A Relative Bandwidth Allocation Method Enabling Fast Convergence in XCP

Hanh Le Hieu[1], Kenji Masui[2], and Katsuyoshi Iida[2]

[1] Graduate School of Science and Engineering, Tokyo Institute of Technology
[2] Global Scientific Information and Computing Center, Tokyo Institute of Technology
hanhlh@netsys.ss.titech.ac.jp,{kmasui,iida}@gsic.titech.ac.jp

Abstract. The eXplicit Control Protocol (XCP), which explicitly sends feedback information on the current network resource condition to end-hosts to control congestion, is a promising protocol for the network supported congestion control that will be needed in the future Internet. Because of its flexibility, XCP can provide relative bandwidth allocation service that is impossible with other protocols. However, with the existing version of this XCP service, the convergence time to network stability is long due to the gradual resource allocation. We have analyzed the existing allocation method and propose a new allocation method for use at XCP routers. The effectiveness of the proposed method has been evaluated through simulation and the results are discussed here.

Keywords: convergence, relative allocation, QoS, congestion control, XCP, NGN

1 Introduction

Technology trends indicate that the use of high-bandwidth and high-latency links (e.g., satellite links) with throughput of up to tens of Gb/s will be widespread in the future Internet [2]. However, theory and experimental confirmation show that in such a high bandwidth-delay product network environment end-host congestion control in Transmission Control Protocol (TCP), in which packet drop is used as a signal to implicitly control congestion to end-hosts, will become inefficient and prone to instability [7]. To deal with this issue, starting with [4], recently a number of network supported congestion controls have been performed. Among of them, the eXplicit Control Protocol (XCP) [5], which controls network congestion by returning feedback information in packet headers to end-hosts, is a promising means of congestion control.

Furthermore, as the Internet has grown, relative bandwidth allocation has received much attention. Relative bandwidth allocation is a service to differentially allocate network bandwidths to flows transversing the network according to predefined weights. This service is offered because users differ in the value they assign to network reliability. A number of users or application/content service providers are willing to pay more than others to obtain better bandwidth at times of congestion. For instance, in the future Internet, the demands of high-definition movie viewing, high-quality live streaming, huge bulk-data exchange,

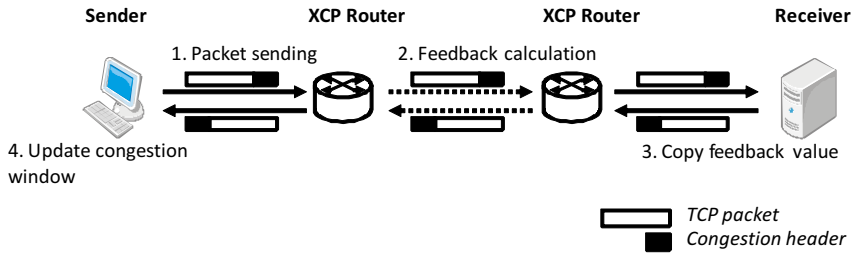


Fig. 1. XCP Illustration

etc., will be considerable. In such cases, users from a higher class who pay more will obtain better quality of service than ones from a lower class. This will make the relative bandwidth allocation service more and more important and requisite.

In the current Internet, where TCP dominates, several relative bandwidth allocation solutions have been reported, such as RIO[3], CSFQ [9] and WPFRA [6]. XCP compares favorably to these solutions in that its unique flexible resource allocation schemes make it possible to provide relative bandwidth allocation service to targeted flows transversing a network with higher performance and it is easier to implement [5]. However, the relative bandwidth allocation method in XCP treats such targeted flows equally with the remaining ones in a network when converging to stability. In XCP, it takes multiple round-trip times (RTTs) for most flows to reach their fair rates, so when a new flow starts arriving in a network with a low transmission rate, the convergence time of this flow to stability, as well as that of the total network, becomes quite long.

