http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (4): 600-604, 2009
ISSN 1812-5638

Performance Comparison of UDP-based Protocols Over Fast Long Distance Network

[1,2]Yongmao Ren, [1]Haina Tang, [1]Jun Li and [1]Hualin Qian
[1]Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
[2]Graduate University of Chinese Academy of Sciences, Beijing, China

Abstract: As massive data generated in large scale e-Science projects such as High Energy Physics (HEP) and astronomical observation (e-VLBI) needs to be transported internationally over fast long distance network, high performance transport protocol is needed. Based on UDP, some reliable transfer protocols are designed. This research mainly studies the principles of these protocols and compares their performance by experiments. It is found that they far outperform TCP, but still have some limitations and can't satisfy the requirement of bulk data transfer perfectly.

Key words: Fast long distance network, RBUDP, TCP, Tsunami, UDT

INTRODUCTION

There are many large scale e-Science projects such as LHC particle accelerator in High Energy Physics (HEP), ITER experiment in Fusion Energy Science and e-VLBI in astronomical observation fields. During these projects, the movement of massive amounts of data between labs, facilities and instruments distributed all over the world are becoming common. These e-Science applications have high requirement for network performance on bandwidth, stability and reliability and so on. Other emerging applications like HDTV transfer and e-Health also require high speed data transfer over long distance.

To satisfy the high requirement of these applications, high performance network transport is needed. First, high speed network is needed. Traditional packet-switched IP network is difficult to satisfy this requirement. New emerged circuit-switched optical network technology is being used to build Fast Long-Distance network (FLDnet). There are many optical FLDnets and related projects now, such as Dynamic Resource Allocation via GMPLS Optical Network (DRAGON), Hybrid Optical and Packet Infrastructure (HOPI), the Global Ring Network for Advanced Applications Development (GLORIAD) and User Controlled Light Paths (UCLP) and so on. End-to-end lighpath can provide high bandwidth and QoS guarantee for these special applications (Ren, 2008a). Second, high performance transfer protocol is needed. Traditional TCP protocol performs poorly over FLDnet and UDP is not reliable. So, research on new transfer protocol has become a hot research topic.

In this study, the issue of bulk data transfer over FLDnet is investigated. After identifying the shortcoming of TCP and TCP variants, the principle of several UDP-based transfer protocols are studied.

BULK DATA TRANSFER ISSUE

Many applications need to transfer bulk data, but their requirements are not completely the same. A basic and general requirement is high speed. Most applications require stable and reliable transfer. Some applications like e-VLBI and HDTV can tolerate few data loss. And some applications like e-VLBI require real-time transfer, but others like HEP don't require.

Traditional TCP performs very poorly over FLDnet. The main reason is its conservative congestion control mechanism like slow start and congestion avoidance and flow control mechanism (complete analysis is given by Ren *et al*. (2008b)). UDP can transfer fast over FLDnet, but it's unreliable. Furthermore, most application level programs adopt TCP. Modification on TCP generally does not result in that the application programs need to be modified. So, many researchers focus on modification on TCP to enhance TCP performance on high speed network. Up to now, several enhanced TCP protocols have been proposed, such as High Speed TCP (HSTCP) (Floyd, 2003), Binary Increase Congestion TCP (BICTCP) (Xu *et al*., 2004), FAST TCP (Jin *et al*., 2004), eXplicit Control Protocol (XCP) (Dina *et al*., 2002) and Variable-structure congestion control Protocol (VCP) (Yong *et al*., 2005) and so on. These TCP variants mainly make modification on TCP's AIMD congestion control algorithm to make TCP congestion window size increase

Corresponding Author: Yongmao Ren, Computer Network Information Center, Chinese Academy of Sciences, Beijing, China

more quickly and reduce recovery time from loss. Some TCP variants like XCP and VCP modify TCP's congestion detection and notification scheme. However, they usually need router to support. So, it's difficult to deploy them. Researchers have evaluated the performance of these TCP variants by NS2 simulation. The results show that most of the TCP variants have higher throughputs than the standard TCP, but when the Round Trip Time (RTT) is high (such as RTT >300 m sec), their throughputs drop quickly. So, these TCP variants still can not satisfy the requirement of bulk data transfer over FLDnet.

