Using Simulation to Test Manned-Unmanned Teaming

Michael J. O'Connor
Trideum Corporation
Huntsville, AL
moconnor@trideum.com

Kenneth LeSueur, Mark Ebert, Sean Millich
Redstone Test Center
Huntsville, AL
kenneth.g.lesueur.civ@
jonathan.m.ebert2.civ@
sean.d.millich.civ@
mail.mil

Fred Ventrone
CTSi
Lexington Park, MD
fred.ventrone@
goctsi.com

Tom Punihaole
Scalable Network Technologies
Huntsville, AL
tpunihaole@
scalable-networks.com

ABSTRACT

Manned-Unmanned Teaming (MUM-T) allows helicopter pilots to link with Unmanned Aircraft Systems (UAS) and receive video feeds, control payloads, and direct UAS movements. Robust communications protocols are used for this process, however, communications loss does occur due to signal strength, terrain, weather, and jamming. Using live testing to create all of these conditions is difficult and has potential safety and cost issues. To address the need for repeatable testing of these issues, a simulation environment was created to replicate the conditions using live aircraft on the ground.

The Joint Unmanned Aircraft Systems Mission Environment (JUAS-ME) is a Central Test and Evaluation Investment Program (CTEIP) for creating test environments for UAS testing. The Army's portion of the program is focused on MUM-T testing.

Two key technologies were developed to support this testing, Hardware-In-the-Loop (HWIL) tactical network simulation and MIL-STD-1553 bus extension. A real-time network simulation that emulates the Standard Common Data Link (SCDL) was built to perturb the data links between the helicopter and UAS. The communications effects server can operate in a physics based mode, fault inject mode, and a hybrid mode. A key test requirement was to exercise the use of the UAS's onboard lasers that receive commands over a MIL-STD-1553 bus. These non-eye safe lasers could not be fired in the UAS HWIL facility, resulting in the need to move the UAS sensor to another test lab. This extension necessitated the development of a MIL-STD-1553 bus extender to link the test facilities that are 4 miles apart.

This paper describes the issues encountered in creating a tactical network simulation with HWIL interfaces and the MIL-STD-1553 bus extender. In particular, several approaches were tried with the network simulation before a workable solution was found. The techniques describe in this paper can be applied to other programs with similar requirements.

ABOUT THE AUTHORS

Michael J. O'Connor is a Senior Program Manager at Trideum Corporation. Mr. O'Connor has more than 25 years' experience in Modeling and Simulation (M&S). He has been a key participant in the development of distributed modeling and simulation standards, including IEEE 1278 and IEEE 1516. He has held many positions in the community, including Chairman of the SISO Standards Activities Committee, Chairman of the SISO Executive Committee, and the chair of the I/ITSEC Simulation Subcommittee. Mr. O'Connor currently supports the technical integration of the "Always On – On Demand" program. He has led the development of multiple simulations using DIS, HLA, and TENA. Mr O'Connor has led the technical integration of several large multi-architecture distributed events including the Multi-Served Distributed Event (MSDE) in 2005 that used DIS, HLA, and TENA. MSDE included over 20 sites and included organizations from the Army, Navy, and Air Force. He holds a bachelor's degree in Computer Engineering from Auburn University, and a master of science in Computer Science from the University of Alabama in Huntsville. Mr. O'Connor is a CMSP.

Kenneth G. LeSueur, Ph.D. serves as the chief technologist in the Subsystems Test Division at the US Army Redstone Test Center (RTC). His work and research have been concentrated in HWIL testing, distributed testing, modeling and simulation, and high performance computing. He received his master's degree and doctorate in computer engineering at the University of Alabama in Huntsville.

Jonathan (Mark) Ebert serves as team lead for the Distributed Test Control Center at the Subsystems Test Division at RTC. He received his bachelor's degree in computer information systems at the Jacksonville State University.

Sean Millich serves as computer engineer in the Distributed Test Control Center at the Subsystems Test Division at RTC. He has more than ten years' experience in distributed M&S in complex network environments. He received his bachelor's degree and master's degree in computer engineering at the University of Florida in Gainesville. While working in the High-performance Computing and Simulation (HCS) Research Laboratory at Florida, his research focused on high-performance interconnects for cluster computing and modeling cluster scalability.

Fred Ventrone holds a Master of Science degree in Computer Science from The Johns Hopkins University, a Bachelor's of Science Degree in Electrical Engineering from The Pennsylvania State University, and a Bachelor's of Arts degree in Physics from Slippery Rock University. Mr. Ventrone has over 25 years of experience in systems engineering, systems development, integration efforts with landing systems (both ground and airborne segments), Global Positioning System simulation and testing, MIL-STD-1553 development, and developing and integrating hardware/software systems. Mr. Ventrone has a very significant technical history dating back to the early 1990's on the GPS testing/simulation and mid 1990's on the JPALS/WAAS programs, supporting the NAWC-AD ground and flight testing and most recently developing the Mudbucket system since 2007.

Thomas Punihaole is a Software Engineer at SCALABLE Network Technologies. He received his bachelor's in Computer Engineering from the University of Southern California. He also earned a Master's in Computer Science from the University of California, Los Angeles in 2012. He has designed and developed novel routing techniques for tactical MANETs and researched a variety of methods to make MANET networks more scalable. He is currently supporting the development of Joint Network Emulator (JNE).

