

The Stream Control Transmission Protocol (SCTP) and its Potential for Amateur Radio

Eduardo Gonzalez
FlexRadio Systems
4616 W. Howard Lane Suite 1-150
Austin, TX USA 78728
Email: ed.gonzalez@flex-radio.com

Dr. Stan McClellan
Texas State Univeristy
601 University Drive
San Marcos, TX USA 78666
Email: stan.mcclellan@txstate.edu

Dr. Wuxu Peng
Texas State University
601 University Drive
San Marcos, TX USA 78666
Email: wuxu@txstate.edu

Abstract

The Stream Control Transmission Protocol (SCTP) provides several unique features not found in TCP (Transmission Control Protocol) and in UDP (User Datagram Protocol) while maintaining the useful aspects of both. This paper provides an introduction to SCTP and several features that should be very attractive to the Amateur Radio Community. These features are particularly useful with the rise of Software Defined Radio which use network interfaces to provide both control and data streams. The main features explored are multi-homing, multi-streaming and the ability to select reliable vs unreliable and ordered vs non ordered delivery of application messages.

I. INTRODUCTION

SCTP is a relatively young transport protocol which grew out of several projects/research that were trying to replace traditional telephone signaling methods with packet based protocols. TCP was found to have some reliability and timing weaknesses that were unacceptable to the industry, hence a new protocol was drafted. It became an IETF Proposed Standard in 2000 and was published as RFC 2960 [1] which was later replaced by RFC 4960 [2] in 2007. It is now mandatory to use SCTP for SS7-based PSTN signaling over IP as well as in the IETF Reliable Server Pooling framework [3]. SCTP is part of the transport layer of the TCP/IP model which mean

SCTP is a reliable, message oriented transport protocol that bridges and adds to a lot of the features found in UDP and TCP.

Like TCP, SCTP

- 1) **is connection oriented** A connection is established at both endpoints prior to sending any data. This is unlike UDP where the sender starts sending data towards an IP address with no prior knowledge of the connection.
- 2) **provides reliable data delivery** Data given to the protocol is assured to reach its destination as long as there is a connection. The protocol handles lost packets, congestion, etc.
- 3) **provides ordered data delivery** Data will be presented to the application in the order that it was sent. This does not mean that packets are assured to get to an endpoint in a certain order.
- 4) **provides flow and congestion control** The protocol is “smart” in the way it sends data so that unfavorable circumstances do not arise such as over congesting the network or overloading the receiver’s resources.
- 5) **provides path MTU discovery** MTU is the Maximum Transmission Unit which is the maximum packet size that a network can handle without fragmenting (splitting into several pieces) packets.

Like UDP, SCTP

- 1) **is message oriented** Unlike TCP, which treats all data as a format-less stream of bytes, SCTP and UDP preserve message boundaries given to it from the application.
- 2) **provides unordered data delivery** Although this seems contradictory to a previous statement it is true. SCTP supports both ordered and unordered data delivery.

Unlike TCP and UDP, SCTP

- 1) **provides message bundling** Small messages are automatically bundled together in the same packet if they fit. TCP does this at the Byte level with no awareness of the actual message boundaries.
- 2) **provides multi-stream support** Within a single connection (in SCTP terms - single association) there can exist many independent streams. They may be used for concurrent transfer or for independent data transfer.
- 3) **provides multi-homed hosts support** A single association can support multiple IP’s. These may exist on a single network interface card or on multiple network interface cards with each assigned an IP address.

- 4) **provides builtin reachability check** Similar to TCP’s “keep-alive” mechanism, SCTP’s heartbeat periodically checks to see whether a path/connection is still reachable. The main goal is to aid in a failover situation in contrast to TCP’s long-term state cleanup goal.
- 5) **provides state cookie mechanism** This security cookie is used to prevent SYN flood attacks where phony connections are requested in order to use up server resources.

