Open Source Analyzer for SISO-J Tactical Data Link Simulation

William Robertson[1], Peter Ross[2] and Andrew Robbie[2]
(1) Monash University aliask@gmail.com
(2) Defence Science & Technology Organisation
{andrew.robbie, peter.ross}@dsto.defence.gov.au

Abstract: Tactical Data Links (TADIL or TDL), such as Link-11 and Link-16, are increasingly being seen as essential for synthetic training exercises. Protocol analysis and debugging tools have traditionally focused on the Test and Evaluation communities' needs, specifically for actual data link hardware on military platforms. Air Operations Division of DSTO have a requirement for TDL protocol analysis tools which are flexible, easy to use and low cost. The open-source Wireshark network protocol analyser has been extended to support inspection of Link 16 J-series messages encapsulated in the SISO-J and STANAG 5602 SIMPLE protocols.

1. INTRODUCTION

Air Operations Division is presently upgrading two of its distributed mission training simulators to include representations of tactical data links. These simulators are the *Desktop Air Combat Simulator* (DACS) [1] and *Air Defence Ground Environment Simulator* (ADGESIM) [2]. Prior to this upgrade, the need for a simulated tactical data link analysis tool was identified.

Previous experience with simulated TDL nets as part of Royal Australian Navy (RAN) and United States Navy (USN) training exercises [3] highlighted the value of tools to analyse TDL traffic at a low level. However, existing tools are focused on the requirements of the Test & Evaluation (T&E) community, and are closed-source and expensive.

The goal of this work is to develop an open-source test suite for development and integration of simulated TDL systems. An initial prototype was developed as part of the DSTO Summer Vacation Scholarship program.

2. WHAT ARE DATA LINKS

Tactical Data Links (TDLs) enable the military to exchange tactical information in a precise, efficient and timely manner. Uses include building a Common Operating Picture, or exchanging targeting information to close the sensor/shooter gap. They complement, and in many instances replace, traditional voice-based communication mediums such as VHF radio.

TDLs comprise message standards for encoding information, distribution systems for exchanging messages between participants of a tactical data link network, and procedures for the management of link networks.

Link 16 is the primary TDL operated by the United States, NATO members and allied countries. Our work involves the use of *simulated* Link-16.

3. LINK-16

When operating within line of sight of another, the participants of a Link 16 network exchange messages using frequency hopping Time Division Multiple Access (TDMA) transceiver terminals operating in the L radio band (960 - 1215 MHz). The distribution system provides resistance against electronic countermeasures whilst maintaining security and integrity of the message content.

Link 16 supports the exchange of fixed length messages, known as J-series messages, and variable messages format (VMF) for applications requiring the transmission of free-text.

Each J-series message comprises an initial word, followed by extension word(s) and optional continuation words. Each word is 75 bits long, containing 70 bits of data and 5 bits parity.

Messages are identified by a label and sub-label. For example, label 2, sub-label 2, identified in shorthand as J2.2, conveys the Precise Participant Location and Identification (PPLI) of an air platform. This message is published on the network to advise other interested participants of an air platform's location and identifying information. United States Department of Defense MIL-STD-6016 and NATO Standardisation Agreement (STANAG) 5516 define how the information is stored and retrieved from the 75-bit words, and rules governing the issuance and receipt of those words [4].

4. SIMULATED LINK-16

The approach taken to simulate Link-16 in distributed mission training has been to simulate the distribution system using UDP and TCP network protocols, whilst retaining use of the J-series message format. These are protocols are also referred to as *wrappers*, as their primary task is to encapsulate J-series messages in a format suitable for transmission [5].

There are two primary distribution protocols found within distributed mission training exercises.

Simulation Interoperability Standards Organization SISO-STD-002 Standard for Link 16 Simulation [6]. Otherwise known as SISO-J, this protocol was developed by the simulation community to address the needs of training and research simulations. SISO-J is an extension of the Distributed Interactive Simulation (DIS) application protocol, and High Level Architecture (HLA) Real-time Platform Reference Federation Object Model (RPR-FOM).

NATO STANAG 5602 Standard Interface for Multiple Platform Link Evaluation (SIMPLE) [7]. SIMPLE was developed by the NATO Tactical Data Link Interoperability Test Syndicate to support TDL interoperability testing. The protocol operates over serial line or Internet Protocol. It does not simulate the characteristics of a real Link-16 network — its only role is encapsulation.

SISO-J encapsulates J-series messages inside the DIS Transmitter and Signal Protocol Data Units (PDUs), allowing line of sight, signal propagation and interference to be modelled within the simulation. The standard defines optional rules for modelling the characteristics of the TDMA network.