Using Simulation to Test Manned-Unmanned Teaming

Michael J. O'Connor	Kenneth LeSueur, Mark Ebert, Sean Millich	Fred Ventrone	Tom Punihaole
Trideum Corporation	Redstone Test Center	CTSi	Scalable Network Technologies
Huntsville, AL	Huntsville, AL	Lexington Park, MD	Huntsville, AL
moconnor@trideum.com	kenneth.g.lesueur.civ@, jonathan.m.ebert2.civ@, sean.d.millich.civ@ mail.mil	fred.ventrone@ goctsi.com	tpunihaole@ scalable-networks.com

INTRODUCTION

In testing of systems that rely on wireless communications (comms) it is imperative to have a representative environment to ensure that the measured mission-level performance accurately depicts the true full-scale tactical situations. For complex Systems of Systems (SoS) testing, the ability to have a complete employment of all systems while operating in an operationally realistic environment is often an unobtainable goal due to limitations of funding, system availability, or safety. When testing aircraft, all three of these categories are constraints and test technologies and techniques are needed that produce quality test data that augments the live flight testing. The test and evaluation community routinely relies on techniques such as Hardware-in-the-Loop (HWIL) testing, modeling and simulation, Live-Virtual-Constructive (LVC) testing, and distributed testing to mitigate these limitations and risks [1].

HWIL testing is often the highest fidelity option, other than live systems-level testing, but it comes with its own set of technical challenges. One such challenge is the inability or difficulty in changing the tactical software and network interfaces to accommodate specialized test configurations and data acquisition. This paper presents an architecture to integrate distributed HWIL laboratories while performing real-time communication link emulation between systems without the necessity of modifying the tactical software or interfaces.

MUM-T BACKGROUND

Manned/Unmanned Teaming (MUM-T) allows Rotary-Wing Aircraft (RWA) pilots to use and/or control an Unmanned Aircraft System (UAS) as part of their mission. MUM-T supports several levels of control or interoperability of the UAS and its payloads. STANAG 4586 defines the following 5 Levels of Interoperability (LOI) [2]:

Level 1 -- Indirect receipt/transmission of UAS-related payload data
Level 2 -- Direct receipt of Intelligence, Surveillance and Reconnaissance (ISR) data
Level 3 -- Control and monitoring of the UAS payload in addition to direct receipt of ISR and other data
Level 4 -- Control and monitoring of the UAS, less launch and recovery
Level 5 -- Control and monitoring of the UAS, plus launch and recovery

This test environment is addressing Level 1 through 4. In addition to live flight test, ground testing of MUM-T capabilities with live UAS and RWA has been performed by co-locating both aircraft and using RF links. Live testing is very expensive and has limitations due to safety, security, and test areas/air space approved for UAS operations. One issue of interest in MUM-T operations is how the systems perform when messages are lost due to communications interference between the aircraft. This type of testing is very difficult to control, measure, and repeat using real RF links. The MUM-T systems implement a sequence of events where the RWA pilot gains control of the UAS through a hand-over process. This process is a sequence of messages exchanged between the aircraft within a defined period of time. The hand-over is approved via the UAS ground control station, but the message traffic is between the UAS and RWA.

3.0 MUM-T TEST REQUIREMENTS

The test environment had to support testing with live UAS and RWA on the ground in their respective HWIL test locations. Neither aircraft could be modified to support this testing. The test environment required the simulation of three critical elements:

- Provide communications effects for SCDL messages between aircraft
- Remote the Common Sensor Payload (CSP) to a location separate from the UAS
- Overdrive (overwrite data fields) the MIL-STD-1553 [3] bus data so the RWA would perceive and respond as it would in a live flight

The test environment must stimulate real hardware systems and provide physics based comms effects and to inject faults into the data link channels. The test environment is required to model the full communications stack, including message packaging, error correction, radio performance, antenna performance, and atmospheric/terrain effects. The test environment must support communications between multiple systems. These include the Ground Control Station (GCS), UAS, Video Display Terminal (VDT), and RWA. Figure 1 shows an overview of the systems and communications paths that are required to be communications affected. The UAS, GCS, Video Display Terminal, and RWA are all actual tactical systems.

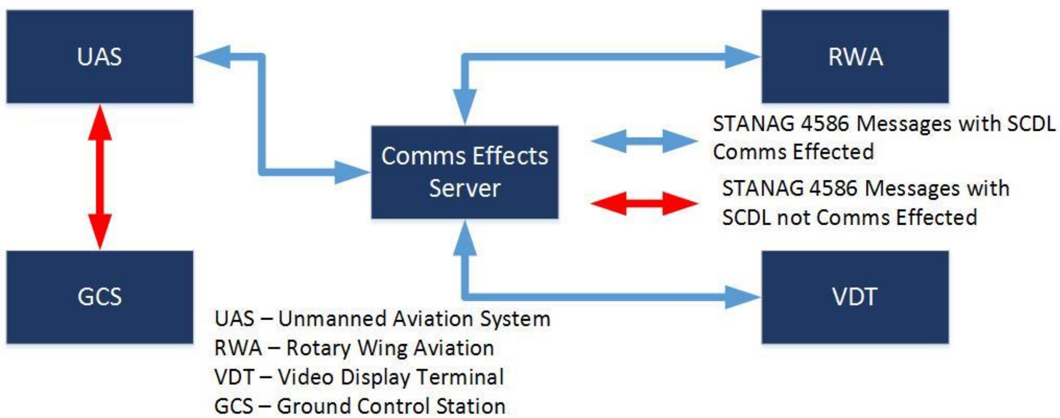

Figure 1. Data Links Requiring Communications Affects

The high bay the UAS is operated in is not configured for laser firing so the CSP was removed and located in a laser test laboratory. This distributed operation required extending the MIL-STD-1553 bus over Ethernet several miles. MIL-STD-1553 buses operate with tightly controlled timing which complicates the extension because of the timing delays in the Ethernet network. Remoting the CSP from the UAS also required that the video feed from the CSP be sent back to the UAS which is normally done via a FireWire connection in the aircraft. Meta-data is sent from the CSP to the UAS using RS-422 where it is combined in a video signal sent via SCDL.