As seen in the lists above SCTP has some features of both TCP and UDP while adding significant improvements such as multi-homing, multi-streaming and even security measures such as the state cookie mechanism. A closer inspection of the inner workings of the protocol is out of scope for this paper but the reader can look at [4] and [3] as well as the SCTP RFCs [2] for a more detailed explanation. For Amateur radio, SCTP’s potential lies within three of its unique features: multi-streaming, multi-homing and data delivery options.

II. MULTI-STREAMING

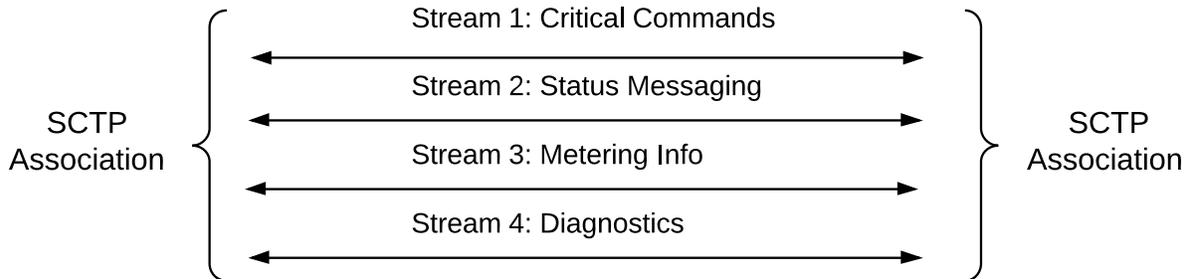


Fig. 1: Using SCTP Streams for partitioning and prioritization of radio control data

Within every SCTP association one or more streams can be created that have several independent properties. The most important is that messages within one stream will not block messages in another. This problem is known as head of line blocking and within TCP the entire connection is stalled if a packet is lost or comes in out of order. When dealing with time critical control streams that are independent of each other this can cause delays and make an application seem unresponsive. In today’s applications multi-streaming is often done by opening several TCP sessions per data stream which significantly increases code complexity as well as resource overhead. Each TCP connection has its own kernel RX and TX buffers as well as requiring independent management. With SCTP this is all built in to a single association, on a single port with one pair of kernel buffers. You can provide data redundancy across several paths to deal with packet loss or provide stream prioritization within a data transfer. For example, in the FLEX-6000 Series Radios [5], a single TCP connection is used as the control channel for the radio. This channel carries status messages for clients as well as commands. Some commands have a much higher priority than others but using TCP any packet that is delayed will block the other from arriving at the radio. As shown in Figure 1, using SCTP, several streams could be created for each type of traffic. Stream 1 could be dedicated to critical commands and have the highest priority. Stream 2 could handle status messages such as transmit on/off and power commands while the last two streams deal with lower priority traffic which in this case would be diagnostic information and metering data. When dealing with equipment that can transmit large amounts of energy the multi-streaming capability of SCTP provides another failsafe so that low priority traffic does not impede critical power related commands.

III. MULTI-HOMING

Another SCTP feature with significant potential within Amateur Radio is the support for multi-homed hosts. This means that an SCTP association can be attached to several different routes or network by using multiple IP addresses across different NICs. In the telecom industry this is critical since there are many nodes that can create different routes for information. If a section of the network goes down then the services and applications do not have to be restarted but switch automatically when a link is down. SCTP allows software writers to take full advantage of these different routes without having to handle multiple TCP connections and failover criteria in their application. SCTP can potentially be used over radio created mesh networks such as the ones proposed by KD6OZH in [6]. Mesh networks inherently have several routes that can be used with differing characteristics. SCTP bound to different nodes could provide safe failover to a different path depending on propagation changes or link utilization metrics. Multi-homing can also be used to add redundancy for critical applications or for convenience. Critical systems cannot afford single points of failure which in the case of networked systems usually requires several connections to different networks. This way, if one network goes down then the system can still use the remaining network. Figure 2 shows how SCTP manages these two paths and uses