5. ANALYSIS TOOL REQUIREMENTS

The first part of the analysis tool suite is a network monitor. The tool will be used to diagnose interoperability problems that occur between two or more simulators attached to a distributed mission training network. Example questions that tool may be used to answer include:

- Are the simulators outputting SISO-J and/or SIMPLE messages on the correct network destination address or port number?
- Are the header fields of the SISO-J and SIMPLE messages populated correctly?
- Which J-series messages are being transmitted by the simulator?
- Do the JU and track numbers being transmitted correspond to the OPTASK Link?

Detailed analysis of the J-series message content is not required of this tool; deep analysis of these message formats is catered for existing data link testing tools used by the Test & Evaluation community.

The tool should be flexible, such that future distribution systems (such as JREAP, the Joint Range Extension Application Protocol) or future message formats (such as Link 22 F-series messages) can be implemented.

The tool should be portable across different operating systems and computer hardware, including Linux and Microsoft Windows.

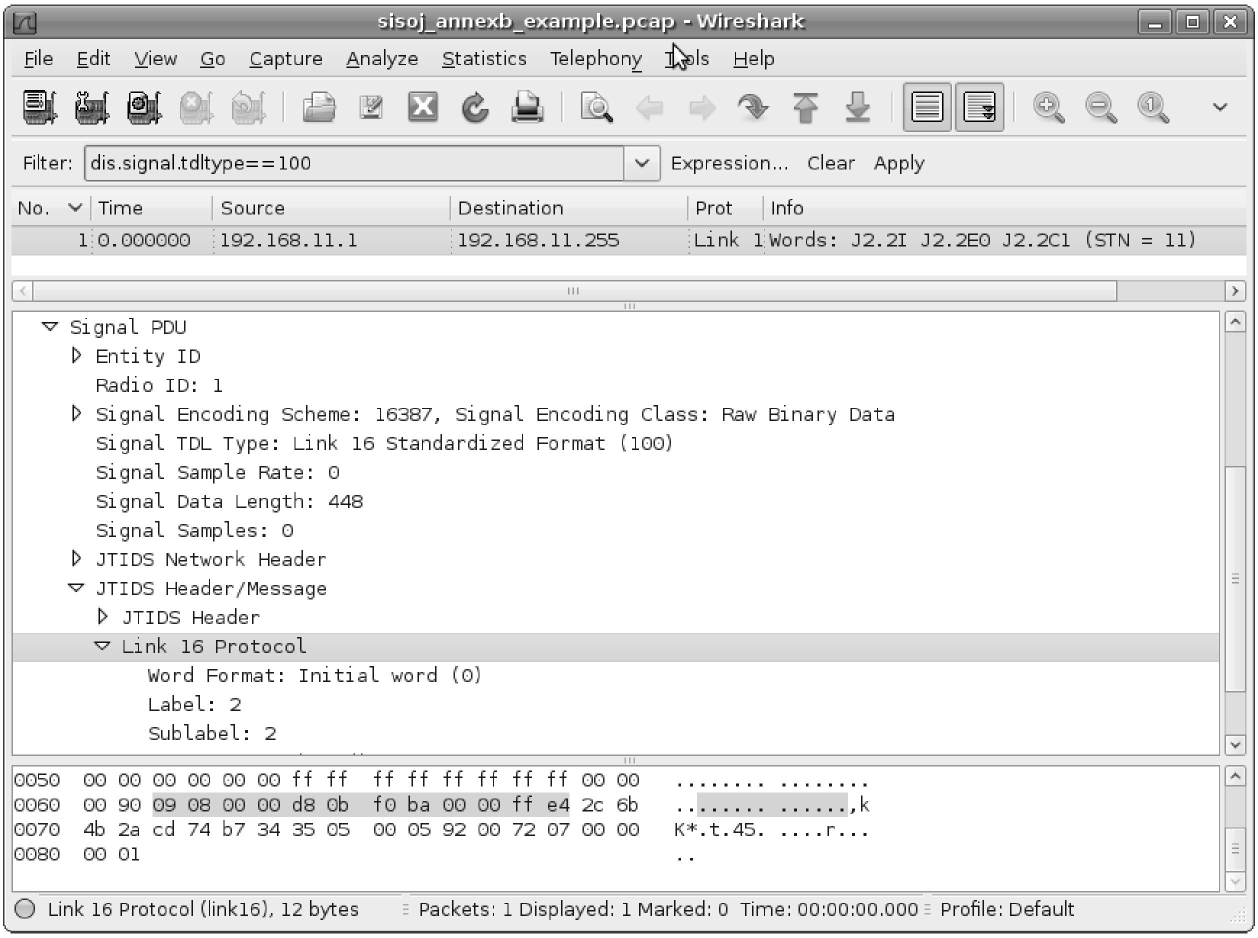
6. WIRESHARK

Wireshark is an open-source network protocol analyser. It provides a comprehensive filtering and query systems, utilities for performing statistical analysis, and dissectors to analyse inspect protocol content [8].

