PAPER

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

Yusuke SAKUMOTO[†a)], ***Student Member***, **Hiroyuki OHSAKI**[†b)], ***and*** **Makoto IMASE**[†c)], ***Members***

SUMMARY In this paper, we reveal inherent robustness issues of XCP (eXplicit Control Protocol), and propose extensions to XCP for increasing its robustness. XCP has been proposed as an efficient transport-layer protocol for wide-area and high-speed network. XCP is a transport-layer protocol that performs congestion control based on explicit feedback from routers. In the literature, many performance studies of XCP have been performed. However, the effect of traffic dynamics on the XCP performance has not been fully investigated. In this paper, through simulation experiments, we first show that XCP has the following problems: (1) the bottleneck link utilization is lowered against XCP traffic dynamics, and (2) operation of XCP becomes unstable in a network with both XCP and non-XCP traffic. We then propose XCP-IR (XCP with Increased Robustness) that operates efficiently even for dynamic XCP and non-XCP traffic.

key words: *XCP (eXplicit Control Protocol), traffic dynamics, robustness, stability, congestion control*

1. Introduction

XCP (eXplicit Control Protocol) has been proposed as an efficient transport-layer protocol in a wide-area and high-speed network [1]. XCP controls congestion using the explicit feedback from routers. As a method of sending router's congestion indication 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 routers to source hosts using multiple bits. XCP performs a window-based flow control similar to that of TCP Reno [3]. XCP adjusts the window size appropriately by receiving congestion information from routers.

In XCP, a *congestion header* added to the header of a packet is used for carrying congestion information. Congestion control using the congestion header is illustrated in Fig. 1. The XCP sender notifies the XCP router of its current window size and the measured round-trip time through 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 through the congestion header. Feedback from XCP routers to the XCP sender is performed via ACK packets returned by the XCP receiver.

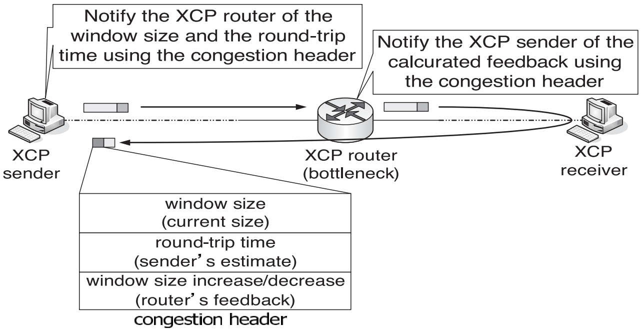

Notify the XCP router of the window size and the round-trip time using the congestion header
Notify the XCP sender of the calcurated feedback using the congestion header
XCP sender
XCP router (bottleneck)
XCP receiver
window size (current size)
round-trip time (sender' s estimate)
window size increase/decrease (router' s feedback)
congestion header

Fig. 1 Overview of XCP congestion control using congestion header; the XCP sender notifies the XCP router of its current window size and the measured round-trip time, and the XCP router notifies the XCP sender of the appropriate amount of window size increase/decrease.

In the literature, many performance studies of XCP have been performed [1], [4]–[8]. 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 [5], [6], [8]. In [5], it is pointed out that XCP performance degrades in a network with a high bit-error rate. In [6], the problem of XCP performance degradation when many ACK packets are discarded is pointed out. The authors of [8] point out that XCP performance degrades in a network with shared access media such as radio communication. The solutions for those problems are proposed in [5], [6], [8].

However, to the best of our knowledge, the effect of traffic dynamics on XCP performance has not been fully investigated. As we will discuss in Sect. 2, XCP has the following problems: (1) the bottleneck link utilization is lowered against XCP traffic dynamics, and (2) operation of XCP becomes unstable in a network with both XCP and non-XCP traffic, which causes drastic degradation of the throughput of XCP as we will show in Sect. 5. 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 partly discussed in [1]. However, since it is a simple mechanism of statically dividing the link bandwidth to XCP and non-XCP traffic, there is a problem that the network resource cannot be efficiently utilized. Namely, neither XCP nor non-XCP traffic can fully utilize the link bandwidth even when there exist no competing traffic.

In this paper, we therefore propose XCP-IR (XCP with Increased Robustness) that operates efficiently for dynamic

Manuscript received October 27, 2009.
Manuscript revised June 5, 2010.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.
a) E-mail: y-sakumt@ist.osaka-u.ac.jp
b) E-mail: oosaki@ist.osaka-u.ac.jp
c) E-mail: imase@ist.osaka-u.ac.jp

DOI: 10.1587/transcom.E93.B.3013

XCP and non-XCP traffic. XCP-IR prevents instability of the XCP control caused by non-XCP traffic dynamics while preventing loss of the bottleneck link utilization caused by XCP traffic dynamics. In this paper, we evaluate the effectiveness of XCP-IR through extensive simulations. Through investigation of both steady state and transient state performances, we show that XCP-IR operates efficiently even for dynamic traffic.

The organization of this paper is as follows. First, Sect. 2 briefly explains the XCP protocol and its inherent robustness issues. Section 3 explains the basic ideas and the operation algorithm of XCP-IR. Sections 4 and 5 evaluate the effectiveness of XCP-IR through extensive simulations. Finally, the conclusion and future works of this paper are discussed in Sect. 6.

2. XCP (eXplicit Control Protocol)

2.1 Overview

In this section, we briefly summarize the operation algorithm of XCP. Refer to [1], [9] for details of the XCP protocol.

- XCP end host algorithm
  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.
  An XCP receiver receives the packet, and sends an ACK packet back to the XCP sender. The XCP receiver simply copies the congestion header of the received packet to the congestion header of the ACK packet.
  When an XCP sender receives the 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. More specifically, 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 [3].
- XCP router algorithm
  The control mechanism of an XCP router is composed of the *efficiency controller*, which tries to maximize the link bandwidth utilization, and the *fairness controller*, which tries to realize fairness among competing XCP flows. The efficiency controller and the fairness controller are invoked every 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, we briefly explain algorithms of the efficiency controller and the fairness controller.
  The efficiency controller calculates the *aggregate feedback value* ϕ, which is the amount of total rate increase/decrease for all XCP flows, from the packet arrival rate at the XCP router and the 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. 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 distributes the aggregate feedback value ϕ to all XCP flows based on an AIMD (Additive Increase and Multiplicative Decrease) discipline. Namely, if ϕ is positive, the fairness controller evenly allocates ϕ to all XCP flows so that the increase in throughput of all XCP flows is the same. Otherwise, the fairness controller unevenly allocates ϕ to all XCP flows so that the decrease in throughput of an XCP flow becomes proportional to its current throughput.

