

Space Link Extension (SLE) Emulation for High-Throughput Network Communication

Robert Murawski ^{*}, Nicole Tchorowski [†], and Bert Golden [‡]
NASA Glenn Research Center
Cleveland, OH 44105

As the data rate requirements for space communications increases, significant stress is placed not only on the wireless satellite communication links, but also on the ground networks which forward data from end-users to remote ground stations. These wide area network (WAN) connections add delay and jitter to the end-to-end satellite communication link, effects which can have significant impacts on the wireless communication link. It is imperative that any ground communication protocol can react to these effects such that the ground network does not become a bottleneck in the communication path to the satellite. In this paper, we present our SCENIC Emulation Lab testbed which was developed to test the CCSDS SLE protocol implementations proposed for use on future NASA communication networks. Our results show that in the presence of realistic levels of network delay, high-throughput SLE communication links can experience significant data rate throttling. Based on our observations, we present some insight into why this data throttling happens, and trace the probable issue back to non-optimal blocking communication which is supported by the CCSDS SLE API recommended practices. These issues were presented as well to the SLE implementation developers which, based on our reports, developed a new release for SLE which we show fixes the SLE blocking issue and greatly improves the protocol throughput. In this paper, we also discuss future developments for our end-to-end emulation lab and how these improvements can be used to develop and test future space communication technologies.

I. Introduction

Increasing data rate requirements for space communications can strain not only wireless communication channels, but also the ground communication networks which forward user data to satellite ground stations (GS). The National Aerospace and Aeronautics Administration (NASA) has recently undertaken a substantial update to the NASA Space Network (SN) which will support high data rate satellite communication, as well as improved support for reliable and high data rate ground network communications. The proposed ground network communication protocol for the SN Ground Segment Sustainment (SGSS) program is the Space Link Extension (SLE) protocol, a set of recommended standards developed by the Consultative Committee for Space Data Systems (CCSDS).^{1,2,3,4,5,6}

In the past, the CCSDS SLE protocol has primarily been used for reliable, low-data rate transfer of data link layer frames remote users to their supporting GS. For future NASA integrated network development, this protocol has been proposed as a standardized interface for all future high and low data rate traffic. Sending high data rate traffic over proposed SLE implementations, however, can be problematic as was shown for various SLE software implementations.⁷ In this paper, the authors studied how the SLE implementation developed by VEGA could handle the transfer of high-data rate SLE traffic over a variety of transport layer protocols. They found that characteristics of the wide area network (WAN) connection, such as non-negligible delay, jitter, and packet errors could lead to significant disruptions in the transfer of SLE data.

In this paper, we focus on the SLE implementation that was developed by Ingenicomm for use in the NASA SGSS system to support the future high data rate Ka-Band Single Access (SA) satellite links. These

^{*}Corresponding Author, contract employee working for Vantage Partners, LLC, Member AIAA

[†]NASA Glenn Research Center Summer Intern, currently at The Ohio State University, OH, Student Member AIAA

[‡]NASA Civil Servant

future high data rate links could potentially support data throughputs of up to 1.2 *Gbps*. To test the performance of this SLE implementation, a ground network emulation testbed was developed at the NASA Glenn Research Center (GRC) Space Communication and Navigation (SCaN) Compatibility Environment for Networks and Integrated Communications (SCENIC) Emulation Lab. In this environment we are able to emulate realistic ground network characteristics such as delay, jitter and packet loss, across a high-data rate SLE connection and observe how the protocol implementation reacts to these network impairments. In this lab we can also emulate wireless communication channels as was demonstrated in previous work.⁸ In future work, we plan on integrating the SLE ground network emulation testbed with a wireless channel emulation for detailed testing of a full end-to-end wireless communication path.

This paper is organized as follows. In Section II we will give a brief overview of the CCSDS SLE protocol and how certain protocol features can affect high data rate communications. In Section III we discuss the SCENIC Emulation Lab and the network setup used to test the SLE implementation. Finally, in Section V we will give insight into our observations and details on our planned future work.

II. Space Link Extension (SLE)

The Consultative Committee for Space Data Systems (CCSDS) Space Link Extension (SLE) protocol was developed to support the forwarding of layer 2 frames from remote users to the ground stations that provide wireless communication services to destination spacecraft. In general, SLE supports CCSDS formatted layer 2 frames, such as the Advanced Orbital Systems (AOS), Telecommunications (TC), and Telecommand (TM) data, as well as the forwarding of binary data streams for unsupported frame types. The layer 2 framing protocols used are out of the scope of this document and should have negligible effect on the performance of the network communications between the SLE User and Provider systems. For all results presented in Section IV, binary data streams were used which pass the time stamping information required to measure the end-to-end SLE performance.

