Improving Testing Capability of Interoperability for Link-11 by building a Gateway for a TCP/IP Network

Ho Jun Lee
Taekyu Kim
Bernard P. Zeigler
Arizona Center for Integrative Modeling and Simulation (ACIMS) at the University of Arizona
Tucson, AZ, 85721
hjlee@ece.arizona.edu, taekyuk@ece.arizona.edu, zeigler@ece.arizona.edu

Dale Fulton
BAE Systems
Dale.fulton.ctr@disa.mil

Doohwan Kim
RTSync Corp.
dhkim@rtsync.com

Keywords:
Link-11, Test of interoperability, TCP/IP Networks, DQPSK, DEVS framework

ABSTRACT: *Link-11 is a tactical digital communication network standard deployed in military platforms now for several decades. The standard employs the digital modulation technique Differential Quadrature Phase Shift Keying (DQPSK) to encode formatted binary messages (M-series messages) onto an audio channel, which are then modulated and transmitted over a radio communications network. Testing of the various systems that use this communications network requires either a fully operational RF network or exchange of the audio-encoded data over a voice channel. An alternative is to send and receive over an IP network, by encoding and decoding the audio from a Data Terminal Set (DTS) that is part of the normal operational equipment. This allows the test community to interconnect distant test sites over IP networks using this robust connectivity for message exchanges. This facilitates interoperability testing at the data and applications level where RF and audio connectivity is not available or is unreliable. We explore this approach to interoperability testing of Link-11 by implementing a gateway between the Link-11 network and a TCP/IP network. Such a gateway has been built and tested at the Joint Interoperability Test Command (JITC) by a team of contractors and government. A review of the engineering and implementation is provided, along with a proposed future expansion of the prototype within a given experimental frame in the Discrete Event System Specification (DEVS) framework.*

1. Introduction

Link-11 is a popular variation of Tactical Digital Information Links (TADIL) series. It transmits binary data over RF network using a digital modulation technique such as Quadrature Phase Shift Keying (DQPSK). This enables participants in the network to communicate through HF Radio equipment in the normal operational environment (ship-to-ship). When we test for interoperability of these Link-11 systems, however, this RF environment is not always necessary nor is it easily available. To facilitate testing, therefore, we devise different methods to connect various players using Link-11 messaging (TADIL A) over analog wireline or digital link and satellite [1].

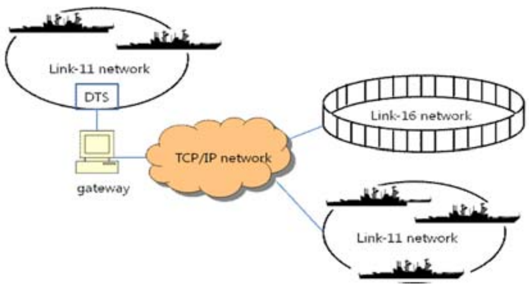


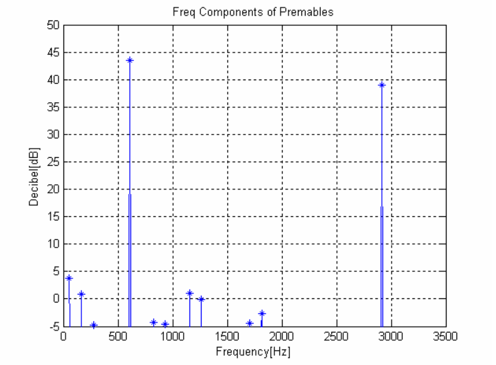
Figure 1. Overview diagram of the gateway concept

Interoperability of Link-11 remains a requirement in a variety of military systems, so it is important for the test community to cost effectively implement this kind of connectivity and networking over widely distributed networks, that today are often connected primarily by TCP/IP networks. Current off-the-shelf hardware and software allows us to connect the Link-11 Data Terminal Sets audio (analog) input and output through analog-to-digital conversion and decoding, which is the approach we have implemented, and which we describe within the subsequent pages of this paper. Figure 1 depicts benefits of the gateway over a TCP/IP network by enlarging network footprint and enhancing connectivity to other networks including Link-11 and Link-16.

