

# A Full-Protocol-Stack Testbed for Space Network Protocol Emulation

Xiaoqin Ni, Kanglian Zhao<sup>(✉)</sup>, and Wenfeng Li<sup>(✉)</sup>

School of Electronic Science and Engineering, Nanjing University,  
22 Hankou Rd., Nanjing 210093, Jiangsu, People's Republic of China  
nixiaoqin44@163.com, zhaokanglian@nju.edu.cn, leewf\_cn@hotmail.com

**Abstract.** With the rapid development of space networks, new space communications protocols are emerging, for which emulation is an essential step during design and test. In this paper, we propose a lab-based testbed, in which software and hardware tools are utilized together to emulate full network protocol stack. A software protocol gateway is implemented to preform protocol conversion like IP over CCSDS in Data Link Layer. A specified hardware, Cortex CRT-Q is adopted for accurate emulation of space links, which connects upper layers with Physical Layer. Thus, our testbed benefit from both the fidelity provided by hardware and flexibility brought by software.

**Keywords:** Lab-based testbed · Full network protocol stack  
IP over CCSDS · Space link

## 1 Introduction

A space communications protocol is a communications protocol designed to be used over a space link, or in a network that contains one or multiple space links. According to the CCSDS blue book, the space communications protocols are defined for the following five layers of the ISO model [1]:

- (a) Physical Layer;
- (b) Data Link Layer;
- (c) Network Layer;
- (d) Transport Layer;
- (e) Application Layer.

During design, implementation and utilization of space communications protocol, emulation is an essential step. Various testbeds for emulating space network have been proposed in different works. The key point of the testbed is how to

---

This work is supported by the National Natural Science Foundation of China (No. 61401194), the Fundamental Research Funds for the Central Universities (021014380064) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

reproduce the space links in laboratory. A space link is a communications link between a spacecraft and its associated ground system or between two spacecraft. [1] The space communication link displays special characteristics different from those of terrestrial ones: larger link delay, higher bit error rates, bursts of errors, packet disordering, etc. To emulate a space-ground link, field testing equipment can be prohibitively expensive and deployment scheme is inflexible. [2] Another method is using software like netem to control the delay, the BER and the rate. [3] The problem of this kind of testbed is that data flows through Ethernet network, which is different from specialized transceivers running dedicated Data Link Layer protocols, like AOS [7].

Experiments with full network protocol stack is preferred for system level performance emulation. To emulate a space communications protocol, a widely used method is utilizing network simulator software like OPNET [4], which is confined to state machine in a single PC lack of fidelity. Another solution is utilizing protocol gateway based on FPGA [5] to perform IP over CCSDS, in which a certain threshold and difficulties for developing exits, with protocol configuration lack of flexibility.

In this paper, a full-protocol-stack testbed is proposed for network protocol emulation. Hardware components like Cotrex Command/Ranging/Telemetry-Quantum (hereinafter referred to as Cortex CRT-Q) [6] is applied to provide accurate space-ground link and software protocol gateway (hereinafter referred to as SPG) provides flexible configuration and emulation of full protocol stack. The remainder of this paper is organized as follows. Section 2 describes the design of our testbed including facilities and equipment. Verification experiments and relevant results are presented in Sect. 3. The conclusions and future works are drawn in Sect. 4.

## 2 Design for Testbed

### 2.1 Overview of the Architecture

The hardware architecture of the testbed is showed in Fig. 1, which consists of one Cortex CRT-Q, two SPGs and several PCs. Cortex CRT-Q provides space links, which incorporates powerful built-in simulation capabilities for functional and performance test purposes including: receiving data as simulation resource and sending out data after demodulation over the ETHERNET LAN, IF modulation (PM, FM, BPSK, QPSK, OQPSK or AQPSK) and demodulation, noise generation, etc. In this paper, Cortex CRT-Q works in the local loop-back mode to emulate a space-ground link in a lab environment. SGP functions as a border gateway, which performs protocol conversion like IP over CCSDS simultaneously. PCs act as communication nodes in space network such as ground stations, spacecraft, users, etc.