The final requirement of the test environment is for both live aircraft systems to be in virtual flight. The UAS used for the testing was already instrumented and linked to simulations to provide this behavior. The control surfaces on the UAS move in response to flight commands. The UAS engines are not in operation, but are simulated. At the beginning of the program, a system did not exist to provide this level of stimulation for the live RWA in a hanger. In the RWA, translation and rotational state information that defines the flight state is distributed from the inertial navigation systems through the MIL-STD-1553 bus to the flight software. From the software perspective, the parameters that defined the flight state are typically treated independent of the engine state or rotor deflections since these are controlled directly by the pilot. The solution to placing the RWA in virtual flight is to use the translation and rotation parameters from a 6 degree of freedom simulation that were then injected onto the aircraft's 1553 bus. This is referred to as "overdriving" the bus.

TACTICAL NETWORK SIMULATION

The tactical network simulation is required to simulate the real-time communications between the live aircraft, GCS, and VDT. There were several aspects of communications effects required for this test environment:

- Model physics based radio propagation
- Model the SCDL communications stack
- Inject selected errors in the data stream

The program selected the Joint Network Emulator (JNE) from Scalable Network Technologies (SNT) as the basis for the comms effects model. JNE already supported the physics based radio propagation. The program added emulation of the SCDL communications stack to the comms effect model. A separate tool was developed to support intelligent real-time error injection. RTC's implementation of new wave forms and protocols in JNE and associated tools is named Joint Scalable Tactical Emulated Networks (JSTEN).

Standard Common Data Link (SCDL) Emulation

The SCDL [4] is a Department of Defense (DoD) IP-based datalink protocol for wideband communications. The RWA and UAS systems communicate using SCDL radios over Ku-Band. Modeling the effects of terrain, weather, noise and other communications effects between aircrafts located on the ground is made possible by a physics-based radio emulation. Since the aircraft have to remain stationary on the ground, the same effects couldn't be achieved using live radios. The MUM-T subsystems use the aircraft's onboard IP-based network to send multicast data and control messages through the communications stack. It was possible to intercept these multicast packets on the live systems and inject them into simulated radios.

Since SCDL is an IP compatible protocol, the simulated radio utilized the existing emulated stack for the network layer and above. The radios also use several communications standards such as General Framing Procedure (GFP), Header Compression and Packet Transfer Frame Format (PTFF). Figure 2 shows the modular breakdown of SCDL and the flow of information through the stack.

The physical and datalink layers of SCDL implement a very robust protocol using multiple techniques to prevent and detect packet errors. These error correction techniques add reliability at the expense of bit redundancy and transmission latency. The trade-offs depend on the combination of interleavers, multiplexers and encoders being used. This is modeled by adding delays and changing the data rate on the radio link. The error correction is implemented via a probabilistic Bit-Error Rate lookup table. Radio transmissions are modeled by sending simulated radio signals in the physics-based propagation model. This propagation model takes into account noise and interference from other simulated aircraft, as well as interference caused by terrain and weather.

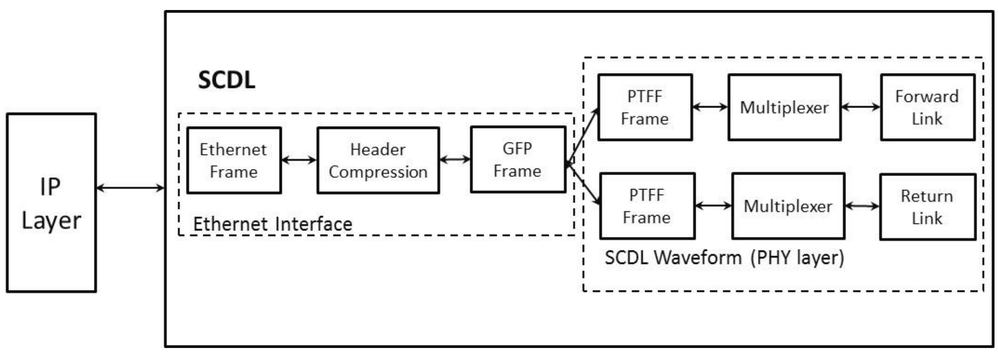

Figure 2. SCDL environment

Initial Approach

The two aircraft are located in separate HWIL facilities on Redstone Arsenal. The communications effects server is located in a third facility, the Distributed Test Control Center (DTCC), along with a VDT that receives video data from the UAS. The sensor payload is located in a fourth facility, the Automated Laser Instrumentation Measurement System (ALIMS) lab, that allows the laser to be fired. The rotary-wing aircraft are located in Aviation System Test and Integration Lab (AvSTIL). RTC's network supported creating a Virtual Local Area Network (VLAN) between these facilities. RTC and the UAS HWIL facility were connected via the Defense Research and Engineering Network (DREN). A Generic Routing Encapsulation (GRE) tunnel was established between RTC and the UAS facility to allow the messages on the private IP network to be routed between the facilities.