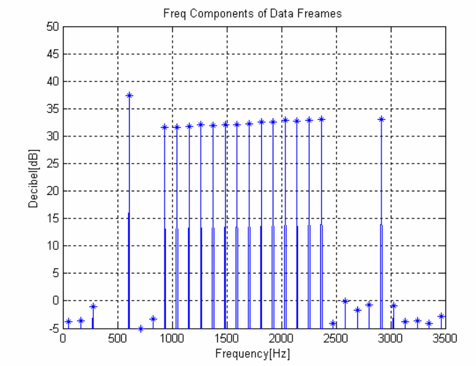
We provide a description of this gateway to connect the Link-11 Data Terminal Sets to TCP/IP networks, and the engineering steps and details to build and validate the equipment and the approach. We begin with section 2, which is an overview of the fundamental characteristics of Link-11 signals, including the encoding and decoding process. In section 3, we describe our alternative method for encoding and decoding as well as the bit data structure definition for handling the bit streams to construct a working gateway. In validation experiments, we confirm the performance of the gateway. Finally, in section 4, we discuss parameter tuning and our solution using Discrete Event Simulation (DEVS) framework.

2. Fundamentals of Link-11 signals

Link-11 employs binary data format (all information encoded into ‘0’ and ‘1’), with a fundamental data structure, or frame consisting of 30 bits. This framing makes it possible to encode and decode binary using 16 discrete audio frequencies, ranging from 605 Hz tone to 2,915 Hz tone with odd harmonics of 55 Hz (Figure 2.).



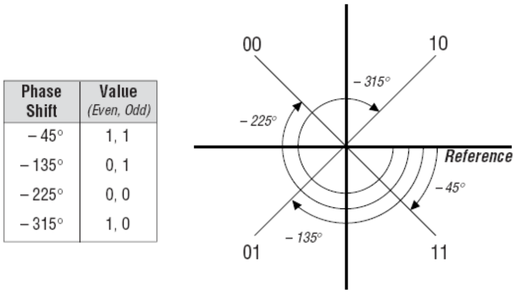
(a) The preamble frame



(b) The phase reference and data frame

Figure 2. Frequency representation of Link-11 signal

This encoding using audio frequencies allow us to transmit and receive digital data within an audio channel that has traditionally been provided over a long-haul HF radio channel. Frames are thus audio encoded using Differential Quadrature Phase Shift Keying - DQPSK.



Phase Shift	Value (Even, Odd)
– 45°	1, 1
– 135°	0, 1
– 225°	0, 0
– 315°	1, 0

Figure 3. Phase shift and bit relationship in Link-11

This encoding uses the phase difference between 2 consecutive frames to represent 2 bits, as shown in Figure 3. Multiple consecutive frames comprise a message, each of which starts with five preamble frames (Figure 2a). The next frame is called the phase reference frame (Figure 2b), which is used as the reference for phase difference to the following frame. Data frames follow the phase reference frame. Phase differences are all subsequently measured using the preceding frame as the reference frame for the next frame.

Each preamble is a two-frequency audio signal. This signal is composed of a 605 Hz tone and a 2,915 Hz tone. The power level of 605 Hz tone is higher than another tone by 6 dB. The phase reference frame is composed of all 16 frequencies. Each data frames’ structure is same as the phase reference frame. So in the data frame, the power level of 605 Hz tone is higher than the other 15 tones by 6 dB, just like the preambles.

Link-11 supports two frame times referred to as the fast data rate and the slow data rate. The fast data rate operates at 75 frames per second whereas the slow data rate operates at 45.45 frames per second. In this paper, we address only the fast data rate, since it is our primary operational rate.

3. The gateway to a TCP/IP network for Link-11

3.1 The concept of substituting a TCP/IP network for an RF network

In general, Link-11 transmits or receives data over either a RF network (HF Radio) or a voice channel (audio to and from the Data Terminal Set), and the corresponding terminal equipment to participate.

Our approach uses the Link-11 DTS, as participants normally do, but digitizes the audio for transmit into a digital network (IP), or conversely converts IP packets into the equivalent audio into the DTS. This enables distributed Link-11 players to participate over an IP network connection, which provides higher reliability than a dial-up audio line that we have traditionally used. This is especially useful for distantly connected players that have caused us recurring problems with variable audio levels, call drops, and high error rates. This alternative, then, offers a superior approach to widely distributed interoperability testing, which is becoming more the rule than the exception.