### 2.2 The Design of the Software Protocol Gateway

The implementation of the SPG is based on the concept of protocol layering principle. Figure 1 also shows the protocol architecture. Notice that, SPG works

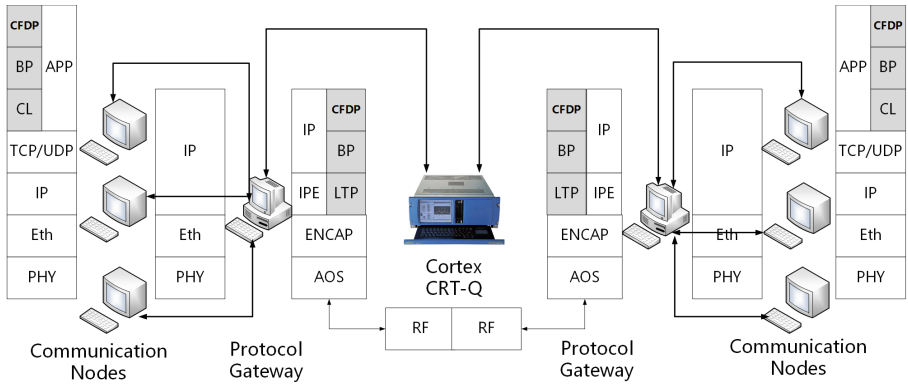


Fig. 1. Overview of the architecture

upon two protocol stacks as a border gateway, solving the protocol conversion problem. Reserving interface like socket port in different layers is convenient for debugging and monitoring with the help of wireshark or tcpdump. SPG operates in a Linux environment and program in each layer will be explained later.

In Network Layer, we write program based on libpcap and libnet, to accomplish capturing and sending IP packet. The Application Layer data extracted with IP header, needs fragmentation and reassembly sometimes when the payload length is bigger than MTU.

In Data Link Layer, Advanced Orbiting Systems (AOS) [7] protocol has been designed to meet the requirements of space missions for efficient transfer of space application data of various types and characteristics over space-to-ground, ground-to-space, or space-to-space communications links. Thus it is selected as Data Link Layer protocol in our testbed because of its maturity and universality, certainly could be replaced by others. The IP OVER CCSDS SPACE LINKS blue book [8] describes the recommended method for transferring IP PDUs over CCSDS SDLPs including AOS. IP PDUs are transferred by encapsulating them, one-for-one, within CCSDS Encapsulation Packets. The Encapsulation Packets [9] are transferred directly within one or more CCSDS SDLP Transfer Frames. This method uses the CCSDS Internet Protocol Extension (IPE) convention in conjunction with the CCSDS Encapsulation Service over CCSDS AOS. We program according to relevant books and RFCs to perform protocol conversion.

In Physical Layer, with different configuration parameters set, Cortex CRT-Q could provide different physical link. It reads data as simulated data from port 3021 or 3022. After local real-time modulation and demodulation, port 3070 is used to send telemetry data out, which is triggered by a request command. Working mode provided by Cortex CRT-Q is oriented to data stream, however, the data form involved in protocol emulation is mainly intermittent data packet. Generally, if the transmission rate of the Application Layer does not match with the bit rate of the Physical Layer, phenomenon occurs as follows: if transmission rate is higher, problems of congestion, delay and packet loss would be serious;

if bit rate is higher, in order to avoid modulation blank, given amount of data is required to wait, which resulted in unnecessary delay. The difference between data stream and data packets makes requests for link adaption. Therefore, socket programming of non-blocking mode is applied. When there is no data to send, idle data is sent to maintain channel synchronization. In brief, we program to put and get AOS frames in CRT frame format and accomplish link adaption.

### 2.3 Data Flow on the Testbed

Make an introduction to the data flow on the testbed. Figure 2 shows only one-direction communication process, the other direction is similar. Two subnets are representing terrestrial and space network respectively, for example, 192.168.0.0/24 (hereinafter referred to as subnet 1) and 192.168.10.0/24 (hereinafter referred to as subnet 2). SPG connects the PCs of each subnet with the Cortex CRT-Q.

