International Journal of Modern Engineering Research (IJMER)

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1034-1038 ISSN: 2249-6645

Improved Multiple File Transfer Protocol using Extended features of SCTP

Prabhanshu Jaiswal [1],Gaurav Agrawal [2],Pushpendra Singh [3], Dr. A.K. Sharma [4]

(CSED, MMMEC Gorakhpur/ GBTU, INDIA)

ABSTRACT

The Stream Control Transmission Protocol (SCTP) is a reliable transport layer protocol, standardized by Internet Engineering Task Force (IETF). Transport layer is responsible for reliable delivery of messages from one host (source) to another host (destination). We are using the key features of SCTP (multi-streaming and multi-homing) with some addtional characteristics for transmission of multiple files. In current scenerio, SCTP creates streams only at the initial association establishment phase (before the transmission of files strats) and afterwards the steams are not allowed to be changed, but in this paper, we propose that these streams can be modified dynamically by using dynamic stream addition (DSA) policy of multi-streaming. And to speedup the throughput of SCTP, we use split fast retransmission (SFR) policy for the transmission of multiple files. We are also proposing a FTP architecture over this extended version of SCTP and illustrating algorithms for both ends (Server site and Client site). The peformance evaluation of our test is studied through ns-2. Our tests shows that the addition of these extended features of SCTP, significantly improve the performance of multiple file transfer.

Keywords: **Dynamic stream addition, FTP protocol, multi-homing, multi-streaming, SCTP.**

1. INTRODUCTION

Modern world is tightly attached with the Internet. The mainstream of communication now-a-days is via internet based applications. Internet is the worldwide, publicly accessible network of interconnected computer networks that transmit data through packet switching using the standard Internet Protocol (IP). This nature of internet demands the exchange of data be fast and reliable. Internet is an infrastructure that comprises of interconnections and by technical specifications or protocols that describe how to exchange data over the network. Technically there are 3 kinds of protocols depending at which layer they are being used:

Network layer protocols: At the lowest level is IP (Internet Protocol), which defines the datagrams or packets that carry blocks of data from one node to another.

Transport layer protocols: The protocols by which one host sends data to another. TCP and UDP are the intermediate layer protocols.

Application layer protocols: This describes the specific messages and data formats sent and understood by the applications running at each end of the communication.

In 1998, an IETF working group (SIGTRAN) was formed to design a mechanism for reliably transporting call control signaling over the Internet. SIGTRAN's goal was to create an IP complement to the telephone switching's SS7 network. SIGTRAN's work was focused to address few key problems in the use of TCP [1]**:**

1.1. Head of line blocking - A problem where sending independent messages over an order-preserving TCP connection causes delivery of messages sent later to be delayed within a receiver's transport layer buffers until an earlier lost message is retransmitted and arrives. These later messages often establish independent telephone calls. For call control signaling, the delay on later messages caused critical call control timers to expire, thus resulting in undesirable call setup failures.

1.2. Message Boundary– Since TCP is a byte stream oriented transfer protocol, the message boundaries are not saved multiple messages from the server to the client could be combined into one, making it difficult for the client to split the messages.

To remedy these situation, SCTP came into existence. The first key feature of SCTP is multi-streaming as describe in RFC 4960. Multi-streaming provides an aggregation mechanism to multiplex logically separate message streams to the same association. While the standard SCTP [1] creates streams only at the initial association establishment phase and afterwards the streams are not allowed to modify. When transmitting web objects concurrently with multi-streaming, it causes unnecessary waiting, as a result of large objects occupying the streams while small ones having no streams allocated.

To improve throughput of multiple file transmission, we add an extension of SCTP to reduce web response time by DSA-SCTP [3].

The second key feature of standard SCTP is multi-homing. A multi-homed SCTP host is accessible through multiple IP addresses. If one of its IP addresses fails-possibly from an interface or link failure, severe congestion, or slow route convergence around path outages—the destination host can still receive data through an alternate source interface.

We allow multiple files to transfer concurrently from multiple paths at the sender ends. To prevent the side effects as follows ,

(i) unnecessary fast retransmissions by a sender, (ii) overly conservative cwnd growth at a sender, and (iii) increased ACK traffic due to fewer delayed ACKs by a receiver, we use the three algorithms as described in [4], which facilitates

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1034-1038 ISSN: 2249-6645

to inflate the congestion window and to retransmit the lost data packets as quickly as possible.