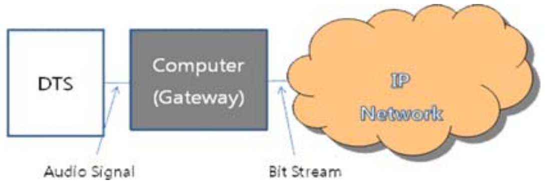


Figure 4. Data flow

Actual implementation of this IP gateway is accomplished by decoding modulated audio signal through a PC-sound card, and packing and sending the bit stream over IP network. Conversely, the gateway receives packets from the IP network, which it encodes into audio and sends to the DTS (Figure 4.). Thus the gateway is encoder, decoder and client on the IP network.

3.1 The new approach of encoding and decoding techniques for Link-11

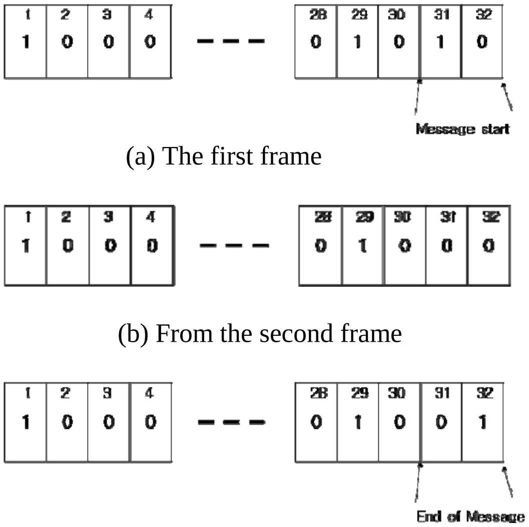
Conventional encoding and decoding processes are based on matched filter or correlation techniques. An alternative is based on Digital Signal Processing (DSP) using a Fast Fourier Transform (FFT), to manipulate frequency components within the real-time audio stream. We already know the specific frequencies that contain information. The FFT makes a set of complex numbers corresponding to frequency components. The complex numbers have information about the signal for both magnitude and phase, so we can extract phase information from the frequency components. All this is well within the reach of current technology to perform this processing within real-time, which means we can handle every signal from DTS using a Discrete Fourier Transform [2].

For the encoding process, we use an inverse FFT that converts the phase change to audio signal. The audio signal is a real-time signal so we must accordingly place complex numbers in the bin of the FFT in a symmetric pattern. For the required magnitudes, we perform automatic gain control to ensure the power difference of 6 dB between tones. The FFT method greatly simplifies analysis and handling of the audio signals without any loss of information. The audio signal coming from DTS is sampled through the PC sound card at a 44,100 Hz sampling rate. We then process these digital discrete samples based on the method of DSP as described.

3.2 Interface design to communicate with a TCP/IP network

The TCP/IP network interface acts like a client. It sends/receives files over socket with a standardized TCP/IP protocol stack. We do require that the interface maintain a connection with a server's IP address and port number, i.e., an IP address of "150.135.218.210" and a port number of 4,000. The interface consists of two parts, receive and transmit, that run independently (receive and transmit are each assigned a different process), enabling fully duplex operation. A continuously open TCP connection is therefore required to provide continuous receive processing, i.e., the interface receive function awaits bit data from the DTS audio decoder. When bit data files are available, the interface then transmits the files to a server. On the other hand, the interface's transmitting part waits for data from the server, from which it then generates bit data for the encoder to process the files and send them to DTS as audio.

Our IP side encoding uses efficient packing into 32 bit words. Within each 32 bit word, 24 bits represent data and 6 bits is hamming code for error correction (30 bit Link-11 frames), and in the remaining two bits we introduce 2 bits for indicators of start and end of message come, after the 30 bits such as ‘10’ for start and ‘01’ for end of the signal (Figure 5.).