To provide comms effects between the HWIL systems, a mechanism had to be developed to intercept and forward the messages between the systems without modifying the tactical software. The approach also had to allow messages from the sender to be comms affected differently to multiple receivers (i.e. RWA receives message, VDT does not).

The initial approach to prevent direct unaffected comms between the systems was to use a different multicast address for communications between each system and the comms effects server. MUM-T uses four different multicast addresses for communications between the tactical systems. For each multicast group used by the MUM-T system, additional multicast groups are defined for each of the other tactical systems receiving the data. Each of the multicast groups provides one-way communication between the systems. By configuring the receivers to listen on a different multicast group, the comms effects server can be inserted into the communications stream. This approach allows comms effects to be applied to messages for each receiver separately. Figure 3 shows this approach. This concept is repeated for the four multicast addresses used by the MUM-T system.

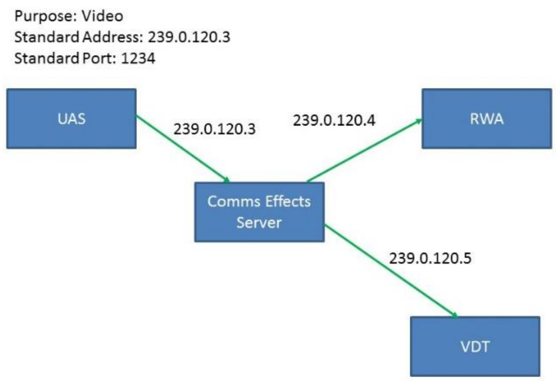

Figure 3. Initial Approach Diagram

This approach has the advantage of keeping all of the systems on the same network space, which simplified the network configuration. Testing was performed using the comms effects surrogate, UAS, VDT, and an emulator for the RWA. The VDT allowed the standard multicast address to be changed and worked correctly with the messages passing through the comms effects server. The RWA emulator also allowed the multicast address to be changed. However, by changing the multicast address, the standard Interoperability Profile (IOP) was not loaded by the emulator. In this mode the emulator does not respond to the UAS's health messages at the correct rate, which causes the UAS to drop the link with the rotary-wing emulator. Modifying the behavior of the emulator was not an option. Because the emulator would not work correctly with the modified multicast groups a new approach had to be developed.

Final Approach

The final approach taken separates the message traffic by using separate networks. This approach has the advantage of not requiring any changes to the tactical systems. However, this approach imposes additional requirements on the comms effects server and on the test network. Since a GRE Tunnel was already in place between the RTC network and the UAS facility network to route the private IP addresses, no additional work was required for that connection.

This approach required creating new VLANs on the RTC network to support the tactical systems. This also imposed a hardware requirement for the computer running the comms effects server. The comms effects server computer needed a Network Interface Card (NIC) for each network to support the MUM-T tactical message interception and retransmission. Because the comms effects server also requires real-time position data for each aircraft and ground terminal to determine the communications effects, an additional simulation network was used. Figure 4 is the network diagram used for this approach. The diagram shows the configuration for a single MUM-T multicast address where the UAS is sending information. The pattern is the same for the reverse path from the RWA to the UAS.

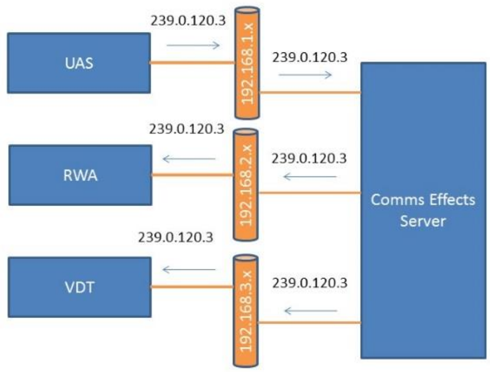

Figure 4. Final Approach Diagram

Similar configurations apply for each source-destination combination, with other multicast addresses used on the same four VLANs. This approach also required internal enhancement and configuration of comms effects server to support routing between the networks.

Failure Injection

The physics based comms effects server is useful for exploring message loss based on the RF phenomenology and setup of the test. To support functional testing of the interface between manned and unmanned systems, a capability was needed to intentionally block particular messages on the communications link. This capability allows for specific messages types or values to be blocked between the communications systems. This is implemented in the non-physics based approach in the Multicast Multiplexer (MM) tool. The MM filter and the comms effects server can be combined to use both a deterministic filter as well as use the non-deterministic physics based model. Figure 5 shows the configuration with the filter only. Figure 6 shows the filter and the physics based model.

Figure 5. Filter Only Example

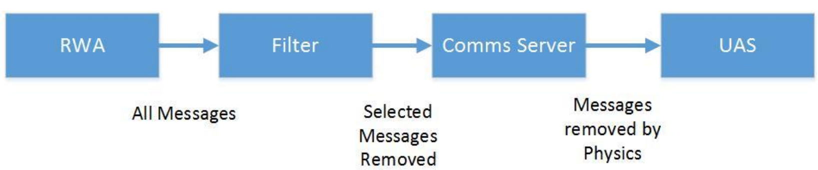

Figure 6. Filter and Comms Effect Server Example

Both the filter and physics based comms effects server support multiple channels. This allows the messages to be treated differently based on the sender/receiver pair.

