Splitting TCP Connections Adaptively Inside Networks

Masayoshi SHIMAMURA\textsuperscript{†}\textsuperscript{a),} Takeshi IKENAGA\textsuperscript{b),} and Masato TSURU\textsuperscript{†}\textsuperscript{c),} Members

\textbf{SUMMARY} \ The explosive growth of Internet usage has caused problems for the current Internet in terms of traffic congestion within networks and performance degradation of end-to-end flows. Therefore, a reconsideration of the current Internet has begun and is being actively discussed worldwide with the goals of enabling efficient share of limited network resources (i.e., the link bandwidth) and improved performance. To directly address the inefficiency of TCP’s congestion mitigation solely on the end-to-end basis, in this paper we propose an adaptive split connection scheme on advanced relay nodes; this scheme dynamically splits end-to-end TCP connections on the basis of congestion status in output links. Through simulation evaluations, we examine the effectiveness and potential of the proposed scheme.

\textbf{key words:} new generation networks (NwGN), advanced relay nodes, adaptive split connection scheme, TCP, QoS

\section{Introduction}

The explosive growth of Internet usage along with greater diversification of communication technologies and applications have caused serious problems in the current Internet\cite{1,2} and these problems are approaching a threshold beyond which they will become intolerable. The Internet needs to be a flexible and dependable network that efficiently uses limited network resources.

The current Internet and TCP/IP protocol suites, however, evolved through incremental approaches and cannot adequately meet the above requirement due to their traditional constraints; this is the well known end-to-end principle\cite{3}. There are two significant problems in mitigating congestion with the current standard protocol, TCP, in end-to-end approaches. First, TCP cannot respond quickly to the congestion due to delayed congestion feedback along the end-to-end path. This may result in considerable packet losses on the congested link and thus retransmissions over preceding links especially in high-speed and/or long-delay networks. Second, the link utilization of preceding links between a TCP source and congested intermediate nodes may become low because the packet sending rate at TCP source is restricted by the congested bottleneck link on the end-to-end path using TCP congestion control algorithms. Therefore, limited network resources are used inefficiently.

As a step towards future dependable networks, a reconsideration of the Internet has been actively discussed (e.g.,\cite{4}). Many new paradigms for achieving high-performance functionalities have emerged in the form of new generation networks (e.g.,\cite{5}), network virtualization technologies (e.g.,\cite{6–8}), OpenFlow architectures (e.g.,\cite{9}), high-performance and flexible routers (e.g.,\cite{10,11}), and so forth. Along these lines with network virtualization and high-performance and flexible router paradigms, in previous work, we proposed a concept of adaptive network services enabled by advanced relay nodes\cite{12} capable of advanced functionalities (e.g., data cache, adaptive packet compression\cite{13}, etc.) partly located within networks rather than simple IP-forwarding processing.

In the present work, assuming that intermediate nodes with large-capacity storage and high-performance processing ability are appropriately located within networks that is the same assumption in our previous work\cite{12}, we propose an adaptive split connection scheme on advanced relay nodes. In this scheme, each advanced relay node monitors the output queue length as congestion-related information, and then the node splits some TCP connections only if heavy congestion is detected. Notably, the proposed scheme adapts to traffic conditions to deal with dynamic changes of the locations of bottleneck, in contrast to TCP performance-enhancing proxy (PEP) mechanisms (e.g.,\cite{14}) that are statically located and performed at the edges of expected bottleneck links (e.g., satellite and wireless links). By splitting an end-to-end TCP connection into multiple TCP connections arranged in series, unlike enhanced end-to-end approaches (e.g.,\cite{15}), each splitting TCP connection can independently control its packet sending rate by adjusting its congestion window size to each bottleneck link bandwidth. Furthermore, we focus on resource usage efficiency at a time scale much shorter than the extremely large time scales considered in bulk transfer approaches in delay-tolerant networks (e.g.,\cite{16}). Consequently, the proposed scheme directly addresses the inefficiency of TCP’s congestion mitigation solely on the end-to-end basis, that is, the scheme can respond immediately to the congestion and avoid unnecessary packet losses; and can help the TCP flows traversing the congested node to release the preceding links earlier for efficient use by other TCP flows. Thus the proposed scheme is expected to flexibly improve network resource efficiency and end-to-end performance.