Finally, we propose a model for transmitting multiple files using our proposed FTP protocol. Basically, FTP is an application layer protocol, used for exchanging files over any network that supports the TCP/IP protocol. Our proposed FTP over extension of SCTP has much benefits upon FTP over standard SCTP, the performance of transmission of files through transport layer will remarkably increase. We also give the algorithms for transmission multiple files through FTP over the extended version of SCTP.

The comparison of multiple file transfer over TCP, standard SCTP and Extension of SCTP is shown in the next coming section, which clearly shows that the transmission of multiple files using extension of SCTP improve the performance.

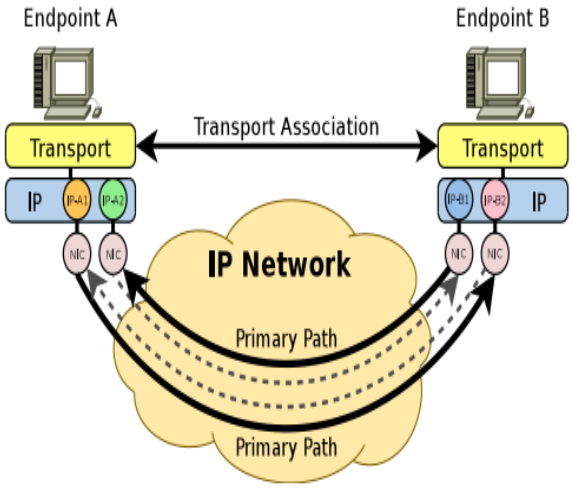

Fig 1. File transfer from multiple-paths

2. EXTENSION OF SCTP

2.1. MULTI -STREAMING IN EXTENDED SCTP

SCTP multi-streaming provides an aggregation mechanism to multiplex logically separate message streams to the same association. While the current SCTP creates streams only at the establishment phase while making the initial association and afterwards the streams are not allowed to be modified. In this paper, we add an extension of the standard SCTP to support dynamic stream addition during the communication [3].

Using independent streams in a SCTP association(), SCTP decouples the reliable data transfer from the strict order-of-transmission data delivery. The messages from the application layer are assign to different streams according to the requirement of the SCTP supported client or the scheduler of the server.

In this paper, we present the necessary modification to the SCTP specification to support the function of dynamic stream addition. In the modified SCTP, both sides of an association are allowed to add new out-streams and in-streams dynamically during the communication. Dynamic stream addition scheme is able to follow the application's requirement to create new streams when it is necessary to match a bursty objects concurrent transmission. We make use of DSA-SCTP [3] to the web server to reduce its response time.

2.2. MULTI-HOMING IN EXTENDED SCTP

In the standard SCTP, multi-homed hosts are used only for the purpose of redundancy and continuous transfer of control/data chunks from the sender host to destination host. Initially one path is chosen as a primary path among the set of available paths and rest are as secondary paths.

In the extension of SCTP, it enhance the concurrent multi-paths transfer of files in CMT-SCTP's multi-homing feature to distribute data across multiple end-to-end paths in a multi-homed SCTP association. The three negative side effects of reordering introduced by CMT are :

(i) unnecessary fast retransmissions by a sender,
(ii) overly conservative cned growth at a sender,

and (iii) increased ACK traffic due to fewer delayed ACKs by a receiver.

These drawbacks can be removed by implementing the three algorithms as described in [4], under the strong assumption that the bottleneck queues on the end-to-end paths used in CMT are independent, which is further modified by load sharing [5].

3. PROPOSED FTP ARCHITECTURE

3.1. FTP OVER EXTENDED SCTP

FTP over Extension of SCTP is similar to that of over TCP, the major change would be the introduction of the association and usage of multi- streaming features. In this system multi-streaming combines the FTP control and data connections in a single association. So only one association is needed for one session and this eliminates the need of more than one connection for a transfer to take place. The architecture that we are proposing consists of the following steps :

3.1.1. INITIATION PROCESS: Instead of establishing a connection with the server, now the client will request an association and during this initiation process both the hosts will open a stream which will be used for exchange of control commands and replies, say *Stream 0*. Also they agree on how many stream they will be using for the actual date transfer.

The number of streams can be dynamically increased as the need of application and the extra needed stream can be added with the help of DSA-SCTP scheme[3].

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1034-1038 ISSN: 2249-6645

3.1.2. EXCHANGE OF CONTROL MESSAGES: Once the association is set up and the hosts have the agreed upon a control stream the exchange of the control messages take place. For instance, server sends the list of files available for download and the client sends back the list of files that it wants to upload.

