37

Multi connection TCP mechanism for high performance transport in an ATM Network

Masami Ishikura, Yoshihiro Ito, Katsuyuki Yamazaki, Tohru Asami
KDD R & D Laboratories
2-1-15 Ohara Kamifukuoka-shi, Saitama 356, JAPAN
Telephone: +81-492-78-7892 Fax: +81-492-78-7510
e-mail: ishikura@lab.kdd.co.jp

Abstract

This paper proposes a new transport mechanism, multi-connection TCP, for high-speed and long delay networks. With to the appearance of ATM network services, wide area data communication networks(WANs) are becoming faster and more world wide oriented. In this situation, the performance of window-based, flow-controlled and connection-oriented transport protocols, such as TCP, is degraded far from the optimum. The throughput of TCP is limited by its window size and round trip delay time(RTT), and the ideal throughput of TCP with 64Kbyte maximum window size over a trans-Pacific ocean link such as TPC-4 is only about 2.6Mbps for a 45Mbps DS3 line with 200ms RTT. In order to achieve high throughput of inter-LAN communications via a high speed, long delay WAN, such as B-ISDN, we propose a multi-connection TCP, and verify its performance by applying it to FTP. A multi-connection FTP can establish up to 16 connections simultaneously, and transfer the data in parallel. The experiment shows multi connection TCP (16 connections and 24Kbytes of each window size) achieves about 8.5Mbps end-to-end throughput via an emulated OC3 155Mbps WAN with 200ms RTT.

Keywords

TCP, transport protocol, high speed network, long delay network, ATM

1 INTRODUCTION

ATM network services are now available in many countries, and data communication networks, such as enterprise networks, are becoming faster and more world-wide oriented. It is, however, difficult to achieve high throughput in a high speed and long delay environment. The performance of window-based, flow-controlled and connection-oriented transport protocols, such as TCP, is extremely degraded due to the short window size in such an environment. The window size of the ordinal TCP implementation is limited to 64Kbytes from the protocol specification[1], and almost all applications, including FTP, uses a TCP window size of less than 32Kbytes. The other problem, for high performance throughput, is traditional error control mechanism of TCP. The existing TCP uses simple re-transmission of the remaining data on an

L. Mason et al. (eds.), *Broadband Communications*

errored packet which is detected. In order to obtain high performance of error protection in a high-speed and long delay environment, a more efficient method, such as a selective acknowledge re-transmission mechanism, is necessary[2].

In order to resolve these problems, several solutions have been investigated. The first solution is to develop a completely new protocol which works efficiently even in a high-speed, long delay network and this is one of the best methods for the future. XTP[3] is one of those protocols. However at present, these new protocols have not been fully standardized, and there is no compatibility with the existing TCP. They can be implemented only in a few types of OSs, hence will not be used widely in the market. The second solution is to enhance the existing TCP protocol[4]-[7]. It should be noted, however, that the window size will be limited up to several hundreds of Kbytes due to the available resource of workstations, and cannot be easily expanded to an order of Mbytes. In addition, the expanded TCP does not support a selective re-transmission mechanism. The extended TCP method is evaluated by [8] in an error free environment with a moderate delay (up to 68ms).

The third solution, proposed and evaluated in this paper, is to establish multiple TCP connections simultaneously to obtain high throughput as an aggregated performance. The idea of a multiple connection mechanism itself has been commonly used in the lower layers, e.g., a multi-link procedure[9] in the HDLC datalink layer, and a digital channel aggregation[10] in the physical layer. In a transport layer, multi-connection TCP is proposed here to achieve high throughput on an end-to-end basis.

This paper presents a multiple connection TCP mechanism, which differs from the previous approaches as follows : (1) the transport layer contains multiple windows, each of which corresponds to a connection, (2) each connection may follow a standard TCP mechanism, (3) this method can also work like a selective acknowledgment mechanism in an erroneous environment, (4) the TCP slow start mechanism in this method works more rapidly than that in the extended TCP method. Furthermore, this solution could be applied to the extended TCP method to obtain higher performance.