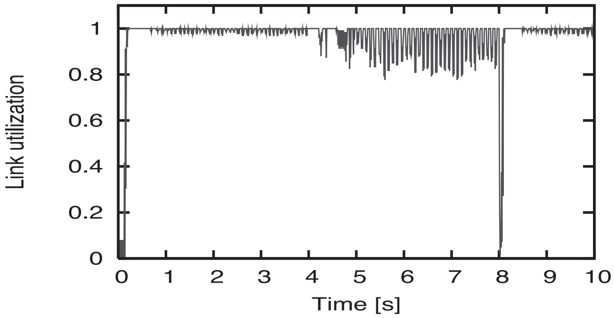


Fig. 2 Time evolution of bottleneck link utilization when changing the number of active XCP flows at t = 4 and 8; the bottleneck link utilization degrades significantly when XCP flows terminate their transfers at t = 8.

2.2 XCP Problems for Dynamic Traffic

We investigate the implications of both XCP traffic dynamics and non-XCP (e.g., TCP and UDP) traffic dynamics on the XCP protocol.

First, we consider XCP traffic dynamics. The time evolution of the bottleneck link utilization (i.e., link utilization measured for every 10 [ms]) for the same simulation model with [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. Also, 100 XCP flows are deactivated at t = 8.

Figure 2 shows that the bottleneck link utilization degrades significantly when XCP flows terminate their trans-

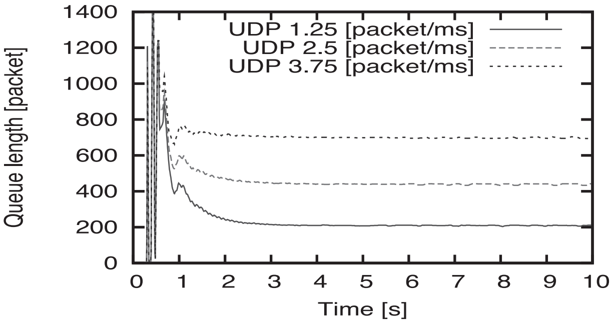


Fig. 3 Time evolution of queue length at XCP router when transmitting non-XCP (UDP) traffic; the queue length of the XCP router becomes large as the amount of non-XCP traffic increases.

fers at $t = 8$. This is because XCP controls the queue length of a router to become zero. Since the router's buffer is almost always empty, the bottleneck link utilization will degrade even when the amount of incoming XCP traffic slightly decreases. In a network with rapid XCP traffic dynamics like realistic XCP traffic dynamics, this phenomenon leads low bottleneck link utilization. As we will see in Sect. 5.1, this phenomenon becomes problematic for realistic XCP traffic dynamics.

Second, we consider non-XCP traffic dynamics. Simulations are performed using the same network topology with that in Sect. 4. The time 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 1.25, 2.5, and 3.75 [packet/ms] is shown in Fig. 3.

Figure 3 shows that the queue length of the XCP router becomes large when the amount of 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 (see Eq. (1)). If the amount of non-XCP traffic increases, the XCP router will be overloaded and many packets will be queued at the buffer. In a network with rapid non-XCP traffic dynamics like realistic TCP traffic dynamics, this phenomenon leads unstable XCP control. As we will see in Sect. 5.2, this phenomenon also becomes problematic for realistic non-XCP traffic dynamics.

From these observations, we conclude that XCP has the following problems: (1) the bottleneck link utilization is lowered against XCP traffic dynamics, and (2) operation of XCP becomes unstable in a network with both XCP and non-XCP traffic.

3. XCP-IR (XCP with Increased Robustness)

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

3.1 Basic Ideas

First, how the degradation of the link utilization caused by XCP traffic dynamics can be prevented? To prevent degradation of the link utilization caused by XCP traffic dynamics, a router's buffer can be utilized effectively. Namely, XCP traffic dynamics can be absorbed at the router's buffer. If a certain amount of packets are always stored in the XCP router's buffer, degradation of the link utilization caused by XCP traffic dynamics can be prevented. In the literature, congestion controls with storing a certain amount of packets in the bottleneck router's buffer have been proposed [10]–[12]. In these congestion controls, storing a certain amount of packets in the bottleneck router's buffer is effective for preventing the degradation of the link utilization.

Second, how instability of the XCP router caused by the increase in non-XCP traffic can be prevented? Instability of the XCP router control caused by the increase in non-XCP traffic can be prevented if the XCP router estimates the available bandwidth to XCP traffic correctly. For this purpose, an XCP router should measure the departure rate of non-XCP traffic. The XCP router estimates the available bandwidth to XCP traffic by subtracting the departure rate of non-XCP traffic from the physical link bandwidth. The XCP router then distributes the available bandwidth for XCP traffic to all XCP flows. Since the XCP router can correctly calculate the bandwidth assigned to each XCP flow, 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 necessary to change neither an XCP sender nor a packet format. Hence, the burden of deploying XCP-IR into a real network is quite low.

3.2 XCP-IR Algorithm

In what follows, we explain the algorithm of XCP-IR. 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

$$\phi = \alpha\, d\,\{(C - D_N) - A\} - \beta\,(Q - Q_T), \tag{2}$$

where D_N is the departure 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 queue length) introduced to prevent degradation of the link utilization caused by XCP traffic dynamics. Thus, Q_T packets are always stored in the XCP router's buffer. Thereby, degradation of the link utilization can be prevented even for XCP dynamic traffic. The increase in Q_T may cause side-effects (e.g., the increase in a queuing delay and a packet loss probability, and less responsiveness). We discuss these side-effects with Sects. 4 and 5 for details.

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

$$D_N = \frac{T}{d}, \tag{3}$$

where T is the total amount of non-XCP packets departed

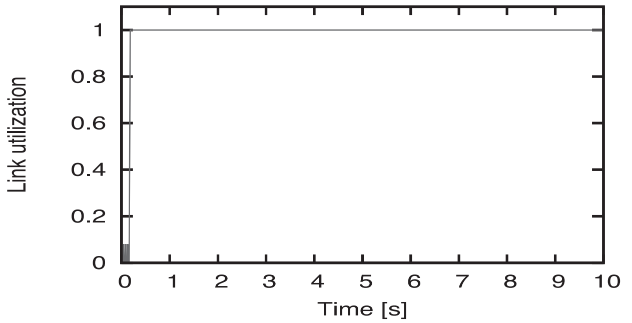


Fig. 4 Time evolution of bottleneck link utilization when changing the number of active XCP flows at t = 4 and 8; XCP-IR completely prevents degradation of the bottleneck link utilization caused by XCP traffic dynamics.