Among practical applications, some application programs like GridFTP can use parallel TCP streams to transfer. By running multiple parallel TCP streams, the throughput can be improved obviously. But too many parallel TCP socket connections increase the handling complexity in end system. Under the limitation of the end system performance, this method is still not perfect. What's more, some applications like e-VLBI can only use a single TCP stream. Another method is using large MTU, but it needs end systems and all the routers between them to support Jumbo Frame and, large packet also increases the probability of packet error result from line error. Additionally, adjusting TCP buffer size is also a good method. By experiment, we find that if there is no packet loss, by adjusting TCP buffer size, the throughput can be 361.28 Mbps on 1000 Mbps link. But, even if there is very small loss rate, the throughput drops quickly. In practice, it's hard to guarantee no loss rate at all.

In conclusion, the performance of TCP and TCP variants is still not good enough to satisfy the requirement of bulk data transfer over FLDnet.

A QUICK REVIEW OF UDP-BASED PROTOCOLS

Up to now, most researchers focus on modification on TCP. But, there are still some researchers try to design new protocols based on UDP. UDP is much simpler than TCP. It merely adds ports to identify individual application processes and a checksum to detect erroneous packets and simply discard them. UDP has no reliable transfer mechanism like sequence number, ACK and retransmission. Additionally, UDP makes no flow control and congestion control like TCP. So, it runs fast over FLDnet. But it's unreliable. If reliability is added into UDP and the merit of fast is kept, it will satisfy the requirement of many e-Science applications.

There is no protocol directly modified on UDP known to the authors. While, based on UDP, several new transfer protocols with reliability are designed, such as Reliable Blast UDP (RBUDP) (He *et al*., 2002), UDP-based Data Transport (UDT) (Yunhong *et al*., 2004, 2007) and Tsunami (Mark, 2002) and so on. They are usually above the TCP and UDP and use UDP to transfer data and TCP to transfer control information. A quick review of these UDP-based protocols is given as follows.

RBUDP: Based on UDP, RBUDP adds simple ACK and retransmission mechanism to guarantee reliability. But it is different from TCP's ACK. In TCP, receiver sends an ACK to sender for each or every other received segment. This is too frequent. Sending ACK takes up handling time and the ACK packets also take up bandwidth. On other hand, in FLDnet, because the RTT is large, after the sender sends out all the packets allowed by window, it may has to wait for a long time to receive the ACK. While, RBUDP firstly uses UDP to continually transfer all the data, the receiver keeps a tally of the packets that are received but gives no ACK until it receives the finish signal DONE. Then, the receiver sends an ACK consisting of a bitmap tally of the received packets by TCP. The sender resends the missing packets and the process repeats until no more packets need to be retransmitted.

RBUDP needs user to set the sending rate. User has to measure the available bandwidth before transfer. Another problem is that because the sender has to keep all the packets that have been sent for retransmission if needed, if the file size is bigger than the memory, it can't be done.

Tsunami: The basic principle of Tsunami is the same as RBUDP. Tsunami mainly makes two points of improvement on RBUDP. First, Tsunami receiver does not wait for finishing of all the data transfer, but periodically (every 50 blocks) makes a retransmission request and periodically calculates the current error rate and sends it to the sender. Second, Tsunami adds rate-based congestion control mechanism. The sender control the sending rate by adjusting the inter-packet delay according to the error rate received. Additionally, if the number of packets that need to be retransmitted is too large, the sender will restart transmission of the file at the given block number.

