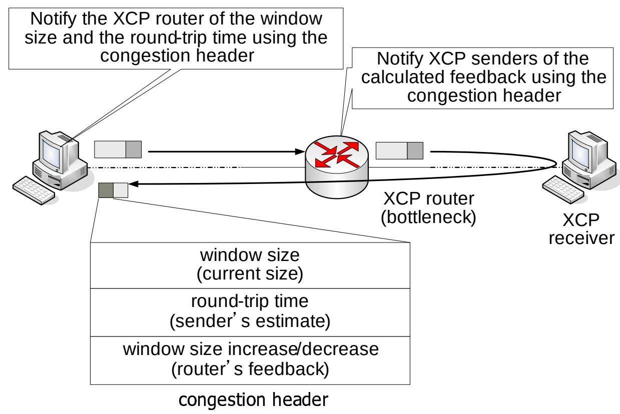

Fig. 1: Overview of XCP congestion control using congestion header

I. Introduction

XCP (eXplicit Control Protocol) has been proposed as an efficient transport-layer communication protocol in a wide-area and high-speed network [1]. XCP is a transport-layer communication protocol that controls congestion using the explicit feedback from a router. As a method of feeding back congestion status of a router to source hosts, ECN (Explicit Congestion Notification) [2] has been used. XCP is a type of ECN; i.e., XCP feeds back congestion status from a router to source hosts using multiple bits. XCP performs a window-based flow control similar to TCP Reno. XCP adjusts the window size appropriately by exchanging congestion information between a router and source hosts. In XCP, a "*congestion header*" added to the header of a packet is used for exchanging congestion information. The overview of congestion control using the congestion header between a source host and a router is illustrated in Fig. 1. The XCP sender notifies the XCP router of its current window size and the measured round-trip time using a congestion header. From the information notified by the XCP sender and current status of the router usage, the XCP router calculates the appropriate amount of window size increase/decrease and notifies the XCP sender of it using the congestion header. Feedback from the XCP router to the XCP sender is performed via the ACK packet returned by the XCP receiver.

In the literature, many simulation-based performance studies of XCP have been performed [1], [3]–[6]. In [1], the authors show that XCP operates more efficiently than TCP Reno in a high-speed network. On the contrary, several problems of XCP are pointed out in [4]–[6]. In [4], it is pointed out that XCP performance degrades in a network environment with a high bit-error rate. In [5], the problem of the XCP performance degradation when many ACK packets are discarded is pointed out. The authors of [6] point out the problem that XCP performance degrades in a network environment with shared access media such as radio communications. The solutions for these problems are also proposed in [4]–[6].

However, the effect of traffic dynamics on the XCP performance has not been investigated. As we will discuss in Section II, XCP has the following problems: (1) utilization of the bottleneck link is lowered due to XCP traffic dynamics, and (2) in environment where non-XCP traffic and XCP traffic coexist, control of XCP becomes unstable. To operate XCP efficiently under diverse network environments, such problems of XCP regarding traffic dynamics have to be solved. Note that the solution for the latter problem (2) is proposed in [1], [7]. However, since it is a simple mechanism of statically dividing the link bandwidth to XCP traffic and non-XCP traffic, there is a problem that the network resources cannot be utilized effectively.

In this paper, we therefore propose XCP-IR (XCP with Increased Robustness) that operates efficiently for dynamic traffic. XCP-IR prevents instability of the XCP control due to non-XCP traffic dynamics while preventing loss of the bottleneck link utilization due to XCP traffic dynamics. In this paper, we evaluate the effectiveness of XCP-IR by simulation.

By investigating steady state, stability, and transient state performances of XCP-IR, we show that XCP-IR operates efficiently even for dynamic traffic.

The organization of this paper is as follows. First, Section II briefly explains the algorithm of XCP. Section III explains the basic ideas and the operation algorithm of XCP-IR. Section IV evaluates the effectiveness of XCP-IR by simulation. Finally, the conclusion and future work of this paper are summarized in Section V.

