A New High-Performance Data Transfer using SCTP

Hyuncheol Kim[1], *Doo-Won Seo*[2]

[1] Dept. of Computer Science, Namseoul Univ., Cheonan, Korea
[2] Dept. of Computer Network, Daeduk College, Deajon, Korea

Summary
SCTP (Stream Control Transmission Protocol) is a new IETF transport protocol over the IP network. Transport protocol is responsible for reliable delivery of a message form a host to another host. We proposed and implemented an efficient file transfer system using the SCTP multiple file transfer and modified STCP congestion control mechanism to guarantees a stable data transfer with multi-streaming and to solve the problems such as server overloading due to multiple connection and the HOL (Head Of Line) blocking that the existing TCP-based file transfer. The result shows that proposed system reduces transmission latency for huge data and improves the performance.

Key words:
SCTP, Revised SCTP, Performance Evaluation

1. Introduction

The transport layers were designed for the terminal communication among the applications that operate in different hosts. This is an attempt to simplify the application development by eliminating the application layer's dependency on network layers.

Transport layers, which provides functions such as flow control, error solution, and secure transfer, has satisfied most of what the network infrastructure needed only with 2 protocols, TCP and UDP, for the last 20 years.

However, the increasing requests could not be fulfilled merely by TCP and UDP due to attempting to provide the telephone service, which was provided in the existing communication network, over the internet and by the appearance of real-time multimedia application service [1][2]. Therefore IETF (Internet Engineering Task Force) revised SCTP (Stream Control Transmission Protocol) to a new standard protocol [3][4][5].

A critical error could occur when transferring large files using the existing TCP. First, when TCP transfers a file, it executes a byte-stream communication. Byte-stream means to divide data into bytes and to transfer regardless of their meanings. The application layers should distinguish the difference between a message and a data. Thus, even if a data processed through TCP is the data of another program, its sectors might become scrambled or its priority or security may be compromised because it is divided into bytes and then transferred.

Moreover, the byte-stream communication induces the HOL (Head of Line) blocking problem. To solve this problem, one application program data can be allocated to one connection. Data priority and security problems could be resolved through this method as well as the scrambling of data and the HOL blocking issue. However, too many connections to application program servers could result in server overloading which could lead to the server malfunctioning.

As an attempt to fix this, this paper proposes the new SCTP transport layer protocol. When a packet loss occurs during the transfer of a stream, the multi-streaming ability of SCTP transport protocol, which supports stream-based communication, allows the other streams to continue with their transfer immediately, not affected by the packet loss of another stream. Being able to transfer data with multiple streams increases the transfer speed, which is the strength of SCTP.

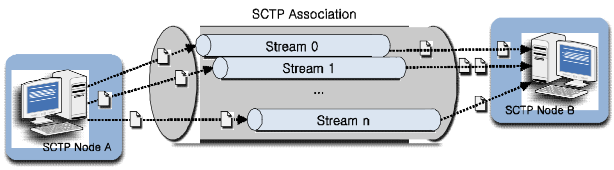

Fig. 1 SCTP multi-streaming concept

The remainder of this paper is organized as follows. Section 2 introduces some characteristics of SCTP. Section 3 and Section 4 describes the proposed system and its design process and compares and contrasts the TCP-based FTP with the SCTP-based FTP that is proposed in this paper. Section 5 concludes this paper by summarizing some key points made throughout and assessing the representation of analyzed results.

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

2. Related Works

The basic service offered by SCTP is the reliable transfer of user messages between peer SCTP users. It performs this service within the context of an association between two SCTP endpoints. SCTP is connection-oriented in nature, but the SCTP association is a broader concept than the TCP connection [3]. This section summarizes some of the key features that provide to the ULP (Upper layer protocol).

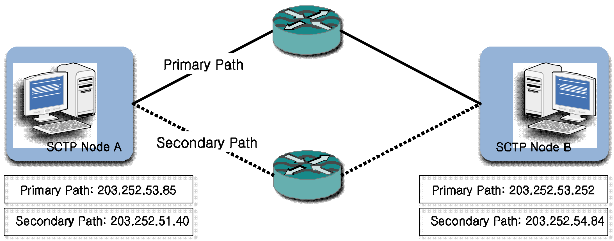

Fig. 2 SCTP multi-homing concept

2.1 SCTP Multi-Streaming

The SCTP transfer protocol has a function in which one SCTP association has multiple streams. Multi-streaming allows for multiple streams on an association to send data. A SSN (Streaming Sequence Number) is allocated to each stream in SCTP so that the order of the streams could be maintained to manage the data being transferred thereby providing secure data transfers.