Dissectors are modules that exist within Wireshark to decode specific network protocols and present the protocol content in a human-readable format. A diverse range of protocol dissectors are included with Wireshark: base protocols such as Ethernet and Internet Protocol and over 500 other application protocols, including IEEE DIS. While it includes routines for dissecting DIS PDUs, at the time of writing only the most common of PDU types were supported, namely Entity State, Fire and Detonation PDUs.

Wireshark was chosen as foundation for this project for the following reasons.

- **Industry standard**. Wireshark is an industry standard for network protocol troubleshooting. Any effort spent improving Wireshark is likely to benefit its other users.
- **Familiarity**. For distributed simulation exercises Air Operations Division regularly uses Wireshark to diagnose faults and capture packets in simulation exercises.
- **Heritage**. Wireshark has been under development since 1998. Infrastructure to decode byte sequences, display information, express filters and manage configuration files has been previously built and thoroughly tested.
- **Portability**. Wireshark is written in the portable ANSI 89 C and runs on Linux, Mac OS X, BSD, Solaris and Microsoft Windows operating systems.
- **Active open-source project**. The software is distributed using the GNU General Public License and is supported by an active developer and user community. The project features a high volume email reflector, a wiki, an extensive library of packet captures, regression testing system and issue tracker.


sisoj_annexb_example.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: dis.signal.tdltype==100

```
No.  Time      Source        Destination     Prot    Info
   1 0.000000  192.168.11.1  192.168.11.255  Link 1  Words: J2.2I J2.2E0 J2.2C1 (STN = 11)
```

```
▽ Signal PDU
  ▷ Entity ID
    Radio ID: 1
  ▷ Signal Encoding Scheme: 16387, Signal Encoding Class: Raw Binary Data
    Signal TDL Type: Link 16 Standardized Format (100)
    Signal Sample Rate: 0
    Signal Data Length: 448
    Signal Samples: 0
  ▷ JTIDS Network Header
  ▽ JTIDS Header/Message
    ▷ JTIDS Header
    ▽ Link 16 Protocol
        Word Format: Initial word (0)
        Label: 2
        Sublabel: 2
```

```
0050  00 00 00 00 00 00 ff ff  ff ff ff ff ff ff 00 00   ........ ........
0060  00 90 09 08 00 00 d8 0b  f0 ba 00 00 ff e4 2c 6b   ........ ......,k
0070  4b 2a cd 74 b7 34 35 05  00 05 92 00 72 07 00 00   K*.t.45. ....r...
0080  00 01                                              ..
```

Link 16 Protocol (link16), 12 bytes   Packets: 1 Displayed: 1 Marked: 0 Time: 00:00:00.000   Profile: Default

Figure 1: *Example dissection of SISO-J Signal PDU containing J2.2 PPLI message.*

Table 1: Typical SISO-J protocol stack.

Application	J-Series Messages
Application	DIS (Signal PDU)
Transport	UDP
Network	Internet Protocol
Data Link	Ethernet II
Physical	100 TX (Cat5E)

Table 2: Typical SIMPLE Link 16 protocol stack.

Application	J-Series Messages
Application	SIMPLE
Transport	TCP
Network	Internet Protocol
Data Link	Ethernet II
Physical	1000 FX (LC)

7. EXTENDING WIRESHARK

Our approach to extending Wireshark has been to develop separate dissectors modules for each of *message* and *distribution* protocol, and then use the facilities within Wireshark to decode the stack appropriately. Common protocol stacks are shown in tables 1 and 2.

The SIMPLE dissector was written in 2006 to decode Link-11 message headers, in support on RAN/USN synthetic training exercises [3]. The SISO-J and J-message dissector was completed as summer vacation student project over a 10 week period. Publicly releasable components of the extension, specifically those concerning SISO-J, are being published separately in the form of a patch against the Wireshark source code [10].