II. XCP (EXPLICIT CONTROL PROTOCOL)

A. Overview

In this section, the algorithm of XCP is briefly summarized. Refer to [1] for details of the XCP algorithm.

- XCP Sender Algorithm
  First, we explain the algorithm of an XCP sender. At the time of packet transmission, an XCP sender stores its estimated round-trip time, its current window size, and the initial value of the feedback value (i.e., the amount of window size increase requested by the XCP sender) in the congestion header of the packet. Moreover, when an XCP sender receives an ACK packet, the XCP sender updates its window size and re-calculates the estimated round-trip time. The window size is updated based on the amount of window size increase/decrease notified by XCP routers. The window size is set to the sum of the current window size and the window-size increase/decrease. The round-trip time is re-calculated using the algorithm similar to that of TCP Reno. The difference between XCP and TCP Reno regarding the algorithm for calculating the round-trip time is that XCP calculates with finer granularity than TCP Reno.
- XCP Router Algorithm
  Next, we explain the algorithm of an XCP router. The control mechanism of an XCP router is composed of the *efficiency controller*, which tries to maximize utilization of the link bandwidth, and the *fairness controller*, which tries to realize fairness among competing XCP flows. The efficiency controller and the fairness controller are invoked every the average round-trip time of all XCP flows. The efficiency controller estimates the amount of total rate increase/decrease for all XCP flows. The fairness controller then calculates the amount of rate increase/decrease for each XCP flow. An XCP router calculates the *feedback value* based on the amount of rate increase/decrease calculated by the fairness controller and information stored in the congestion header of arriving packets. In what follows, algorithms of the efficiency controller and the fairness controller are briefly explained.
  The efficiency controller calculates the *aggregate feedback value* ϕ (i.e., the amount of total rate increase/decrease for all XCP flows) from the packet arrival rate to the XCP router and the current queue length as

  $$\phi = \alpha\, d\, (C - A) - \beta\, Q, \qquad (1)$$

  where d is the average round-trip time of XCP flows accommodated in the XCP router, C is the output link bandwidth of the XCP router, A is the packet arrival rate at the XCP router, Q is the minimum queue length observed during the average round-trip time, and α and β are control parameters of the XCP router. The efficiency controller controls so that: (1) the link bandwidth is fully utilized, and (2) the number of packets in the buffer becomes zero.
  The fairness controller splits the aggregate feedback value ϕ to all XCP flows based on AIMD (Additive Increase and Multiplicative Decrease). Namely, when $\phi \geq 0$, the fairness controller equally distributes ϕ to all XCP flows . On the contrary, when $\phi < 0$, the fairness controller distributes ϕ to all XCP flows, such that the bandwidth allocation to each XCP flow is proportional to its throughput. The fairness controller introduces the shuffle traffic h defined by the following equation so that the aggregate feedback ϕ converges to 0 when the throughput of each XCP flow is equal.

  $$h \;=\; [\gamma\, T - |\phi|]^{+} \qquad (2)$$

  where γ is a control parameter of an XCP router, and T is the total size of packets arrived during the average round-trip time of XCP flows.

B. XCP Problems for dynamic traffic

In what follows, we discuss problems of XCP for traffic dynamics. We consider both XCP traffic dynamics and non-XCP traffic (e.g., TCP and UDP) dynamics. We show that XCP has the following problems: (1) utilization of the bottleneck link is lowered due to XCP traffic dynamics, and (2) in the environment where non-XCP traffic and XCP traffic coexist, control of XCP becomes unstable.

First, XCP traffic dynamics is considered. The evolution of the utilization of the bottleneck link (i.e., link utilization measured for every 10 [ms]) when using the same simulation model as [1] is shown in Fig. 2. The amount of XCP traffic is fluctuated by changing the number of active XCP flows in a network similarly to [1]. In this simulation, 10 XCP flows are activated at $t = 0$, and 100 XCP flows are activated at $t = 4$, 100 XCP flows are deactivated at $t = 8$, and 10 XCP flows are deactivated at $t = 10$. Figure 2 shows that the utilization of the bottleneck link degrades significantly when XCP flows terminate their transfer. This is because XCP controls the queue length of a router to become zero. Since the router's buffer is empty, the utilization of the bottleneck link will degrade when the amount of XCP traffic decreases.