We have implemented a multi connection mechanism to evaluate the performance of a data transfer method in a high-speed, long delay network. Implementation has been firstly done as an FTP application level, i.e., Multi-connectioned FTP(MFTP). MFTP aggregates the multiple TCP connections. In the following Section 2, we clarify the problems of TCP dynamics over a high-speed, long delay network, and examine several solutions of the problems in detail. In Section 3, we present the mechanism of MFTP and discuss features and applicability of the multi-connection method to other applications such as HTTP. In Section 4, experiments on MFTP and results are shown. In Sections 5 and 6, a discussion and conclusion are given.

2 TCP OVER HIGH SPEED, LONG DELAY NETWORKS

2.1 Problems of TCP dynamics

Because the TCP was not designed to be used in recent high-speed, long delay network, such as ATM wide area networks, the maximum window size is insufficient for those network applications. The maximum window size of TCP is specified in [1] and is limited to 64Kbytes. In order to make use of network bandwidth efficiently, the round trip time(D) bandwidth(BW)

products(D *BW) of the network must be smaller than the maximum window size of TCP. The ideal throughput(Ti) of TCP, where the bandwidth is not limited, is obtained by the equation(1).

$$Ti = W / D \quad (1)$$

Ti(bps) : throughput of TCP
W(bit) : maximum window size of TCP connection
D(s) : Round Trip Time (RTT)

For example, using a DS3(45Mbps) link between the United States and Japan(the round trip time is about 200ms via submarine cable), the D*BW product is about 9Mbits(1.12Mbytes), which is much larger than the TCP maximum window size. In this case, a situation called window close occurs. This causes degradation of network utilization, and the throughput with a 64Kbyte TCP window size achieves only about 2.6Mbps. The bottleneck of the throughput is the shortage of TCP window size.

2.2 Features and advantages of multi connection method

In order to reduce the network idle time caused by window close, there are three approaches. The first is a completely new protocol for a high-speed, long delay network such as XTP. The second is the extension of TCP window size using TCP options described in [7], and the third is the aggregation of multiple TCP connections.

The third one, which is called multi connection TCP mechanism, lets an application use the multiple existing TCP connections in the aggregate. This multi connection mechanism could apply to the extended TCP described above as well as the traditional TCP. Furthermore, this mechanism supports a pseudo selective acknowledgment mechanism.

In the case of a single TCP connection, when an error is detected, a go-back-N error control mechanism starts and re-transmits all data after errored data. As mentioned in [5], it may cause the degradation of throughput to expand the window size in an erroneous environment. However, in the case of the multi connection method, if data transmit errors occur in some connections, the go-back-N error control mechanism starts simultaneously only for those connections and re-transmits all data after errored data for each errored connection respectively. This speeds up the re-transmission.

When all the errors have been re-transmitted, TCP starts a slow start. As shown in Figure 1, the time Tr to recover from the slow start condition to the congestion control mode of a multi

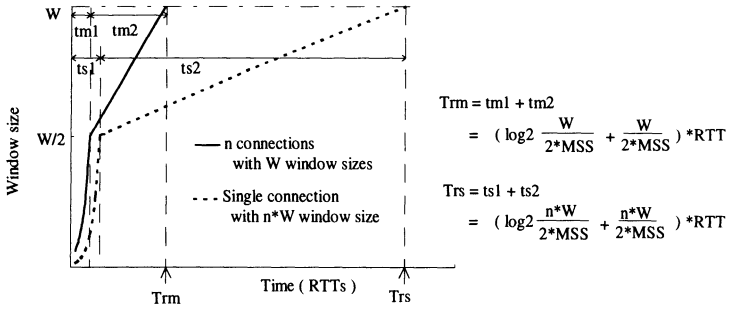

Figure 1 Illustrated recovery time from Slow Start.