As for the single-stream TCP, in the case where packets of a data that was transferred to the stream is lost, the Head of Line blocking occurs. The data cannot be transported to an application when waiting for the lost packet to be retransferred.

Fortunately, SCTP supports the multi-streaming function in which even if the data transfer in one stream encounters the packet loss of a data, the transfer process in other streams are not affected and therefore able to continue transferring data to an application. Fig. 1 shows the SCTP multi-streaming concept [6][7].

Therefore, SCTP's strength is in its multi-streaming ability that makes the data transfer faster. In addition, SCTP contains a U flag that can select the data chunk to be ordered or unordered. If a data transfer does not to be ordered, the U flag can simply be set to one for unordered data transfer. The receiving end of the terminal that identifies the U flag can reduce the waiting time because it does not need to wait to receive the packets in order or to receive any lost data. Therefore, using the ordered, unordered function of the data chunk in association with the multi-streaming ability can bring a significant increase in the performance.

2.2 SCTP Multi-Homing

An SCTP endpoint is considered multi-homed if there are more than one transport addresses that can be used as a destination address to reach that endpoint. Moreover, the ULP of an endpoint shall select one of the multiple destination addresses of a multi-homed peer endpoint as the primary path. By default, an endpoint should always transmit to the primary path, unless the SCTP user explicitly specifies the destination transport address to use. Fig. 2 shows the SCTP multi-streaming concept [6][7].

3. Revised SCTP

3.1 Control Architecture

In order to resolve the issue that the existing TCP-based FTP has, the following method was used to enhance the performance efficiency. First, SCTP is used as the transport layer protocol. Its standard multi-homing and multi-streaming functions contribute to its stability over TCP. Three parameters have to be adjusted in order to convert the TCP application to the SCTP application.

The third parameter allocated when generating a socket has to be changed from IPPROTO_TCP to IPPROTO_SCTP, and the netinet/sctp.h file has to be included rather than netinet/tcp.h. Moreover, the existing TCP_NODEALY option has to be changed to SCTP_NODEALY to adjust to SCTP in using the setsockopt function that configures the socket options.

3.2 Multi-Streaming Design

The performance efficiency of TCP-based FTP dropped when transferring multi-data therefore having to open connections every time each data is transferred. In this paper, as a solution to this issue, the connection after completing a data transfer is not closed but it waits for the next data transfer.

At the initial connection, the server identifies the number and the size of the data to be transferred and the connection is terminated at the completion of transferring all the data. This way, the buffer of the transport layers only need to be generated once thereby reducing the host's overloading whereas it used to be generated for every file there is in the data.

When using TCP, PORT, RETR, SIZE commands have to be exchanged in the control connection after each data transfer and the connection and the termination of a data connection also need to be repeated. Since the number of PORT, RETR, SIZE, SIZE response, data connection setup, and data connection teardown processes increase as the number of transfer increase in the TCP-based FTP, the overhead also increases as the number of files to transfer increases.

Meanwhile, when using SCTP, the names of all the files to receive is included within a single RETR and as for SIZE, the transfer is made the number of files to receive by each data stream. The data connection is initiated when the file transfer is initiated and is left open until the completion of all file transfer.

3.3 Modified Congestion Control

Proposed high-performance file transfer system adopts the congestion control scheme that is proposed by [8]. The cwnd is not exponentially increased as that of classical TCP algorithms like TCP Reno [9][10][11]. During congestion avoidance algorithm for each acknowledgement received in a round trip time the congestion window is increased by

$$cwnd = cwnd + [0.01 * cwnd] \quad (1)$$

If a packet is lost in the network, four duplicate SACK chunks are required at the SCTP sender. On the first detection of the congestion, the congestion window is reduced by

$$cwnd = cwnd - [0.125 * cwnd] \quad (2)$$

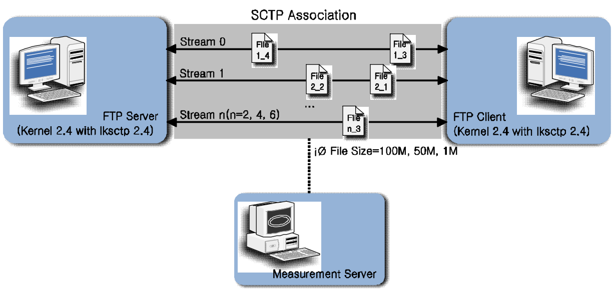

Fig. 3 Experimental Setup

4. Performance Evaluation

The simplest network is built to investigate the multi-streaming feature. Two end nodes were used for the measurement process, one for the FTP client and the other for the FTP server. All nodes run Linux. Fig. 3 shows the testbed to measure the performance of the proposed architecture.