Next, non-XCP traffic dynamics is considered. In what follows, we show that the control of an XCP router becomes unstable when non-XCP traffic increases. Simulations are performed using the same network topology as that in Section IV. The evolution of the number of packets in the XCP router's buffer (i.e., the queue length) when transmitting UDP traffic with the average transfer rate of (10, 20, 30 [Mbit/s]) is shown in Fig. 3. Figure 3 shows that the queue length of the XCP

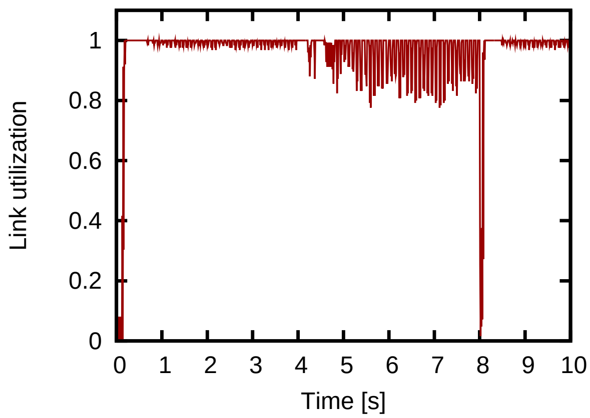

Fig. 2: Evolution of the utilization of the bottleneck link when changing the number of active XCP flows

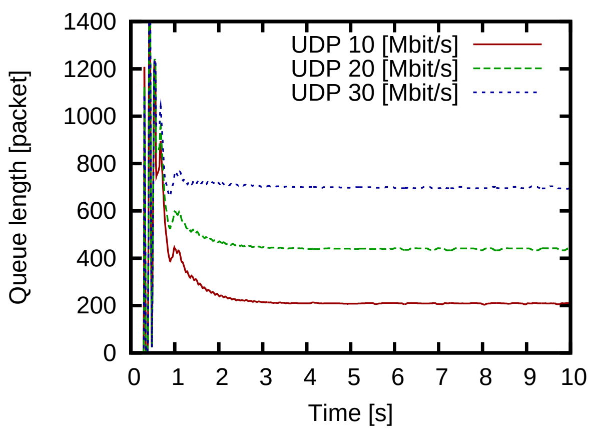

Fig. 3: Evolution of the queue length in XCP when transmitting non-XCP traffic (UDP traffic)

router becomes large when non-XCP traffic increases. This is because the XCP router performs its control by assuming that the available bandwidth of its output links is known. If the amount of non-XCP traffic increases, the XCP router will be overloaded and many packets will be queued at the buffer.

III. XCP-IR (XCP with Increased Robustness)

In this section, the operation algorithm of XCP-IR and the basic ideas for improving the robustness of XCP for traffic dynamics are explained.

A. Basic ideas

First, the idea for preventing degradation of the link utilization due to XCP traffic dynamics is explained. To prevent degradation of the link utilization due to XCP traffic dynamics, a router's buffer is utilized effectively. Namely, XCP traffic dynamics is absorbed at the router's buffer. Specifically, a certain amount of packets are always stored in the XCP router's buffer. Thereby, degradation of the link utilization due to XCP traffic dynamics can be prevented, thanks to the packets stored in the buffer.

Next, the idea for preventing instability of the XCP router control due to increase in non-XCP traffic is explained. To prevent instability of the XCP router control due to increase in non-XCP traffic, the XCP router just estimates the available bandwidth of XCP traffic correctly. For this purpose, an XCP router measures the arrival rate of non-XCP traffic. The XCP router estimates the available bandwidth of XCP traffic by subtracting the arrival rate of non-XCP traffic from the physical link bandwidth. The XCP router distributes the available bandwidth for XCP traffic. Since the XCP router can correctly calculate the bandwidth assigned to each XCP flows, stable control can be realized regardless of non-XCP traffic dynamics.