This paper describes two relative bandwidth allocation methods implemented at XCP routers to enable fast convergence to stability. First, a Basic Method of utilizing the correct feedback value provided by a normal XCP router is tried. After that, an Improved Method that overcomes a problem in the Basic Method is proposed. The remainder of this paper is organized as follows. First, we provide an overview of XCP and the existing relative bandwidth allocation method proposed in [5] in Section 2. We describe our proposed methods for use at XCP router in Section 3, and summarize the simulation experiment and discuss the results in Section 4. Concluding remarks and possibilities for future work are given in Section 5.

2 XCP and Relative Bandwidth Allocation Method in XCP

2.1 XCP

While TCP concurrently allows the two congestion-control goals of high utilization and fairness to be achieved, XCP [5] is a protocol for Internet congestion control that is based on decoupling congestion control from the resource allocation policy. The XCP protocol is illustrated in Fig. 1. Each XCP packet carries a three-field congestion header. The **H_throughput** and **H_RTT** fields show the current throughput and RTT estimates of senders. They are filled in by

the senders and never modified in transit. The remaining field **H_feedback**, takes positive or negative values and is initialized by senders. Routers along the communication path modify this field to directly and explicitly control the throughput of the sources in a way that causes the system to converge to optimal efficiency and max-min fairness. When the feedback reaches the receiver, it is returned to the sender in an acknowledgment packet and the sender accordingly updates its rate.

2.2 Relative Bandwidth Allocation Method in XCP (XCP Method)

The relative bandwidth allocation method provided in [5] works as follows. XCP uses the current throughput of each flow to converge each flow's rate to a fair rate, so the sender replaces the **H_throughput** field with the current throughput divided by the weight of the flow. As a result, at times of congestion, the throughput of each flow becomes $\frac{throughput_i}{weight_i} = \frac{throughput_j}{weight_j}$, hence $\frac{throughput_i}{throughput_j} = \frac{weight_i}{weight_j}$, which shows the relative bandwidth allocation between flows in the network.

This paper focuses on the relative bandwidth allocation at routers, where each flow's throughput replacing process is shifted to XCP routers. Specifically, we suppose that whenever a packet arrives, XCP routers identify the source of the packet and change the throughput value using a corresponding predefined weight. The method that enables this is called XCP Method.

Note: A provision algorithm to determine weights in relative bandwidth allocation has been proposed in [8].

3 Proposals

In this section, two methods to enable fast convergence for relative allocation of network bandwidth to targeted flows at XCP routers are proposed.

These two methods have a common first step of estimating the specific bandwidth of a target flow. Next, the Basic Method converges the targeted flow's congestion window by using a corrected feedback value for both the increasing and decreasing process of the targeted flow's congestion window, while the Improved Method only uses this value in the decreasing process.

3.1 Basic Method

Targeted Flow Throughput Estimation First, whenever a packet arrives, XCP routers identify its source (e.g., from the flow ID, flow source, etc.).

Then, if the packet belongs to a targeted flow, the optimized congestion window corresponding to the desired bandwidth of the flow is calculated at the router through Eq. (1):

$$cwnd_{opt} = \frac{Bandwidth \cdot RTT}{PacketSize} \cdot \frac{weight}{\sum\limits_{all\ flow} weight}, \tag{1}$$

where $cwnd_{opt}$ is the optimized congestion window, $Bandwidth$ is the possible bandwidth of a network link, RTT is round-trip time, $PacketSize$ is the size of

Table 1. Definitions

$weight$	$0 < weight < 1$	$weight = 1$	$weight > 1$
Type of flow	Targeted flow	Background flow	Targeted flow
Type of service	Deprioritizing		Prioritizing

each packet, and $weight$ is a parameter to adjust the desired bandwidth for the targeted flow. In this paper, based on the value of $weight$, the type of flows and service in a relative bandwidth allocation service are defined as shown in Table 1.