TCP connection (n connections, each with window size W and maximum segment size MSS) is much shorter than a single TCP connection with window size n*W and maximum segment size MSS. The former consists of n independent connections with maximum window size W in parallel. Tr for each congested connection is given as (log2(W/(2*MSS))+ W/(2*MSS)) times RTT, which is the Tr for the former. Tr for the latter case is given as (log2(n*W/(2*MSS))+n*W/(2*MSS)) times RTT. This means that Tr for the former is almost 1/n of the latter Tr for large W/MSS.

These advantages described above show the multi-connection method is more durable than single connection method in an erroneous environment.

Considering software portability at present and applicability to extended TCP in the future, we also present the application layer multiple TCP connection method as a near term solution to realize high performance file transfer application over high speed, long delay networks.

3 MULTI CONNECTION TCP MECHANISM

3.1 Two Implementations of Multi connection TCP mechanism

There are two ways to implement the multi TCP connection mechanism as in shown in Figure 2. One is to implement this mechanism inside the application layer, such as Multi-connectioned FTP(MFTP). The other is to implement it inside the transport layer as a Multi TCP connection sub-layer which provides an existing socket-like interface to the applications. The mechanism of multi TCP connection is the same in both implementations, but a difference exists in the interface to applications.

We first developed MFTP to evaluate a multi-connection mechanism due to the following reasons. The traffic of HTTP applications is increasing exponentially in the INTERNET but the FTP[11] is still one of the highest traffic applications[12]. It has become popular in exchanging larger files, such as graphical images or photos between sites via high speed intercontinental WANs. Furthermore, in closed user networks, such as enterprise networks, many kinds of huge amount of data are transferred world wide. In such cases, a high performance file transfer protocol, which still works well over high speed, long delay networks, is required. Therefore, we firstly applied the multi TCP connection mechanism to FTP to evaluate this mechanism. We implemented the core of the mechanism in the FTP application layer, and called it Multi connectioned FTP(MFTP).

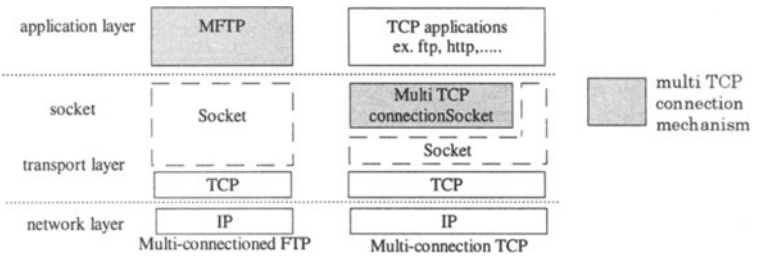

Figure 2 Two typical implementations of multi TCP connection mechanism.

MFTP uses multiple existing TCP connections to utilize the aggregated large window size. The theoretical throughput of MFTP is obtained by the equation (2).

$$Tm = N * W / D \quad (2)$$

Tm(bps)	: throughput of MFTP
N	: the number of TCP connections used in MFTP
W(bit)	: window size of each TCP connection
D(s)	: Round Trip Time

3.2 Functional Blocks of Multi-connectioned FTP

The functional blocks of MFTP are illustrated in Figure 3. The shaded parts of Figure 3 are new additions or modifications to the existing FTP to support the MFTP. We used the source code of FTP programs of FreeBSD as the basis of MFTP. The newly added part of the source code is about 200 steps in each user and server program. The MFTP is upper compatible to the existing FTP. The existing FTP uses the Telnet protocol to send and receive FTP commands. The MFTP adds new commands to establish multiple connections for data transfer. The function of each block is as follows.

● **User Interface** : This module inputs and outputs the MFTP commands between the user and User PI. The user interface is the same as the existing FTP.