For the performance evaluation, 2, 4, and 6 100Mbyte files were transferred and their respective durations were recorded. In addition, we measure the time in 50Mbyte, 100Mbyte multiple files transmission.

In the case of SCTP, the file transfer time was recorded from the open control connection binding a data connection to the disconnection after transferring all the data. Whereas in the case of TCP, the file transfer time was recorded from the data connection of the first file, with the control connection open, to the data disconnection of the last file.

The results were divided into upload, which is the transfer of data from client to server, and download, which is the transfer of data from server to client. Table 3 and Table 3 are showing result about download and upload respectively.

Looking at the result of performance evaluation when TCP and SCTP transfer files in mass, SCTP can be said to have the higher throughput than TCP.

File Size	Num. of File		Orig. TCP	Orig. SCTP	Modified SCTP (b:0.5, a:0.04)	Modified SCTP (b:0.25, a:0.01)	Modified SCTP (b:0.125, a:0.01)	Modified SCTP (b:0.0625, a:0.005)
100M	2	Time(s)	38.71	34.65	26.38	26.31	26.67	30.8
		Mbytes/s	5.81	7.65	8.94	9.01	7.95	7.87
		Num. of Packet	186863	276113	276127	275747	244668	275491
	4	Time(s)	74.53	63.69	55.42	53.14	51.22	50.95
		Mbytes/s	5.02	7.21	8.42	8.98	8.99	9.53
		Num. of Packet	356395	567154	549263	561651	537385	572294
	6	Time(s)	111.98	98.25	88.85	87.88	82.6	83.43
		Mbytes/s	5.26	6.84	7.34	7.61	8.15	6.85
		Num. of Packet	570458	809431	769255	787212	783717	808557
50M	2	Time(s)	13.9	12.01	12.09	11.9	11.72	12.06
		Mbytes/s	8.55	8.78	8.94	9.37	9.02	9.24
		Num. of Packet	110148	128451	125733	129324	123259	129551
	4	Time(s)	26.32	25.48	22.83	22.01	23.89	22.25
		Mbytes/s	8.03	8.54	9.08	9.88	8.46	9.61
		Num. of Packet	203402	253310	242741	254883	236740	250217
	6	Time(s)	43.9	35.66	34.73	34.11	35.13	32.77
		Mbytes/s	6.67	8.37	9.08	9.5	8.65	10.1
		Num. of Packet	286799	372953	368895	380122	356668	388367
1M	2	Time(s)	0.2	0.43	0.26	0.22	0.22	0.34
		Mbytes/s	11.45	9.09	9.29	10.13	10.15	8.27
		Num. of Packet	2295	2712	2835	2700	2778	2823
	4	Time(s)	0.4	0.67	0.47	0.63	0.46	0.46
		Mbytes/s	11.67	7.93	9.76	7.06	9.96	9.96
		Num. of Packet	4615	5573	5552	5544	5550	5598
	6	Time(s)	0.6	0.72	0.78	0.73	0.72	0.76
		Mbytes/s	11.76	10.3	9	9.4	9.48	9.2
		Num. of Packet	6936	8422	8472	8313	8257	8408

Table 1: Modified SCTP Performance – Download

File Size	Num. of File		Orig. TCP	Orig. SCTP	Modified SCTP (b:0.5, a:0.04)	Modified SCTP (b:0.25, a:0.01)	Modified SCTP (b:0.125, a:0.01)	Modified SCTP (b:0.0625, a:0.005)
100M	2	Time(s)	35.13	32.98	27.04	26.94	27.79	27.61
		Mbytes/s	6.76	6.54	9.5	9.09	8.99	8.87
		Num. of Packet	234050	321176	333388	309280	323764	318026