Targeted Flow Throughput Convergence In this step, the current congestion window of the flow is converged to the optimized congestion window calculated from Eq. (1) at the routers. This is achieved by simply using the feedback value that was earlier correctly calculated at the XCP routers. In detail, the congestion window of a target flow is increased when smaller than the optimized congestion window and decreased when larger. Furthermore, to ensure smoothness when converging the adjusting flow's congestion window, the current congestion window status of this flow and the feedback value that is correctly calculated at the routers are also taken into consideration. Equation (2) expresses this process:

$$feedback_{adj}(\alpha_{inc}, \alpha_{dec}, \varepsilon) = \begin{cases} d^{\alpha_{inc}} \cdot |feedback| & cwnd < (1-\varepsilon) \cdot cwnd_{opt}, \\ -d^{\alpha_{dec}} \cdot |feedback| & cwnd > (1+\varepsilon) \cdot cwnd_{opt}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where $cwnd$ is the current congestion window, $d = \left| \frac{cwnd - cwnd_{opt}}{cwnd_{max}} \right|$, and $cwnd_{max} = \frac{Bandwidth \cdot RTT}{PacketSize}$. Here, α_{inc} and α_{dec} are used to decide whether or not the converging process considers the difference between the current adjusting congestion window and the optimized one. In addition, ε is a parameter to adjust the precision of the method. The smaller this parameter is, the more accurate the method becomes. The sender uses this $feedback_{adj}$ to adjust its own congestion window.

3.2 Improved Method

Targeted Flow Throughput Estimation The first step in this method is the same as that of the Basic Method.

Targeted Flow Throughput Convergence When the $weight$ parameter in Eq. (1) for a targeted flow is higher than one, the Basic Method will take a longer time to converge to the desired value (a detailed analysis is provided in the next section). To solve this problem, in this step, instead of using the correct feedback value, right from first RTTs, the method aggressively allocates the desired bandwidth to a specific flow as shown by Eq. (3):

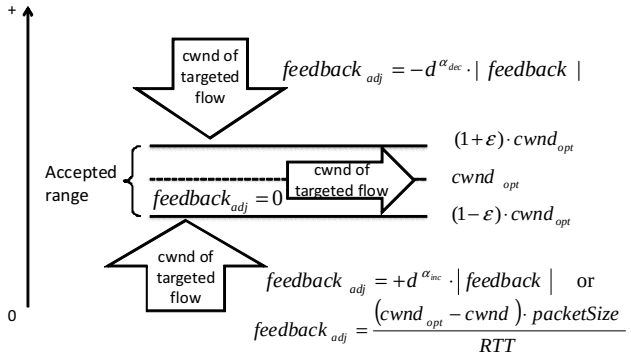


Fig. 2. Illustration of calculation in the convergence process of proposals

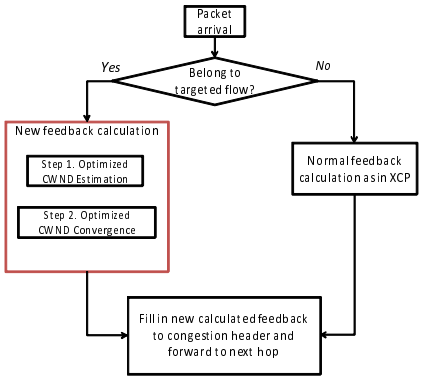


Fig. 3. Proposal overview

$$feedback_{adj}(\alpha_{dec}, \varepsilon) = \begin{cases} \frac{(cwnd_{opt} - cwnd)\cdot packetSize}{RTT} & cwnd < (1-\varepsilon)\cdot cwnd_{opt}, \\ -d^{\alpha_{dec}} \times |feedback| - & cwnd > (1+\varepsilon)\cdot cwnd_{opt}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The image of convergence calculation in this step of the two proposed methods is shown in Fig. 2.

An overview of our proposals is given in Fig. 3.

4 Experiments and Analysis

The goal of this work has been to develop relative bandwidth allocation methods for use at XCP routers to enable fast convergence. In this section, the performance of the proposed methods is verified in terms of the convergence time compared with that of the XCP Method, how precisely they allocate bandwidth to target flows as expected, and the stability of the whole network.