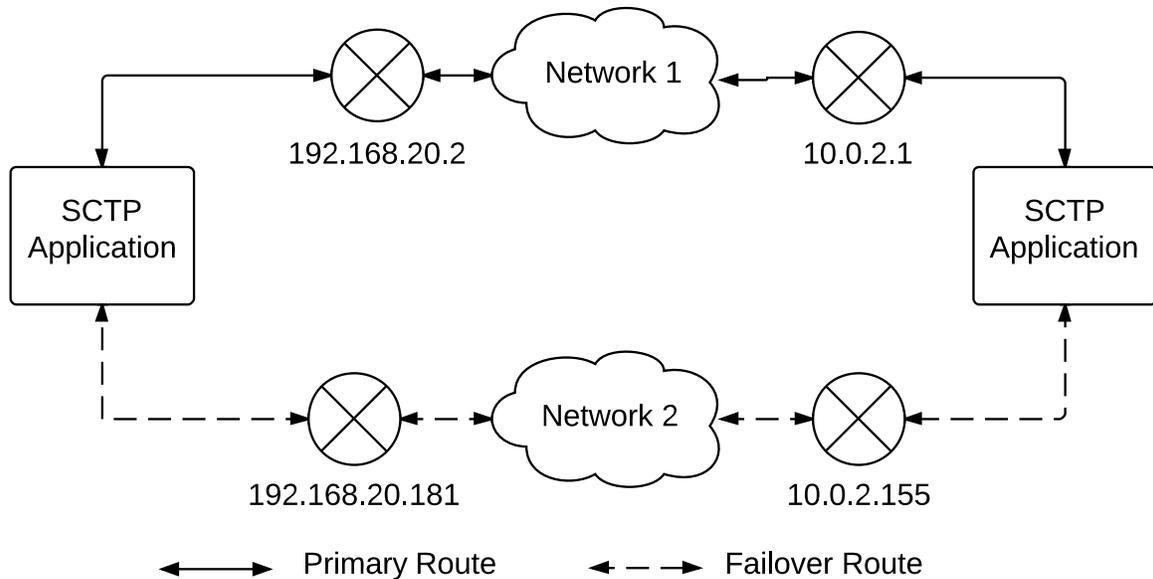


Fig. 2: Sctp Multi-homing path redundancy

one as a failover case. The primary route is Network 1 using IP addresses 192.168.20.2 and 10.0.2.1. As the application is sending data the Sctp protocol constantly monitors the state of Network 1, if it becomes unreachable then it begins to use Network 2 with IP addresses 192.168.20.181 and 10.0.2.155. To the application the switch is transparent and automatic. From a non-critical perspective this same failover mechanism can be convenient when an operator switches networks as in going from in home WiFi to on the road LTE. A TCP connection will time out and the operator will have to restart his applications. Using Sctp the primary network becomes the in home WiFi and the failover path becomes the LTE connection. If the operator leaves home his application seamlessly switches over and there is no operational interruption.

IV. DATA DELIVERY

When dealing with network data transfers, software engineers often choose the protocol they will use based on several trade offs one of which is how much custom software will have to be built on top of the protocol. For some types of data UDP is preferred since it provides a fast and simple way to transfer data. UDP provides high throughput with its low overhead scheme of simply sending packets into the network without worrying whether they arrive or if there is even a listener on the other end of the connection. This is very convenient for streaming data that has quick expiration characteristics. For example, panadapter information is only useful until the next frame of data arrives. There is no need to block rendering if a new frame has arrived but an old one was lost on the network. On the other hand as soon as you need reliable transfer or the data being transferred is dependent on ordering then those features must be built on top of UDP since it does not provide them natively. Depending on the requirements a shift to TCP might be required and with it a complete shift on how data is transferred. As mentioned in the Introduction UDP is message oriented while TCP is byte oriented. The move from UDP and TCP often includes a painful rewrite of how data is being passed to the protocol. With UDP you give the protocol a message and you get a message out on the other end. Using TCP a message is given to the protocol and you get a stream of bytes on the other end which may include several messages put together. TCP is simply transferring a stream of bytes and it is up to the application to handle boundaries and fragmentation. Sctp avoids this and the code change associated with it by supporting both facilities from each protocol. It is message oriented like UDP so the application does not have to handle boundaries and it provides the options for ordered/un-ordered and reliable/un-reliable data transfer.