These improvements are realizable only by changing the control algorithm of an XCP router. Namely, it is not necessary to change either an XCP sender nor a packet format. Hence, the burden of deploying XCP-IR into a real network is quite low.

B. XCP-IR algorithm

In what follows, the algorithm of XCP-IR is explained. With XCP-IR, only the method of calculating the aggregate feedback value ϕ (Eq. (1)) is different from that of XCP. XCP-IR calculates the aggregate feedback value ϕ as

$$\phi = \alpha\, d\, \{(C - A_N) - A\} - \beta\, (Q - Q_T), \tag{3}$$

where A_N is the arrival rate of non-XCP traffic, and Q_T is the target value of a queue length.

Q_T is a control parameter (i.e., the target value of the the queue length) introduced to prevent degradation of the link utilization due to XCP traffic dynamics. Thus, Q_T packets are always stored in the XCP router's buffer. Thereby, degradation of link utilization can be prevented even for XCP dynamic traffic.

A_N is an internal variable introduced to prevent instability of the XCP control due to the increase in non-XCP traffic. An XCP router calculates A_N as

$$A_N = \frac{T}{d}, \tag{4}$$

where T is the total amount of non-XCP packets arrived at the XCP router during the control interval d of the XCP router. By changing the available bandwidth of XCP traffic from C to $C - A_N$, XCP-IR can absorb XCP router's temporaly overload when non-XCP traffic increases.

In what follows, we discuss characteristics of XCP-IR. Specifically, we discuss how steady state stability, and transient state performances are affected by changing the calculation method of the aggregate feedback ϕ from Eq. (1) to Eq. (3). First, the round-trip times of XCP flows in XCP and XCP-IR are compared. The round-trip time in XCP-IR is larger than that in XCP. XCP-IR controls so that the queue length becomes Q_T. In XCP-IR, the queuing delay of Q_T/C therefore occurs in the router.