4.1 Basic Method

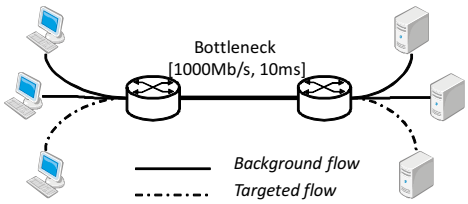


Fig. 4. Dumbbell topology

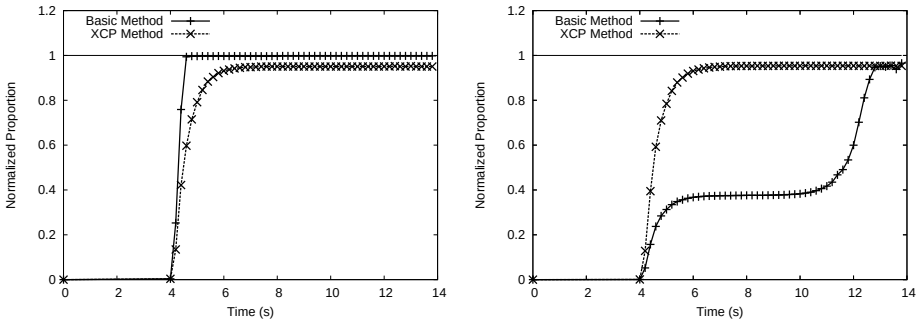
Experiment Setup For the experiments, the classes relating to XCP in ns-2.33 [1] were modified to enable control of feedback value at routers.

In these experiments, the typical dumbbell topology was used, as shown in Fig. 4. To evaluate the effect that applying this method to a specific flow had on the remaining background flows, a scenario of starting *Normal* background flows at 0.1-s intervals and then initiating a targeted flow after a 3-s interval was set. The bandwidth at the bottleneck link, delay, and packet size were respectively fixed to $1,000$ Mb/s, 10 ms, and $1,000$ Byte. The buffer size at each router (i.e., the product of the link's bandwidth and delay) was kept as in normal XCP [5]. The experiments were performed with 10 background flows and one targeted flow. Furthermore, to evaluate the proposed method for both prioritized and deprioritized targeted flows, the *weight* parameter in Eq. (1) was fixed to 0.5 or 3. Also, based on preliminary experiment results, α_{inc}, α_{dec}, and ε were set to 0, 1, and 0.1, respectively. We did not consider the difference between the current congestion window and the optimized one in the increasing case because *distance* is always smaller than one (and greater than 0), which diminishes the value of $feedback_{adj}$. This is contrary to the purpose of increasing congestion windows for a targeted flow.

Experiment Results and Analysis The experiment results for comparing the convergence time between the Basic Method and the XCP Method are shown in Fig. 5. These results were averaged over a 0.1-s period. As the number of background flows was 10, the starting time of the targeted flow was at 4 s.

In Fig. 5, the x-axis shows the time and the y-axis indicates the normalized proportion of the target flow's throughput to the desired one. Figures 5(a) and 5(b) show the results when the *weight* was 0.5 and 3, respectively. In Fig. 5(a), the Basic Method outperformed the XCP Method by nearly 2 s, but when the *weight* was 3, the Basic Method converged target flows to the optimized value much more slowly than the XCP Method. The reason for this can be understood as follows by referring to Fig. 6.

Although the XCP Method relatively allocates bandwidth to flows, it actually converges to the weighted allocation status as it does in normal allocation. Figure 6 describes the convergence to fair status process for a simple case of two flows with throughput (X_1, X_2) in a network. In this figure, $X_{goal-deprior}$ is the estimation rate in deprioritizing, X_{fair} represents the fair rate, and $X_{goal-prior}$ corresponds to the estimation rate in prioritizing. The targeted flow's throughput is X_1 and the XCP Method needs time to converge X_1 from the initial status,