UDT: UDT is more complex than RBUDP and Tsunami and it's similar to TCP. Based on UDP, in addition to add reliability, UDT also adds congestion control and flow control mechanisms. For reliability, UDT receiver sends selective acknowledgment (SACK) at a fixed interval and sends negative acknowledgment (NAK) once a loss is detected to explicitly feed back packet loss. For congestion control, UDT adopts DAIMD (AIMD with decreasing increase) algorithm to adjust sending rate (not congestion window size like TCP). To differentiate congestion and error, UDT does not react to the first packet loss, while decreases the sending rate if there is more than one packet loss in a congestion event. Additionally, UDT uses receiver-based packet pairs to estimate the link capacity. Like TCP, UDT also uses a flow control window to limit the number of unacknowledged packets.

EXPERIMENTAL CONFIGURATIONS

The FLDnet testbed is built by two terminals connecting to the network emulator with Gigabit Ethernet links (1000 Mbps) as shown in Fig. 1.

The bottleneck bandwidth is 1000 Mbps. The RTT and loss rate can be changed by the network emulator. In the experiments, some typical RTT values corresponding to practical networks ranging from 2-600 m sec^{-1} and the loss rates ranging from 0-1% are used. For general FLDnet, RTT of 300 m sec^{-1} and loss rate of 0.1% are used in experiments by default.

In order to avoid forming bottleneck of performance in the terminals, two high performance IBM servers are placed. As shown in Table 1, the disk speed is faster than the link bandwidth and the utilizations of CPU and memory are found to be less than 100% during the experiments.

RBUDP and UDT also provide memory-to-memory transfer mode. All the following experiment results are from practical file transfer in disk-to-disk transfer mode. A high definition TV (HDTV) file with file size of 4.5 GB is used in experiments by default.

RBUDP and Tsunami need user to set the sending rate and other parameters like buffer size and block size can be set as well. So, usually, the same experiment is made many times under different parameters settings and the best result is recorded. For TCP, a kind of TCP variant named BIC-TCP (default TCP protocol used in Linux 2.6.18 kernel) is used and the TCP buffer size is set to be larger than BDP (Bandwidth Delay Product) in order to get optimal performance.

Table 1: Configuration of terminal servers in the experimental testbed

Server	CPU	Memory	DISK speed	OS
A	Dual Intel (R) Xeon (R) 2.00 GHz	4 GB	253 MB sec^{-1}	Linux 2.6.18
B	Dual Intel (R) Xeon (R) 2.00 Ghz	3 GB	261 MB sec^{-1}	Linux 2.6.18

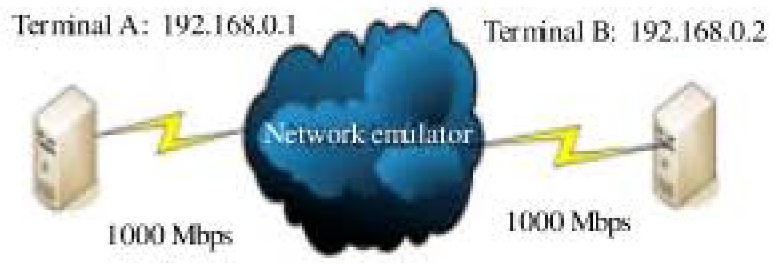

Fig. 1: FLDnet testbed

EXPERIMENTS RESULT

Currently, the most important requirement of e-Science applications is the transfer speed and these applications usually run on dedicated end-to-end lightpath link, which means that fairness and TCP-friendliness is not a problem needed to be considered. So, in this study, throughput is main comparative metric.

Efficiency and stability: Utilization of the available network resource is used to measure the protocol efficiency. For FLDnet, the link is not cheap, so the utilization of bandwidth is important. And for most e-Science applications, high speed and stable throughput is necessary. A series of experiments are made under typical FLDnet configuration (RTT>300 m sec, loss rate = 0.1%). The throughput of TCP is very low (about 1.05 Mbps). Figure 2 shows the throughput of these UDP-based protocols when transferring different size of payloads.