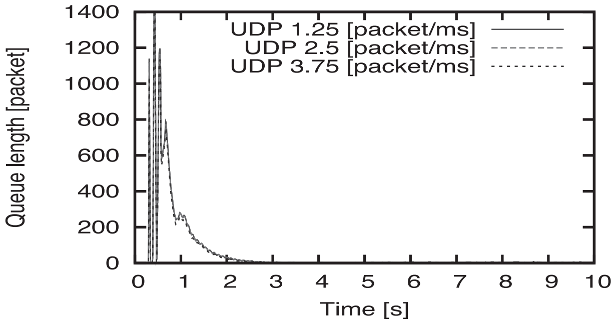


Fig. 5 Time evolution of queue length at XCP-IR router when transmitting non-XCP (UDP) traffic; XCP-IR succeeds to control the queue length regardless of the average transfer rate of UDP traffic.

at the XCP router during the control interval d of the XCP router. By changing the available bandwidth of XCP traffic from C to $C - D_N$, XCP-IR can absorb XCP router's temporary overload when the amount of non-XCP traffic increases.

Note that such a simple modification to the XCP router itself may cause a fairness issue between XCP and non-XCP traffic. Specifically, changing the available bandwidth of XCP traffic from C to $C - D_N$ implies that XCP traffic suffers lower priority than non-XCP traffic. However, priority control between XCP and non-XCP traffic can be easily realized by introducing one of service differentiation mechanisms at an XCP router. For instance, a multi-level RED mechanism [13] can be applied to the XCP router for realizing priority control between XCP and non-XCP (e.g., TCP and UDP) traffic. By configuring parameters of the weighted round-robin queue appropriately, fairness among XCP and non-XCP traffic can be realized. Note that a multi-level RED mechanism is for realizing fairness among traffic classes (i.e., XCP and non-XCP traffic) rather than among traffic flows (i.e., XCP and non-XCP flows).

In what follows, we briefly demonstrate that the problem of XCP for XCP and non-XCP traffic dynamics can be solved using XCP-IR.

Figure 4 shows the time evolution of the bottleneck link utilization when using XCP-IR in the same simulation model with that of Fig. 2. The target value Q_T of the queue length is set to 2,000 [packet]. By comparing Figs. 2 (XCP) and 4 (XCP-IR), one can find that XCP-IR completely prevents degradation of the bottleneck link utilization caused by XCP traffic dynamics.

Figure 5 shows the time evolution of the queue length of the XCP-IR router when transmitting the UDP traffic with the average transfer rate of 1.25, 2.5, and 3.75 [packet/ms]. To simply focus on the tolerance to non-XCP traffic dynamics, the target value of the queue length Q_T is set to 0 [packet]. By comparing Figs. 3 (XCP) and 5 (XCP-IR), one can find that XCP-IR succeeds to control the queue length regardless of the average transfer rate of UDP traffic.

In what follows, we discuss characteristics of XCP-IR. Specifically, we discuss how steady state and transient state performances are affected by changing the calculation method of the aggregate feedback ϕ from Eq. (1) to Eq. (2).

First, the round-trip times of XCP flows in XCP and XCP-IR are compared. The round-trip time of XCP-IR is larger than that of XCP. XCP-IR controls the queue length to Q_T whereas XCP controls to 0. In XCP-IR, the queuing delay of Q_T/C therefore occurs in the router's buffer. In other words, XCP-IR sacrifices the round-trip time for improving robustness. As we will discuss in Sect. 5, we believe such an increase in the round-trip time is acceptable for most applications.

Second, stability and transient state performance of XCP and XCP-IR are compared. We analyzed stability and the transient state performance of XCP and XCP-IR using the analytic approach in [14]. Although derivation and results are not presented due to space limitation, we found that the stability and the transient state performance of XCP-IR around the equilibrium point are same as those of XCP. Namely, our findings indicate that the stability and the transient state performance around the equilibrium point do not change, even if the calculation method of the aggregate feedback value is changed from Eq. (1) to Eq. (2). This implies that sensitivity of XCP-IR control parameters on stability and transient state performance is the same with that of XCP-IR.

4. Performance Evaluation Using Synthetic Traffic Dynamics

In this section, we evaluate the performance of XCP-IR through extensive simulations using synthetic traffic dynamics. We investigate the performance of XCP-IR from the following viewpoints.

1. Stability and transient state performance
2. Robustness for XCP traffic dynamics
3. Robustness for non-XCP traffic dynamics

Figure 6 shows the network topology used in simulation. Except the existence of UDP traffic, we use the same simulation model with that in [1]. Multiple XCP flows and the single UDP flow share the single bottleneck link. For compact notation, in what follows, the bandwidth and the propagation delay of the bottleneck link are denoted by C and τ, respectively. Unless explicitly stated, parameter values shown in Table 1 are used throughout our simulations. We used ns-2 version 2.28 with several modifications to implement XCP-IR.

4.1 Stability and Transient State Performance

Stability

In Sect. 3, the stability of XCP-IR around the equilibrium point was discussed. In the following simulations, we investigate the stability after initiating data transfers rather than the stability around the equilibrium point.

The stability regions of the control parameter (α, β) in XCP and XCP-IR are shown in Fig. 7. We performed a large number of simulations with different sets of control parameters (α, β). To focus on the effect of dynamic XCP traffic, 20 XCP flows started their transfers simultaneously (i.e., UDP traffic is not generated). This figure means that operation of XCP or XCP-IR was stable, when a set of control parameters (α, β) was in the stability region (i.e., the region surrounded by the boundary line and XY-axes). We determined XCP or XCP-IR was stable when the window sizes of all flows were stabilized within ± 5% of their equilibrium values.

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 XCP flows initiate their data transfers. However, with the recommended parameter configuration of $(\alpha, \beta) = (0.4, 0.226)$ [1], both XCP and XCP-IR operate stably. Hence, as long as the recommended parameter configuration is used, there should be no stability problem in XCP-IR.

Transient state performance

We also investigate the transient state performance after initiating data transfers rather than the transient state performance around the equilibrium point.

