4 High-speed Transmission and Interoperability Technology

4-1 Transport Protocols for Fast Long-Distance Networks: Comparison of Their Performances in JGNⅡ

KUMAZOE Kazumi, KOUYAMA Katsushi, HORI Yoshiaki, TSURU Masato, and OIE Yuji

The majority of network applications adopt the Transmission Control Protocol (TCP) as the transport layer protocol on IP networks. This is because the TCP has important two functions; one is the error control function providing a reliable, error free data transmission and another is the congestion control mechanism realizing a modest sharing of network resources. However, the current TCP is not always suitable for applications which require highly reliable and high speed transfer of a huge amount of data over long haul networks, such as in the Grid Computing environment. Therefore, various new transport protocols have been proposed with the aim of the efficient use of abundant resources in fast long-distance networks. In this paper, we discuss the results of throughput characteristics of some practical high-speed transport protocols on JGN (Japan Gigabit Network)Ⅱ.

Keywords
High-speed transport protocol, Fast long-distance network, Testbed

1 Introduction

The majority of network applications have adopted Transmission Control Protocol (TCP) as the transport layer protocol on IP networks. However, the current TCP (Standard TCP) does not provide efficient data transmission over fast long-distance networks. This is because Standard TCP restricts the window size at the start of the flow (i.e., "slow start") and performs flow control and congestion control based on the Round Trip Time (RTT). For example, to fill up 10-[Gbps] in a single connection where the Maximum Transmission Unit (MTU) is 1,500 [bytes] and the RTT is 100 [ms], Standard TCP connection would have for an hour and forty minutes without a single packet loss, which is unrealistic[1]. On the other hand, with the increasing bandwidth of core networks, applications demanding long-distance high-speed data transmission on fast long-distance networks have become a reality (as seen in grid applications, for example). These applications demand high-speed, reliable transmission of large amounts of data, and these demands have in turn led to various proposals for high-speed transport protocols capable of efficient data transmission, to replace Standard TCP.

This article shows the examples of the

experimental results conducted on JGNⅡ using existing high-speed transport protocols. Section **2** introduces the targeted high-speed transport protocols, Section **3** illustrates the experimental environment, and Section **4** describes the experimental results.

2 High-speed transport protocols

The targeted high-speed transport protocols based on Standard TCP are: (1) HSTCP[1], (2) Scalable TCP[2], (3) FAST TCP[3], (4) BIC[4], and (5) HTCP[5]. UDT[6] is an example of a protocol developed based on the UDP protocol.

Protocols (1)–(5) are required to be installed only on the sender side machine, while Protocol (6) is required to be installed on both the sender and receiver machines. These protocols are implemented by several researchers and we conducted an experiment using these implementations.

HSTCP and Scalable TCP change their congestion window size (cwnd) depending on the AIMD algorithm with the following equations, the same as the current TCP (which we will hereinafter refer to as the Standard TCP (RFC2581: TCP Congestion Control)) during the congestion avoidance phase, where a is the increase parameter and b is the decrease parameter. For the Standard TCP, the values of a and b are 1/cwnd and 0.5, respectively.

When ACK is received:

$$\text{cwnd} = \text{cwnd} + a * \text{MSS} \qquad (1)$$

When packet loss is detected:

$$\text{cwnd} = \text{cwnd} * (1 - b) \qquad (2)$$

The HSTCP behaves identically to the Standard TCP for a small cwnd, but in the area of window size larger than a threshold (for example, 38 MSS), cwnd is governed by a modified AIMD algorithm, where a and b vary in a complex way depending on the current value of the cwnd. Scalable TCP also modifies the characteristics of AIMD behavior of the Standard TCP and has a threshold window size (the default size is 16 MSS). When the cwnd exceeds the threshold, it will be updated using a=0.01 and b=0.125.

BIC updates its cwnd based on binary search and additive increase functions. In other words, cwnd is increased through Additive Increase until packet loss is detected. When packet loss is detected, BIC-TCP assumes that available bandwidth is nearby and performs a binary search, aiming at convergence with the available bandwidth.