(a) Weight equals 0.5 (b) Weight equals 3

Fig. 5. Convergence time comparison between the Basic Method and the XCP Method

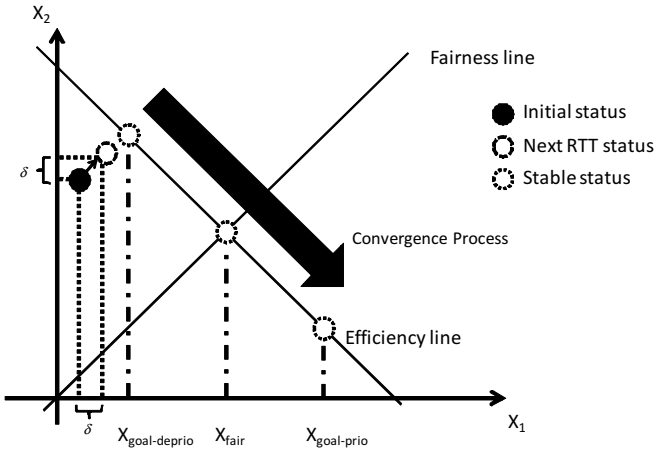
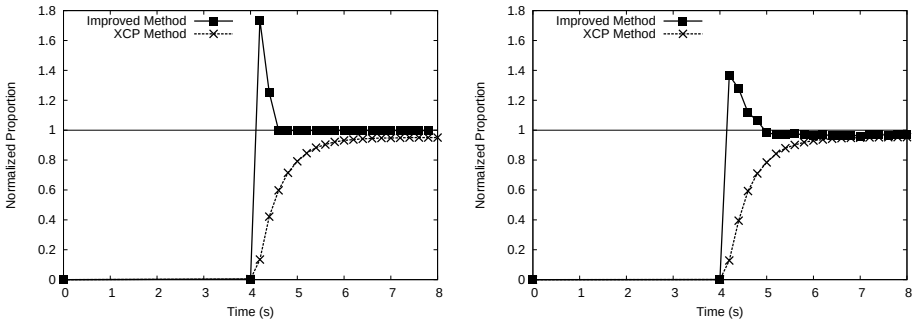


Fig. 6. XCP convergence process

normally near 0, to X_{fair}. The feedback value at XCP routers is calculated in a way that allocates bandwidth from flows whose throughput is greater to flows whose throughput is smaller than fair one. As the Basic Method simply uses this feedback value, the throughput of X_1 can consequently reach $X_{goal-deprio}$, then X_{fair}, and finally $X_{goal-prio}$. Given the situation that the convergence time in the XCP Method is the same for all cases (fair, deprioritizing, prioritizing), it is easy to understand why the convergence time with the Basic Method was faster/slower than that with the XCP Method in the deprioritizing/prioritizing cases.

4.2 Improved Method

The goal of this experiment was to evaluate the performance of the Improved Method described in Sec. 3.2. First, a simple simulation experiment with the same scenario used to evaluate the Basic Method's performance is discussed.



(a) Weight equals 0.5 (b) Weight equals 3

Fig. 7. Convergence time comparison between the Improved Method and the XCP Method

Next, the effect of parameter ε from Eq. (3) on the precision of this method is evaluated. The influence of the number of background flows on the performance of the proposed method is then considered. Finally, the impact of the number of targeted flows is discussed.

Experiment Setup In this experiment, the network topology and other parameters (such as the link bandwidth, delay, packet size, buffer size at router, and α_{dec}) were kept as in the experiment described above.

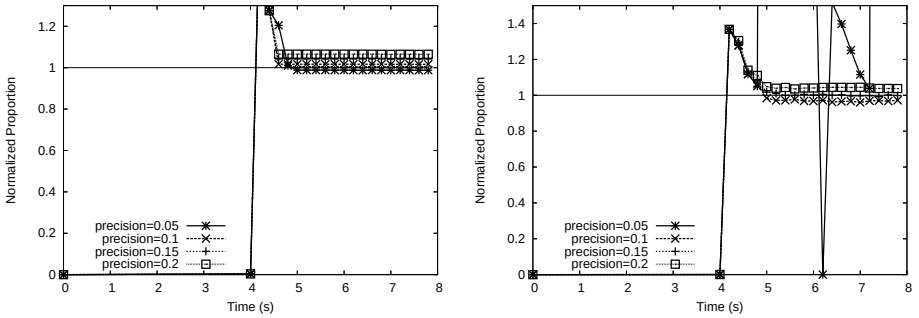
Experiment Results and Analysis In this part, in the coming figures showing the experiment results, the x-axis and y-axis represent the simulation time and the normalized proportion to the desired throughput of a targeted flow, consequently.