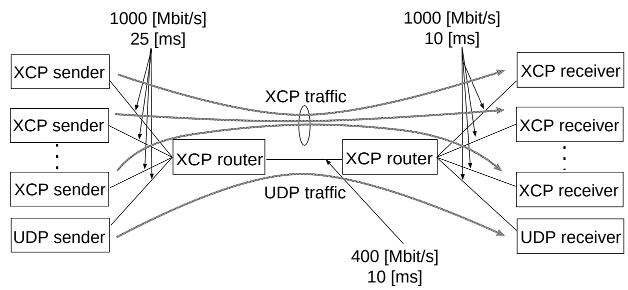

Fig. 4: Network topology used in simulation

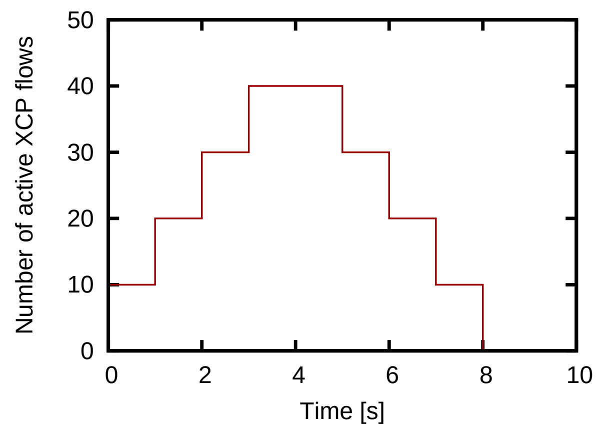

Fig. 5: The number of active XCP flows changed in simulation

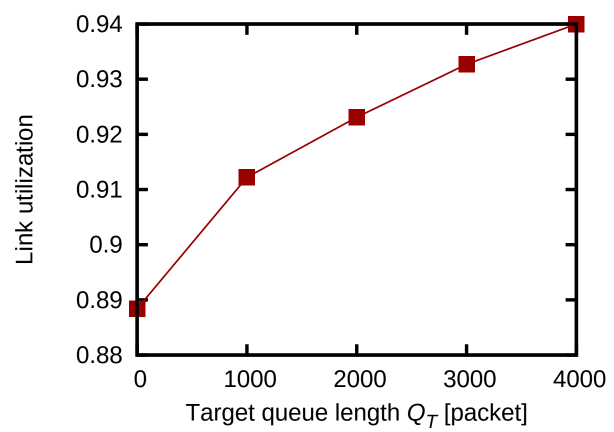

Fig. 6: Utilization of the bottleneck link for a different target value of the queue length Q_T (XCP corresponds to the case of Q_T = 0)

Stability and transient performance of XCP and XCP-IR are investigated using the analytic approach in [8]. Due to space limitation, although derivation and results are not indicated, by investigating stability and transient state performance of XCP-IR, we found that the stability and the transient state performance around the equilibrium point were same as that of XCP. Namely, we found that the stability and the transient state performance around the equilibrium point did not change, even if the calculation method of the aggregate feedback value was changed from Eq. (1) to Eq. (3).

IV. Simulation

Figure 4 shows the network topology used in simulation. Multiple XCP flows and the single UDP flow share the single bottleneck link. Unless explicitly stated, in the following simulations, we use the following parameters: the control parameters of XCP routers (α, β) are (0.4, 0.226), γ is 0.1, and the packet size is 1,000 [byte]. Note that we used ns-2 [7] version 2.28 with modification for XCP-IR.

First, we investigate the utilization of the bottleneck link in XCP and XCP-IR. In simulation, dynamic XCP traffic was generated by changing the number of active XCP flows as shown in Fig. 5. Namely, 10 XCP flows are activated every 1 [s] after $t = 0$, and 10 XCP flows are deactivated every 1 [s] after $t = 5$. To focus on the effect of dynamic XCP traffic, background UDP traffic is not generated. The utilization of the bottleneck link when changing the target value of the queue length Q_T is shown in Fig. 6. Note that the result of XCP corresponds to the case of $Q_T = 0$ [packet]. Figure 6 shows that the utilization of the bottleneck link increases as the target value of the queue length Q_T increases. For instance, when the results of XCP and XCP-IR with Q_T= 4,000 [packet] are compared, one can find that link utilization in XCP-IR is approximately 5% higher than they in XCP.

Next, we investigate the stability of XCP-IR. In Section III, the stability of XCP-IR around the equilibrium point was investigated by using the analytic approach in [8].

In the following simulations, we investigate the stability after starting transfer from the initial state rather than the stability around the equilibrium point. The stability region of the control parameter (α, β) in XCP and XCP-IR is shown in Fig. 7. We performed many simulations with different sets the control parameters (α, β) $(0.1 \leq \alpha \leq 1.5,\ 0.1 \leq \beta \leq 5.0)$. This figure means that operation of XCP or XCP-IR was stable, when a set of control parameters (α, β) is in the stability region (under the boundary lines). Note that 20 XCP flows started their transfers simultaneously in simulation. Background UDP traffic is not generated to focus on the effect of dynamic XCP traffic.

Figure 7 shows that the stability region of XCP-IR is smaller than that of XCP. Namely, the stability of XCP-IR is lower than that of XCP when an XCP flow starts transfer from the initial state. However, with the recommended parameter configuration of $(\alpha\ \beta)$ = (0.4, 2.26) [1], both XCP and XCP-IR operate stably. Hence, as long as the recommended parameter configuration is used, there is no problem in the stability of XCP-IR.