3.1.3. FILE TRANSFER: After receiving the list of file(s) from the client the server opens multiple streams and then starts transferring multiple file simultaneously by using Concurrent Multipath Transfer (CMT) using SCTP Multi-homing Over Independent End-to-End Paths[4] and load sharing scheme [5]. On the client side, we maintain an array that has the pointers to the file streams to which the data has to be written. Depending on the stream on which the data is coming the same is written to the associated file stream.

3.1.4. TRANSFER TERMINATION: Termination of file transfer has to be considered on each stream at each path. Once the file is transferred a CLOSE command is passed on the stream, since SCTP maintains the message boundaries it is possible to send a termination message. Client upon receiving the termination message from a stream, it goes and closes the file stream as well as the path associated with that stream and then clean the array so that a new file could be sent over the same stream.

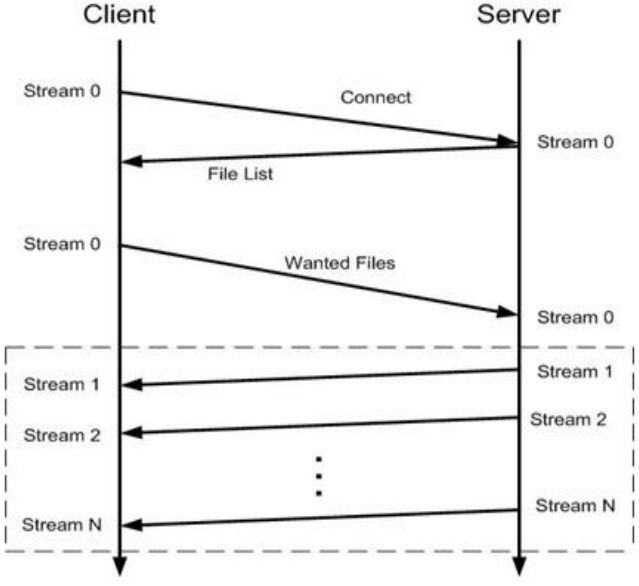

Fig 2. Transmission of files

3.2 ALGORITHM USED FOR TRANSMITTING MULTIPLE FILES USING EXTENDED SCTP

These algorithms run at their respective hosts. The Server site is always in running mode and waiting for a request. Whenever a client wants to access some file from the server site, it sends a request (INIT chunk)to the server and the server responds with INIT-ACK chunk to the client. If the client is an authentic user only then it replies with a COOKIE chunk. And then server sends a COOKIE-ACK chunk to the client in the encrypted form to avoid the intrusion from attackers. In this way the connection between user and server will be established by using four-way handshaking.

Termination of connection is done by using Full Shutdown at the both side. Hence, this procedure also provides some kind of security from the attackers because all the chunks are in encrypted from and only valid user can get access of the resources of server.

3.2.1. AT THE SERVER SITE :

step 1. *Server waits for Clients.*
step 2. *Accept a request from a client and fork a child.*
step 3. **if** PORT_NO already is in use, ***then***
generate an error massage
else
{
generate varification tag for specific Port
and inform the client;
}
step 4. *Get list of files from Client.*
step 5. *Send the list to the Cllient and initializes* ***stream_no*** .
step 6. Get requested file size information from the Client
step 7. **do {**
If ***stream_no*** <***objects_requested***,
{
create extra streams dynamically and
inform the clients;
Add the newly created streams in the
existing association;
update the stream_no;
}
Send file object in the open streams.
}while(! SHUTDOWN);
step 8. *Close all the streams.*
step 9. *End.*

3.2.2. AT CLIENT SITE :

step 1. *Clients request to the Server.*
step 2. ***if*****"the request is not fulfilled by server",** *then terminate.*
step 3. *Get PORT_NO and verification tag information from the Server.*
step 4. *Get list of files from Server.*
step 5. *Formate and display the user.*
step 6. *Get wanted list of files from user.*
step 7. *Prepare for incoming message and create an array of file pointer* ***FP*** *and initialize to all by* ***NULL*** **.**
step 8. *Get the message and cache the* streams, *let* **i**
step 9. ***If 'i' is a control stream,*** then
If *it has* ***terminate message,*** *then*
{
Close all stream;
terminate the connection;
}
else
write to ***FP[i].***
step 10. ***GOTO*** *step 9.*
step 11. *End.*

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1034-1038 ISSN: 2249-6645

3.3. Experimental Result and Analysis

We setup a simple network for investigating the features of the extended SCTP. In our network we use two nodes – one as Server and other as a Client. Both are running on Linux Kernel 2.4.19. We have performed various experiments on transferring multiple files, having different size, using the features of TCP, SCTP and extended SCTP as we proposed earlier. For the performance evaluation we took less than 1Mb, greater then 1Mb and less than 5Mb, and greater than 5Mb files for transmission and their respective durations were recorded. All experiments are done repeatedly and the average of that is taken and shown in the table of fig. 3.

