Comparing Study of Transport Layer Protocols SCTP with TCP and UDP

Mr. Pradip G. Vanparia [#1], Dr. Yogesh. R. Ghodasara [*2]

[#1] Asst. Professor
Department of Computer Science & Technology
Shree M & N Virani Science College Rajkot
[1] *pradeep.vanparia@gmail.com*

[*2] Principal and Dean,
College of Agricultural Information Technology,
Anand Agricultural University, Anand
[2]yrghodasara77@yahoo.co.uk

Abstract— **One of the main requirements of 4G systems is that the users should not feel any difference between a wired and a wireless network and they should have multiple options for connectivity over heterogeneous networks. In other words, it needs a new adaptive, reliable, and efficient TCP that is able to fulfill these requirements. The direct approach to enhance the performance of TCP for the new generation networks is to modify the TCP itself because TCP still represents one of the major causes of poor performance in a wireless environment.**

The transport layer's primary role is to provide end-to-end communications service between two or more applications running on different hosts. The transport layer can also perform sophisticated actions such as flow control, error recovery, and reliable delivery, which might be necessary.

TCP (Transmission Control Protocol) has been around for quite some time and it has provided us with the protocol to move data from one point to another in our computer networks. TCP is a connection-oriented protocol, a connection can be made from client to server, and from then on any data can be sent along that connection. Despite its success, TCP also has many limitations.

UDP stands for User Datagram Protocol. It is a minimal message-oriented transport layer protocol. It enables two hosts to connect and send short messages to one another. Unlike Transmission Control Protocol (TCP), it does not guarantee that data is received or that it is received in the order in which it was sent.

SCTP (Stream Control Transmission Protocol) does everything that TCP does while adding new features that TCP doesn't have. SCTP is, in many ways, superior to TCP. SCTP Stands for Stream Control Transmission Protocol. SCTP is connection oriented protocol. But due to our long term use of TCP, it is quite hard to transition to the superior one.

Keywords— SCTP, Stream Control Transmission Protocol, TCP, HTTP, Multihoming, Multistreaming, HOL, UDP

I. INTRODUCTION

The transport layer is the fourth layer of the OSI Model and exists between the Session Layer and Network Layer. The transport layer is used to actually transfer data from one application to another and can be used to stream data from one device to another. The transport layer is used to assemble, compile, and encode data so that it is ready to be transported.

The transport layer is advantageous because it can assemble data extremely fast and open a connection between two or more applications even if they are located on different devices. The transport layer is also advantageous because it provides error checking and data recovery as well as both connection-oriented and connection less communication.

In the Internet protocol suite this function is achieved by the connection oriented Transmission Control Protocol (TCP) and connection less User Datagram Protocol (UDP).

TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until the message to be exchanged between client and server.TCP checks packets for errors and submits requests for re-transmissions if errors are found; it will also return the multiple packets of a message into a proper, original sequence when the message reaches its destination.

A TCP has several limitations. Head-of-line locking, timer granularity, a lack of built-in support for multihoming, the lack of a built-in heartbeat mechanism. SCTP offers recognized error-free, non-duplicated transfer of messages. Detection of data loss, data corruption and duplication of data is achieved by using checksums and sequence numbers.

UDP is connectionless protocol. When you a send a data or message, you don't know if it'll get there, it could get lost on the way. There may be corruption while transferring a message. If we send two messages out, you don't know what order they'll arrive in (no ordered).

The comparison studies of SCTP protocols. The results of studies show that SCTP has more advantages than TCP which are less delay, less packet loss, less HOL blocking, more efficient congestion control, and higher throughput of data transfer.

II. HANDSHAKE

Initiating a new connection in TCP, UDP and SCTP occurs with a packet handshake. In TCP, it's a called a

three-way handshake. The client sends a SYN packet (short for Synchronize) for which the server responds with a SYN-ACK packet (Synchronize-Acknowledge). Finally, the client confirms receipt with an ACK packet.

The problem that can occur with TCP is when a rogue client forges an IP packet with a bogus source address, and then floods a server with TCP SYN packets. The server allocates resources for the connections upon receipt of the SYN, then under a flood of SYN packets, eventually runs out and is unable to service new requests. This is called a Denial of Service (DoS) attack.