Next, we investigate the transient state performance of XCP-IR. Similarly to the previous case, we investigate the stability after starting transfer from the initial state rather than the stability around the equilibrium point. The settling time of the window size of a XCP flow while changing the target value

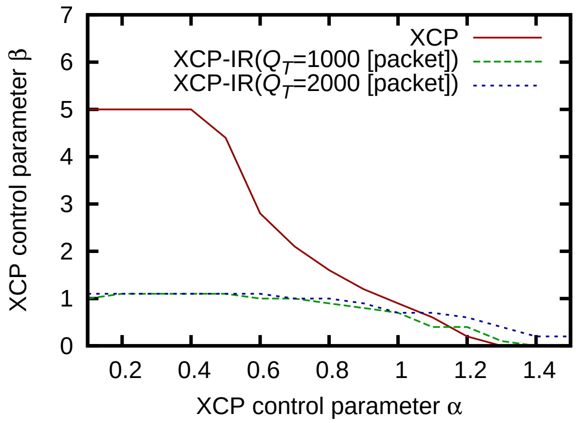

Fig. 7: Stability region when an XCP flow starts transfer from the initial state (XCP control is stable when (α, β) is the boundary line)

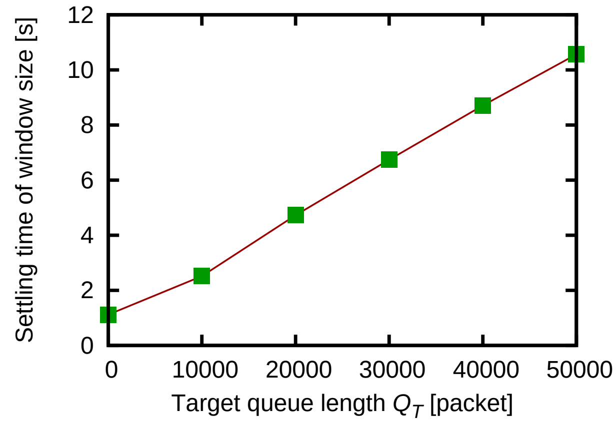

Fig. 8: Settling time of the window size when an XCP flow starts transfer from the initial state

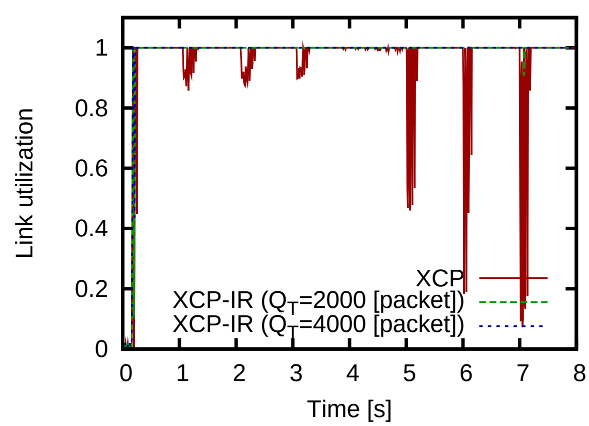

Fig. 9: Evolution of utilization of the bottleneck link for a different target value of the queue length Q_T

of the queue length Q_T from 0 to 50,000 [packet] is shown in Fig. 8. The settling time of the window size is defined as the time for the window size to be stabilized settled within $\pm 5\%$ of the equilibrium value (i.e., the window size in steady state). Note that only one XCP flow starts transfer in this simulation. Background UDP traffic is not generated to focus on the effect of dynamic XCP traffic.

Figure 8 shows that the settling time of the window size of the XCP flow increases as the target value of the queue length Q_T becomes large. The cause of increase in the settling time can be explained as follows. To prevent degradation of link utilization due to dynamic XCP traffic, XCP-IR is controlled so that Q_T packets are queued at the XCP router's buffer. Storing a certain amount of packets at the XCP router's buffer results in non-negligible queuing delay, leading increase in the round-trip time of XCP flows. Since the XCP router controls every the average round-trip time, when the round-trip time of an XCP flow increases, the control frequency of an XCP router becomes low. Consequently, the responsiveness of the XCP router degrades and the settling time of the XCP flow increases. From these observations, we find that the target value of the queue length Q_T should be determined taking account of trade-offs between robustness and responsiveness.