● **User/Server Protocol Interpreter(PI)** : User PI initiates the control connection to the Server PI and interprets FTP commands. MFTP PIs are an additional function to interpret the new MFTP command.

● **User/Server Data Transfer Process(DTP)** : User and Server DTP implement the interface to the file system and control TCP connections according to commands from User/Server PI. They contain the following functional blocks.

· Parameter Collection function

This function exists only in User DTP and gathers the parameters which are necessary to determine the appropriate number of TCP connections for a specific destination. It uses a ICMP Echo/Reply function to measure the RTT. Parameters are stored to use on the next occasion to access the same destination.

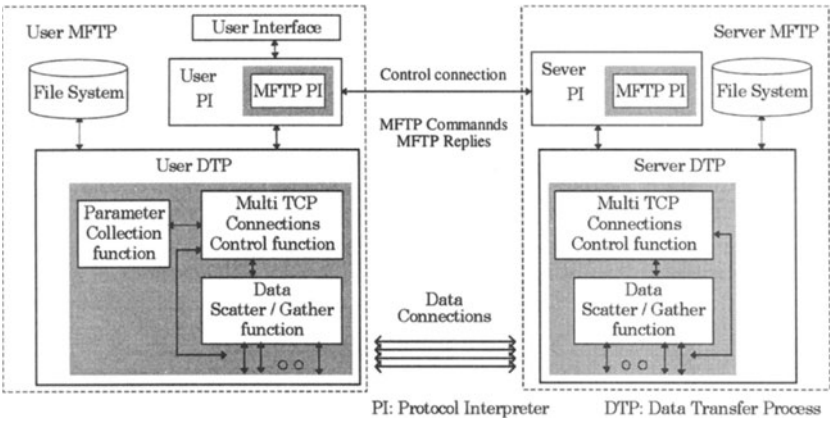

Figure 3 Functional blocks of multi-connectioned FTP (MFTP).

- Multi TCP Connections Control function
  This function calculates the number of TCP connections for data transmission according to equation(2) using parameters which were collected by the Parameter Collection function. The calculated number of TCP connections may be overridden by a MFTP user using a MFTP command. This function negotiates the number of connections between the user and server DTP and establishes, controls or disconnects multiple TCP connections.
- Data Scatter and Gather function
  At the sender's site, this function sequentially scatters the data to be transmitted into N TCP connections block by block. At the receiver's site, this function obtains blocks of data from N TCP connections and reconstructs them into the original data. The receiver's site needs a large size of receive buffer to reconstruct the gathered data in a correct sequence.

3.3 Two methods for Data scattering/Gathering

There are two ways to scatter data as shown in Figure 4. One is the static assignment method and the other is the dynamic assignment method.

In the static assignment method, each data block is assigned to each connection permanently before the transmission. This method is easy to implement and the load of the scattering process is low. However, the total performance of multi TCP connections is degraded by the one with the lowest performance, and this method requires a larger size of receive buffer to deliver received data to an FTP application in sequence. It is difficult to set up a stream lined service for applications using this method.

In the dynamic assignment method, each data block is assigned to the first non-blocked connection found at that time. This method requires a smaller buffer size than the former to reconstruct the data in sequence and the processing time becomes shorter.

In our implementation, the MFTP uses the dynamic assignment method considering the applicability of this mechanism to other applications, such as HTTP.

4 EXPERIMENTS AND RESULTS

4.1 Configurations and Protocol Stacks of the Experimental Network

The configurations and protocol stacks of the experimental network are shown in Figure 5. The performance evaluation of MFTP was made over the interconnected ATM LANs. MFTP user

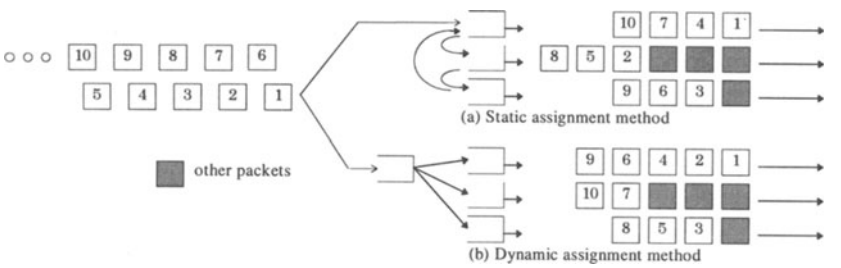

Figure 4 Data assignment method to multi TCP connections.