The filters are defined before execution with all filter details stored in XML files. More than one filter can be applied at the same time. Each time filter is applied and the time it is removed it is removed is logged for analysis. All messages that are blocked, the time they are blocked, and the filter that blocked them are also logged.

SCDL is designed to be very robust, and as such it will rarely deliver corrupted messages. In addition, based on the way messages are duplicated and divided into multiple packets it is unlikely a disruption in the communications link will only cause a single message to be lost. To support functional testing of different behaviors, three types of filtering have been defined: time-based, message type, and message type plus. Each filter has a name, multicast address, sender, receiver, filter type, and filter data.

Time based filters are used to block all messages on a specified multicast group from a sender to a receiver for a user defined time period. The filter allows the user to specify a pattern of link blockages that repeat until the filter is removed. The pattern is a defined set of durations when the link will be blocked and when it is not blocked. The following is an example pattern: 5 seconds blocked, 2 seconds not blocked, 3 seconds blocked, 5 seconds not blocked, etc. This pattern would repeat until the filter is removed. There is no limit on the length of the pattern.

Message type filters are used to block messages of a selected type on a specified multicast group from a sender to a receiver. The filter may block all messages of the type, every other message of the type, or every nth messages of the type. The filter is in effect until it is removed.

Message type plus filters are used to block messages of a selected type and all messages (from specified sender to receiver) after the specified message for a specified period of time. A time is specified for MM to block messages after the receipt of the message. The filter specifies the multicast group, sender, and receiver. Once the specified message is detected, MM will block all messages for the time defined after the receipt of the selected message. This filter simulates the case of the CDL error correction not being able to reconstitute the message and an entire block of data is lost.

REMOTE CSP

The CSP was remoted (removed from the UAS and moved to another location) to a facility that was cleared to allow laser firing. The laser firing commands are a critical element of the MUM-T and some of the operations are safety critical. Therefore, use of the laser commands is important to the test environment and cannot be performed at the facility housing the UAS. This required addressing two technical issues: MIL-STD-1553 bus extension and video distribution with metadata. The following sections describe how these issues are addressed.

Bus extension to CSP

The extension of the MIL-STD-1553 bus on the UAS, located in the JSIL HWIL, and the CSP, located in the ALIMS lab several miles away, was developed to allow minimal delay in the communication between the Bus Controller on the UAS and the Remote Terminal (RT) of the CSP. The communication link between the two locations is Ethernet over copper and fiber optic networks. The MIL-STD-1553 bus extension was developed using two Mudbucket MIL-STD-1553 bus extenders from Coherent Technical Services Inc., which support up to 10 channels of dual redundant MIL-STD-1553 bus communication (unique data overdrive, standard Bus Monitor (BM), standard Bus Controller (BC), and standard RT operations). The Mudbucket was chosen so as to have a single unit that would support not only the extended MIL-STD-1553, but for the unique ability to overdrive data in a way that cannot be done with Commercial Off The Shelf (COTS) equipment. The overdrive functionality will be discussed below in the Overdrive Section.

Figure 7 shows the overall MIL-STD-1553 Bus Extension architecture. There are two bus extenders involved, one at each of the two physical locations. The bus extender takes the single (Blue 1553 Bus in Figure 7) and turns it into two separate busses. Mudbucket-1 acts as an emulated RT, emulating the CSP in order to properly respond to the real UAS BC (within the MIL-STD-1553 specifications). Mudbucket-2 acts as an emulated BC sending the commands received at Mudbucket-1 to the real CSP on the second MIL-STD-1553 bus. In addition, Mudubcket-2 also relays the responses of the real RT data back to Mudbucket-1 which makes it available as the response data in the emulated RT, thus completing the round trip communication between the UAS BC and the remoted CSP RT.