A. Ordered vs Un-ordered

With an option change to the Sctp stream, it can switch between enforcing ordered delivery of messages to the application. This allows for a very flexible interface between Sctp applications. Once again this is useful since the application has the ability to choose depending on the type of data being exchanged with minimal code changes.

B. Reliable vs Un-reliable

Very similarly an SCTP stream can change from reliable data transfer like in TCP and unreliable data transfer like UDP. Situations where unreliable data transfer is suitable would include data with very fast expiration. This could include metering data since a meter's value expires when there is new information and graphic rendering. On the other hand command information, and status information may be critical for an application and therefore the protocol needs to guarantee that it has arrived on the other end. SCTP's flexibility truly shines when all the different types of data traffic are considered.

V. BARRIERS TO SCTP USE

There are some barriers in using SCTP for everyday applications but they are progressively getting smaller. SCTP is still relatively young for a transport protocol but it can be used on most operating systems. Currently SCTP is supported in Linux through the `lksctp-tools` package and corresponding Linux Kernel module [7] but Windows and Mac operating system support is not native. Support for Windows comes from a third party driver as well as for Mac OS. As an alternative there is a Google user-land implementation that runs on all platforms [8]. This provides all the benefits of SCTP without relying on the underlying operating system for support. Network Address Translation can also be problematic since some consumer grade equipment don't handle SCTP sessions that are multi-homed. Encapsulation methods such as SCTP over DTLS over UDP are a way around this and are being touted by projects such as WebRTC [?] which provide real-time audio and video through web browsers. Although wide protocol adoption is a slow moving process SCTP is finally beginning to branch out from its telecom origins into mainstream usage and Amateur radio is not exempt from benefitting from its features.

VI. CONCLUSION

SCTP is an attractive protocol for Amateur Radio use based on its unique features. The fact that they are provided at the protocol level frees developers and users from creating those same features at the application layer and allows faster development of software that leverages cutting edge Software Defined Radio. With more and more radios becoming network aware, SCTP should be considered as a viable and robust candidate for networked data exchange. The applications listed here touch a very small subset of all the applications that SCTP could extend to and the hope is that further interest in the protocol spurs innovation of even more use cases.

REFERENCES

- [1] 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, RFC2960," Online, October 2000.
- [2] R. Stewart, "Stream Control Transmission Protocol, RFC4960," Online, September 2007.
- [3] T. Dreibholz, E. Rathgeb, I. Rungeler, R. Seggelmann, M. Tuxen, and R. Stewart, "Stream Control Transmission Protocol: Past, current, and future standardization activities," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 82–88, April 2011.
- [4] P. Natarajan, F. Baker, P. Amer, and J. Leighton, "SCTP: What, Why, and How," *Internet Computing, IEEE*, vol. 13, no. 5, pp. 81–85, sept.-oct. 2009.
- [5] F. Systems, "Flex Signature Series," Online at <http://www.flexradio.com/amateur-products/flex-6000-signature-series/>, 2014.
- [6] K. John B. Stephensen, "A Software Defined Radio for Mesh Networks," in *32nd ARRL and TAPR Digital Communications Conference*, September 2013.
- [7] L. M. H. Yarroll and K. Knutson, "Linux Kernel SCTP : The Third Transport," Online at old.lwn.net/2001/features/OLS/pdf/pdf/sctp.pdf, 2001.
- [8] B. Penoff, A. Wagner, M. Tuxen, and I. Rungeler, "Portable and performant userspace SCTP stack," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, August 2012, pp. 1–9.