UDP is a connectionless protocol, which means that it doesn't require a formal handshake to get the data flowing. A frame that contains UDP data simply communicates to another station without any prior warning or fanfare. Of course, the receiving station must be configured to receive data on the appropriate UDP port, but no formal handshaking process is required. To send UDP data, it's packaged up in an IP frame and sent on its way.

SCTP protects against this type of attack through a four-way handshake and the introduction of a cookie.

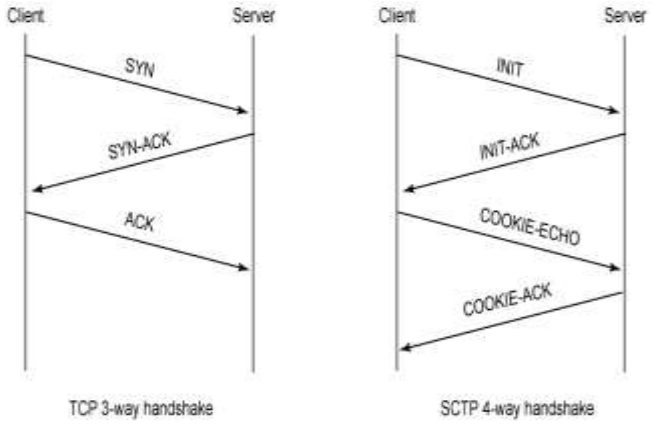

Figure 1. TCP three-way handshake & SCTP four-way handshake

In SCTP, a client initiates a connection with an INIT packet. The server responds with an INIT-ACK, which includes the cookie (a unique context identifying this proposed connection). The client then responds with a COOKIE-ECHO, which contains the cookie sent by the server. At this point, the server allocates the resource for the connection and acknowledges this by sending a COOKIE-ACK to the client.

To solve the problem of delayed data movement with the four-way handshake, SCTP permits data to be included in the COOKIE-ECHO and COOKIE-ACK packets.

III. ORDERING

TCP transmissions are sent in a sequence and they are received in the same sequence. In the event of data segments arriving in wrong order, TCP reorders and delivers application.

In the case of UDP, sent message sequence may not be maintained when it reaches receiving application. There is absolutely no way of predicting the order in which message will be received.

Messages in SCTP are transferred reliably but not necessarily in the desired order. TCP guarantees that data is delivered in order (which is a good thing, considering TCP is a stream protocol). UDP guarantees no ordering. But, you can also configure streams within SCTP to accept unordered messages if desired. This feature can be useful in message-oriented protocols in which requests are independent and ordering is not important. Further, you can configure unordered delivery on a stream-by-stream basis within an association.

IV. MULTI-HOMING

Multi-homing provides applications with higher availability than those that use TCP. A multi-homed host is one that has more than one network interface and therefore more than one IP address for which it can be addressed. In TCP, a connection refers to a channel between two endpoints (in this case, a socket between the interfaces of two hosts). SCTP introduces the concept of an association that exists between two hosts but can potentially collaborate with multiple interfaces at each host.

At the top is a TCP connection. Each host includes a single network interface; a connection is created between a single interface on each of the client and server. Upon establishment, the connection is bound to each interface.

At the bottom of the figure, you can see an architecture that includes two network interfaces per host. Two paths are provided through the independent networks, one from interface C0 to S0 and another from C1 to S1. In SCTP, these two paths would be collected into an association.

SCTP monitors the paths of the association using a built-in heartbeat; upon detecting a path failure, the protocol sends traffic over the alternate path. It's not even necessary for the applications to know that a failover recovery occurred.