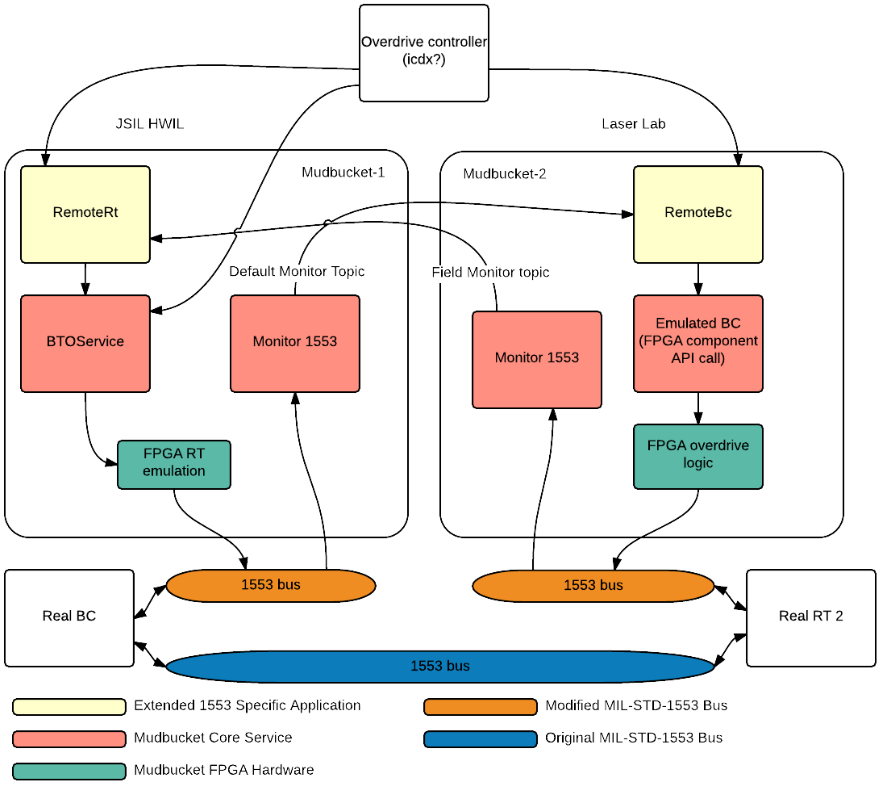

Figure 7. Mudbucket MIL-STD-1553 Bus Extension

All inter-process and inter-Mudbucket communication are done using a third party middleware publish/subscribe network software called RTI Connext. It complies with the Data Distribution Service (DDS) standard from the Object Management Group. One feature of the DDS standard is to separate peer-to-peer communication via domains. Using this and configuration parameters defined by the user, Mudbucket-1 only sends the MIL-STD-1553 bus traffic related to the remoted RT across the network to Mudbucket-2 so as to minimize the overall network traffic. Furthermore, the communication message between the two bus extenders is the exact same message, but kept separate via the domain identification. This is key as the exact same process runs on both bus extenders to generate this data and it is common throughout which makes consuming the data very simple within different applications.

A number of challenges were faced during this development, such as the differing Terminal Addresses (TA) of the CSP on the UAS and the CSP in the ALIMS lab, responding to the first command messages intended for the target RT, dealing with infrequent messages, and handling firewall port definitions for the RTI Connext middleware.

The differing TAs of the on aircraft CSP and the CSP in the ALIMS lab was allowed as a configurable parameter at the start of the application. It was a fairly simple fix to allow the change of the command word on the remoted side to account for the new TA. The status word generated on the JSIL HWIL side has the TA portion fixed upon setting up the emulated RT.

The issue with how to respond to the first message had some possible serious implications, the worst of which could be that the BC would issue a subsequent command to turn on the lasers unexpectedly. So, it was important to determine how to handle this safely. The initial implementation did the simple thing, just respond with all zero data on the first of each message requested. This was fine for implementation in the lab. However, not knowing how the real UAS would respond, it was determined that something more robust was required to test on the aircraft. The possibility of responding with a set of known valid data was considered. This may be a viable solution, but it was not clear there would always be the same answer to give. For instance, if there were some positional data based last location from the last run, this could change from one startup to the next. It was finally determined to not respond to the first command from the BC until valid data was received from the remoted RT. The measured latency between the original command from the BC to the remoted RT and back to the real BC was a single message delay, which was determined to be acceptable.

Dealing with infrequent messages could have an impact on the system getting old data. The first occurrence of this is covered by the fact that system does not respond to a first-time message until valid data is received from the real RT. However, subsequent command for this message would respond with latent data that could be harmful to the system. Therefore, a timeout was implemented for the remoted BC for each message sent to it from the real BC. If a message has not been requested on the remoted BC side for greater than the timeout period, the remoted BC would insert a command requesting that message from the real RT, thus keeping a local updated copy of that data in the event the real BC requests that message.

Overall analysis showed that at most there was a single message delay in the remoted MIL-STD-1553 application. However, due to the asynchronous operation there were very infrequent times when the remoted RTs data did not arrive in time for the next requested message from the real BC. In this case the previous messages' data was repeated in the data requested. Furthermore, all of the normal operations of using the remoted CSP were successful including operating in all modes of the CSP.

Video

The video from the CSP is sent to the UAS processing via a FireWire connection with the metadata sent via a RS-422 connection. The RS-422 data is embedded as Key Length Values (KLV) in the video stream. Remoting this process required two different approaches.

The first was to get the video from the CSP to the UAS. This was accomplished with a video encoder. The same type of encoder used on the UAS was used in the ALIMS lab. This was a relatively simple hardware solution, however documentation was not available as to which video codex is used. It took some experimentation to determine the correct codex and video size. Lessons learned from the video transfer are in the Results section.

The second aspect was the metadata associated with the video stream. This includes the location the CSP is looking at and other information. This data is send via a RS-422 connection and combined with the video on the UAS. RS-422 extenders over IP are used to remote the metadata.

This solution allows the remote video to be processed and sent as though it was on the UAS. There is an added latency for the remote CSP to UAS link, but it is small enough to not introduce problems for this type of testing.

RWA OVERDRIVE

There are two parts of the Mudbucket overdrive capability developed for this test environment. The first part is related to the RWA in virtual flight controlling the CSP via the UAS. The second part is related to making the remoted CSP perceive that it is flying at altitude with the simulated flight of the UAS while it is sitting static on top of a tower tracking targets on the ground below it. The targets on the range will actually be much closer than they would be if the CSP was at altitude observing the same targets.