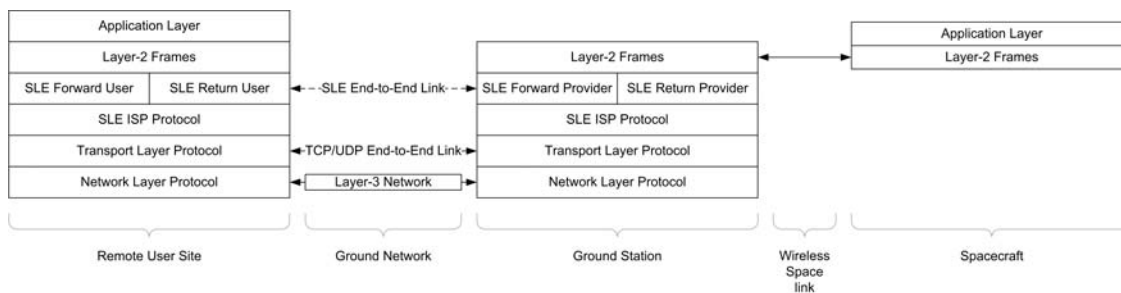


Figure 1. Space Link Extension Network Diagram

Figure 1 shows the general network protocol stack diagram used in an end-to-end space communication link which utilizes the CCSDS SLE protocol for communication over the IP-based layer-3 ground network. As you can see, for forward traffic (i.e. ground user to spacecraft) layer 2 frames are generated at the remote user site, typically referred to as the User Mission Operations Center (User-MOC), and are sent to the SLE Forward User protocol for tunneling over the layer-3 ground network. The SLE protocol encodes these layer 2 frames using the ASN.1 encoding as specified in the CCSDS standards^{4,5} and forwards the encoded SLE Protocol Data Units (PDUs) to the specified transport layer protocol. The CCSDS protocol does not explicitly state what transport layer protocol should be used for the ground network communications, however, the NASA Space Network (SN) Ground Segment Sustainment (SGSS) program has specified that the Transport Control Protocol (TCP) should be used to ensure reliable and in-order delivery of the SLE PDUs. For SLE return traffic,^{1,2} the layer-2 frames received by the satellite modem are passed to the SLE Return Provider which again encapsulates the frames into an SLE PDU and sends these PDUs to the specified transport layer protocol.

In addition to the CCSDS Recommended Specifications for SLE, CCSDS has released application program interface (API) specifications for both the forward⁹ and return^{10,11} services. The API specifications give greater detail as to how an SLE implementation should handle the various commands in the SLE specifications. For data transfer between SLE user and provider servers, the *CLTU_TRANSFER_DATA*, *RAF_TRANSFER_DATA* and *RCF_TRANSFER_DATA* invocations are defined for the SLE FCLTU, RAF,

and RCF specifications, respectively. How this data call is handled can have a significant effect on the performance of the SLE implementation as it is this call that forwards the ASN.1 encoded SLE PDU to the underlying transport layer protocol. In Section 2.2.8.3 of the SLE F-CLTU API specifications,⁹ it states that this invocation can be called in one of three ways.

1. an implementation can always pass invocations for which a check has failed to the application
2. an implementation can prevent queuing of invocations by withholding an invocation until the previous invocation has been confirmed by the application. In that case, it can always generate the appropriate return when needed; or
3. an implementation can decide to pass invocations to the application on a case by case basis

The CLTU-TRANSFER-DATA invocation sends the formed SLE protocol data unit (PDU) to the underlying transport layer for transmission to the SLE Provider. In the first case, whenever a packet transmission fails, the SLE ISP1 service will pass a CLTU-THROW-EVENT call to the SLE FCLTU application, which will result in packet retransmissions. Option #2 above, on the other hand, prevents queuing of PDUs by preventing any CLTU-TRANSFER-DATA invocations until the packet transmission is confirmed. This is commonly referred to as a blocking protocol, as it blocks all future data transmissions until the current data transmission is acknowledged by the receiver.