PCs in subnet 1 sets the routing table, enabling all data whose destination is subnet2 are converged in SPG 1. SPG 1 receives the data from network card that would be sent to pcap program to filter out IP packets, which encapsulated into AOS frame later and sent to Cortex CRT-Q in simulated data format. Cortex CRT-Q modulation frequency is set as 70M, with different modulation parameters configured. SPG 2 keeps sending request commands to the Cortex CRT-Q. Once telemetry data is received, which would go through CRT-unpack and AOS-unpack program. Original IP packets would be sent to the PCs in subnet 2 through libnet program, after adding Ethernet frame header.

Based on Cortex CRT-Q and SPGs, data flow contains space-ground links, upon which full network protocol stack are emulated.

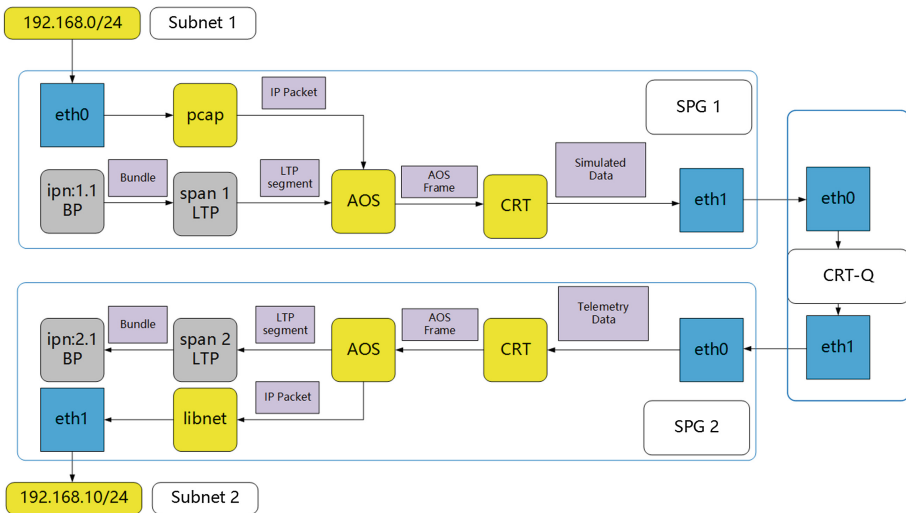


Fig. 2. Data flow

### 3 Test and Discussion

#### 3.1 Fidelity of the Testbed

The fidelity of testbed mainly depends on the accuracy of bit rate and bit error rate (hereinafter referred to as BER) that Cortex CRT-Q supplies. The following two experiments is performed to verify these indicators. Because in Cortex CRT-Q, frame size is fixed after configuration. In this paper, frame size is 1024 B. Make analysis according to AOS frame format (Table 1) without noise. We could reason out:

$$Bandwidth = \frac{Payloadsize}{Framesize} BitRate \tag{1}$$

**Table 1.** AOS frame format

AOS frame 1024 B						
Sync word	AOS	AOS data field				AOS
1ACFFC1D	Head	MPDU	Encap	IPE	Payload	CRC
4 B	6 B	2 B	4 B	1 B	1005 B	2 B

If payload is smaller than 1005B, it will be filled with idle data until the total length is 1005 B. Otherwise, it will be split into several frames with length of 1005 B. At this time, the formula is revised to:

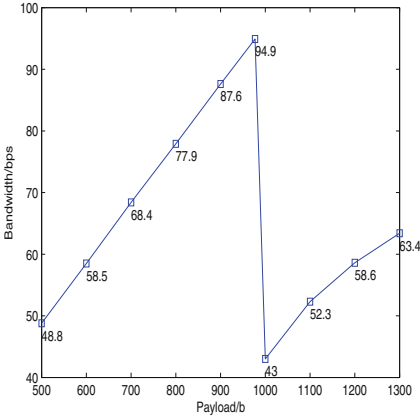
$$Bandwidth = \frac{Payloadsize}{Framesize * N} BitRate \tag{2}$$