The remainder of this paper is organized as follows.
Section 2 explains the adaptive split connection scheme. Section 3 examines the effectiveness and potential of the proposed scheme through simulation evaluations. Section 4 concludes the paper.

2. Adaptive Split Connection Scheme

To resolve issues in terms of network resource efficiency and improved end-to-end performance, we propose an adaptive split connection scheme. In this study, we assume that, along the lines of research on network virtualization and in-network intelligence with new router architecture concepts for future Internet, certain intermediate nodes within networks have large-capacity storage and high-performance processing ability. Regarding such a router architecture, some prototype implementation was reported to achieve high-performance by exploiting a separation of fast and slow forwarding and processing elements [10]. Note that we do not assume that all intermediate nodes have sufficient computational and/or storage resources. Instead, such special intermediate nodes are partly and appropriately located within networks.

In the proposed scheme, an advanced relay node splits an end-to-end TCP connection on the basis of congestion-related information at its output queue. Congestion at the advanced relay node means that a bottleneck link exists between the advanced relay node and receiver on an end-to-end path; this results in unused network resources between the sender and advanced relay node on the end-to-end path because the TCP adjusts its congestion window size to the available bottleneck link bandwidth. Our split connection strategy is aimed at efficiently utilizing the unused network resources through split TCP connections. Note that the main difference of the proposed scheme from existing TCP PEP mechanisms is that it can adapt to congestion-related information dynamically and quickly at advanced relay nodes within networks.

Figure 1 compares end-to-end TCP and the proposed scheme. Figure 1 (a) represents an example of a topology and scenario where TCPS T_1 and T_2 transfer data from S to D_1 and from S to D_2, respectively. At the start of transfer, links L_1 and L_2 have the same capacity. Then L_2 becomes a bottleneck due to starting congestion (e.g., caused by other background traffic on L_2 that consumes half of the capacity of L_2). After T_1 starts at time t_0, L_2 is congested from time t_1, and then T_2 starts at time t_2. In the case of end-to-end TCP depicted in Fig. 1 (b), T_1 can use all the available bandwidth until t_1. Then, T_1 decreases its packet sending rate because L_2 is congested from t_1; this results in unused link bandwidth in L_1 after t_1. Consequently, T_1 cannot complete data transmission before T_2 starts at t_2 due to the bottleneck of L_2.

Figure 1 (c) depicts the behavior of the proposed scheme. Advanced relay node R detects congestion after L_2 is congested from t_1. R splits T_1 at time t_1 + t_δ. Thus, T_1 adjusts its congestion window size to link bandwidth L_1 rather than L_2. Therefore, the unused link bandwidth of L_1 can be efficiently utilized from t_1 + t_δ. Consequently, competition between T_1 and T_2 at L_1 can be avoided because T_1 can complete data transmission before t_2.

Figure 2 shows the behavior of the proposed method in terms of a time sequence. The proposed scheme determines to split a TCP connection when the output queue length exceeds a pre-specified threshold. First, advanced relay node R sends a SPLIT control packet to source node S. If S accepts the split connection, S then stops data transmission temporarily and sends a REQUEST control packet to destination node D. Next, D sends a REPLY control packet to S if D also accepts the split connection. Finally, S resumes data transmission when S receives the REPLY control packet.
packet. Consequently, the TCP connection can be split into two TCP connections although this procedure imposes an additional delay overhead for coordination among R, S, and D, especially in cases with a long round-trip time between S and D. Note that the split TCP connection between S and R or one between R and D can be split again if intermediate advanced relay nodes detect any other congestion.