Based on the network round-trip-time (RTT) delay and the size of the SLE PDUs the SLE protocol is sending to the underlying transport layer, we can calculate the maximum theoretical throughput of a blocking SLE implementation. For the SLE Return all Frames (RAF) and Forward Command Layer Transfer Unit (F-CLTU) protocols, the maximum SLE PDU size that can be handled by the CTLU_TRANSFER_DATA command or RAF_TRANSFER_DATA command is 64kB. In a blocking connection, the SLE protocol implementation will wait for a successful acknowledgment of a packet transfer before executing additional transfer data commands. Therefore, the maximum theoretical data rate is given by:

$$R = S/T, \tag{1}$$

where R is the maximum theoretical data rate, S is the size of the SLE PDU transferred in the blocking transfer data call, and T is the time required to receive and acknowledgment for a successful packet transfer. In our network, the delay T includes the emulated round trip time as well as any additional processing and transmission delays at both servers and at the intermediate network switch. A diagram detailing the timing delays associated with a blocking transmission is shown in Figure 2.

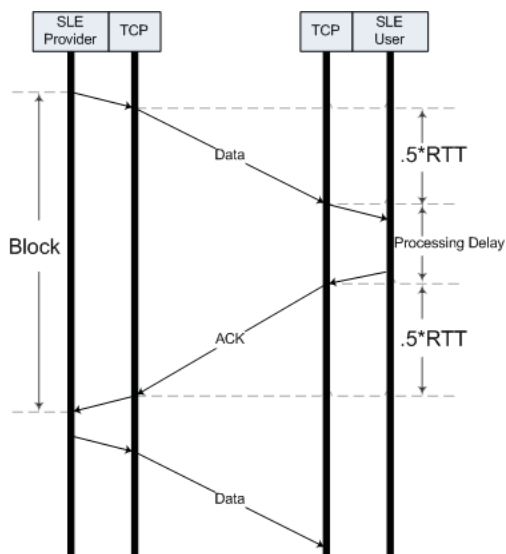


Figure 2. SLE Blocking Network Delay

This is significant as networks which have a large bandwidth-delay product, i.e. networks with long delays and high throughput requirements, will spend a significant amount of time in this blocking state, which as a result, will significantly reduce the achievable network throughput.

III. SCENIC Emulation Lab

The NASA Glenn Research Center (GRC) SCENIC Emulation Lab is currently under development and is being designed to enable real-time emulation of end-to-end communications in the future NASA Integrated Network Architecture (INA). The INA is being developed as part of the NASA Space Communication and Navigation (SCaN) program to integrate the three NASA space communication networks: Near Earth Network (NEN), Space Network (SN), and the Deep Space Network (DSN). As new technologies are being developed to support the future integrated network, they must be thoroughly tested in a safe emulation environment to ensure that they are compatible with the current network and can operate under non-ideal conditions that may be experienced when in operational use.

In this paper, we demonstrate the first ground network emulation configuration. This configuration was developed to test new ground network communication protocols, such as the Space Link Extension (SLE), in networks with real delay, jitter and packet loss. In SCENIC we can emulate these expected network characteristics a wide area network (WAN) network link. Because these network effects are emulated in real-time on real networking hardware, we can test the actual SLE protocol implementations, rather than simulating the SLE characteristics mathematically or through discrete event based simulation tools, such as OPNET or QualNet. Thus, our emulation environment can test for issues in the implementation, or in the CCSDS SLE standards themselves.

For the emulation tests discussed in this paper, we have installed the Ingenicom Configurable Ground System (CGS) software on two IBM x3550 servers, each running Red Hat Enterprise Linux (RHEL) 6.4. These servers are connected to a local fiber test network through a Brocade MLXe4 10-gigabit fiber switch as shown in Figure 3.