All of the three UDP-based protocols get much larger throughput than TCP. Among them, RBUDP and Tsunami have very high throughput up to 660 Mbps. Note that the throughput of RBUDP drops obviously when transferring large file (>2 GB), that's because RBUDP has to use stream to transfer large file (This point has been explained hereinbefore). When using Tsunami, the file size has no limit and the throughput is high. But some parameters like block size and buffer size are needed to be adjusted to be

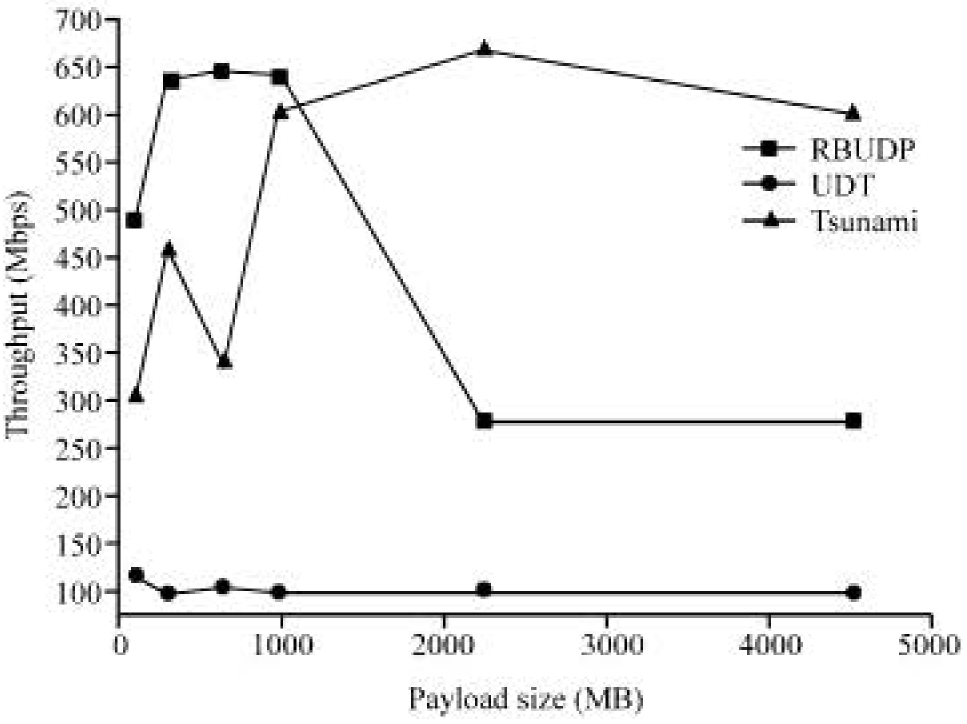

Fig. 2: Throughput vs. payload size

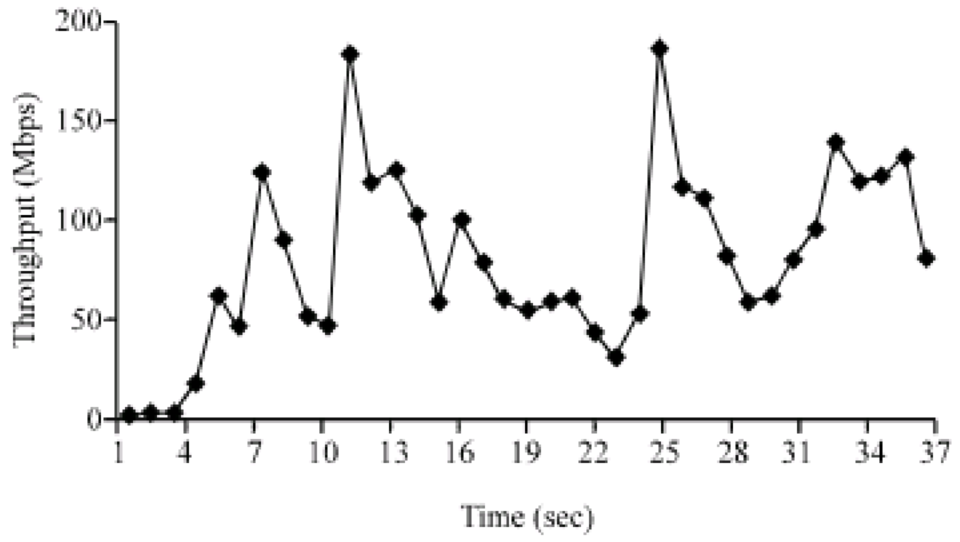

Fig. 3: UDT throughput changes with time

optimal value. The throughput of UDT is relatively not high. And the payload size has no big effect on UDT throughput.

As for stability, the throughputs of RBUDP and Tsunami don't change dramatically with time. But UDT is not stable. Figure 3 shows the UDT throughput changes with time.

The sawtooth throughput of UDT is similar with TCP, that's because UDT also makes congestion control by using DAIMD algorithm. When there is packet loss, the sending rate reduces dramatically. This makes it difficult for UDT to keep stable throughput in noisy link.

Throughput vs. loss: On FLDnet, if it's end-to-end lighpath link, the packet loss rate is extremely low since that there is no congestion on this kind of dedicated link and the error rate of optical link is very low. But, sometimes, there are packet losses due to security devices, network equipment problems or terminal system performance problems. So, the throughputs under different loss rate conditions are investigated.

It can be shown that (Fig. 4), if there is no loss rate at all, TCP can get high throughput (361.28 Mbps) too (TCP buffer size is adjusted). However, even if there is very small loss rate, the throughput drops dramatically. Comparatively, UDP-based protocols have much higher throughput than TCP. For RBUDP and Tsunami, the packet loss rate does not impact throughput greatly (but when the loss rate is 1%, the throughput of Tsunami drops quickly.). But, for UDT, the impact is obvious. That's because the congestion control mechanism of UDT responses sensitively for packet loss. If the loss rate is high, UDT also performs poorly like TCP.

Throughput vs. RTT: On FLDnet, bulk data is transferred between different terminals all over the world. So, different distances between terminals result in different propagation delays. Therefore, throughputs of these protocols under different RTT conditions are tested.