After this procedure, the TCP connection is split into one between S and R and one between R and D. When R receives packets from S, R stores them. Then R sends the stored packets to D in accordance with TCP congestion control between R and D. Thus, both TCP connections are independently established and each TCP also independently controls its congestion window size. Consequently, the proposed scheme improves the network resource efficiency because the TCPs can efficiently utilize unused link bandwidth.

3. Simulation Evaluation

In previous sections, we mentioned the issues regarding network resource inefficiency caused by traditional end-to-end TCP and proposed the adaptive split connection scheme. To test its potential, we evaluated the scheme through simulations.

First, we compare traditional TCP and the proposed scheme. Figure 3 depicts a simulation topology where source node S communicates with destination node D by TCP with a bulk transfer application via multiple advanced relay nodes R_i. The bottleneck link between S and D is link L_3 where a congested link is simulated. In the proposed scheme, the end-to-end TCP connection is split at R_2 into two TCP connections on the basis of congestion-related information. The bottleneck link between S and R_2 then becomes link L_2. Therefore, TCP connection between S and R_2 is also split at R_1 into two TCP connections. Regarding parameters, we set a threshold for split connection equal to 50% of the queue length, 20 packets of buffer size at the output queue of L_2, and a TCP congestion control algorithm of TCP SACK implemented in simulator ns-2.33 [17].

Figure 4 shows a simulation result. The x-axis and y-axis represent the elapsed time of simulation and the sequence number of TCP packets received at each node, respectively. First, the end-to-end data transfer of 10,000 packets finishes at 12.04 seconds. In this scenario, it means all the links L_i are also consumed during the 12.04 seconds (hereafter referred to as the bandwidth consumption period). In contrast, since the proposed scheme split TCP connections, data transfers toward R_1, R_2, and D are finished at 1.27, 2.48, and 12.03 seconds, respectively. Therefore, the bandwidth consumption period at L_1 and L_2 is reduced by 89.45% and 79.40%, respectively. Consequently, network resource efficiency is improved.

Next, we examine the performance of the proposed scheme. Figure 5 depicts a simulation topology similar to that of Fig.3. In this scenario, the end-to-end TCP connection is split at R. We vary the link bandwidth b from 10 to 100 Mb/s and the propagation delay time d from 1 to 20 ms. Regarding other parameters as well as the previous scenario, we use the same threshold for split connection, the same buffer size at the output queue, and the same TCP congestion control algorithm. As an evaluation index in this analysis, we focus on the throughput calculated from the data transfer completion time at D.

The impact of link bandwidth is shown in Fig. 6. In the range between 10 and 80 Mb/s (x-axis), the proposed scheme splits the TCP connection and improves throughput compared to that with the end-to-end TCP. Since the end-to-end TCP increases its congestion window size until packet loss detection regardless of congestion, it degrades throughput performance. On the other hand, the proposed scheme splits the connection into two TCP connections on the basis of congestion detection. Since the two TCPs independently...
adjust their congestion window size, they can reduce packet losses.

The impact of delay time is shown in Fig. 7. As the propagation delay time increases (x-axis), each throughput performance is degraded because the TCP feedback loop between sender and receiver nodes lengths. However, in comparison with the end-to-end TCP, the proposed scheme achieves better performance throughout the range. This shows that the proposed scheme can improve the performance, by splitting connections adaptively, even if the delay time is long.

Through the simulation evaluations, we have confirmed the effectiveness of the proposed scheme. In particular, even if congestion occurs due to a bottleneck link, the proposed scheme can improve end-to-end throughput performance and efficiently use the link bandwidth with a short bandwidth consumption period by using split TCP connections on the basis of congestion-related information.