In contrast to the above three protocols, which update cwnd on detection of packet loss, HTCP determines the increase in the AIMD parameter as a function of the time elapsed after the last packet loss. In other words, taking Δ as the elapsed time following detection of the last occurrence of congestion, the HTCP protocol updates the parameter “a” in the AIMD algorithm to a(Δ).

FAST TCP protocol updates its cwnd according to Equation (3), using packet loss and the RTT of the packet. For every RTT,

$$cwnd = cwnd(baseRTT \,/\, avgRTT) + \alpha \qquad (3)$$

Here, baseRTT is the measured minimum RTT, and avgRTT is the average RTT.

The UDT protocol increases the reliability of the standard UDP protocol to enable high-speed data transmission. UDT operates in the application layer and is based on assumed use in an environment with a limited number of flows. UDT measures the available bandwidth using packet pairs and applies rate-base control, which adaptively varies the transmission rate based on information on available bandwidth.

The performance of these protocols has been already studied and reported on through various experiments on testbeds globally. However, a wide variety of characteristics of these protocols in more realistic situations arising in actual shared network environments have not been fully covered so far, and thus should be investigated, with an aim at practical use in the current/future Internet. As the bandwidth of the Internet increases (both in access networks and core networks) users may come to want to transfer larger amounts of data using the above-mentioned protocols, regardless of the intention of the original developers of those protocols. Originally, these high-speed transport protocols were not oriented toward use in networks like the Internet, where various kinds of flows coexist. And there have been no experiments or evaluations assuming these circumstances thus far.

In this paper, therefore, we investigate the throughput characteristics of several promising high-speed transport protocols: HighSpeed TCP (HSTCP), Scalable TCP, FAST, BIC, CUBIC, HTCP and UDT, through experiments on the Japan Gigabit Network (JGNⅡ), an Ethernet-based open testbed network over Japan to be adopted for the next-generation Internet. Supposing these high-speed transport protocols will be adopted for next-generation Internet, it is important to consider their performance, focusing especially on (i) how changes in network conditions (e.g., the amount of background traffic, bandwidth, propagation delay, packet loss, and packet mis-ordering) affect the performance of high-speed transport protocol flows; and (ii) how the high-speed transport protocol flows affect the performance of the coexisting multiple heterogeneous flows, and vice versa.

3 Experimental environment

Figure 1 shows the experimental environment. Two types of JGNⅡ paths were employed in the experiment: domestic and international. Table 1 lists the characteristics of each path[1]. The experiments made use of the network emulators shown in Fig.1 (b) and (c) to test the effects of delay variation and path switching.

Linux was adopted as the operating system (OS) for the sender and receiver side end host terminals. We selected Linux because protocol implementations are provided as patch codes for the Linux kernel (except for the case of UDT). Throughput was used as the performance measure.