(a) The first frame

(b) From the second frame

(c) The last frame

Figure 5. 32 bits frame structure

The framing format is 32-bits of binary data in little-endian. The binary data means that these 32-bits are 4 bytes with these 4 bytes being the digital representation of the analog 30 bits. The little-endian means that byte 0 contains the first 8 bits of the frame such that bit 0 of byte 0 is equal to bit 0 of the frame and bit 0 of byte 1 is equal to bit 8 of the frame. For example, 8-bits of data, "01000110", is to be 0x46 (1-byte) and one frame (32-bits) is “0x46 0x46 0x46 0x46”, or bit-wise: “01000110 01000110 01000110 01000110”. In the receiving process, the interface sends either 3*21*32-bit frames (three words) or 4*21*32-bit frames (four words) at one time since four words (4*21*32-bit frames) are the maximum data size that the gateway can generate in one second. Also, the same data format would be used for transmitting, but the data size is two frames (2*32-bit frames). This is because one significant piece of information in normal mode is contained within two frames and the data portion of the transmissions is a multiple of two frames.

4. Experiments

We implement the gateway in MATLAB and we compile it in order to operate in stand-alone machines that may not have MATLAB. We use Sound Blaster (Audigy) sound cards with a sampling rate of 44100 Hz. The experiments were done with one or more DTS at the JITC.

Our gateway connects to a remote monitoring computer where THE THEATER AIR AND MISSILE DEFENSE INTER-OPERABILITY ASSESSMENT CAPABILITY (TIAC) TOOLS runs within an IP network. TIAC analyzed the bit pattern received from the gateway and generated and transmitted data over IP networks. The test was carried out in NETTEST and ROLL CALL modes of the DTS, showing the capability to both interpret and generate known bit patterns from the DTS.

In NETTEST mode, the gateway decoded the signal from DTS and correctly sent them over the IP network. TIAC compared the bit pattern with NETTEST pattern and checked that they matched. TIAC also generated NETTEST data and sent that back to the gateway, which generated the correct audio encoded signal through the sound card and out to the DTS. DTS recognized the incoming signal as NETTEST.

In ROLL CALL mode, the same process was repeated and verified. This process is similarly used for message traffic and further validation of the concept and method. This is subsequently used in the operational mode of the gateway.

5. DEVS framework for parameter tuning of sound card

In the experiments, we found that the sampling rate of a commercial sound card varies by a very small amount. In other words, the sampling rate does not exactly follow the hardware specification given by manufacturer (tolerances, temperature, etc). For example, the gateway has run at the sampling rate of 43,998.86 Hz not 44,100 Hz on the test machine. This variation does not affect most sound card applications, but in our case, the time length of one frame is so short that the performance of the gateway is very much dependent on the accuracy of sampling rate of the sound card. Our solution to this involves computation of the inherent sampling error, and application of a compensating value to correct for the variations in commercial equipment.

The small amount of adjustment of the sampling rate of the hardware is not possible since sound card come with fixed sampling rates such as 44,100, 48,000 and 96,000 Hz. Therefore, we have conducted our initial experiments using trial & error values in our compensating (correcting) value. We could also have used techniques to recognize the number of samples using conventional optimization algorithms and genetic algorithms. Whatever our final method might be will require us to vary the number of samples per frame instead of sampling rate, since it is simpler to implement in software. In fact, we devised an approach using the DEVS formalism that allows us to do a one-time computation of the compensation value each time we use a different sound card (normally at installation of the gateway software).

Discrete Event System Specification (DEVS) is an advanced modeling and simulation formalism. The formalism is shown below [3]:

$$M = <X, S, Y, \delta_{int} , \delta_{ext} , \lambda, ta>$$

where

X: set of input values.
S: set of states.
Y: set of output values.
δ_{int}: Internal transition function.
δ_{ext} : External transition function.
λ: Output Function.
ta : Time advance function.

DEVS provides an efficient testing framework, including the experimental frame, in which we can define the model of a certain system configuration and run a simulation. This framework can also be plugged into the target system that we use for the gateway.