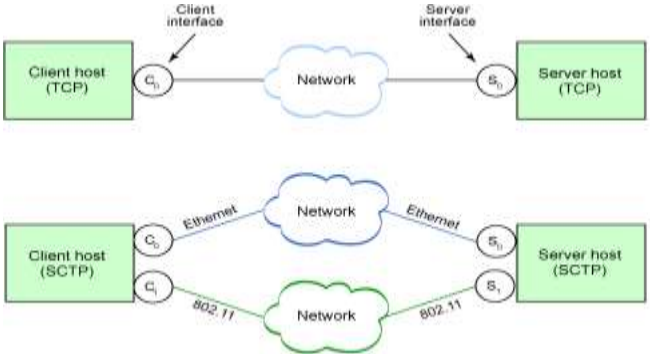

Figure. 2. The difference between a TCP connection and an SCTP association

Failover can also be used to maintain network application connectivity. For example, consider a laptop that

includes a wireless 802.11 interface and an Ethernet interface. When the laptop is in its docking station, the higher-speed Ethernet interface would be preferred (in SCTP, called the primary address); but upon loss of this connection (removal from the docking station), connections would be failed over to the wireless interface. Upon return to the docking station, the Ethernet connection would be detected and communication resumed over this interface. This is a powerful mechanism for providing high availability and increased reliability.

V. MULTI-STREAMING

Using TCP, only one single data stream is allowed per connection. All of the information must be passed through that one stream. SCTP allows multiple simultaneous data streams within a connection or association. Each message sent to a data stream can have a different final destination, but each must maintain message boundaries. For example, systems cannot send parts of the same message through different streams; one message must go through one stream.

When running an ordered data delivery system, if one of the packets is out of order or missing, the stream is blocked pending resolution to the order. This is called "Head-of-Line blocking." With the use of multi-streams, only the stream that is affected would be blocked; the other streams would continue to flow.

Multi-streaming is an important feature of SCTP, especially when you consider some of the control and data issues in protocol design. In TCP, control and data typically share the same connection, which can be problematic because control packets can be delayed behind data packets. If control and data were split into independent streams, control data could be dealt with in a timelier manner, resulting in better utilization of available resources.

Relationship of an SCTP association to streams

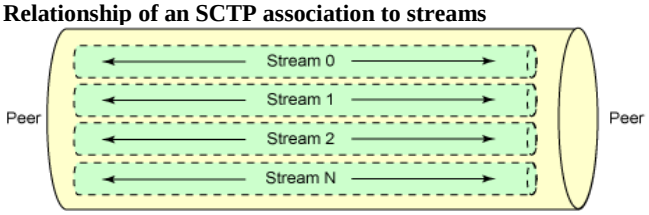

Figure 3. SCTP Association

VI. MESSAGE FRAMING

With message framing, the boundaries in which messages are communicated through a socket are preserved; this means that if a client sends 100 bytes to a server followed by 50 bytes, the server will read 100 bytes and 50 bytes, respectively, for two reads. UDP also operates in this way, which makes it advantageous for message-oriented protocols.

In contrast, TCP operates in a byte-stream fashion. Without framing, a peer may receive more or less than was sent (splitting up a write or aggregating multiple writes into a single read). This behavior requires that message-oriented protocols operating over TCP provide data-buffer and message framing within their application layer (a potentially complex task).

SCTP provides for message framing in data transfer. When a peer performs a write on a socket, it is guaranteed that this same-sized chunk of data will be read at the peer endpoint (see Figure 5).

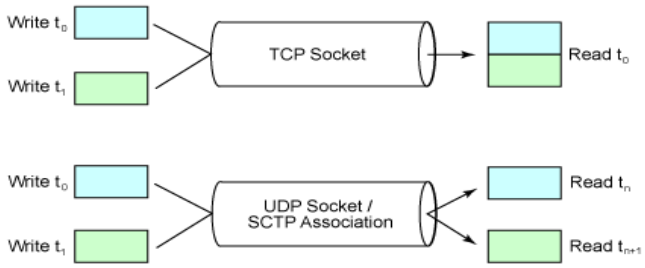

Figure4. Message framing in UDP/SCTP vs. a byte-stream-oriented protocol

VII. GRACEFUL SHUTDOWN

TCP and SCTP are connection-based protocols, while UDP is a connection-less protocol. Both TCP and SCTP require connection setup and teardown between peers.