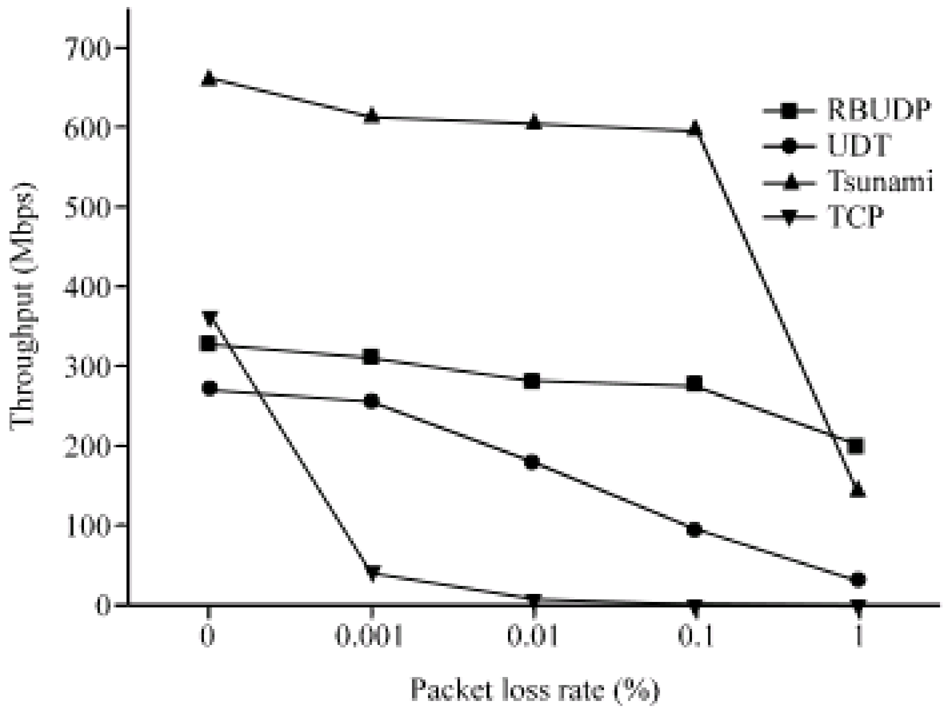

Fig. 4: Throughputs under different loss rate conditions

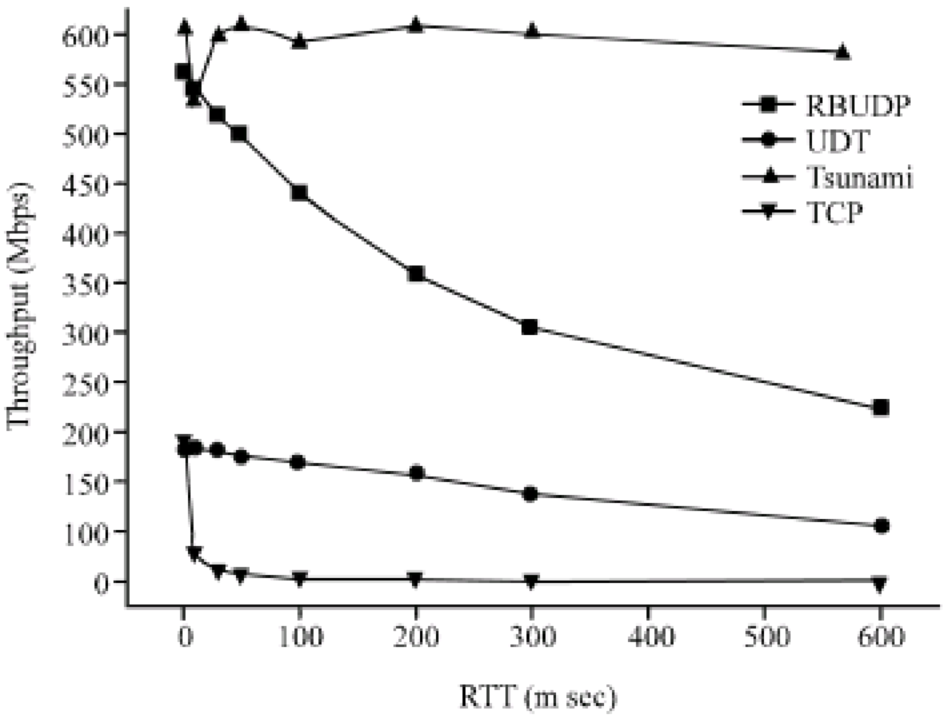

Fig. 5: Throughputs under different RTT conditions

It can be seen that TCP also performs well if the RTT is very small (<2 m sec in LAN), but performs poorly if the RTT is big (in WAN and FLDnet). Still, UDP-based protocols have much higher throughput than TCP (Fig. 5). The throughputs of RBUDP and UDT reduce as the increasing of RTT. The throughput of Tsunami is much higher than others and changes little as the increasing of RTT.

CONCLUSIONS AND FUTURE WORK

UDP-based transfer protocols RBUDP, Tsunami and UDT use UDP to transfer data and TCP to transfer some control information. Currently, throughput is the focus because it is the urgent requirement for transferring bulk data on dedicated link over FLDnet. By the analysis of experiment results from the FLDnet testbed, it can be concluded that all the three UDP-based transfer protocols have mach better performance than TCP. Among them, Tsunami has the best performance and RBUDP also performs well when transfer file which is not too big.