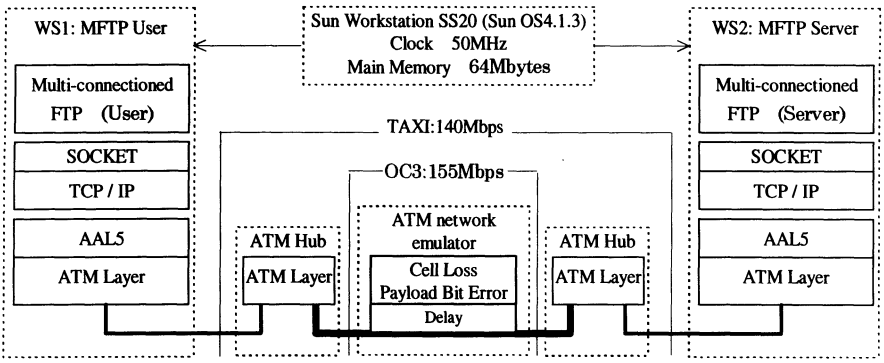

Figure 5 Configurations and protocol stacks of the Experimental Network.

and server programs were installed in WS1 and WS2. These WSs are SUN SparcStation20(Sun OS 4.1.3), and their specifications are depicted in Figure 5. WS1 and WS2 were accommodated in ATM-Hubs(Fore Runner ASX-200) using a 140Mbps TAXI interface respectively. These ATM-Hubs were interconnected via the ATM network emulator[13] using OC3 interfaces to introduce the propagation delay, ATM cell loss and payload bit errors artificially. WS1 and WS2 used AAL5(VBR) over PVC.

4.2 Testing Parameters

The throughput of MFTP was measured by transferring 16Mbyte data files from WS1 to WS2. In order to reduce the effect of disk access speed, the user side put the test data to '/dev/null' on the server side. The MFTP worked in a block transfer mode. The parameters of measurements are listed in Table 1. The existing FTP of SUN OS 4.1.3 adopts the value of 24Kbytes as the maximum window size, so, we used the value of 24Kbytes as the maximum window size in these measurements. To investigate the relation between the maximum window size and the throughput for reference, MFTP throughput is measured in various windows in an error free and RTT 100ms environment.

Each measurement was performed over 10 times and the average of the results was calculated. However, there was little variance in the measured data, and each standard deviation was at most under 10% from the average for each value.

4.3 Results of MFTP throughput

MFTP throughputs measured in an error-free environment are shown in Figure 6. The throughputs for file transfer in different RTTs are plotted for each number of connections. The

Table 1 Measurement parameters

Parameters	Range
Number of connections	1~ 16
Inserted Round Trip Time (ms)	0 ~ 500
Cell Loss Ratio	$2.4*10^{-4}$ ~ $3.8*10^{-6}$
Payload Bit Error	$4.8*10^{-7}$ ~ $3.0*10^{-8}$
Window size / connection	24Kbytes,(4K ~ 51Kbytes)

maximum window size of each connection is 24Kbytes for an RTT of 0~500ms. In Figure 7, the relation between maximum window size and throughput of MFTP is shown in an environment of RTT 100ms.

MFTP throughputs measured in an erroneous environment are shown in Figures 8 to 10. In order to evaluate the performance of MFTP in the erroneous environment, two types of artificial errors were injected. One is cell loss error to simulate the error which occurs in ATM switches, and the other is payload bit error to simulate the link error.