As mentioned above, the effectiveness of XCP-IR is dependent on the target value of the queue length Q_T. Therefore, in what follows, we discuss how the target value of the queue length Q_T should be configured to maximize the XCP-IR performance. An XCP router controls every the average round-trip time d. Hence, when the number of active XCP flows decrease rapidly, the time of d is taken for the XCP router to notify XCP senders of the re-calculated feedback value. Let us assume that M XCP flows terminate their transfers among N XCP flows. In this case, $d\,C\,M/N$ packets are sent from the buffer until XCP senders are notified of the re-calculated feedback value. By denoting the two-way propagation delay of an XCP flow by τ, the queue length in steady state is given by Q_T, and the round-trip time is given by $\tau + Q_T/C$. To prevent degradation of the utilization of the bottleneck link, it is necessary to satisfy $d\,C\,M/N \leq Q_T$;

$$Q_T \geq \frac{\tau\,C\,M}{N - M}. \tag{5}$$

In what follows, the validity of Eq. (5) is investigated by simulation. The number of active XCP flows was changed as shown in Fig. 5. Evolutions of the utilization of the bottleneck link in XCP and XCP-IR are shown in Fig. 9 when changing the propagation delay of the access link between an XCP sender and the XCP router to 5 [ms]. In this case, the target value of the queue length Q_T that satisfying Eq. (5) is $Q_T \geq 2,500$ [packet]. Figure 9 shows that degradation of the utilization of the bottleneck link can be prevented when Q_T is configured to a larger value than 2,500 [packet] (e.g., Q_T=4,000 [packet]).

Finally, we investigate the robustness of XCP-IR to non-XCP traffic. The evolution of queue length in XCP-IR when transmitting the UDP traffic with the average transfer rate of

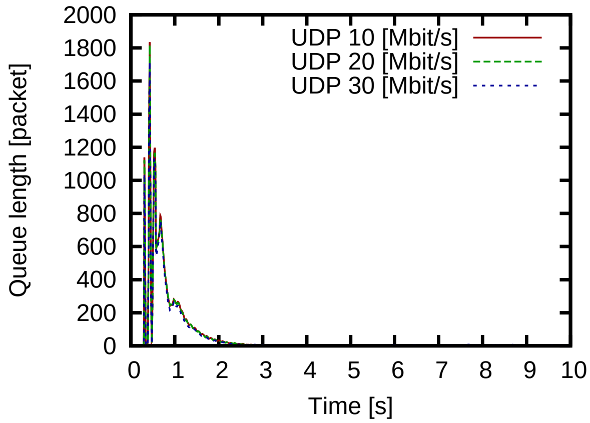

Fig. 10: Evolution of the queue length in XCP-IR when transmitting non-XCP traffic (UDP traffic)

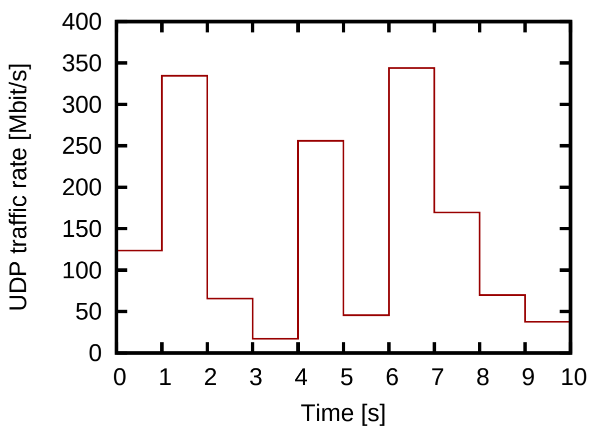

Fig. 11: Average transfer rate of background UDP traffic changed in simulation

10, 20, and 30 [Mbit/s] is shown in Fig. 10. To focus on the tolerance to non-XCP traffic dynamics, the target value of the queue length is set to $Q_T = 0$ [packet]. By comparing Figs. 3 (XCP) and 10 (XCP-IR), one can find that XCP-IR can stabilize the queue length regardless of the average transfer rate of background UDP traffic.