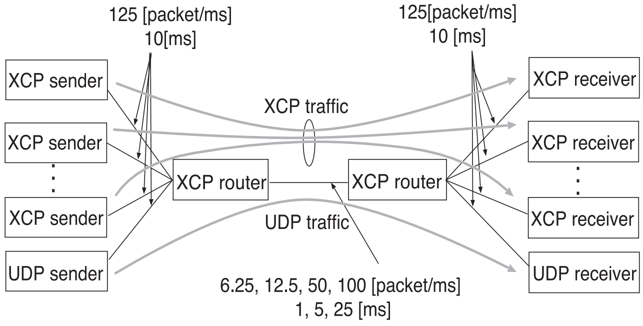


Fig. 6 Network topology used in simulation; multiple XCP flows and the single UDP flow share the single bottleneck link.

Table 1 Parameters used in simulation.

packet size	1,000 [byte]
socket buffer size	delay-bandwidth product
buffer size of XCP router	10,000 [packet]

The settling time of the window size of an XCP flow while changing the target value of the queue length Q_T from 0 to 5,000 [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 within ±5% of its equilibrium value (i.e., the window size in steady state). To focus on the effect of dynamic XCP traffic, 20 XCP flows start their transfers simultaneously in this simulation.

Figure 8 shows that the transient state performance of XCP-IR is always better than that of XCP unless the target queue length Q_T is too large. As the target queue length Q_T increases, the settling time of the window size once decreases. But it then increases as the target queue length becomes too large. This is because that an increase in the target queue length Q_T has both positive and negative effect on the transient state performance. Namely, increasing the target queue length makes XCP-IR more aggressive since the aggregate feedback value ϕ becomes large as Q_T increases. At the same time, increasing the target queue length makes XCP-IR less responsive since the round-trip time becomes large as Q_T increases. With an appropriate setting of the target queue length Q_T, the positive effect is much stronger than the negative effect, leading XCP-IR's better transient performance than XCP.

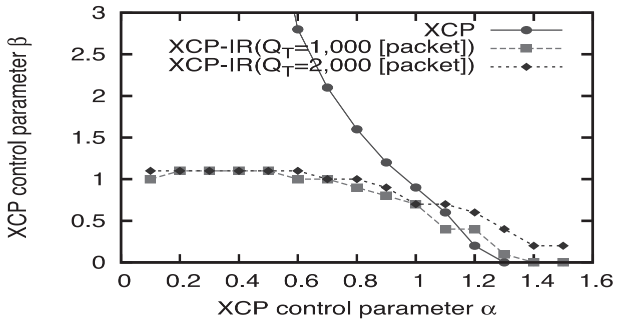


Fig. 7 Stability region when XCP flow starts data transfers (XCP control is stable when (α, β) is under the boundary line); both XCP and XCP-IR operate stably with the recommended parameter configuration of (0.4, 0.226).

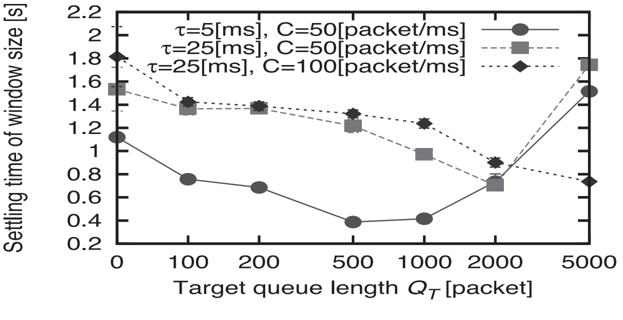


Fig. 8 Settling time of window size when an XCP flow starts its data transfer; the settling time of the window size is a concave function for the target value of queue length Q_T.

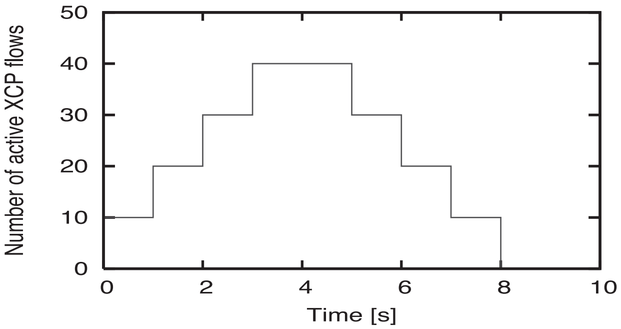


Fig. 9 Synthetically generated XCP traffic dynamics; the number of active XCP flows are synthetically changed during simulation.

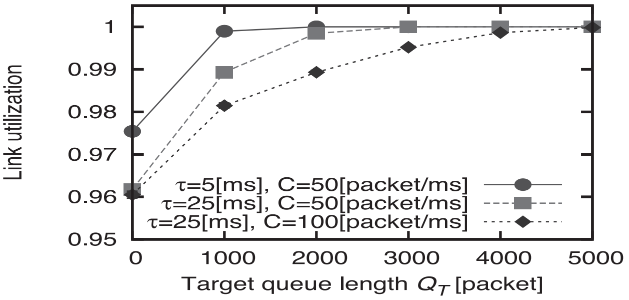
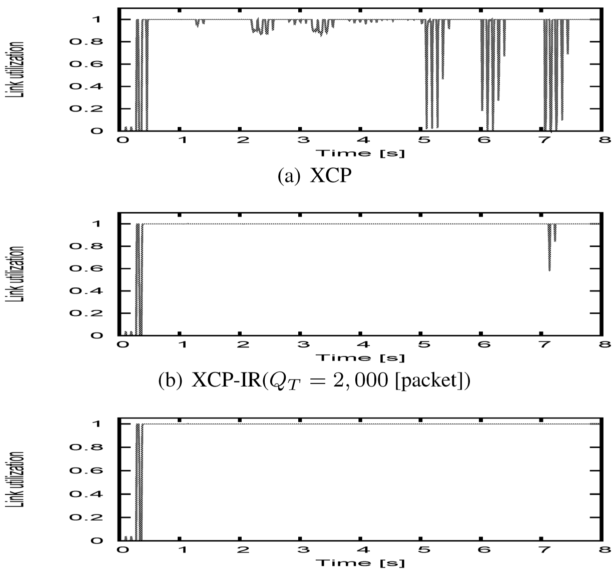


Fig. 10 Bottleneck link utilization for a different target value Q_T with synthetic XCP traffic dynamics; XCP-IR always shows higher bottleneck link utilization than XCP.