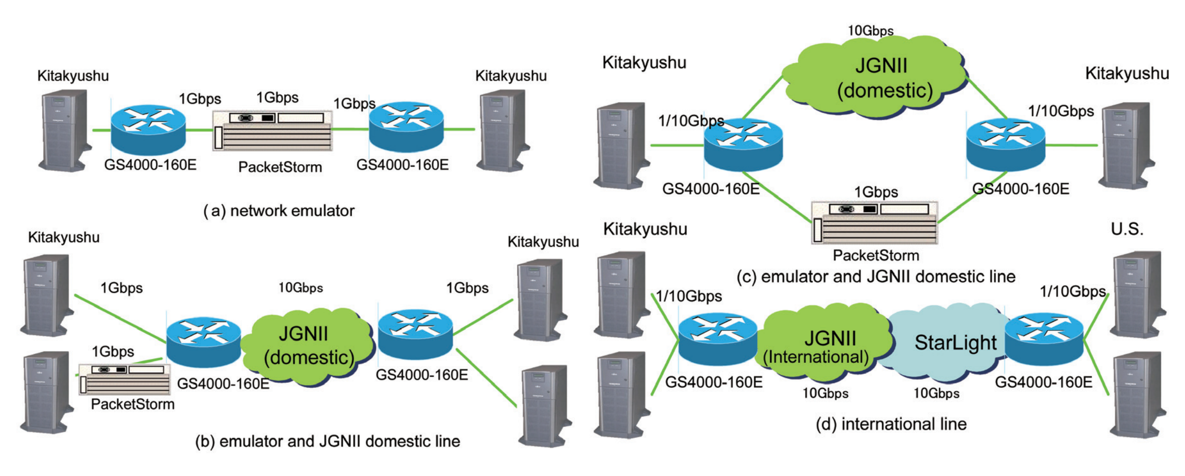


Fig.1 Network configuration

Table 1 Types of experimental paths

	RTT[ms]	Bandwidth[Gbps]
Network Emulator	0–10000	1
JGNII domestic path	38	1
JGNII international path	180	1

---

[1] Although the available bandwidth in JGNⅡ is 10 [Gbps], this article reports on experiments conducted in a 1-[Gbps] environment. Experiments in the 10-[Gbps] environment will be scheduled at a later date.

4 Results of experiments

4.1 Basic characteristics

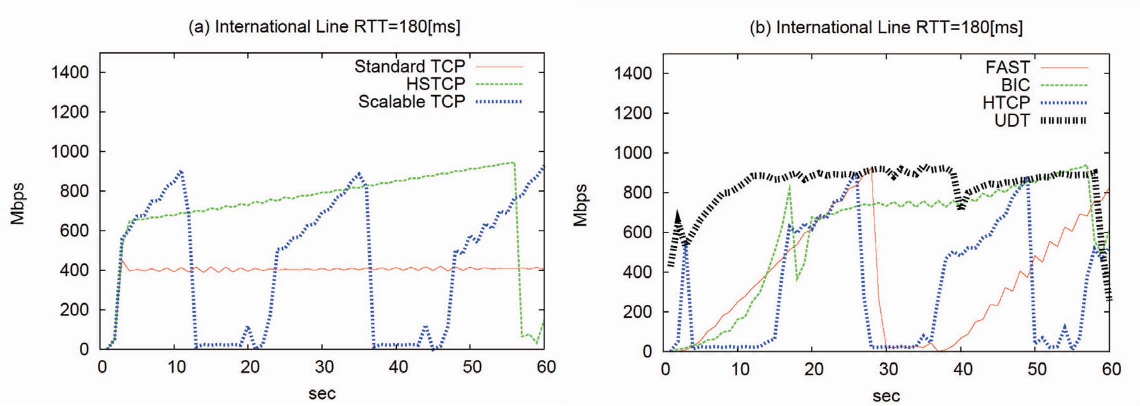
Figure 2 shows the throughput characteristics observed in the JGNⅡ international line with the use of a single protocol.

As shown in Fig.2 (a), the standard TCP, HSTCP, and Scalable TCP exhibit very similar throughput behavior in the slow start phase and can reach their maximum throughput. In the congestion-avoidance phase after the occurrence of packet losses, however, the throughput of the standard TCP increases very slowly, halving each time a packet loss is detected by a triple-duplicated ACK, and thus is finally reduced to a point where it is extremely small, while HSTCP and Scalable TCP are able to recover their maximum throughput again after a while, mainly due to their aggressiveness in increasing throughput. As shown in Fig.2 (b), FAST, which is able to control the congestion window size by measuring RTT variation (as queueing delay), demonstrates a rapid recovery of throughput to its maximum after a drastic decrease in throughput. On the other hand, BIC increase and decrease throughput gently, by controlling its congestion window size in a binary search manner. UDT, a user-level protocol implemented over the standard UDP, seems to stably keep a high throughput during the run time, by controlling its sending rate based on estimation of the available bandwidth of the end-to-end path and reception of NACK packets.