Simulation Result in Simple Scenario Figure 7 shows the simulation results with the same scenario as in Sec. 4.1, where 10 background flows were started at 0.1-s intervals and one targeted flow was entered at 4 s. Figures 7(a) and 7(b) show the results when the *weight* was set to 0.5 and 3, respectively.

The Improved Method significantly affected performance in the prioritizing case, i.e., when the *weight* was fixed to 3. Specifically, it shortened the convergence process time by nearly 1.5 s compared with that of the XCP Method. This result confirmed that it is possible to aggressively allocate network bandwidth to a certain flow right from the beginning of connection.

Effect of Precision Parameter The effect of precision parameter ε was evaluated by changing ε in Eq. (3) from 0.05 to 0.1, 0.15, and 0.2. As ε becomes smaller, the method becomes more precise.

Figures 8(a) and 8(b) show the results when weight parameters were fixed to 0.5 and 3, respectively. For ease of evaluation, the normalized throughputs of



(a) Weight equals 0.5 (b) Weight equals 3

Fig. 8. Effect of precision parameter ε to the Improved Method's performance

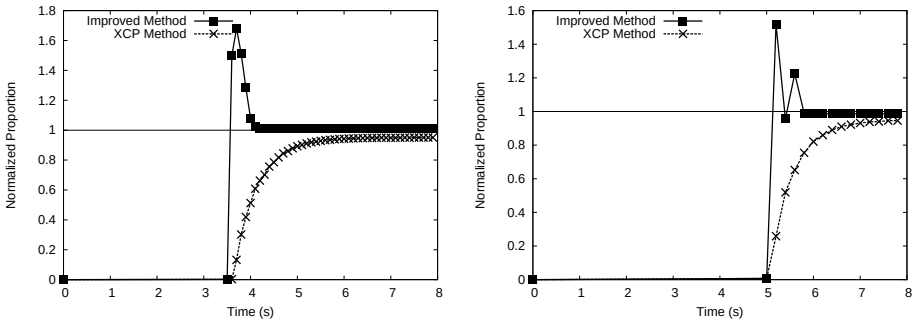
the targeted flows were averaged over a 0.2-s period. Figure 8(a) shows that the proposed method worked well enough in precisely allocating bandwidth to the targeted flow for all cases. However, if the convergence time is also taken into consideration, 0.1 was the best option.

Figure 8(b) shows that the method had trouble ensuring convergence to the goal value when ε was 0.05, which was not the case when the *weight* was 0.5. The reason for this is that the desired value in prioritizing is larger than that in deprioritizing, and this made it harder for the method to converge to the targeted congestion window within that precise range. The best performance was when ε equaled 0.15, but the difference from when ε was 0.1 was not large. Therefore, an optimized range of ε from 0.1 to 0.2 seems reasonable.

For convenience, for the later experiments the precision parameter was consequently fixed to 0.1 or 0.2.

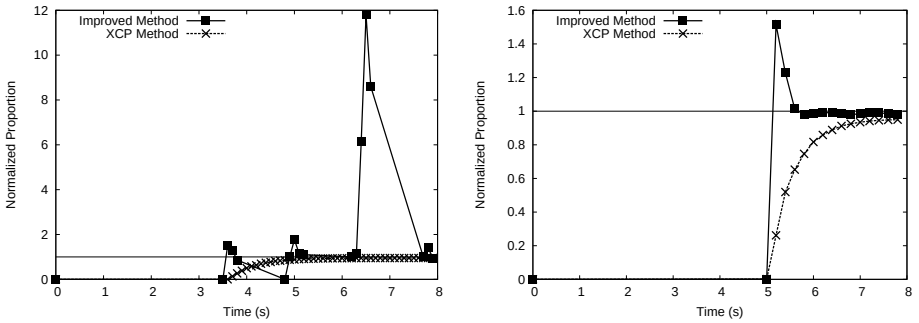
Effect of the Number of Background Flows This experiment was to evaluate the effect of the number of background flows on the performance of the Improved Method. The number of background flows was fixed to 5 or 20, and both the convergence time and the stability of the whole network were taken into consideration. Because of the difference in the number of background flows, the entering time for the targeted flow was approximately 3.5 s or 5 s.