This experiment shows that TCP and standard SCTP took much more time than the Extended SCTP for transferring multiple files between two nodes. Although Extended SCTP has some over-head of managing incoming and outgoing packets/chunks, but the overall performance of Extended SCTP much better than the standard SCTP and TCP.

NUMBER OF FILES	Time Taken for TCP	Time Taken for SCTP	Time Taken for Extended-SCTP
File Size: less than 1Mb.			
1 File	0.085	0.070	0.067
2 Files	0.165	0.089	0.072
3 Files	0.247	0.118	0.096
4 Files	0.314	0.156	0.129
5 Files	0.435	0.246	0.184
File Size: Greater than 1 Mb and Less than 5 Mb.			
1 File	0.511	0.557	0.048
2 Files	1.035	0.864	0.792
3 Files	1.75	1.048	0.917
4 Files	2.681	1.532	1.082
5 Files	3.152	1.946	1.207
File Size: Greater than 5 Mb.			
1 File	0.942	0.950	0.839
2 Files	1.844	1.094	0.954
3 Files	2.865	1.561	1.051
4 Files	3.512	1.937	1.537
5 Files	4.836	2.720	1.964

Fig 3. Statistics of Multiple File Transfer

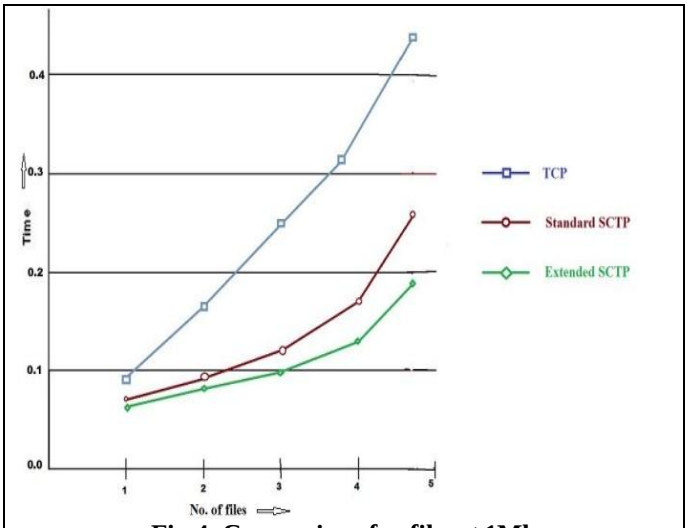

Fig 4. Comparison for files < 1Mb

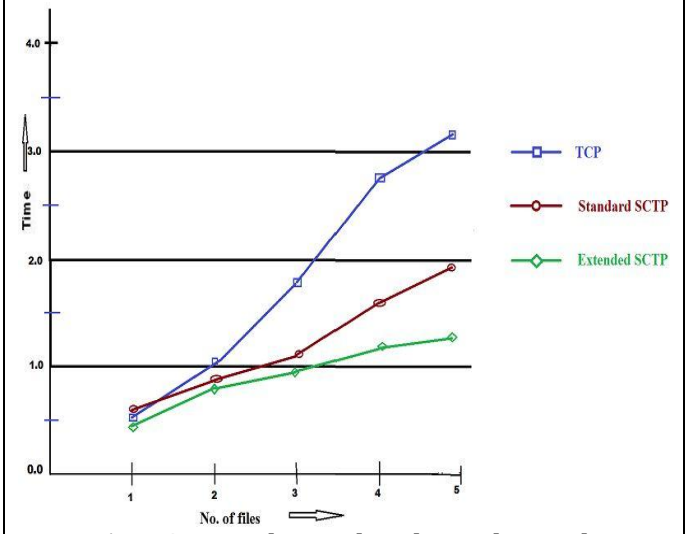

Fig 5. Greater than 1Mb and Less than 5Mb

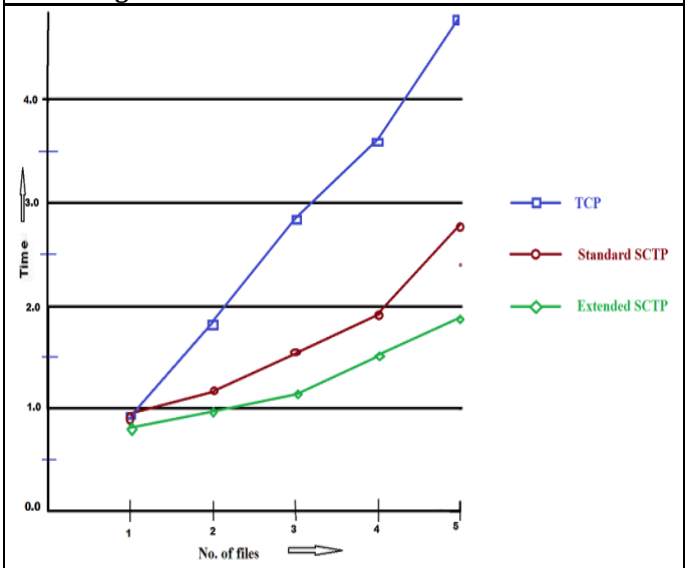

Fig 6. Greater than 5Mb

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-1034-1038 ISSN: 2249-6645

4. FUTURE WORKS

4.1. FTP OVER SCTP ON INTERNET: The only constraint that this project had was, the SCTP protocol was not matured enough when the work on this project began. So the availability was a problem, this protocol was only available in the CS LAN so all the modules were run on INTRANET. Now the SCTP protocol could be installed on a LINUX box that could be out of the network thus this application could be run over INTERNET and record the Performance gain.

4.2. IMPLEMENTATION OF SCTP IN MOBILE NETWORKS USING MULTI-HOMING: In the recent past, a lot of research was going on SCTP protocol and one of the areas of research is to implement Multi-Homing, a SCTP feature, in Mobile Networks and see whether this feature could do any good for the existing Mobile Networks.

4.3. HTTPS/FTPS OVER SCTP: To overcome the inefficiencies present in the standard protocols like HTTP and FTP currently enhanced and more secured protocols are being used, they are HTTPS and FTPS. An implementation of these protocols over SCTP could make the argument that SCTP could replace TCP or UDP stronger.

5. CONCLUSION