8. USAGE EXAMPLE

Those familiar with Wireshark will find that the Link-16 dissector has the same user interface as existing decoders. Existing Wireshark facilities to capture, sort and dissect packets, and apply filters (through expression syntax) and perform bandwidth utilisation calculations are all available with the Link 16 dissector. Figure 1 shows the main window of Wireshark. This comprises a brief one-line packet summary shown in the *summary window*, the dissector output shown in the *protocol tree window* (center) and a hex-dump of the packet shown in *data view window* (bottom).

Due to the lack of a standard UDP or TCP port number for SISO-J and SIMPLE, the user must specify the port in a configuration page in order for Wireshark to automatically dissect SISO-J or SIMPLE packets. Alternatively the the '*decode-packet-as*' may be used to dissect selected packets.

In our laboratory the Wireshark software is installed on network monitoring computers that are attached to specific Ethernet switch ports that have *port mirroring* enabled. This enables traffic exchanged directly between computers to be captured by the network monitoring.

9. CONCLUSION AND FURTHER WORK

The Wireshark network protocol analyser has been extended to support analysis of simulated tactical data link messages. It has been tested and since used to diagnose interoperability problems between ADGESIM and a commercial SISO-J implementation.

9.1 Further work

Additional dissectors The utility could be enhanced by adding dissectors for other message formats, such as Link-11 M-series, Link 22 F-series, or Australian VMF messages.

J-series content dissector Besides the J-series label and sub-label, the dissector does not currently decode the contents of the J-series messages. Dissection of the Joint Unit (JU) number, Track Number (TN) and coordinate fields may be beneficial.

Link net characterisation It is commonly assumed that the simulated link net will behave in the same way as an actual net of the same configuration, however this is not the case, especially for distributed simulation. Participants in the same virtual link net (in simulation, within UHF range) may be thousands of kilometres apart — perhaps 250ms of delay. In addition, the packet loss rate or jitter may be significantly different. Network characterisation is essential to ensuring fair fight.

Message visualisation Errors in coordinate systems and artefacts like repeated tracks are hard to pick up at operator consoles (which display perceived truth) or network analysers (while positions may be decoded, they are hard to relate to the big picture). Link traffic that is associated with the IEEE DIS ground truth and the map provides a frame of reference to determine validity. Message visualisation is also useful for instructors and game controllers, especially when injecting deliberate errors for operators to handle.

10. REFERENCES

1. McIlroy, D., and Simpkin, G. (2006), *Aerospace Battlelab Framework Fighter Mission Experimentation Environment*, Proceedings of the 2006 SimTecT Conference, Melbourne Australia, May-June 2006.
2. Zalcman, L., (2005) *The Royal Australian Air Force, Air Defence Ground Environment Simulator*, Proceedings of the I/ITSEC 2005 Conference, Orlando, Florida, USA, November 2005
3. Clark, P., Ross, P., Oliver, P., Macdonald, R., and M. MacNeil, (2007), *Coalition Fleet Synthetic Training*, Proceedings of the 2007 SimTecT Conference, Brisbane, Australia, June 2007.
4. Friedman, N., (2006), *The Naval Institute Guide to World Naval Weapon Systems*, Naval Institute Press, ISBN 1557502625.
5. Hill, F., (2003), *Systemic Problems with Data Link Simulation*, Fall Simulation Interoperability Workshop 2003, Paper No. 03F-SIW-002.
6. Simulation Interoperability Standards Organization, (2006), *SISO-STD-002 Standard for Link 16 Simulations (Draft)*, June 2006.
7. North Atlantic Treaty Organization (2001), *Standardization Agreement STANAG 5602 Standard Interface for Multiple Platform Link Evaluation, Edition 1.*
8. Orebaug, A., (2007), *Wireshark & Ethereal: Network Protocol Analyzer Toolkit*, Syngress, ISBN 1597490733.
9. Sorroche, J., (2008), *SISO J to SIMPLE Translation Advice and Lexicon for Enabling Simulations (SIMPLE TALES)*, European Simulation Interoperability Workshop 2008, Paper No. 08E-SIW-046.
10. Robertson, W. and P. Ross, (2010), *Extending the Wireshark Network Protocol Analyser to Decode Link 16 Tactical Data Link Messages*, Draft DSTO Technical Note.

Author Biographies

WILL ROBERTSON is a final year BE (Software) student at Monash University and a DSTO summer vacation scholarship recipient for 2009/10.

PETER ROSS is an Engineer at the Defence Science and Technology Organisation, of the Australian Department of Defence. His research interests include distributed simulation interoperability and audio/video CODEC design.

ANDREW ROBBIE is a Senior Engineer at the Defence Science and Technology Organisation, of the Australian Department of Defence, where he works on architectures for human-in-the-loop simulation. His research interests include low latency networking and image generation.