The first overdrive application is putting the RWA in virtual flight. The Mudbucket interfaces with the AvSTIL Time Space Position Information (TSPI) simulation via a reflective memory network and transparently overdrives the pertinent TSPI parameters on the respective MIL-STD-1553 and Aeronautical Radio INC. (ARINC) 429 buses on the aircraft. This MIL-STD-1553 and ARINC 429 overdrive occurs with minimal latency (approximately 1.5 usec) such that the aircraft system does not know the Mudbucket is in the system. This provides the aircraft with the TSPI required to support tests as if in actual flight.

The second overdrive is currently in development and it will involve overdriving data to and/or from the CSP to support the test scenario where the CSP is mounted on a range tower tracking live targets below. The challenge here is that the collective system must perceive that the CSP is actually on board the UAS and tracking targets from a much higher altitude while the real CSP is statically sitting on the tower tracking targets that are much closer. In addition, this will be accomplished while the RWA is in virtual flight in the AvSTIL hangar. Since the pilot in the RWA can control the CSP, the test systems must be able to overdrive messages to the CSP as well as from the CSP in order to make the entire system useful in this virtual environment.

RESULTS

The work for this effort has been performed over several years. The first year focused on the proving the techniques used to connect the two aircraft using non-radio-frequency links and the physics based comms effects server. The second year focused on remoting the sensor from the UAS. The original approach changed over the course of the program as more information about the aircraft were learned.

The basic connectivity was accomplished between the aircraft. Most of the initial problems in connectivity were related to the ground network used. The comms effects server and RWA were located in labs operated by RTC. The UAS was located in a facility operated by the Aviation and Missile Research, Development and Engineering Command (AMRDEC). Both organizations have access to the Defense Research and Engineering Network (DREN). Because the aircraft use private IP addresses, a tunnel had to be established between the test facilities. All of the traffic is multicast and the routers are configured in sparse mode. During initial testing, data from the AMRDEC site reached RTC, but RTC's traffic was not seen on the AMRDEC side. The problem was determined to be that the tactical systems located on the AMRDEC side did not join the multicast groups. This issue was solved by adding static joins to the routers on the AMRDEC side.

There were two primary cases where the comms effect server would drop the link between the two aircraft. The first was when terrain blocked line-of-sight between the aircraft. When this occurred the loss of connection was instantaneous. This was tested using mountainous terrain to mask the simulated RF signals. This worked as designed. The second case is where the aircraft exceed the maximum range of the radio systems. This loss of connection occurs over time as some packets are delivered and others are dropped. This is based on the radio performance and protocol performance modeled by the comms effect server. Pilots are used in the RWA to operate the MUM-T system during testing. During the extended range testing, they commented that the controls seemed sluggish and the UAS was slow to respond to commands. This was the expected result as it emulated the real world behavior of the systems when operating close to maximum range.

The focus of the second year of testing was the remoting of the CSP to a laser safe lab. This was accomplished with the MIL-STD-1553 bus extender. This performed very well and neither the CSP nor UAS experienced any issues related to the bus extension.

By remoting the CSP to another lab the video from the CSP had to be sent back to the UAS to be processed. This required video encoders that could take the FireWire video and convert it to be sent over an IP network. The video was received directly on the IP network of the UAS. However, this network also had the application that converted the RF signal from the UAS into IP via a device called the Mini-C. The extra video traffic overwhelmed the interface on the Mini-C. This resulted in an incomplete handover process between the UAS and the RWA. The timing sequence required to complete a handover is very tight. Because of unintended packet loss caused by the video stream, the messages for the handover were being lost. This required architectural changes to the network on the RDECOM side. Two separate networks had to be created and the static join on the multicast address was only required on one of the networks. The same tool that is used to filter messages can republish video streams on different multicast groups. The video was sent from the ALIMS lab on one multicast address and then translated to the required group on the UAS internal network. Once this fix was in place the handovers could be completed.

Several different network configurations were used before the final solution was reached shown in Figure 8. The DTCC, AvSTIL, and ALIMS lab are located on the RTC network. The JSIL is located on the RDECOM network. The IP addresses shown are representative of the actual addresses, but are not valid IP addresses.