However, they still have some drawbacks. First, the bandwidth utilization is still not high enough (<70%). This makes the long distance bandwidth still can't be fully used. Second, Both RBUDP and Tsunami have no bandwidth estimation function and the sending rate is needed to be appointed. Before using them, the available bandwidth is needed to be measured by tools like Iperf. Additionally, RBUDP has no congestion control and flow control. Tsunami has no flow control, which makes the sender may send too many packets than the receiver's receiving capacity. UDT has complex congestion and flow algorithms and also considers the fairness and TCP friendliness, but its throughput not high and is not suitable for transfer bulk data on dedicated link over FLDnets.

In these experiments, high performance servers are used, so, the performance of terminals has little effect on transport protocol. But in practical application, the terminal performance should be considered. However, all the three UDP-based protocols have not considered this point. Recently, a new kind of UDP-based protocol named Performance Adaptive UDP (PA-UDP) is proposed by Eckart *et al*. (2008).

In future, these existing UDP-based protocols are still needed to be enhanced. First, an ideal protocol which can get stable high throughput for bulk data transferring on dedicated lightpath of FLDnet is needed to be designed. Then, the issues like fairness and TCP-friendliness when running the protocol on shared Internet link are needed to be considered.

ACKNOWLEDGMENTS

The authors thank the designers or developers of the UDP-based protocols for their explanation and help in experiments. They are Eric He (RBUDP), Yunhong Gu (UDT) and Jan Wagner (Tsunami). This research was supported in part by International Cooperation and Communications Project of National Ministry of Science and Technology of China under Grant No. 2006DFA1990 and the Knowledge Innovation Program of Chinese Academy of Sciences under Grant No. 0814061101.

REFERENCES

Dina, K., H. Mark and R. Chalrie, 2002. Congestion control for high bandwidth-delay product networks. Proceedings of ACM SIGCOMM'02, Aug. 19-23, Pittsburgh, Pennsylvania, USA., pp: 1-14.

Eckart, B., H. Xubin and W. Qishi, 2008. Performance adaptive UDP for high-speed bulk data transfer over dedicated links. Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Apr. 14-18, Miami, FL, pp: 1-10.

Floyd, S., 2003. High speed tcp for large congestion windows. http://portal.acm.org/citation.cfm?id=RFC3649.

He, E., J. Leigh, O. Yu and T.A. DeFanti, 2002. Reliable blast UDP: predictable high performance bulk data transfer. Proceedings of IEEE Cluster Computing, Sept. 23-26, Washington, DC, USA., pp: 317-317.

Jin, C., D.X. Wei and S.H. Low, 2004. FAST TCP: Motivation, architecture, algorithms, performance. IEEE/ACM Trans. Network., 14: 1246-1259.

Mark, R.M., 2002. Tsunami: A high-speed rate-controlled protocol for file transfer. http://steinbeck.ucs.indiana.edu/~mmeiss/papers/ tsunami.pdf.

Ren, Y.M., H.N. Tang, J. Li and H.L. Qian, 2008a. Optical network control and management for grid applications. J. Software, 19: 1481-1490.

Ren, Y.M., G.T. Qin, N.A. Hai, J. Li and H.L. Qian, 2008b. Performance analysis of transport protocol over fast long distance optical network. Chinese J. Comput., 31: 1679-1686.

Xu, L., K. Harfoush and I. Rhee, 2004. Binary increase congestion control for fast long-distance networks. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc., 4: 2514-2524.

Yong, X., L. Subramanian, I. Stoica and K. Shivkumar, 2005. One more bit is enough. ACM SIGCOMM Comput. Commun. Rev., 35: 37-48.

Yunhong, G.U., H. Xinwei and G. Robert, 2004. Experiences in design and implementation of a high performance transport protocol. Proceedings of the ACM/IEEE Conference on Supercomputing, Nov. 6-12, Pittsburgh, PA, USA., pp: 22-22.

Yunhong, G.U. and L.R. Grossman, 2007. UDT: UDP-based data transfer for high-speed wide area networks. Comput. Networks, 51: 1777-1799.