(a) XCP

(b) XCP-IR($Q_T = 2,000$ [packet])

(c) XCP-IR($Q_T = 2,600$ [packet])

Fig. 11 Time evolution of the bottleneck link utilization with XCP, XCP-IR(Q_T = 2,000 [packet]), and XCP-IR(Q_T = 2,600 [packet]) for τ = 25 [ms] and C = 50 [packet/ms].

4.2 Robustness for XCP Traffic Dynamics

For investigating robustness of XCP-IR, dynamic XCP traffic is synthetically generated by changing the number of active XCP flows as shown in Fig. 9. 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, UDP traffic is not generated. The bottleneck link utilization when changing the target value of the queue length Q_T is shown in Fig. 10. Note that the result of XCP corresponds to the case with $Q_T = 0$ [packet].

Figure 10 shows that XCP-IR always shows higher bottleneck link utilization than XCP. This figure also shows the bottleneck link utilization increases as the target value of the queue length Q_T increases. It should be noted that all bottleneck link utilizations in Fig. 10 are convex for the target value Q_T. This implies an advantage of XCP-IR; i.e., a small increase in Q_T significantly improves the bottleneck link utilization. In practice, the target value of the queue length Q_T should be determined by taking account of trade-offs between robustness and responsiveness. For the link utilization, the improvement in XCP-IR is not so significant. However, in Sect. 5.1, we show that XCP-IR will significantly improve throughput.

Discussion on appropriate setting of Q_T

How the target value of the queue length Q_T should be configured to maximize the XCP-IR performance? When the number of active XCP flows decreases rapidly, at least the feedback delay of d_f is taken for the XCP router to notify XCP senders of the changed feedback value. The feedback delay d_f is a fraction of the round-trip time d; i.e., d_f can be obtained by subtracting the queuing delay from the round-trip time d. Let us assume that M of N XCP flows terminate their transfers. In this case, $d_f\,C\,M/N$ packets are drained from the buffer until XCP senders are notified of the re-calculated feedback value. To prevent degradation of the bottleneck link utilization, it is necessary to satisfy $d_f\,C\,M/N \leq Q_T$; i.e.,

$$Q_T \geq \frac{d_f\,C\,M}{N} \tag{4}$$

Figure 10 clearly shows the validity of Eq. (4); i.e., the bottleneck link utilization is almost 1.0 when the target value of the queue length satisfies Eq. (4). Namely, $Q_T \geq 1,250$ for τ = 5 [ms] and C = 50 [packet/ms], $Q_T \geq 2,250$ for τ = 25 [ms] and C = 50 [packet/ms], and $Q_T \geq 4,500$ for τ = 25 [ms] and C = 100 [packet/ms].

We then focus on the time evolution of the bottleneck link utilization for τ = 25 [ms] and C = 50 [packet/ms]. In Fig 11, the time evolutions of the bottleneck link utilization in XCP and XCP-IR with $Q_T = 2,000$ and $Q_T = 2,600$ is plotted. The number of active XCP flows is synthetically changed as shown in Fig. 9. Figure 11 shows that degradation of the bottleneck link utilization can be prevented with Q_T larger than 2,250 [packet] (i.e., Q_T = 2,600 [packet]).

Note that the bottleneck link utilization is stabilized quite rapidly in Fig. 11. This is because XCP can quickly respond to decrease in the number of active XCP flows. When the number of active XCP flows decreases rapidly, the queue length of the XCP router decreases, which results

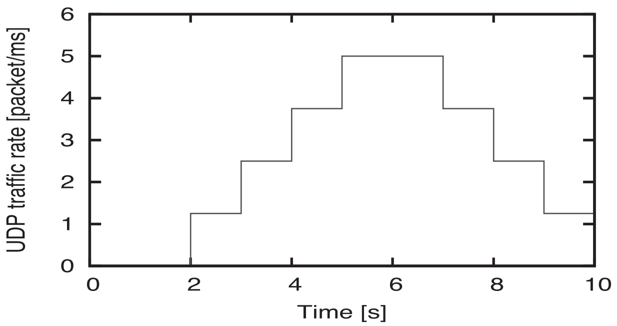


Fig. 12 Synthetically generated UDP traffic dynamics; the average transfer rate of UDP traffic is changed during simulation.

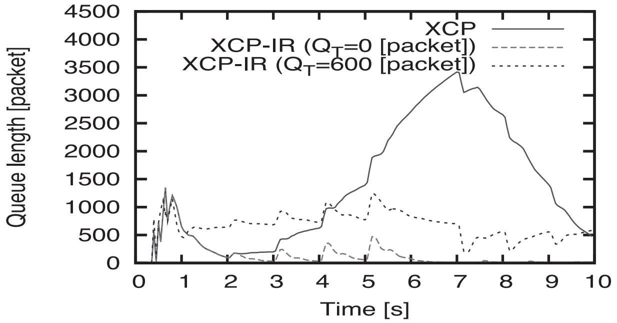


Fig. 13 Time evolutions of queue lengths with XCP and XCP-IR (Q_T = 0 and 2,000 [packet]) for τ = 25 [ms] and C = 50 [packet/ms]; XCP-IR can control the queue length around the target value Q_T.

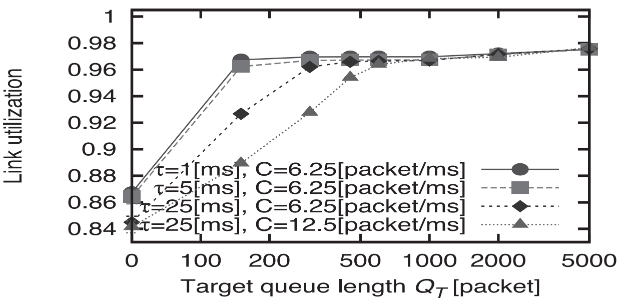


Fig. 14 Bottleneck link utilization for a different C and τ; the bottleneck link utilization with XCP-IR is approximately 10% higher than that with XCP.

in smaller round-trip time to active XCP flows. Thus, XCP flows can quickly respond to congestion indication from the XCP router.

4.3 Robustness for Non-XCP Traffic Dynamics

We first investigate the robustness of XCP-IR for synthetically generated dynamic UDP traffic. The average transfer rate of background UDP traffic is changed every second as shown in Fig. 12.

The time evolutions of the queue lengths with XCP and XCP-IR ($Q_T = 0$ and 600 [packet]) for τ = 25 [ms] and C = 50 [packet/ms] are shown in Fig. 13. A single XCP flow is activated at $t = 0$.