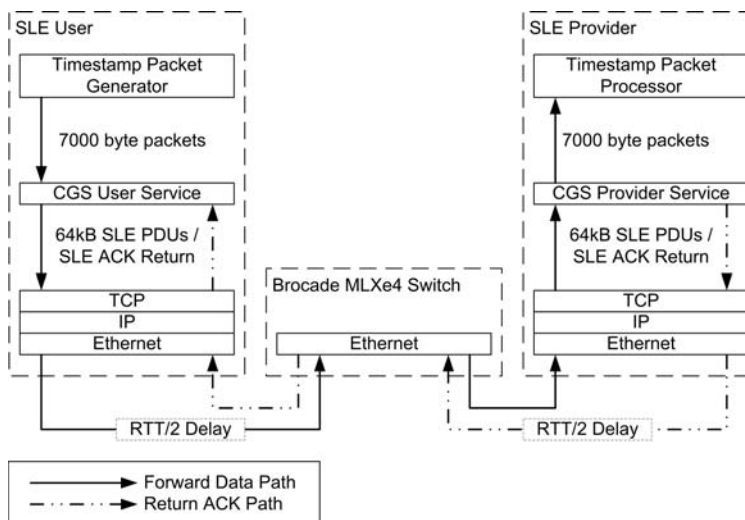


Figure 3. Detailed Network Interconnect

Within the CGS software, we have configured a timestamp generator which generates user data at a predefined bit rate with time stamp information to support delay measurements. This user data is forwarded to the SLE processor which encapsulates the user data into 64 kilobyte SLE Protocol Data Units (PDUs). These PDUs are then forwarded to an active Transport Control Protocol (TCP) socket connection for transmission over the emulated WAN connection. The end-to-end connection between the SLE User and Provider services are all commercial off the shelf (COTS) networking equipment. When transmitting the Ethernet frames from SLE server to the Brocade switch, we utilize the Linux Network Emulation (netem) function to inject $RTT/2$ seconds of delay into all transmitted frames, where RTT is the desired round-trip-time delay of the end-to-end SLE connection. The $RTT/2$ seconds of delay are also configured for the return path, resulting in a full RTT second delay for the SLE connection.

IV. Results

In this section we present some of the performance results of our throughput testing of the CCSDS SLE implementation. First, we present the baseline network performance test results to demonstrate that the underlying network can support the high-throughput data that will be required of the SLE tunneled connection.

A. Baseline Network Performance

For our baseline network performance results we tested a TCP connection through the Ingenicomm Configurable Ground System (CGS). In this case, a simple TCP socket was opened up and a 1.2Gbps packet stream was generated consisting of pseudo-random data and timestamping information used to estimate the one-way packet delay. The 200ms delay was selected based on the maximum expected round trip time for the NASA Integrated Services Network (NISN). According to a study by the NASA Communications Services (CSO) document, the maximum RTT for a NISN connection across the Continental United States is approximately 120ms . To account for higher delays that can be expected for connections to foreign ground stations, we have increased the emulated RTT to a maximum of 200ms .

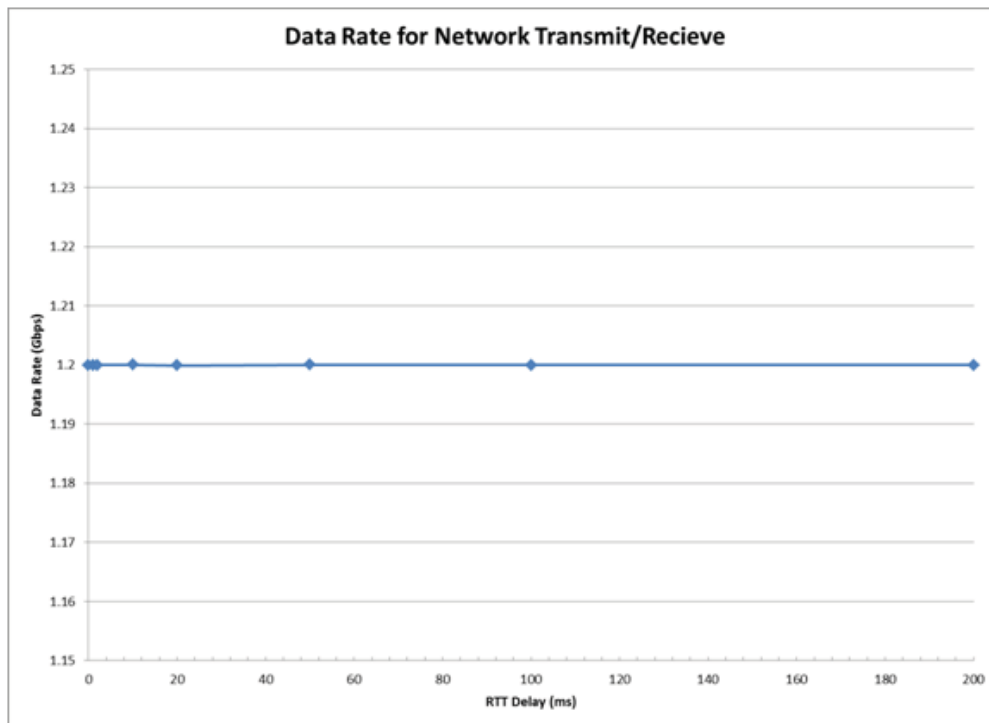


Figure 4. Baseline CGS Network Throughput vs. RTT Delay

As you can see in Figure 4, as the emulated round-trip time (RTT) delay of the underlying network connection is increased from 0 to 200ms the observed average throughput remains constant at 1.2Gbps . These values were collected over a 5-minute simulation run, and the data rate was recorded every second.