	4	Time(s)	71.78	65.26	53.77	54.65	55.04	67.7
		Mbytes/s	5.54	6.57	9.22	9.09	8.93	7.28
		Num. of Packet	399697	657801	643178	644661	636145	597409
	6	Time(s)	107.96	86.14	79.55	94.39	74.23	81.96
		Mbytes/s	6	9.58	9.39	7.89	8.81	8.91
		Num. of Packet	642948	983541	962121	894273	930899	935590
50M	2	Time(s)	15.43	13.97	10.15	12.59	10.26	14.57
		Mbytes/s	8.63	11.33	10.83	9.26	10.52	8.09
		Num. of Packet	119010	147814	142519	151226	139980	148843
	4	Time(s)	26.78	28.65	21.04	39.43	21.37	25.58
		Mbytes/s	8.31	9.61	10.6	4.86	10.26	9
		Num. of Packet	218246	296347	289515	247614	284503	298677
	6	Time(s)	42.95	43.64	31.63	33.84	33.71	40.76
		Mbytes/s	7.74	10.16	10.06	9.92	9.79	8.4
		Num. of Packet	329253	456524	412920	432763	427022	443334
1M	2	Time(s)	0.2	0.31	0.23	0.22	0.23	0.26
		Mbytes/s	10.49	10.59	9.24	9.43	10.41	9.13
		Num. of Packet	2328	2911	2775	2743	3029	2987
	4	Time(s)	0.39	0.51	0.47	0.53	0.44	0.41
		Mbytes/s	11.91	10.86	9.39	8.44	10.37	10.11
		Num. of Packet	4728	6103	5677	5408	5961	5403
	6	Time(s)	0.59	0.71	0.61	0.78	0.61	0.61
		Mbytes/s	11.92	11.68	10.46	8.24	11.26	10.16
		Num. of Packet	7119	8883	8265	8321	8924	8152

Table 2: Modified SCTP Performance - Upload

5. Conclusion

This paper proposes an application program in which the SCTP protocol that guarantees a multi-streaming ability and a stable data transfer is used as a solution to the fact that the existing TCP and UDP-based application programs have critical stability issues whereas the current network traffics are becoming more real-time and larger in volume.

The existing TCP-based FTP induces unnecessary overhead and delay because of the attributes such as the disassociation of data connection and control connection and the non-persistent data connection. The overhead and delay will increase even more when transferring multiple files or large-sized files.

Although SCTP-based FTP is in the process of being researched only the transport layer protocol TCP is replaced by SCTP and there still exists overheads and delays because the file transfer mechanism remains the same. However, the SCTP-based multi-file transfer algorithm proposed in this paper improves the performance by reducing such overheads and delays.

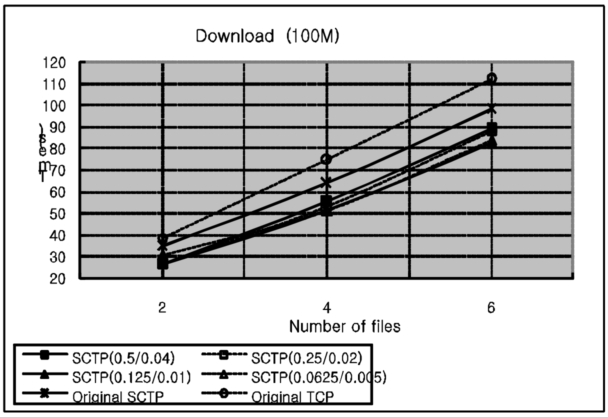

Fig. 4 Transmission time (Download, 100M file)

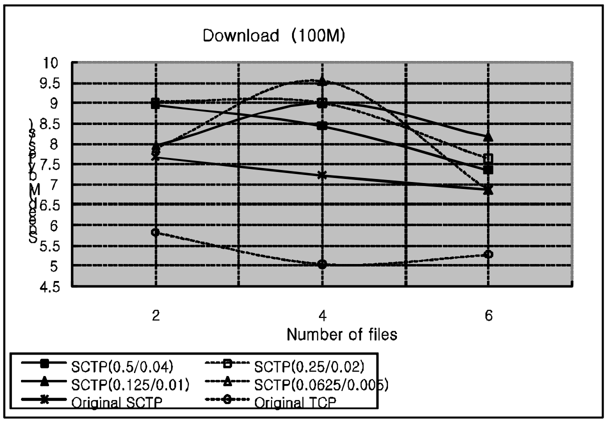

Fig. 5 Transmission speed (Download, 100M file)

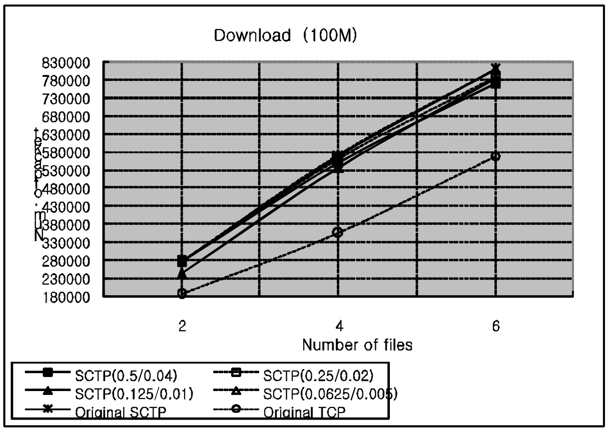

Fig. 6 Number of packets (Download, 100M file)