To investigate the characteristics of XCP-IR under severe environment, the average transfer rate of background UDP traffic is changed suddenly every second as shown in Fig. 11. Evolutions of the queue length in XCP and XCP-IR are shown in Fig. 12. Figure 12 shows that control of XCP becomes unstable because the transfer rate of background UDP traffic fluctuates rapidly. On the contrary, the control of XCP-IR is almost stable regardless of the amount of background UDP traffic.

V. Conclusion

In this paper, we have proposed XCP-IR (XCP with Increased Robustness) that operates efficiently even for dynamic

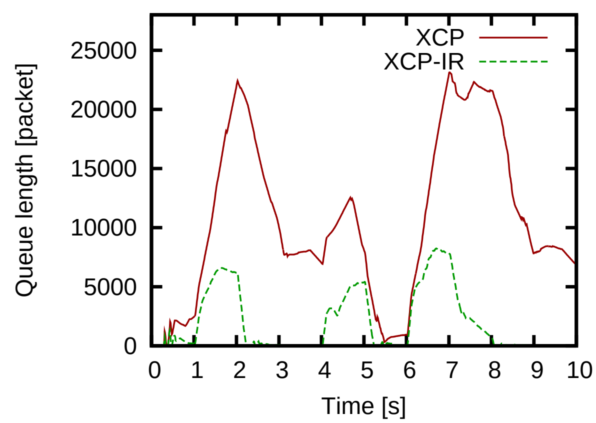

Fig. 12: Evolution of the queue length when suddenly changing the transfer rate of background UDP traffic

traffic. XCP-IR prevents instability of the XCP control due to non-XCP traffic dynamics while preventing degradation of the bottleneck-link utilization due to XCP traffic dynamics. By several simulations, we have shown that XCP-IR operated efficiently even for dynamic traffic. Namely, we found that link utilization of XCP-IR is approximately 5% higher than that of XCP, and that XCP-IR can stabilize the queue length regardless of the average transfer rate of background UDP traffic.

As future work, we are planning to evaluate performance of XCP-IR in more realistic network environments. We are also planning to examine the optimal control parameter configuration for maximizing the XCP-IR performance.

Acknowledgements

The authors would like to thank Prof. Masayuki Murata for his fruitful suggestions.

References

[1] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings of ACM SIGCOMM 2002*, vol. 32, Aug. 2002, pp. 89–102.

[2] K. Ramakrishnan, S. Floyd, and D. B. Rosen, "The addition of explicit congestion notification (ECN) to IP," *Request for Comments (RFC) 3168*, Sept. 2001.

[3] D. Katabi, "XCP's performance in the presence of malicious flows," in *Proceedings of Second International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2004)*, Feb. 2004.

[4] K. Zhou, K. L.Yeung, and V. O. Li, "P-XCP: A transport layer protocol for satellite IP networks," in *Proceedings of IEEE Globecom 2004*, vol. 5, Nov. 2004, pp. 2707–2711.

[5] D. Lopez-Pacheco and C.Pham, "Robust transport protocol for dynamic high-speed networks: enhancing the XCP approach," in *Proceedings of IEEE MICC ICON 2005*, vol. 1, Aug. 2005.

[6] F. Abrantes and M. Ricardo, "XCP for shared-access multi-rate media," *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 27–38, July 2006.

[7] "The network simulator – ns2," available at http://www.isi.edu/nsnam/ns/.

[8] Y. Sakumoto, H. Ohsaki, and M. Imase, "On XCP stability in a heterogeneous network," *Proceedings of 12th IEEE Symposium on Computers and Communications(ISCC'07)*, pp. 531–537, July 2007.