In this paper, we have evaluated the extension features of SCTP – dynamic stream addition behavior[3] in order to show the advantages of multi-streaming in multiple file transfer and by using concurrent multi-path transmission policy, we can use some of the available paths for increasing the throughput of transmission of multiple files. Then we have presented our results comparing the SCTP[1] against the TCP, assuming different scenarios such as head of line blocking problem and Denial of service attack. Furthermore, TCP use multiple connection for simultaneously transfer of multiple files when SCTP needs only one association i.e. one control stream for control connection and multiple data stream for multiple file transfer. Extension of SCTP features such as DSA[3] can also be used to find good response time in web application and multimedia application etc. We also show the comparison results of TCP, standard SCTP and Extended SCTP for multiple file transferring. We use multi-homing features for fault tolerant and load sharing in multiple file transfer.[5]

6. REFERENCES

[1] R. Stewart, Q. Xie, "Stream Control Transmission Protocol (SCTP): A Reference Guide", Addison Wesley, 2001, ISBN: 0-201-72186-4.

[2] JinwookSeo, YoungjuAhn , Minki No, Hyunchel Kim, Seongjin Ahn and Jinwook Chung, "Implementof FTP application using SCTP," *International Journal of Computer Scienceand Network Security*, VOL.6 No.12, December 2006.

[3] MiaoXue, Haisheng Jiang, Ping Dong, Yajuan Qin,Sidong Zhang, Hongke Zhang, "DSA-SCTP: ANEXTENSION OF SCTP TOREDUCE WEBRESPONSE TIME," *in proceedings of IEEE ICCTA,978-1-4244-4817*, Mar. 2009.

[4] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP MultihomingOver Independent End-to-End Paths*,"IEEE/ACM Transactions on Networking, vol. 14, no. 5, pp. 951–964*,Oct. 2006, ISSN 1063-6692.

[5] Inwhee Joe, Sijia YAN, "SCTP Throughput Improvement with Best Load Sharing based on Multi-homing," *Fifth IEEE International Joint Conference on INC, IMS and IDC, 2009.*

[6] P. Natarajan, N. Ekiz, E. Yilmaz, P.D. Amer,and J. Iyengar,"(NR-SACKs) Non-Renegable Selective Acknowledgments for SCTP," *in Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP), Orlando, Florida/U.S.A.*, Oct. 2008, pp. 187– 196, ISBN 978-1-4244-2506-8.

[7] Lin Cui, SeokJooKoh, and Woo Jin Lee, "Fast Selective ACK Scheme for Throughput Enhancement of Multi-Homed SCTP Hosts," IEEE communications letters, Vol. 14, No. 6, JUNE 20i0.