This figure shows that XCP fails to stabilize the queue length. This figure also shows that XCP-IR controls the queue length around the target value of the queue length Q_T. This clearly indicates that XCP-IR has robustness for UDP traffic dynamics.

5. Performance Evaluation Using Realistic Traffic Dynamics

5.1 Robustness for Realistic XCP Traffic Dynamics

For deploying XCP-IR in a real network, its performance should be evaluated under realistic conditions. We therefore investigate the robustness of XCP-IR for realistic XCP traffic dynamics.

Realistic dynamic XCP traffic is generated by aggregating a large number of XCP flows according to the mesurement results reported in [15]. Currently, majority of the Internet traffic is Web and P2P traffic. We generated realistic dynamic XCP traffic by aggregating a large number of XCP flows. Each XCP flow is randomly actived to match the measured distribution of flow activation intervals [15]. It is then deactivated when it finishes its file transfer, whose file size is randomly determined according to the measured distribution of file sizes [15]. More specifically, distributions of flow activation intervals of Web and P2P traffic are given by an exponential distribution with mean 0.015 and 14.98 [s], respectively. Distributions of file sizes of Web and P2P traffic are given by a Pareto distribution with mean $0.421C \times 10^{-3}$ and $0.121C$ [Mbyte], respectively. Note that file sizes are proportional to the bottleneck link bandwidth C. In our simulation scenario, the number of XCP flows carrying Web traffic is as appoximately 1,100 times as those carrying P2P traffic.

The bottleneck link utilization for different settings of C and τ are shown in Fig. 14. This figure shows that XCP-IR can achieve 97% link utilization with an adequate setting of the target queue legnth Q_T while XCP achieves only 86% link utilization. In this case, Q_T = 600 [packet] is sufficient for well utilizing the bottleneck link bandwidth regardless of the bottleneck link bandwidth C and the propagation delay τ. Q_T can take a much smaller value while almost fully utilizing the bottleneck link bandwidth if the propagation delay is small (e.g., τ = 1 or 5 [ms]). Note that since this simulation scenario is very busrty, the bottleneck link utilization in XCP-IR is slightly less than 1.0. An example of realistic traffic dynamics (i.e., the evolution of the number of active XCP flows) generated by aggregating a large number of XCP flows is shown in Fig. 15. Note that the realistic traffic dynamics (Fig. 15) is much more dynamical than the synthetic traffic dynamics (Fig. 9).

The average round-trip delay of all XCP flows for different settings of C and τ are shown in Fig. 16. This figure shows that the average round-trip time of all XCP flows gradually increases as the target queue legnth Q_T increases. This is because XCP-IR controls so that Q_T packets are queued at the router's buffer to prevent degradation of the bottleneck link utilization. Note that since this simulation

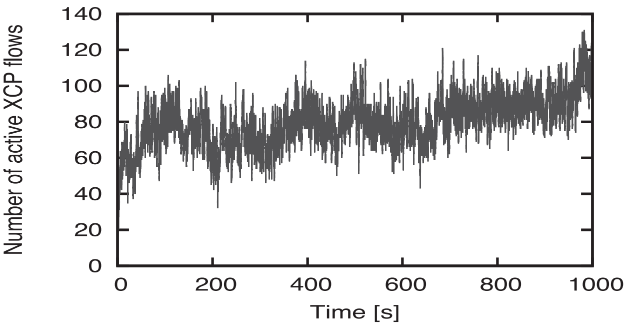


Fig. 15 Time evolution of the number of active XCP flows for τ = 25 [ms] and C = 6.25 [packet/ms].

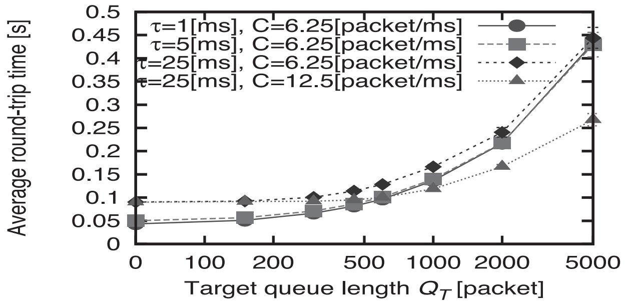


Fig. 16 Average Round-trip time of all XCP flows for a different C and τ; the average round-trip time of all XCP flows gradually increases when Q_T increases.

scenario is very busrty, the queue length with XCP-IR is smaller than Q_T. Therefore, the increase in the round-trip time of all XCP flows is smaller than Q_T/C. To prevent the significant increase in the round-trip delay of an XCP flow, Q_T should be set between 0 and 150 [packet] for LAN environment and between 0 and 600 [packet] for WAN environment.

Average throughputs of all Web flows and all P2P flows for different settings of C and τ are shown in Figs. 17 and 18. Figure 17 shows that the average throughput of all Web flows with XCP-IR is always higher than that of XCP unless the target queue length Q_T is too large. This is because as the target queue length Q_T increases, although increase in the link utilization become small, increase in the round-trip time do not become small. Figure 18 shows that the average throughput of all P2P flows with XCP-IR is always higher than that of XCP. This is because there is almost no effect of the throughput on the increase in the round-trip time since file sizes of P2P traffic are large.

Due to space limitation, results for the packet loss probability in the bottleneck router for a different C and τ are not included. We have confirmed that a packet was not lost in the bottleneck router. If the configuration value of Q_T is sufficiently small as compared with the buffer size of the XCP router, increase in a packet loss probability can be prevented.

We then focus on the time evolution of the bottleneck link utilization for τ = 25 [ms] and C = 6.25 [packet/ms]. In Fig 19, the time evolutions of the bottleneck link utilization with XCP and XCP-IR for Q_T = 150 [packet] and Q_T = 300 [packet] is plotted. This figure shows that degradation of the bottleneck link utilization can be almost prevented with XCP-IR(Q_T = 300 [packet]).

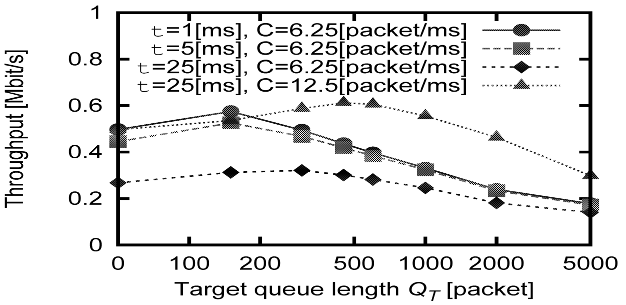


Fig. 17 Average throughput of all Web flows for a different C and τ; the average throughput of all Web flows is a convex function for the target value of queue length Q_T.