B. SLE Network Performance

The following results show the performance of the CCSDS SLE protocol as implemented in the CGS software over the emulated WAN network connection.

In Figures 5 and 6 the observed average delay of the SLE forward and return data traffic is shown. Also plotted is the maximum achievable throughput of the blocking network connection, based on the SLE PDU size of 64 kilobytes and the emulated network delay. The maximum achievable throughput is based on the assumption that the SLE is utilizing a blocking transmission protocol and the throughput is calculated as shown in Section II. For each of these tests, the underlying TCP connection was the same as was used for the

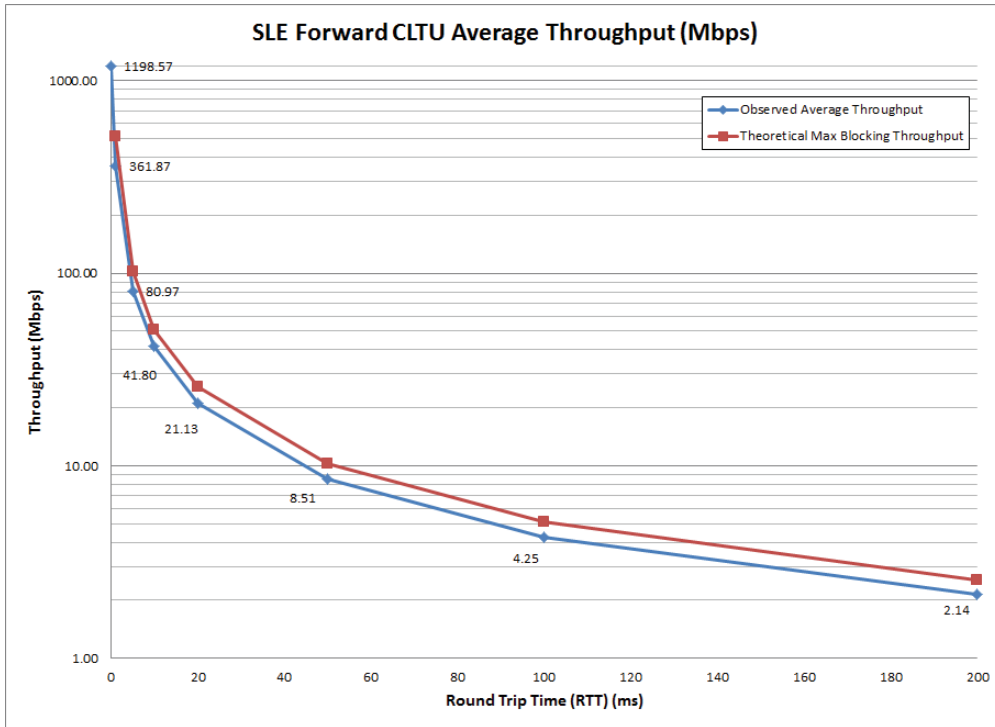


Figure 5. CCSDS SLE F-CLTU Throughput with Emulated Network Delay

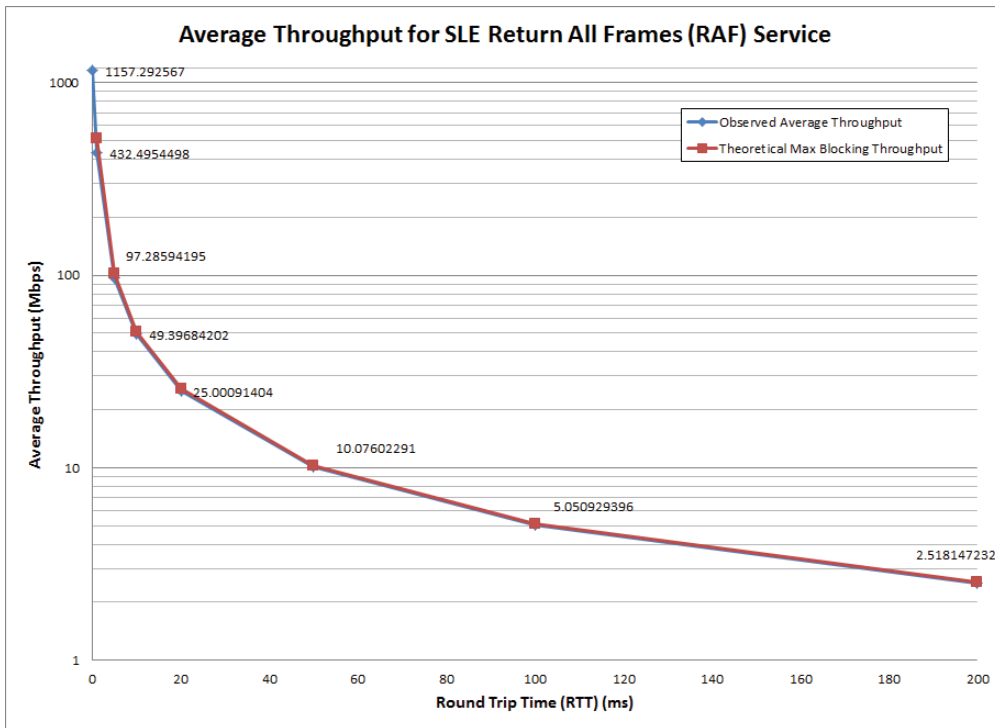


Figure 6. CCSDS SLE RAF Throughput with Emulated Network Delay

baseline network results shown in Figure 4. As you can see in these results, as the emulated RTT network delay is increase, the throughput performance of the CCSDS SLE implementation is significantly reduced. As the delay is increased to a RTT of 200 *ms*, less than 1% of the offered throughput can be transferred via the SLE protocol.

This reduced throughput, based on our observations and discussions with the protocol vendor is due to the blocking nature of this SLE implementation. As you can see from the results, the SLE observed throughput and the SLE expected blocking throughput are closely correlated. While utilizing blocking SLE calls is supported by the CCSDS SLE API specifications, as shown in Section II, this is not the optimal implementation for the SLE transfer frame operations.

C. Updated SLE Performance Results

In the previous section, performance results for SLE were collected and, as you could see from the results in Figures 5 and 6 the emulated network delay significantly reduces the achievable throughput for the tested version of SLE. Based on conversations with the SLE developers, this was believed to be caused by an interpretation of the CCSDS SLE standards which suggests that blocking calls must be used for the SLE calls to the underlying transport layer protocol.

Several months after reporting our findings to the SLE developers and NASA management, a new release of the Ingenicomm SLE software was released. The same throughput tests were performed on the updated, version 1.11 release of the Ingenicomm SLE software. The observed throughput can be seen in Figure 7.