These results are shown in Figs. 9 and 10. Figures 9(a) and 9(b) show the results when *weight* was set to 0.5 and the number of background flows was 5 and 20, respectively. Figures 10(a) and 10(b) describe the results when *weight* was fixed to 3 and the number of background flows was 5 or 20. Figure 9 shows that the method was quite robust in the case of deprioritizing, i.e., when the *weight* was 0.5. The method shortened the convergence time to about 2 s for both cases. This is because the method takes advantage of the convergence process to fair status from normal XCP. However, as shown in Fig. 10(a), the method failed in its attempt to converge the targeted flow's congestion window to the optimized value. To investigate the reason for this result, we further examined the raw data from the simulation result and discovered packet loss at the targeted flow. The



(a) With 5 background flows (b) With 20 background flows

Fig. 9. Effect of the number of background flows to the Improved Method's performance when the weight parameter equals 0.5



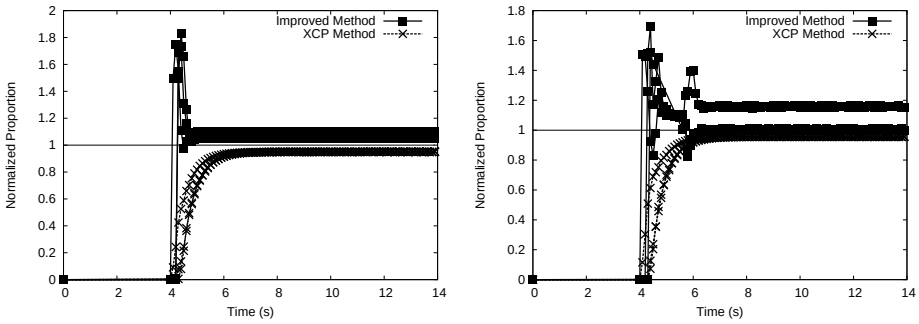
(a) With 5 background flows (b) With 20 background flows

Fig. 10. Effect of the number of background flows to the Improved Method's performance when the weight parameter equals 3

cause of the packet loss was the method suddenly allocating a huge amount of bandwidth to the targeted flow within a short time while the buffer size of routers were fixed as in XCP (i.e., the product of the link's bandwidth and delay). The successful result shown in Fig. 10(b) is consistent with this explanation. In this case, as the number of background flows increased, the optimized value of the targeted flow's congestion window became smaller. Accordingly, packet loss did not occur and the proposed method enabled a convergence time approximately 1 s shorter than that of the XCP Method.

Effect of the Number of Targeted Flows In contrast to the above experiment, in this part we verified the performance of the Improved Method by changing the number of targeted flows to three while keeping the number of background flows the same (i.e., 10 flows).

The results when the *weight* was fixed to 0.5 or 3 are shown in Figs. 11(a) and 11(b), respectively. As in the above experiment, in the prioritizing case the



(a) Weight parameter 0.5 (b) Weight parameter 3

Fig. 11. Convergence time comparison between the Improved Method and the XCP Method with three targeted flows

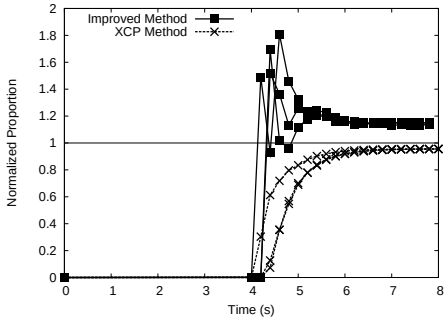


Fig. 12. Convergence time comparison between the Improved Method and the XCP Method with three targeted flows when the buffer size at routers was enlarged

proposed method did not work as well as the XCP Method. As packet loss occurred, it took the last arriving targeted flow about 1 s to re-setup its congestion window and be converged to the optimized value. Hence, the convergence time with the Improved Method was 1 s longer than that with the XCP Method.