Flows with different RTTs may coexist in a network, including the high-speed Internet. Figure 3 shows the throughput characteristics when each protocol flow has different RTTs. Two flows—Flow 1 (RTT1 = 38 [ms]) and Flow 2—were established in the configuration illustrated in Fig.1 (b), and the RTT of Flow 2 was varied as RTT1, 2*RTT1, and 4*RTT1. Figure 3 shows the average throughput rate for each flow. As shown in Fig.3, when the RTT ratio is 1:1, the throughput values of the two flows were the same in all the protocols. However, when the RTT ratio was 1:2, FAST and BIC flows showed greater fluctuations than seen with the 1:2 ratio. When the RTT ratio was 1:4, HSTCP, FAST, and BIC flows showed greater fluctuations than seen with the 1:4 ratio. In contrast, with HTCP, and UDT throughput characteristics did not vary with different RTT ratios.

4.2 Response to change in network conditions

We investigate how the dynamic changes of network conditions (RTT and bottleneck bandwidth) affect the throughput characteristics of each of the high-speed transport protocols, by switching the paths between two routers (an emulator-path and a JGNⅡ domestic region path) on the network as illustrated in Fig.1(b). In this subsection, we set a large



(a) Standard TCP, HSTCP, Scalable TCP Protocol

(b) FAST, BIC. HTCP, UDT Protocol

Fig.2 *Throughput characteristics of a single connection (JGNⅡ international line)*

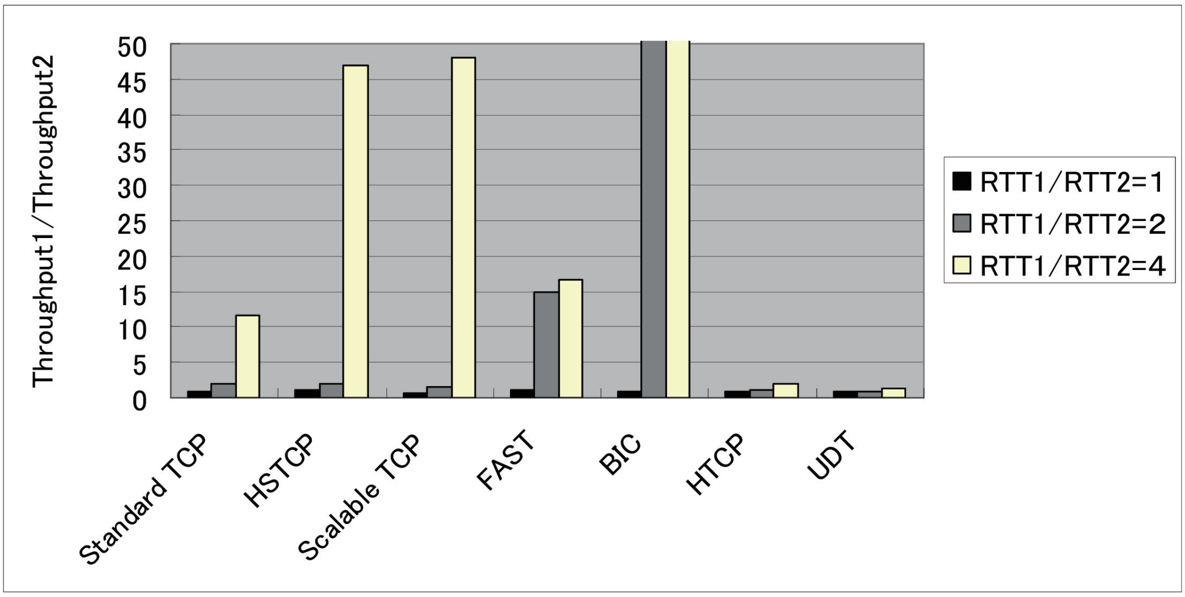
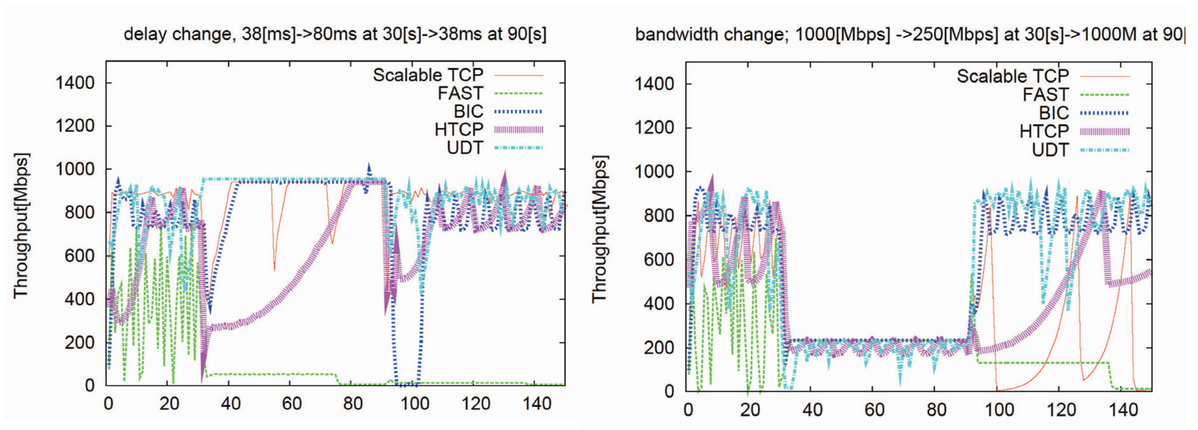