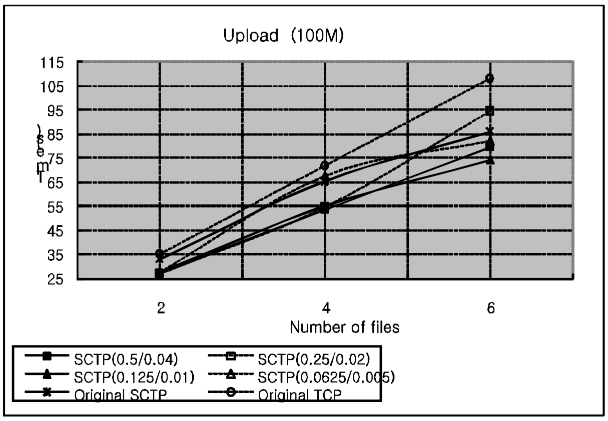

Fig. 7 Transmission time (Upload, 100M file)

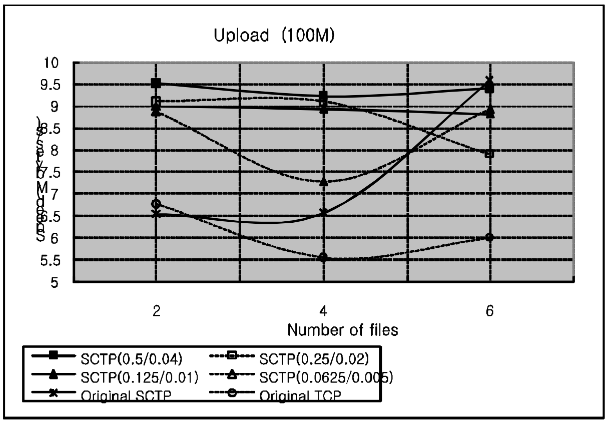

Fig. 8 Transmission speed (Upload, 100M file)

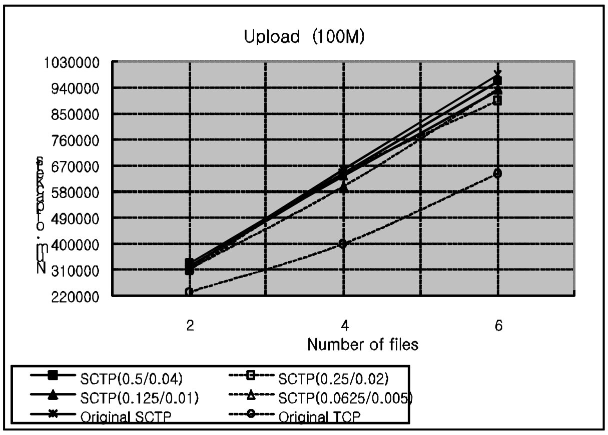

Fig. 9 Number of packets (Upload, 100M file)

References

[1] P.Amer, S. Iren, and P. Conrad, "The Transport Layer; Tutorial and Survey", ACM Computing Surveys, vol. 31, no. 4, Dec. 1999.

[2] J. Postel, "Transmission Control Protocol", IETF RFC 793, Sep. 1981.

[3] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, 2000.

[4] R. Stewart, and Xie Qiaobing, "Stream Control Transmission Protocol", Addison-Wesley, 2001.

[5] S. Fu, and M. Atiquzzaman, "SCTP: State of the art in Research, Products, and Technical Challenges", IEEE Communications Magazine, vol. 42, no. 4, Apr. 2004, pp. 64-76.

[6] S. Ladha, and P. Amer, "Improving multiple file transfers using SCTP multistreaming", 23rd IEEE and Comm Conf (IPCCC), Apr. 2003, pp. 13-22.

[7] Tasneem Yunus Kanpurwala, "SCTP multi-streaming: Study of transmission of partially reliable and reliable data using dynamic stream priorities", The Degree of MASTER of SCIENCE, Oklahoma State University, 2006.

[8] ISHTIAQ Ahmed, OKABE Yasuo, and KANAZAWA Masanori, "Improving Performance of SCTP over Broadband High Latency Networks", Proceedings of the 28th Annual IEEE International Conference on Local COmputer Network(LCN'03), 2003.

[9] Rob Brennan, and Thomas Curran, "SCTP Congestion Control: Initial Simulation Studies", International Teletraffic Congress, 2001.

[10] Gerard J. Heinz II, and Paul D. Amer, "Priorities In Stream Transmission Control Protocol (SCTP) Multistreaming, SCI '04, Jul. 2004.