Figure 8 shows the throughputs in a cell loss error injected environment for each number of connections. Figure 9 plots the throughput for each RTT. Figure 10 plots the characteristics of the throughput for each error rate in a RTT 200ms environment. The throughput values in Figure 10 are normalized by that of 16 connections in an error free and RTT 200ms environment. The window size of each connection is 24Kbytes.

4.4 Multi connection TCP v.s. single large window TCP

To compare the performance of multiple small windows TCP to single large window TCP in erroneous environment, another experiment was performed using 'SUN OS TCP-LFN' option. It supports the extension TCP window size and dose not support selective acknowledgment. The experiment was performed via bridged interconnected ethernet LAN. The testing parameters and the results are listed in Table 2.

Table 2 Throughput (kbps) of Multi connection TCP and single big window TCP

	total window size (Kbytes)				
	64	128	256	384	512
Multi connection TCP (n*64Kbyte)	762	848	883	1187	1465
Single big window TCP(TCP extension)	762	770	764	751	747

added bit error ratio = $3.0*10^{-7}$ added RTT = 200ms

5 DISCUSSION

5.1 Multi connection TCP mechanism in the error free environment

In error-free environments without any inserted delay, the throughput of MFTP in the experimental network achieved over 32Mbps as shown in Figure 6 using any number of connections. Because an ATM link(OC3:155Mbps) is faster than the ATM LAN (TAXI:140Mbps), there is no bottleneck in the ATM link in the experimental network, so it is seen that 32Mbps is the maximum performance of MFTP with the workstations used in our experiments.

In Figure 7, as the maximum window size is increasing, the throughput is increasing step by step. It is considered that the length of TCP packet affects the characteristics of the throughput. The maximum length of IP datagram(MTU size) and TCP packet(MSS size) are 9188bytes and 9148bytes respectively in the ATM network. In the experimental network, the average length of TCP packets used in MFTP is about 9Kbytes calculated by transferred data size and the number of TCP packets. This means that the TCP used in this MFTP experiment sends data near the MSS size only, therefore TCP does not always utilize the whole window size. In the case where

the maximum window size is set to 24Kbytes, the actual window size, which we call segmented window size, works out to about 18Kbytes. The measured throughput therefore increases discretely as shown in Figure 7.

5.2 Effects of multiple connections over long delay network

As shown in Figure 6, when the round trip time is larger than 200ms, the throughputs increase linearly, so the multi TCP connection mechanism works effectively. However, the theoretical

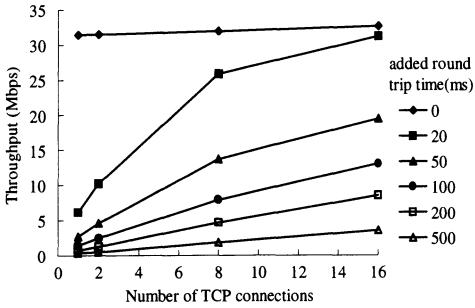

Figure 6 Throughput of MFTP in error free environment

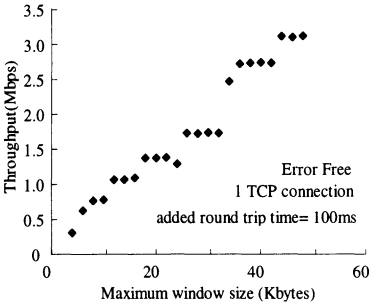

Figure 7 Throughput of MFTP for maximum window size.

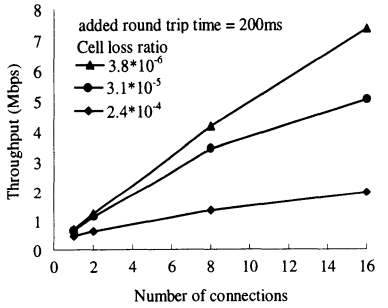

Figure 8 Throughput vs. Number of connections in erroneous environment.