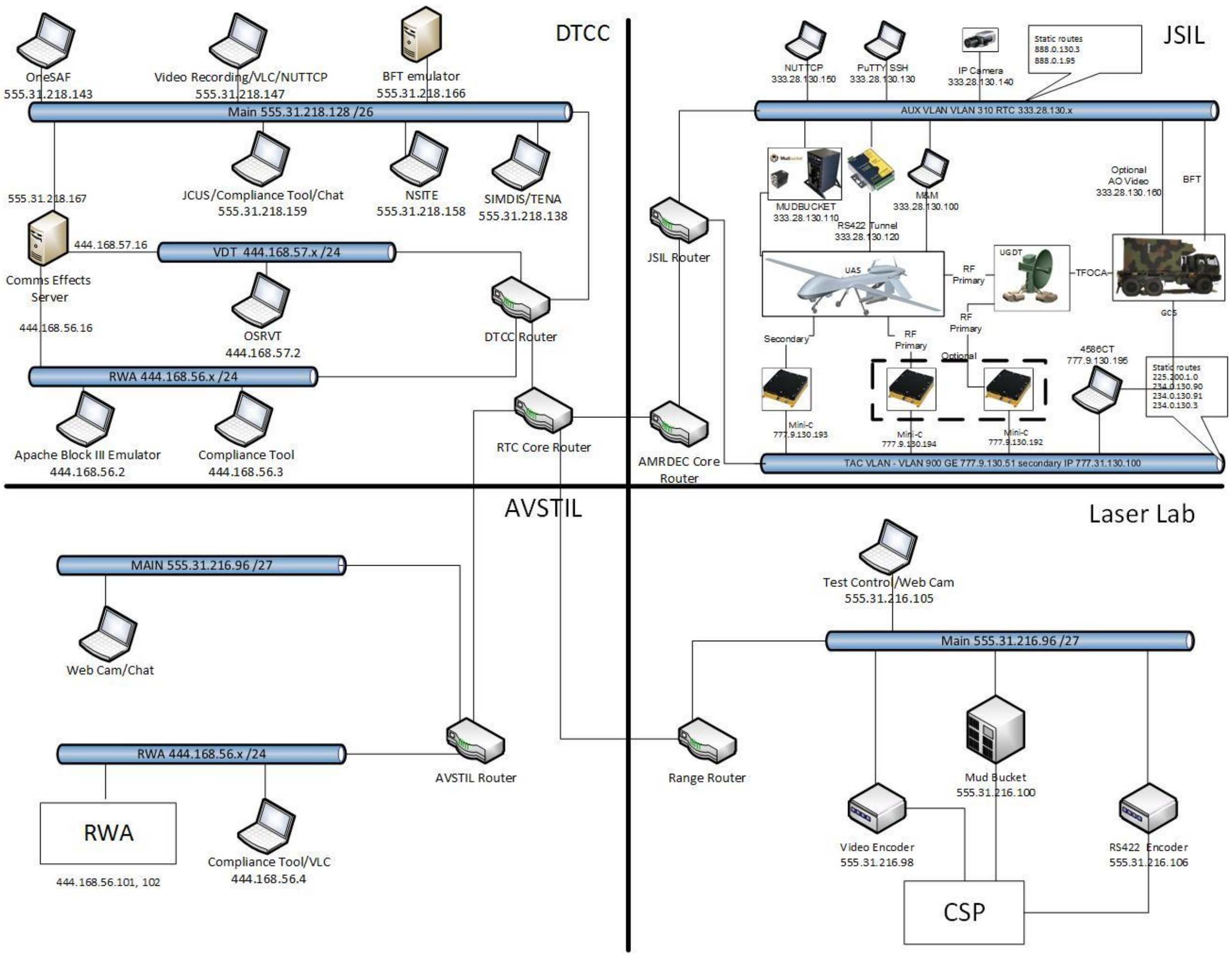

Figure 8. Final Network Configuration

FUTURE WORK

While significant progress has been made in the development of this test capability, future work remains. Some of this will be performed by the current program and some may be performed by future programs.

In this year's effort the CSP will be moved to a tower on the test range and will lase actual targets. The CSP will know its actual location on the test range at RTC and will provide data based on the target it is lasing. This presents an issue in that the simulation flying the UAS will have it located at a different position. At the time this paper is written several approaches are being considered on how to reconcile these different locations.

The connection to the RWA is made using a special port to access the internal IP network. This port is only available on one model of the RWA. Future work may include using a product similar to what is used on the UAS to convert the RF signal to IP allowing different models of the RWA to be used for testing.

There is also the potential to add another type of filtering to the filter application. This type of filter would be used to block messages of a selected type that contain a data field with a specified value on a specified multicast group from a sender to a receiver. This filter will support a subset of SCDL message types. Each selected message type will allow the user to filter based on the values of selected message fields, which will allow finer control of the messages blocked by the filter.

Comms affecting the data link between the UAS and GCS was not required for this test environment, however adding comms affects to that link is a possible future enhancement. The comms effects server could model this link without any changes. To add this capability would require additional changes in the network configuration at the AMRDEC lab. These changes may be made at some time in the future to support new test requirements.

CONCLUSIONS

This effort demonstrated a test environment can be created to test MUM-T communications between a UAS and RWA. The maturity of technologies to support distributed SoS testing and high fidelity tactical network simulations are such that they can be applied to formal test/experimentation events. In performing distributed technical testing, care must be given to analyzing the system(s) latency budget. But as in this case, many applications, even with tight requirements can be supported in a distributed manner if the proper engineering attention is given to the test network architecture, dissection of the system(s), and distance of separation. The approach documented in this paper can be applied to other similar communications test environments. While other types of radios and protocols will have some different issues, many of the same technical issues will have to be addressed.

ACKNOWLEDGEMENTS

This effort was funded by the US Army's Joint Unmanned Aircraft Systems – Mission Environment (JUAS-ME). JUAS-ME is a Central Test and Evaluation Investment Program (CTEIP) funded effort. Mr. Rob Stone is the Program Manager for the US Army JUAS-ME program. James Perillat the UAS Programs Division Project Support Lead of the AMRDEC Joint Technology Center/Systems Integration Lab (JSIL) provided support for the operation of the UAS.

REFERENCES

[1] Kenneth G. LeSueur, Emil Jovanov, Ph.D., "Hardware-in-the-Loop Testing of Wireless Sensor Networks," in International Test and Evaluation Association Journal September 2009, 30: pp. 333-338.

[2] STANAG 4586, "UAV Control System (UCS) Architecture". Edition 2

[3] MIL-STD-1553B, Digital Time Division Command/Response Multiplex Data Bus, United States Department of Defense, September 1978

[4] Specification 7681990, Revision H Performance Specification for the Standard Common Data Link (Std-CDL) Waveform