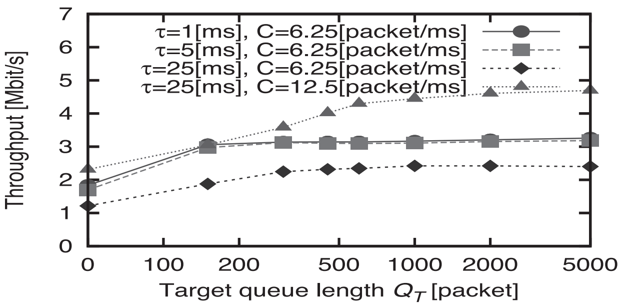
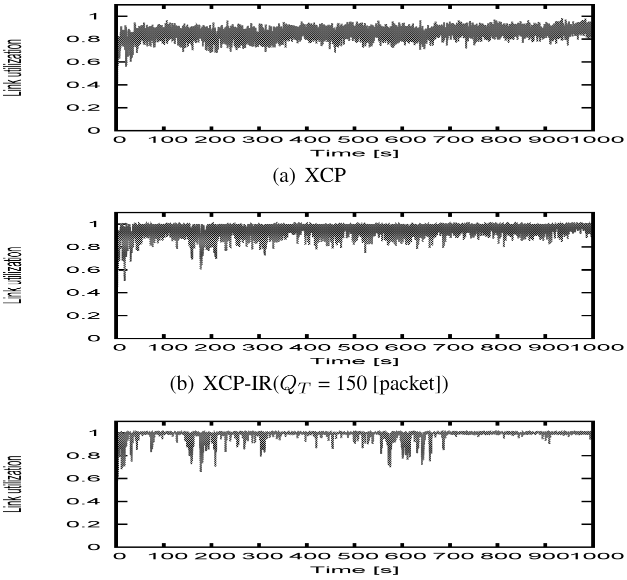


Fig. 18 Average throughput of all P2P flows for a different C and τ; the average throughput of all P2P flows with XCP-IR is approximately 200% higher than that with XCP.



(a) XCP

(b) XCP-IR(Q_T = 150 [packet])

(c) XCP-IR(Q_T = 300 [packet])

Fig. 19 Time evolution of the bottleneck link utilization with XCP, XCP-IR(Q_T = 150 [packet]), and XCP-IR(Q_T = 300 [packet]) for τ = 25 [ms] and C = 6.25 [packet/ms] when generating realistic XCP traffic.

From these observations, we conclude that XCP-IR has

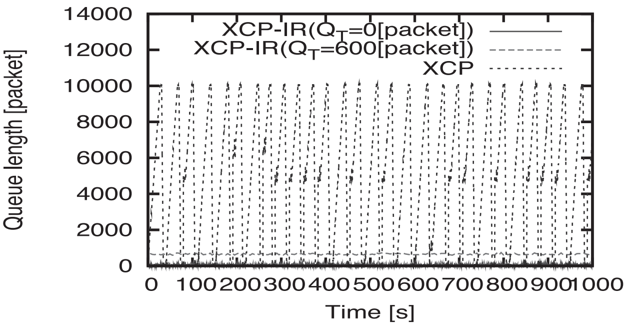


Fig. 20 Time evolutions of queue lengths with XCP, XCP-IR (Q_T = 0 [packet]), and XCP-IR (Q_T = 600 [packet]) for τ = 25 [ms] and C = 6.25 [packet/ms]; the queue length of XCP drastically oscillates, whereas the queue length of XCP-IR is stabilized and minimized.

high robustness for different types of XCP traffic dynamics.

5.2 Robustness for Realistic TCP Traffic Dynamics

We finally investigate the robustness of XCP-IR for realistic non-XCP traffic dynamics. Since the majority of the Internet traffic is carreid by TCP, we consider the impact of dynamic TCP traffic on the robustness of XCP-IR.

Similarly to Sect. 5.1, realistic dynamic TCP traffic is generated by aggregating a large number of TCP flows according to the mesurement results in [15]. Namely, TCP flows are randomly activated and their file sizes are randomly determined based on the measurement results in [15]. A single XCP flows is activated at $t = 0$.

Time evolutions of queue lengths with XCP, XCP-IR (Q_T = 0 [packet]), and XCP-IR (Q_T = 600 [packet]) for τ = 25 [ms] and C = 6.25 [packet/ms] are shown in Fig. 20. This figure shows that the queue length of XCP drastically oscillates, whereas the queue length with XCP-IR is stabilized and minimized. Note that the queue length with XCP is unstable; the queue length is simply upper-bounded by the buffer size, and many packet are lost due to buffer overflows. These result indicate that the lack of robustness is quite costly for realistic dynamic traffic. The oscillatory behavior of the queue length with XCP in Fig. 20 is directly caused by the oscillatory behavior of XCP flows. Figure 21 shows the evolution of the window size of an XCP flow. As explained in Sect. 2.2, when the amount of non-XCP traffic increases, the XCP router is overloaded and many packets are queue at the buffer because the XCP router performs its control by incorrectly assuming that the available bandwidth to XCP traffic is known.

We then investigate how malicious XCP's lack of robustness is in terms of the bottleneck link utilization, the throughput of the XCP flow, and the round-trip time. The bottleneck link utilization, the throughput of the XCP flow, and the round-trip time of the XCP flow for different buffer sizes of the XCP router are shown in Figs. 22 through 24, respectively. Those figures show that the performance of XCP is significantly affected by the buffer size of the XCP router, but XCP still suffers either low bottleneck link utilization or large round-trip time. This can be explained as follows. If the buffer size of the XCP router is small, many packets are lost due to buffer overflows, leading low bottleneck link utilization and low throughput. On the contrary, if the buffer size of the XCP router is large, the queuing delay in the XCP router is large, leading a large round-trip time.

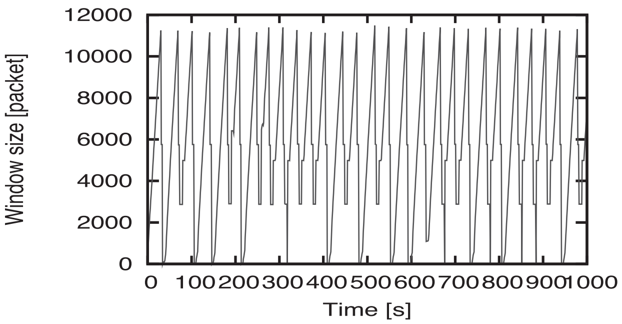


Fig. 21 Time evolutions of the window size with XCP for τ = 25 [ms] and C = 6.25 [packet/ms].