$$N = \lceil \frac{Payloadsize}{1005B} \rceil \tag{3}$$

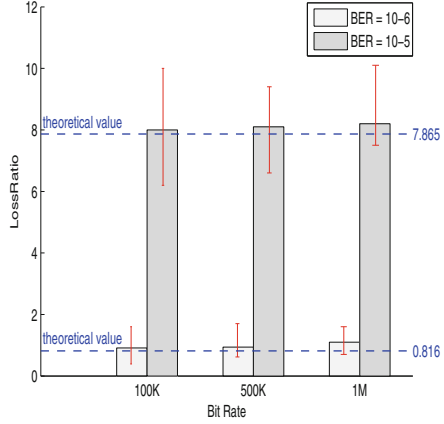
Therefore, when the payload length is set as 1005 B, actually 977 B subtract IP and UDP header, the optimal bandwidth utilization equals 95.41%. In the two subnets, two PCs are running iperf [10] server and client respectively. The bandwidth in 100 Kbps link with different payload size is tested as Fig. 3, and the best bandwidth with different bit rates is tested as Table 2. Since the measured

**Table 2.** Best bandwidth with different bit rate

Bit rate	Bandwidth	Channel utilization
100k	94.9k	94.90%
500k	474.0k	94.80%
1M	942.1k	94.20%



**Fig. 3.** Bandwidth with different Payloadsize (100 kbps)



**Fig. 4.** Loss ratio (different Bit rate)

results are very close to the theoretical value 95.41%, it can be concluded that the indicator of bit rate is valid and accurate.

Now, we analyze validity of configuration for BER. Essentially, software emulator like tc/netem, controls BER in Physical Layer by counting and dropping specific amount of packets in upper Layer (probably Network Layer). On the contrary, Cortex CRT-Q controls BER by setting up noise with different  $C/N_0$  in Physical Layer, which leads to packet loss in Data Link Layer because of failing to pass checksum. The latter is more logical and credible. According to the formula:

$$P_{eBPSK} = \frac{1}{2} \operatorname{erfc}(\sqrt{E_b/N_0}) \tag{4}$$

$$E_b/N_0 = C/N_0 - 10 \lg R \tag{5}$$

After  $C/N_0$  is set to 59.5 dB ( $R = 100$  Kbps, BSPK) in the noise modular of the Cortex CRT-Q,  $E_b/N_0$  is showed around 9.5 dB. The BER now is  $10^{-5}$ , according to the formula (4). As for loss ratio,

$$\text{LossRatio} = 1 - (1 - BER)^{8 \text{packetsize}} \tag{6}$$

Packet size is 1024 B because each frame in Cortex CRT-Q is 1024 bytes, after conversion, the packet loss rate is 7.865%. After backing up data sent to and receive from Physical Layer (CRT), we can calculate BER by making comparisons. Analogically, after backing up data sent to and receive from Data Link Layer (AOS), we can calculate Loss Ratio. File with the size of 10 MB was sent in the configuration of  $BER = 10^{-5}$  and  $BER = 10^{-6}$  with different bit rates, the test results are as Fig. 4. The measured results are very close to the theoretical value, it can be concluded that the indicator of BER valid and accurate.

### 3.2 Flexibility of the Testbed

Firstly, two subnets ping each other to make analysis of delay. When bit rate is 100 Kbps without noise, average RTT is 460 ms. Considering that each ICMP packet is packed into a 1024 B CRT frame, channel delay is 82 ms and one-way program processing delay is about 148 ms.

Based on the design principle of protocol layering, we can flexibly change upper protocols to test other protocol stack for example DTN [11] (gray parts in Fig. 1). To perform testing, the Interplanetary Overlay Network (ION) version 3.5.0 open source software implementation of DTN [12] was used on Linux PCs including SPGs and communication nodes. On the basis of the IP over CCSDS, according to the relevant blue book and RFC [13–15], with ION software providing CFDP/BP/LTP application, new data flow is as follows (gray parts in Fig. 2): Gateway 1 acting as ipn:1.1, splits CFDP file into BP bundles and cuts converged bundles into LTP Blocks, then according to the link layer MTU(1005 B, optimal payload size), LTP Blocks turn into LTP segment. This is a very intuitive process of a CFDP-BP-LTP-AOS-RF protocol emulation.