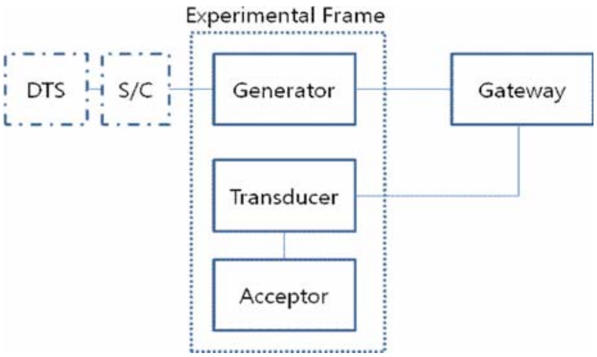


Figure 6. DEVS experimental frame for the gateway

To figure out the exact number of samples, we try repeated run with different values. To do this, we record samples of the audio signal from sound card first. The Generator (shown in Figure 6) generates input segments for gateway. That is done in two ways. It imports the stored audio data or generates the samples by itself. The generator computes the number of samples per frames arbitrarily and gives it to the gateway. The Transducer shown in Figure 6 has a stored reference bit pattern of NETTEST, to which it compares the output bit stream of the gateway. Acceptor monitors the result of the bit-wise comparison and stores it. This process is used to iteratively arrive at minimal errors, and thus obtain optimal samples per frame value. Another benefit of adopting DEVS comes from applying QFFT which is a quantized FFT using DEVS formalism [4]. It reduces computation time compared to that of FFT sustaining required precision.

By putting DEVS and this optimization technique together, we can achieve an automated and effective testing framework to calculate the number of samples per frame. We will study this work in the future.

6. Conclusions

In this paper, we presented our approach to implementing a cost-effective gateway for Link-11 using efficient DSP techniques and commercial platforms. The gateway is currently being validated and performance tested (characterized) within the lab networks. Further experiments are underway to show the performance and reliability of the gateway. Ongoing work will perfect the implementation of the DEVS framework to provide adaptive and robust samples-per-frame calibration to compensate for differences in commercial sound cards.

7. Acknowledgement

This research is funded by work performed under contract through Northrop Grumman Information Technology for the Joint Interoperability Test Command, Ft. Huachuca, AZ.

8. References

[1] Navy Center for Tactical Systems Interoperability (NCTSI): “Understanding Link-11: A Guidebook for Operators, Technicians, and Net Managers”, pp. 2_1- 2_31, September 1996.

[2] John G. Proakis, Dimitris Manolakis: Digital Signal Processing:Principles, Algorithms and Applications (3rd), Prentice Hall, 2005.

[3] B. P. Zeigler, H. Praehofer, and T. G. Kim: Theory of Modeling and Simulation (2nd), Academic Press, 2000.

[4] Harshavardhan Gopalakrishnan : “SCALABLE DEVS MODELING OF QUANTIZED DSP BASED SYSTEMS”, Thesis of M.S. , University of Arizona, 2005.

Author Biographies

Ho Jun Lee is a Ph.D. student in Electrical & Computer Engineering (ECE) at the University of Arizona and a member of ACIMS. His research interests are in modeling & simulation of complex systems based on DEVS such as communication and Radar systems. He received M.S. degree from Korea University.

Taekyu Kim is a Ph.D. student in Electrical & Computer Engineering (ECE) at the University of Arizona and a member of ACIMS. He holds an M.S. in ECE from the University of Arizona. His research interests include DEVS based hybrid system modeling, model based system design, and modeling language.

Bernard P. Zeigler, Ph.D. is Professor of Electrical and Computer Engineering at the University of Arizona, Tucson and Director of ACIMS. He is internationally known for his 1976 foundational text *Theory of Modeling and Simulation*, recently revised for a second edition (Academic Press, 2000), He has published numerous books and research publications on the Discrete Event System Specification (DEVS) formalism. In 1995, he was named Fellow of the IEEE in recognition of his contributions to the theory of discrete event simulation.

Dale Fulton is a technical lead for BAE Systems, for which he designs and develops instrumentation and performs test engineering Over his 25 year engineering career, he has developed numerous automated test approaches. He has worked within commercial, DoD, and DoE environments from concept to end-products for digital and RF applications, including cellular communications, tactical networks, and high-speed data acquisition. Currently he is automating traditionally manual tactical data link testing by applying systems methodology such as DEVS. Mr. Fulton holds a BS degree from the University of Arizona.

Doohwan Kim, Ph.D. specializes in discrete-event modeling and simulation and distributed computing. He had conducted research and development on complex large scale systems integration problems and led many software development projects. He provides strategic direction and manages operations of RTSync. He received his Ph.D. degree in computer engineering from the University of Arizona.