SCTP : AN ALTERNATIVE TRANSPORT LAYER PROTOCOL

DR. YOGESH R. GHODASARA

Associate Professor, College of Agricultural Information Technology
Anand Agricultural University, Anand 388110, GUJARAT (INDIA)

yrghodasara77@yahoo.co.uk

ABSTRACT—*TCP has performed immense service as the primary means of reliable data transfer in IP networks. However, an increasing number of recent applications have found TCP too limiting, and have incorporated their own reliable data transfer protocol on top of UDP. SCTP (Stream Control Transport Protocol) is an answer to some of the limitations imposed by TCP protocol. This paper is a study of TCP limitations found in today's network applications and features of SCTP to resolve these limitations.*

Keywords -TCP, SCTP ,Multihoming, Stream

1. INTRODUCTION

The Application Developers facing following limitations in TCP based Network Application Development.

1. TCP provides both reliable data transfer and strict order-of-transmission delivery of data. Some applications need reliable transfer without sequence maintenance, while others would be satisfied with partial ordering of the data. In both of these cases the head-of-line blocking offered by TCP causes unnecessary delay.
2. The stream-oriented nature of TCP is often an inconvenience. Applications must add their own record marking to delineate their messages, and must make explicit use of the push facility to ensure that a complete message is transferred in a reasonable time.
3. The limited scope of TCP sockets complicates the task of providing highly-available data transfer capability using multi-homed hosts.
4. TCP is relatively vulnerable to denial of service attacks, such as SYN attacks.

Transport of PSTN signaling across the IP network is an application for which all of these limitations of TCP are relevant. While this application directly motivated the development of SCTP, other applications may find SCTP a good match to their requirements.

2. SCTP ARCHITECTURE

The Stream Control Transmission Protocol (SCTP) is an IP transport protocol. It resides at an equivalent level with TCP and provides a reliable transport service, ensuring that data is transported across the network without error, and in sequence. It is designed to transport Public Switched Telephone Network(PSTN) signaling messages over IP networks, but is capable of broader applications.

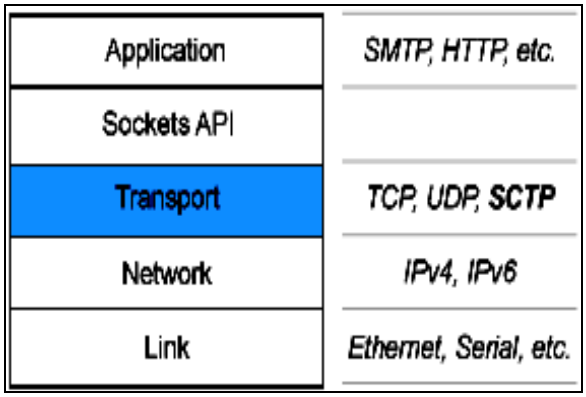

Figure 1 SCTP in Layered architecture of the IP stack

SCTP employs a session-oriented mechanism (like TCP) for transmission of data. A relationship is established between the endpoints of SCTP associations prior to data transfer, and this relationship is maintained until all data transmission has been successfully completed.

3. SCTP FEATURES

There are many good features in SCTP. The most important ones are:

1. Multi-Stream Capability
2. A broder concept known as Association for connection end points
3. Multi-homing Support
4. Path and peer failure detection

3.1 Multi-Stream Capability

While TCP assumes a single-stream data transfer with byte sequence preservation, SCTP offers the multi-streaming facility. The term multi-stream means a sequence of user messages that are to be delivered to the upper-layer protocol in order, with respect to other messages within the same stream. This is in contrast to its use in TCP, where it refers to a sequence of bytes.

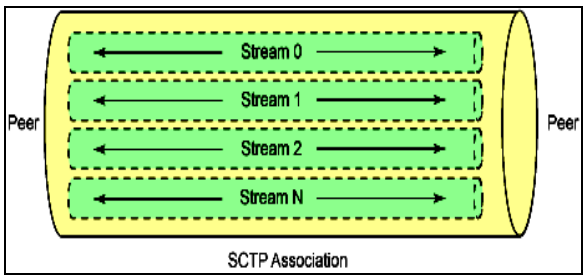

Figure 2 Streams in SCTP

Streams allow user data to be partitioned into multiple 'streams', each with independently sequenced delivery. Therefore, message loss at one stream will not affect the delivery of other streams.

SCTP streams are unidirectional. This reduces the delay caused by 'head-of-line' blocking experienced in TCP, since there is an independence of delivery across 'streams' within an association. Even programmatically, one finds it useful to separate logically different streams of data over the same socket, and SCTP provides this facility.