TCP and SCTP connection termination sequences

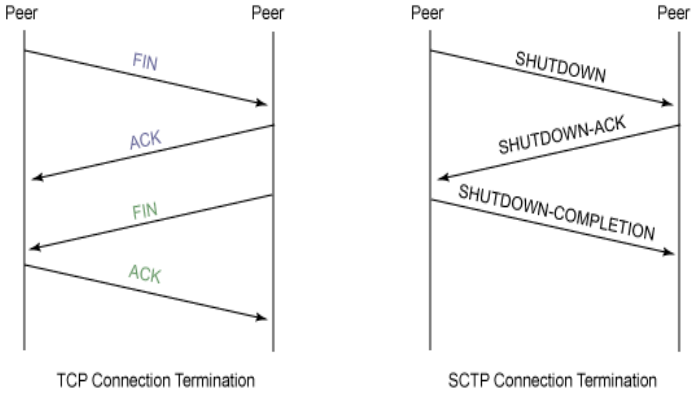

Figure 5. TCP and SCTP connection termination sequences.

In TCP, it's possible for a peer to close its end of a socket (resulting in a FIN packet being sent) but then to continue to receive data. The FIN indicates that no more data is to be sent by this endpoint, but until the peer closes its end of the socket, it may continue to transmit data. Applications rarely use this half-closed state, and therefore the SCTP designers opted to remove it and replace it with a cleaner termination sequence. When a peer closes its socket (resulting in the issuance of a SHUTDOWN primitive), both endpoints are required to close, and no further data movement is permitted in either direction.

VIII. CONCLUSION

In this paper, we present our comparison of the SCTP with TCP and UDP protocol as the transport protocol. TCP is very vulnerable to denial of service attacks, which locks up the resources of the server and prevents others from using it. Rather than TCP's 3-way handshake initiation, SCTP uses a 4-way handshake that allocates resources near

the end of the entire handshake. Because of this, SCTP s not vulnerable to DoS attacks.

The Stream Control Transmission Protocol (SCTP) is a reliable transport protocol that provides stable, ordered delivery of data between two endpoints (much like TCP) and also preserves data message boundaries (like UDP). However, unlike TCP and UDP, SCTP offers such advantages as multi-homing and multi-streaming capabilities, both of which increase availability.

Lastly, SCTP has made delivery ordering optional. Delivery ordering is necessary in many instances. That's why it is mandatory in TCP. But in certain cases where ordering is not needed, resources can be freed by disabling this capability. SCTP provides flexibility to suite each situation with optional ordered delivery.

SCTP is, in many ways, superior to TCP. But due to our long term use of TCP, it is quite hard to transition to the superior one. Nonetheless, the advantages of SCTP would gradually draw attention and users.

REFERENCES

1) Ghassan A. Abed, "Comparative Performance Investigation of TCP and SCTP Protocols over LTE/LTE-Advanced Systems" ,Vol. 1, Issue 6,Aug12,ISSN:2278–1021
2) Protocol Engineering Lab, U Delaware, URL:www.pel.cis.udel.edu/
3) Ladha.S, Amer.P.D. "Improving file transfers using SCTP multistreaming", IPCCC, 2004, Phoenix, Arizona, pp513-522.
4) S. Fu and M. Atiquzzaman, SCTP: State of the Art in Research, Products and Technical Challenges, IEEE Journal Communication Magazine, April 2004
5) F. Perotto, C. Casetti, and G. Galante, SCTP-based Transport Protocols for Concurrent Multipath Transfer, IEEE Wireless Communications and Networking Conference, 2007
6) J. Postel, "Transmission Control Protocol," IETF RFC 793,Sept. 1981; available at http://ietf.org/rfc/rfc793.txt.
7) "Defining Strategies to Protect against TCP SYN Denial of Service Attacks," Cisco Systems, tech. memo,Aug.2001; http://www.cisco.com/warp/public/707/4.html
8) Stream Control Transmission Protocol (SCTP) http://www.sctp.org/implementations.html/
9) SCTP - DataTAG /datatag/WP3/sctp/index.htm
10) H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm,R. Katz, "TCP behavior of a busy Internet server: Analysis and Improvements," IEEE INFOCOM, San Francisco, 3/98
11) OpenBSD Project. OpenSSH Homepage, July 2005. http://www.openssh.org/.