As a result, it seems that to get good performance, the Improved Method needs the buffer size at routers to be enlarged to ensure packet loss does not occur. Figure 12 compares results for the Improved Method and the XCP Method when the router buffer size was enlarged by the packet loss quantity, i.e., about 1500 packets. It can be seen that the convergence time with the Improved Method was slightly shorter, by about 0.5 s, than that with the XCP Method.

5 Conclusion and Future Directions

In this paper, we proposed two relative bandwidth allocation methods for use at XCP routers to achieve faster convergence compared to that with the XCP Method. This is implemented by adjusting the targeted flow's congestion windows. We evaluated the two proposed methods through experiments using network simulation. The results for the Basic Method show that it outperformed

the XCP Method for deprioritized targeted flows, but it took noticeably longer to converge a targeted flow's throughput to the desired one. The results also indicated that the Improved Method succeeded in allocating bandwidth to a targeted flow within a shorter convergence time at precision from 10% to 20% with a small ratio of targeted flows. It also was found in the Improved Method that the network's stability could be effected due to allocating large amount of packets at the beginning to targeted flows in prioritizing.

In the future, we would like to work on the following issues. Firstly, the problem of packet loss with the Improved Method needs to be solved by considering the queue status at routers. We would then like to evaluate the proposed method under more realistic conditions than could be obtained in the research reported in this paper. Specifically, the network topology should be extended to more complex scenarios that include a larger number of flows as well as cross flows. In addition, to deal with the scalability of dynamic network, the possibility of implementing the proposals as in DiffServ model should be considered. The bandwidth estimation is brought to edge routers while the converging process is performed mainly in core routers. Lastly, the applicability of the proposed method to hierarchical link-sharing mechanisms should be investigated. Such mechanisms are widely used by ISPs to provide bandwidth management inside their domains.

Acknowledgment

This work was supported in part by Grant-in-Aid for Young Scientists (A) (1) (21680006) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. The Network Simulator ns-2.33. http://www.isi.edu/nsnam/ns
2. 10 Gigabit Ethernet Technology Overview. http://www.intel.com/network/connectivity/resources/doc_library/white_papers/pro10gbe_lr_sa_wp.pdf (2003)
3. Clark, D., Wenjia, F.: Explicit Allocation of Best-effort Packet Delivery Service. IEEE/ACM Trans. Netw. 6(4), 362–373 (1998)
4. Floyd, S.: TCP and Explicit Congestion Notification. ACM SIGCOMM Comput. Commun. Rev. 24(5), 8–23 (1994)
5. Katabi, D., Handley, M., Rohrs, C.: Congestion Control for High Bandwidth-Delay Product Network. In: Proc. ACM SIGCOMM 2002 (2002)
6. Lee, C., Chen, C., Chen, W.: Weighted Proportional Fair Rate Allocations in a Differentiated Services Network. IEICE Trans. Comm. 85(1), 116–128 (2002)
7. Low, S., Paganini, F., Wang, J., Adlakha, S., Doyle, J.: Dynamics of TCP/AQM and a Scalable Control. In: Proc. IEEE INFOCOM 2002. vol. 1, pp. 239–248 (2002)
8. Shimamura, M., Iida, K., Koga, H., Kadobayashi, Y., Yamaguchi, S.: Provisioning Algorithm for Minimum Throughput Assurance Service in VPNs Using Nonlinear Programming. In: IEEE Australasian Telecommunication Networks and Applications Conference (ATNAC). pp. 311–316 (2007)
9. Stoica, I., Shenker, S., Zhang, H.: Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks. ACM SIGCOMM Comput. Commun. Rev. 28(4), 118–130 (1998)