3.2 Associations

SCTP is connection-oriented in nature. An SCTP association is a broader concept than a TCP connection.

3.2.1 The SCTP handshake mechanism

We are aware of the TCP three-way handshake. We are also aware of how this leads to SYN attacks. Figure 3 briefly illustrates how it happens.

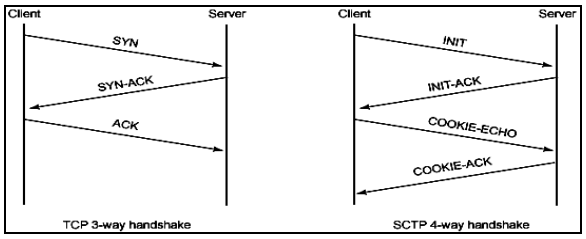

Figure 3 SCTP handshake mechanism

The problem that can occur with TCP is when a rogue client forges an IP packet with a bogus source address, 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.

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

3.2.2 The SCTP association 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. What's different about socket shutdown in SCTP is the removal of TCP's *half-close*. Figure 4 shows the shutdown sequences for TCP and SCTP.

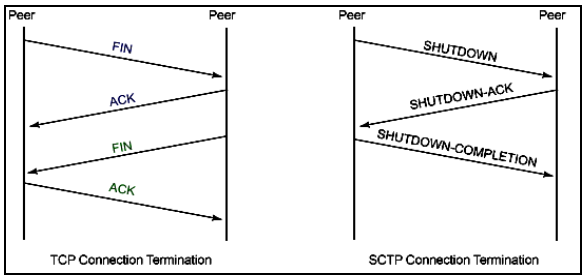

Figure 4 SCTP Shutdown

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, both endpoints are required to close, and no further data movement is permitted in either direction.

3.3 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.

Figure 5 illustrates the difference between a TCP connection and an SCTP association.

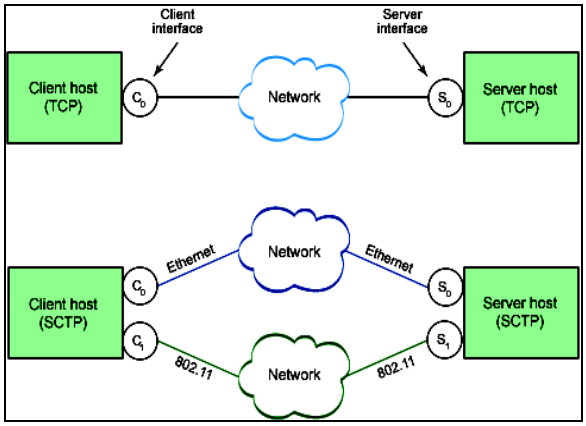

Figure 5 Multi-homing in SCTP

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.

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.

4. The Future of SCTP

SCTP is a relatively new protocol, considering that it became an RFC in October 2000. Since then, it has found its way into all major operating systems, including GNU/Linux, BSD, and Solaris. It's also available for the Microsoft Windows operating systems as a third-party commercial package.

Along with availability, applications will begin to use SCTP as their primary transport. Traditional applications such as FTP and HTTP have been built on the features of SCTP. Other protocols are using SCTP, such as the Session Initiation Protocol (SIP) and the Common Channel Signaling System SS7.

With the inclusion of SCTP into the Linux kernel, it's now possible to build and deploy highly available and reliable networked applications. As an IP-based protocol, SCTP is a seamless replacement for TCP and UDP but also extends new services, such as multi-homing, multi-streaming, and increased security.

5. CONCLUSION

In summary, we can compare SCTP with other two transport layer protocols TCP and UDP with following table.

Features	SCTP	TCP	UDP
Connection Oriented	Yes	Yes	No
Full Duplex	Yes	Yes	Yes
Reliable Data Transfer	Yes	Yes	No
Ordered Data Delivery	Yes	Yes	No
Unordered Data Delivery	Yes	No	Yes
Flow Control	Yes	Yes	No
Multi-streaming	Yes	No	No
Multi-homing	Yes	No	No
Protection against SYN flooding attacks	Yes	No	NA
Allows half-closed connections	No	Yes	NA

REFERENCES

[1] M. Tim Jones. *Better Networking with SCTP.* An IBM developerworks article

[2] http://www.sctp.org

[3] Srikanta Prasanna. *A conceptual Introduction to SCTP*. article in LinuxForYou Magazine Volume 07 Issue 12 pg.76-79

[4] RFC-3286 An Introduction to SCTP

[5] RFC-4960 Stream Control Transmission Protocol

[6] Stream Control Transmission Protocol (SCTP): A Reference Guide (Addison-Wesley, 2002).