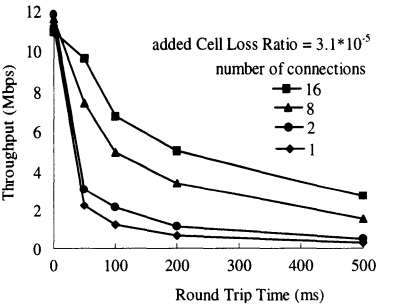

Figure 9 Throughput vs. RTT in erroneous environment.

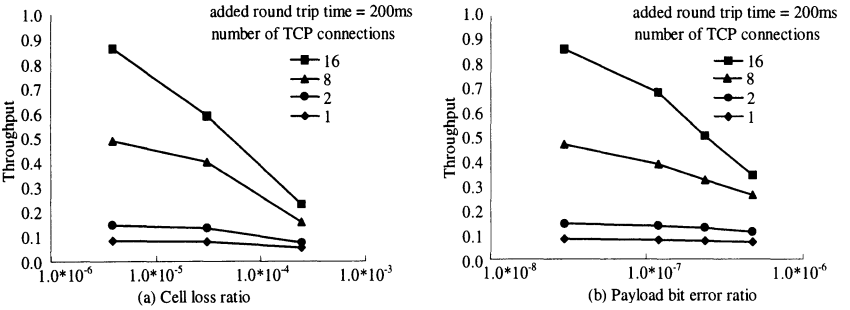

Figure 10 Throughput vs. error ratio (Cell loss, Payload bit error).

throughputs obtained by equation(2) using the maximum window size and the added round trip time are higher than the measured throughput. For example, according to the equation(2), MFTP should achieve 15.7Mbps using 16 connections under the conditions of 200ms of round trip time and 24Kbytes of maximum window size for each connection, while the actual throughput achieved is only 8.5Mbps. It is confirmed that there is no packet loss in the error free environment, hence this degradation is not caused by re-transmission. Equation(2) can obtain only the ideal maximum throughput using the maximum window size and the added round trip time. To find the practical throughput Tp, other factors, for example, the protocol processing time at the work stations, the segmented window size discussed in the last of clause 5.1, should be considered. Equation(2) is modified to (2)' to reflect these factors.

$$Tp = k * N * W' / D \qquad (2)'$$

Tp(bps)	: Real throughput of MFTP.
N	: Number of TCP connections used in MFTP.
W'(bit)	: Segmented window size of each TCP connection.
D(s)	:Round trip time.
k	:System parameter of the experimental workstations and network.

The segmented window size W' is considered to be about 18Kbytes as depicted in Figure 7.

k is a system parameter of the experimental network. *k* is about 0.75 calculated from the measured throughput data for 16 TCP connections, RTT 200ms and the segmented window size of 18Kbytes. Using this value of *k*, equation(2)' also matches the measured data for RTT 500ms. The equation(2)' with the same *k* fits the lower throughput case compared to the maximum throughput of the experimental network(32Mbps). Equation(2)' also obtains the optimal number of connections when the expected throughput is set to Tp.

5.3 Multi connection TCP mechanism in erroneous environment

In the environment RTT 200ms, the throughput increases as the number of connections increases as shown in Figure 8. Except for the no delay environment, the throughput increases as the number of connections increases as shown in Figure 9. Hence, even in the erroneous environment, the multi connection method worked effectively. However, the increasing rate is not linear in high error rate environments.

In the TCP protocol, the error detection is carried out packet by packet. In order to compare the characteristics of MFTP throughput for cell loss error and payload error, the cell loss error rate and pay load error rate should be converted to TCP packet error rate. The average length of TCP packet used in this experiment is about 9Kbytes as described before. A TCP packet consists of 188 cells(48bytes/cell) or consists of 72kbits. Converting the X axis to packet error rate according to the number of cells or bits of a TCP packet, Figure 10 (a) and (b) depict almost the same characteristics.