Fig.3 *Throughput characteristics with coexisting flows of different RTTs*

socket buffer size resulting in the maximum cwnd approximate to the BDP for RTT of 80 [msec]. Figure 4(a) shows the throughput behavior of a single flow in a case where the path is switched from the original path (the JGNⅡ domestic region path) with 38 [msec] of RTT to the alternative path (the emulator-path) with 80 [msec] of RTT 30 seconds after the start of the flow, with the path switched back to the original one 60 seconds after the first switching. At the moment of change to the alternative path with a longer RTT, the throughput of each of the TCP-based protocols decreases considerably, although that of UDT seems unaffected by this change. FAST suffers the largest drop in its throughput until the path is switched back to the original one due to the nature of its RTT-based congestion control, while BIC and HSTCP are able to recover their throughput (BIC is quicker than HSTCP). On the other hand, at the moment of return to the original path with a shorter RTT, the throughput of both UDT and the TCP-based protocols (except for FAST) rapidly decreases to near zero and then quickly recovers its original behavior. Figure 4(b) shows the throughput behavior in a case where the path is switched from the original path with 1 [Gbps] of bottleneck bandwidth to an alternative path with 0.25 [Gbps] of bottleneck bandwidth 30 seconds after the start of the flow, with the path switched back to the original one 60 seconds after the first switching. All of the high-speed transport protocols seem to be able to appropriately adapt their throughput for such changing of the bandwidth. In particular, the adaptation of UDT seems very quick, which may be due to its rate-control, based on measurements of the available bandwidth of the end-to-end path. On the other hand, FAST behaves in a conservative manner, and accordingly throughput is lower than that of the other protocols.

4.3 Characteristics when different protocol flows exist simultaneously

Next we will describe the characteristics for cases in which different high-speed transport protocol flows coexist in the same path. Table 2 shows the total throughput with the establishment of two different protocol flows. The protocol combinations are indicated in color in the table where total throughput of two different protocol flows was smaller than the total throughput for two protocol flows of a single type.



(a) Path Switching (RTT [msec] = 38 → 80 → 38)

(b) Path Switching (Bandwiath [Mbps] = 1000 → 250 → 1000)

Fig.4 *Effects of path switching on throughput characteristics*

For example, the average total throughput for two HSTCP flows was 720 [Mbps], and the average total throughput for two Scalable TCP flows was 492 [Mbps]. On the other hand, HSTCP flow plus a Scalable TCP flow yielded an average total throughput of 381 [Mbps], which is less than either total throughput value with two protocol flows of the same type. These results indicate that different protocol flows have mutual effects on the respective throughput characteristics when present at the same time. As the table indicates, some combinations of high-speed transport protocols reduce total throughput, impairing the performance of remaining protocol flows.