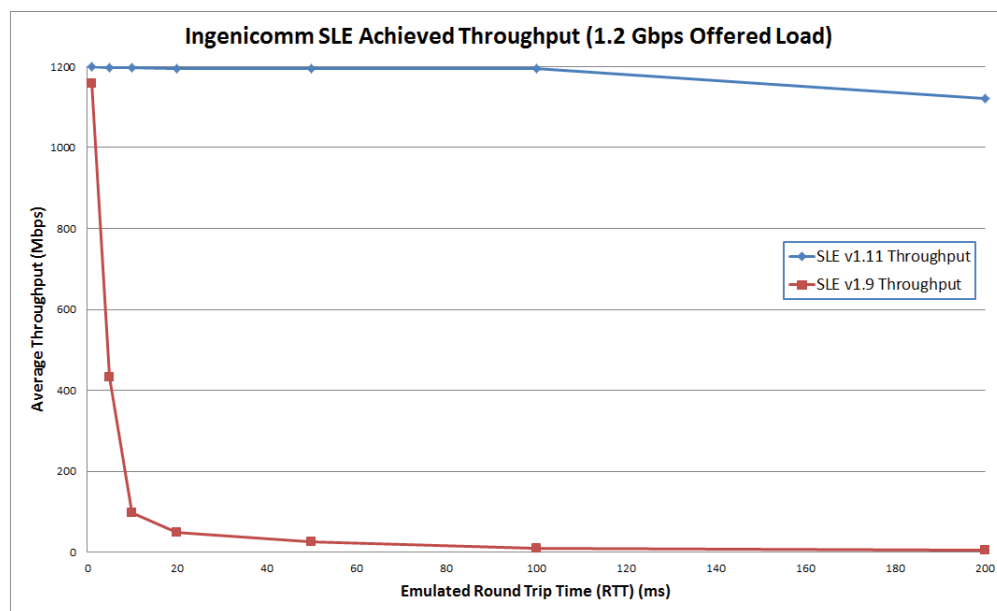


Figure 7. SLE Improved Throughput

As you can see in Figure 7 the throughput has been greatly improved based on improvements to the SLE code. When the emulated RTT delay reaches our maximum tested value of 200 *ms* the delay is approximately 1.1 *Gbps*.

V. Conclusion

In this paper, we discussed performance issues that were observed for the space link extension implementation that was developed for the NASA SGSS program to support future high data rate missions. Based on our observations, these issues were likely due to non-optimal blocking messages being used between the SLE transport mapping layer (TML) and the underlying transport control protocol (TCP). While supported by the CCSDS SLE specifications, blocking calls are not ideal for network connections with a high delay-bandwidth product, and as such, for high delay links, the maximum achievable throughput is greatly reduced. As was seen in Section IV, the loss in throughput was very significant. For high network

latency scenarios where the network round-trip-time (RTT) was approximately 200 milliseconds, less than 1% of the offered throughput could be sent over the SLE link. After discussing this issue at length with the SLE implementation developers, Ingenicomm was able to resolve this issue in their latest release of the Configurable Ground System (CGS) SLE implementation. Re-running the performance tests with the latest SLE implementation showed significant performance improvements as shown in Figure 7.

A. Future Work

Ground network communications, while a vital part of space communications, is only a small part of the end-to-end communication path for a NASA supported mission. We are currently working to expand the capabilities of our emulation lab such that we can support a full end-to-end communication environment. In the short term, this involves the development of a software defined radio (SDR) platform that can process the data tunneled through the SLE protocols described in this paper, and transmit that data over an emulated wireless channel. When this capability is realized, we will be able to test communications between an ground user and spacecraft at a much higher fidelity. This will allow us to test not only intermediate tunneling protocols such as SLE, but investigate issues with application layer protocols that are proposed for future space mission support.

References

- ¹CCSDS Recommended Standard, “Space Link Extension - Return All Frames Service Specification,” *CCSDS 911.1-B-3*, January 2010.
- ²CCSDS Recommended Standard, “Space Link Extension - Return Channel Frames Service Specification,” *CCSDS 911.2-B-2*, January 2010.
- ³CCSDS Recommended Standard, “Space Link Extension - Return Operational Channel Frames Service Specification,” *CCSDS 911.5-B-2*, January 2010.
- ⁴CCSDS Recommended Standard, “Space Link Extension - Forward CLTU Service Specification,” *CCSDS 912.1-B-3*, July 2010.
- ⁵CCSDS Experimental Specification, “Space Link Extension - Enhance Forward CLTU Service,” *CCSDS 912.11-O-1*, July 2012.
- ⁶CCSDS Recommended Standard, “Space Link Extension - Internet Protocol for Transfer Services,” *CCSDS 913.1-B-1*, September 2008.
- ⁷Götzelmann, M., Karch, M., Lucia, D., Abello, R., and Dreihahn, H., “High Rate Telemetry Transfer - CCSDS SLE And Beyond,” *AIAA SpaceOps 2012*, 2012.
- ⁸Murawski, R., Bhasin, K., Bittner, D., Sweet, A., Coulter, R., and Schwab, D., “Hardware and software integration to support real-time space-link emulation,” *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2012 *IEEE 17th International Workshop on*, Sept 2012, pp. 271–275.
- ⁹CCSDS Recommended Practice, “Space Link Extension - Application Program Interface for the Forward CLTU Service,” *CCSDS 916.1-M-1*, October 2008.
- ¹⁰CCSDS Recommended Practice, “Space Link Extension - Application Program Interface for Return All Frames Service,” *CCSDS 915.1-M-1*, October 2008.
- ¹¹CCSDS Recommended Practice, “Space Link Extension - Application Program Interface for Return Channel Frames Service,” *CCSDS 915.2-M-1*, October 2008.