Furthermore, as listed in Table 2, Multi connection TCP obtain higher throughput than that of single connection TCP which has the same total window size.

From the results of the experiments, it is considered that the MFTP still works effectively in erroneous long delay networks.

6 CONCLUSION

This paper has discussed a multi-connection method at the application layer as well as the transport layer to increase the data transfer throughput in the environment of high-speed, long delay networks. We have developed a multi-connection method for FTP, MFTP, to evaluate the performance of data transfer. Our MFTP has the following features and advantages.

1. MFTP uses multiple existing TCP connections in aggregate, so it is easy to implement it in any other work stations.
2. MFTP can adjust the number of connections flexibly using the parameter collection function.
3. The multi connection method applied to a file transfer protocol works effectively. MFTP has achieved over 8.5Mbps end-to-end throughput using 16 connections of 24Kbyte maximum window size via the emulated ATM WAN of an OC3 155Mbps link with 200ms round trip delay.
4. MFTP works effectively even in an erroneous environment, and it is considered that the multiple TCP connection method is more durable than a single large window mechanism such as the TCP extension.

The multi-connection handling mechanisms could be applied to a TCP socket sub-layer with the above advantages from 2 through 4. In this case any TCP applications, such as HTTP, can increase the throughput in the environment of high-speed, long delay networks.

We conclude that the multiple-connection method is, at present, a suitable and desirable solution for the high performance data transfer application over high speed, long delay networks such as inter national ATM networks. We are now implementing the multi-connection mechanism over extended TCP to gain a much greater increase of throughput.

7 ACKNOWLEDGMENT

The authors wish to thank Dr. Y. Urano and Mr. S. Fukumitsu of KDD R&D Laboratories for their continuous encouragement and support.

8 REFERENCES

[1] J. Postel, "Transmission Control Protocol", RFC793, Internet Engineering Task Force, Sep. 1981.

[2] S. O'Malley, L. Peterson, "TCP Extensions considered harmful", RFC1263, Internet Engineering Task Force, Oct. 1991.

[3] XTP Forum, "Xpress Transport Protocol Specification", XTP Revision 4.0, March 1995

[4] V.Jacobson and R.Braden, "TCP Extensions for Long-Delay Paths", RFC1072, Internet Engineering Task Force, Oct. 1988.

[5] R.Fox 'TCP Big Window and Nak Options", RFC1106, Internet Engineering Task Force, Jun. 1989.

[6] V.Jacobson, R.Braden and L.Zhang, "TCP Extension for High-Speed Paths", RFC1185, Internet Engineering Task Force, Oct. 1990.

[7] V.Jacobson, R.Braden and D.Borman, "TCP Extensions for High Performance", RFC1323, Internet Engineering Task Force, May 1992.

[8] C.Villamizar and C.Song, "High Performance TCP in ANSNET", ACM SIGCOMM Computer Communication Review, pp.45-60, Vol.24 Num.5, Oct. 1995.

[9] ISO7478 Information processing systems - Data communication - Multilink procedures

[10] ISO/IEC13781 Information technology - Telecommunications and information exchange between systems - Private Integrated Services Network - Digital Channel aggregation

[11] J. Postel, J. Reynolds, 'FILE TRANSFER PROTOCOL (FTP)", RFC959, Internet Engineering Task Force, Oct. 1985.

[12] V. Paxon, "Empirically Derived Analytic Models of Wide-Area TCP Connections", IEEE/ACM Transactions on Networking, vol.2, No.4, Aug. 1994.

[13] K.Yamazaki, T.Nakajima and S.Hayakawa, "Considerations on Network Performance of 64kbit/s-Based Services in an ATM Network", IEICE TRANS. COMMUN., VOL. E78-B, NO.3 MARCH 1995.