5 Conclusions

In this paper, we have reported the preliminary results of an experiment on several promising high-speed transport protocols on JGNII (the successor of JGN), a new high-speed testbed network set up across Japan. We focused especially on (i) how the changes in network conditions affect the performance of the high-speed transport protocol flows; and (ii) how the high-speed transport protocol flows affect the performance of coexisting multiple heterogeneous flows, and vice versa, which are of practical importance for use over the Internet. We are conducting more extensive experiments and preparing to further examine the throughput characteristics of each of the target protocols on a 10 [Gbps] end-to-end path. Throughout our experiment so far, each of these recently proposed high-speed

Table 2 *Total throughput characteristics when two different high-speed transport protocol flows coexist simultaneously*

	HSTCP	Scalable	FAST	BIC	HTCP	UDT
HSTCP	720	381	319	535	600	814
Scalable	381	492	383	495	625	825
FAST	319	383	402	301	745	790
BIC	535	495	301	711	287	744
HTCP	600	625	745	287	489	580
UDT	814	825	790	744	580	749

transport protocols seems to have some weakness when employed in the global Internet. Each of them has its own objectives and particular scenarios in which the protocol is effectively used. Therefore, we need to further investigate and develop general purpose high-speed transport protocols suitable for deployment over the global high-speed Internet.

Acknowledgements

We would like to thank Yunhong Gu, Michal Sabana, David Turkington, and Richard Grossman for their help in conducting our experiments and their useful comments on UDT. We also thank to Cheng Jin for his useful comments on FAST, and Injong Rhee and Lisong Xu for the information they provided us on BIC and CUBIC. We are grateful to the staff at APAN/JGNⅡ NOC for their network operations.

References

1 S.Floyd, “HighSpeed TCP for Large Congestion Windows”, RFC 3649. Experimental. Dec. 2003.
2 Scalable TCP : http://www-lce.eng.cam.ac.uk/~ctk21/scalable
3 FAST : http://netlab.caltech.edu/FAST
4 BIC :http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/index.htm
5 HTCP : http://www.hamilton.ie/net/htcp
6 UDT : http://ust.sourceforge.net
7 R.L.Cottrell et al., “Characterization and Evaluation of TCP and UDP-based Transport on Real Networks”, PFLDnet2005, Feb. 2005.
8 Experimental Evaluation of TCP Protocols, http://www.hamilton.ie/net.eval
9 K.Kumazoe, K.Kouyama, Y.Hori, M.Tsuru, and Y.Oie, “Transport Protocols for Fast Long- Distance Networks : Evaluation of Their Penetration and Robustness on JGNII”, PFLDnet 2005, Feb. 2005
10 Kazumi Kumazoe, Katsushi Kouyama, Yoshiaki Hori, Masato Tsuru, and Yuji Oie, “Transport Protocols for Fast Long-Distance Networks; Comparison of Their Performances in JGNII”, IEICE Technical Report, IN2005-67, pp.37-42, Sendai, Japan, Jul. 2005.

KUMAZOE Kazumi

Expert Researcher, Kitakyusyu JGNⅡ Research Center, Collaborative Research Management Department

Network Performance Evaluation

KOUYAMA Katsushi

Guest Researcher, Kitakyusyu JGNⅡ Research Center, Collaborative Research Management Department

Network Performance Evaluation

***HORI Yoshiaki**, Dr. Eng.*

Guest Researcher, Kitakyusyu JGNⅡ Research Center, Collaborative Research Management Department

Computer Network Engineering

***TSURU Masato**, Dr. Eng.*

Expert Researcher, Kitakyusyu JGNⅡ Research Center, Collaborative Research Management Department

Performance Measurement, Modeling and Analysis of Computer

***OIE Yuji**, Dr. Eng.*

Collaborative Research Management Department JGNⅡ Project Leader

Information Network Engineering