4. Concluding Remarks

Along the lines of research on in-network intelligence with sufficient storage in future Internet, we have proposed the adaptive split connection scheme that dynamically splits end-to-end TCP connections on the basis of congestion status in output links on advanced relay nodes. Note that the proposed scheme is recursive in a sense that it can handle more than one congested link along the path of a TCP flow. The proposed scheme directly addresses the inefficiency of TCP’s congestion mitigation solely on the end-to-end basis, and the preliminary simulation evaluations have shown the effectiveness and potential of the proposed scheme. When various types of TCP flows, including long-lived and short-lived flows and/or bulk transfer and real-time application flows, traverse the congested link, the appropriate selection of TCP flows to be split is essential but remains as future work. We are planning to implement the proposed scheme on real machines and perform experiments on testbeds.

Acknowledgment

This work was partly supported by the National Institute of Information and Communications Technology, and the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (B) (No. 21300024).

References

[1] Cisco Systems, Inc., “Cisco visual networking index: Forecast and methodology, 2008–2013,” June 2009. Information available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf
[2] L. Qiu, Y.R. Yang, Y. Zhang, and S. Shenker, “On selfish routing in Internet-like environments,” IEEE/ACM Trans. Netw., vol.14, no.4, pp.725–738, Aug. 2006.
[3] J. Saltzer, D. Reed, and D. Clark, “End-to-end arguments in system design,” ACM Trans. Comput. Syst., vol.2, no.4, pp.277–288, Nov. 1984.
[4] J. Rexford and C. Dovrolis, “Future Internet architecture: clean-slate versus evolutionary research,” Commun. ACM, vol.53, no.9, pp.36–40, Sept. 2010.
[5] National Science Foundation (NSF), “FIND: NSF NeTS future internet design initiative,” Information available at http://www.nets-find.net/
[6] A. Nakao, “Network virtualization as foundation for enabling new network architectures and applications,” IEICE Trans. Commun., vol.E93-B, no.3, pp.454–457, March 2010.
[7] N. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” IEEE Commun. Mag., vol.47, no.7, pp.20–26, July 2009.
[8] N. Feamster, L. Gao, and J. Rexford, “How to lease the Internet in your spare time,” ACM SIGCOMM Comput. Commun. Rev., vol.37, no.1, pp.61–64, Jan. 2007.
[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” ACM SIGCOMM Comput. Commun. Rev., vol.38, no.2, pp.69–74, April 2008.
[10] J. Turner, P. Crowley, J. Dehart, A. Freestone, B. Heller, F. Kuhms, S. Kumar, J. Lockwood, J. Lu, M. Wilson, C. Wiseman, and D. Zar, “Supercharging PlanetLab - High performance, multi-application, overlay network platform,” Proc. ACM SIGCOMM07, Aug. 2007.
[11] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. Kaashoek, “The click modular router,” ACM Trans. Comput. Syst., vol.18, no.3, pp.263–297, Aug. 2000.
[12] M. Shimamura, T. Ikenaga, and M. Tsuru, “Advanced relay nodes for adaptive network services — concept and prototype experiment,” Proc. Second International Workshop on Network Traffic Control, Analysis and Applications (NTCAA-2010), Nov. 2010.
[13] M. Shimamura, H. Koga, T. Ikenaga, and M. Tsuru, “Compressing packets adaptively inside networks,” IEICE Trans. Commun., vol.E93-B, no.3, pp.501–515, March 2010.
[14] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, “Performance enhancing proxies intended to mitigate link-related degradations,” RFC 3135 (Informational), June 2001. Information available at http://www.ietf.org/rfc/rfc3135.txt
[15] S. Floyd, M. Allman, A. Jain, and P. Sarolathi, “Quick-Start for TCP and IP,” RFC 4782 (Experimental), Jan. 2007. Information available at http://www.ietf.org/rfc/rfc4782.txt
[16] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram, “Delay tolerant bulk data transfers on the Internet,” Proc. 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS2009), pp.229–238, June 2009.
[17] VINT Project, “The network simulator — ns-2.” Information available at http://www.isi.edu/nsnam/ns/