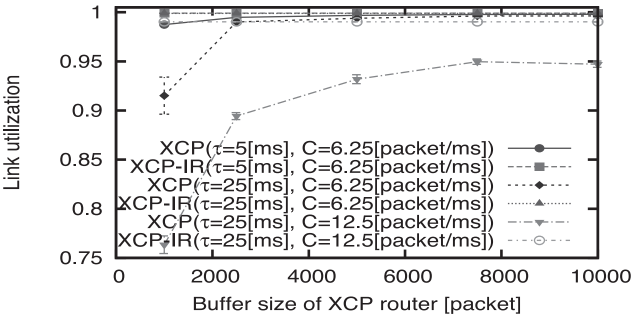


Fig. 22 Bottleneck link utilization for a different buffer size of the XCP router; the bottleneck link utilization with XCP is low when the buffer size of the XCP router is small.

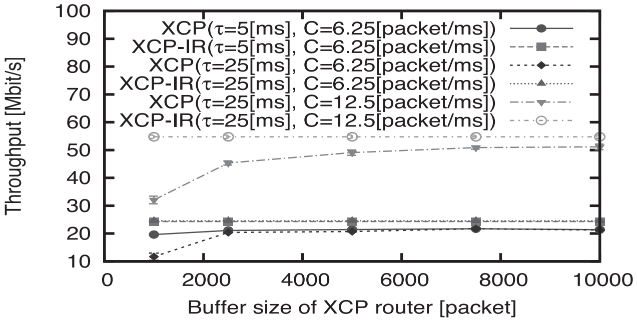


Fig. 23 Throughput of the XCP flow for a different buffer size of the XCP router; the throughput of the XCP flow with XCP is low when the buffer size of the XCP router is small.

From these observations, we conclude that XCP-IR achieves good performance and high robustness for different types of non-XCP traffic dynamics.

6. Conclusion

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

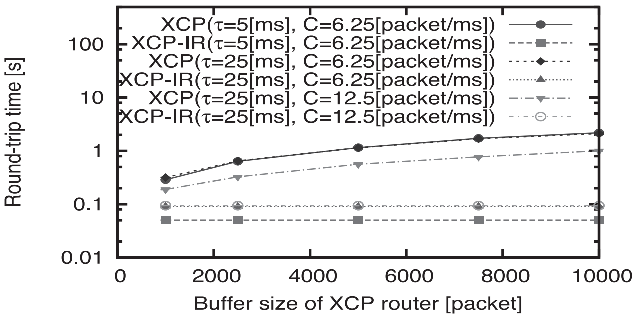


Fig. 24 Round-trip time of the XCP flow for a different buffer size of the XCP router; the round-trip time of the XCP flow with XCP is very large when the buffer size of the XCP router is large.

namic traffic. XCP-IR prevents instability of the XCP control caused by non-XCP traffic dynamics while preventing degradation of the bottleneck-link utilization caused by XCP traffic dynamics. Through extensive simulation experiments, we have shown that XCP-IR operated efficiently even for dynamic traffic. In particular, we have shown that the throughput with XCP-IR is approximately 200% higher than that with XCP.

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

Acknowledgements

We would like to express our appreciation to Prof. Masayuki Murata for joining meaningful discussions. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (A), 19680003, 2007.

References

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

[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] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," Request for Comments (RFC) 2581, April 1999.

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

[5] K. Zhou, K.L. Yeung, and V.O. Li, "P-XCP: A transport layer protocol for satellite IP networks," Proc. 47th IEEE Global Telecommunications Conference (GLOBECOM 2004), pp.2707–2711, Nov. 2004.

[6] D.M. Lopez-Pacheco and C. Pham, "Robust transport protocol for dynamic high-speed networks: Enhancing the XCP approach," Proc. IEEE 7th Malaysia International Conference on Communications and 13th IEEE International Conference on Networks (MICC and ICON 2005), pp.1–6, Nov. 2005.

[7] D.M. Lopez-Pacheco, C. Pham, and L. Lefévre, "XCP-i: Explicit control protocol for heterogeneous inter-networking of high-speed networks," Proc. 49th IEEE Global Telecommunications Conference(GLOBECOM 2006), pp.1–6, Nov. 2006.

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

[9] A. Falk, Y. Pryadkin, and D. Katabi, "Specificaton for the explicit control protocol(XCP)," IETF Internet Draft: Draft-falk-xcp-spec-03.txt, July 2007.

[10] D.X. Wei, C. Jin, S.H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," IEEE/ACM Trans. Netw., vol.14, no.6, pp.1246–1259, Dec. 2006.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," Proc. 25th IEEE International Conference on Computer Communications (INFOCOM 2006), pp.1–12, April 2006.

[12] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Netw., vol.1, no.4, pp.397–413, Aug. 1993.

[13] J. Aweya, M. Ouellette, and D.Y. Montuno, "Service differentiation using a multi-level red mechanism," Int. J. Netw. Manage., vol.12, no.2, pp.81–98, Jan. 2002.

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

[15] T. Mori, M. Uchida, and S. Goto, "Flow analysis of internet traffic: world wide web versus peer-to-peer," Systems and Computers in Japan, vol.36, no.11, pp.70–81, Oct. 2005.

Yusuke Sakumoto received B.E. and M.E. degrees in the Information and Computer Sciences from Osaka University in 2006 and 2008, respectively. He is currently a Ph.D. candidate Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of congestion control protocol. He is a student member of IEEE and IPSJ.

Hiroyuki Ohsaki received the M.E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 1995. He also received the Ph.D. degree from Osaka University, Osaka, Japan, in 1997. He is currently an associate professor at Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of traffic management in high-speed networks. He is a member of IEEE and IPSJ.

Makoto Imase received his B.E. and M.E. degrees in information engineering from Osaka University in 1975 and 1977, respectively. He received D.E. degree from Osaka University in 1986. From 1977 to 2001, he was engaged Nippon Telegraph and Telephone Corporation (NTT). He has been a Professor of Graduate School of Information Science and Technology at Osaka University since 2002. His research interests are in the area of information networks, distributed systems and graph theory. He is a member of IPSJ and JSIAM.