We set up another testbed as a contrast (Fig. 5). A PC utilizes tc/netem acting as a link. At first, delay is configured as 222 ms and rate is 100 kbps. So that RTT of Ping is 460 ms, same as our testbed. But bandwidth measured by iperf (977 B payload size) is 96.1 Kbps, higher than 94.9 Kbps. After revising rate to make sure the result of iperf is the same as 94.9 Kbps, ION software uses different protocols, such as CFDP/BP/LTP protocol, sends the same file (200 kB), delivery time in our testbed compared with software emulator are as Fig. 6.

It is displayed that delivery time of our testbed is still a little longer than software testbed. The reason is that in software testbed, even rate and delay are the same, protocol in Data Link Layer and Physical Layer is different. Because in our testbed, LTP is directly running upon AOS (after simple Encapsulation [9]) and frame size of AOS is fixed (1024 B in our testbed) for both forward

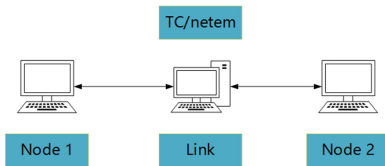


Fig. 5. TC/netem testbed

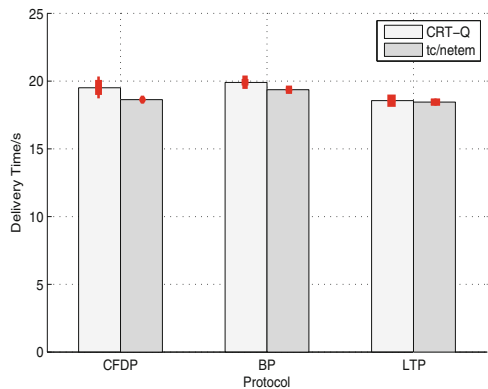


Fig. 6. DTN protocol testing

data packets and backward ACK packets. However, in software testbed, LTP is running upon UDP and frame size of Ethernet is not fixed. It means in software testbed, at least, ACK arrives more quickly because of small size, therefore, the total communication process is shorter. The phenomenon reflects the advantage of our testbed, full-protocol-stack-emulation, in another way.

## 4 Conclusion and Prospect

In this paper, software and hardware tools are utilized together to emulate full network protocol stack. Based on the accurate physical space-ground link provided by hardware components, results of the emulation is more credible. What's more important, the testbed proposed in this paper can provide the researchers and developers the feasibility to emulate or test new protocols in any layer of a reconfigurable full network protocol stack in space networks.

## References

1. Green Book. Overview Of Space Communications Protocols (2007)
2. Murawski, R., Bhasin, K., Bittner, D., et al.: Hardware and software integration to support real-time space-link emulation. In: IEEE, International Workshop on Computer Aided Modeling and Design of Communication Links and Networks. IEEE (2012)
3. Lochin, E., Prennou, T., Dairaine, L.: When should I use network emulation? *Ann. Telecommun.* **67**, 247–255 (2012)
4. OPNET Homepage. [www.opnet.com/products/modeler/home.html](http://www.opnet.com/products/modeler/home.html)
5. Liu, F., Yan, H., Liu, H.: The design and implementation of configurable high-speed IP over AOS gateway. In: Computer Applications and Communications. IEEE (2014)
6. Cortex CRT-Q. <http://www.zodiacaerospace.com/en/cortex-crt-q>
7. Blue Book. AOS Space Data Link Protocol (2006)
8. Red Book. IP over CCSDS space links (2006)
9. Blue Book. ENcapsulation Service (2009)
10. Iperf. <https://sourceforge.net/projects/iperf/?source=directory>
11. Caini, C., Cruickshank, H., Farrell, S., et al.: Delay- and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications. *Proc. IEEE* **99**, 1980–1997 (2011)
12. ION-DTN. <https://sourceforge.net/projects/ion-dtn/?source=directory>
13. Standard, Draft Recommended, and Red Book. CCSDS Bundle Protocol Specification (2010)
14. Red Book. Protocol (LTP) for CCSDS (2011)
15. Sheets BBP. CCSDS File Delivery Protocol (CFDP) (2004)