Message Latency Characterization of a Distributed Live, Virtual, Constructive Simulation Environment

James R. Murphy[*]
NASA Ames Research Center, Moffett Field, CA, 94035

Srba Jovic[†] and Neil M. Otto[‡]
SAIC, Moffett Field, CA, 94035

A distributed test environment incorporating Live, Virtual, Constructive, (LVC) concepts was developed to execute standalone and integrated simulations and flight-tests that support unmanned aircraft research for NASA's Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Project. The LVC components form the core infrastructure that supports simulation of UAS operations by integrating live and virtual aircraft in a realistic air traffic environment. This LVC infrastructure enables efficient testing by leveraging the use of existing distributed assets. The LVC concepts used for the UAS in the NAS project include live aircraft, flight simulators, and virtual air traffic control assets operating at facilities distributed across multiple NASA Centers. With a distributed network, however, there is a concern that message latency could impact the realism of a simulation and its data. The latencies associated with sending data among these distributed facilities were, therefore, measured to ensure that they fall within acceptable parameters. Several live and virtual test assets were integrated into the LVC infrastructure including NASA Armstrong's Ikhana MQ-9 unmanned aircraft, NASA Glenn's S3-B manned aircraft, and the B747 flight simulator at NASA Ames. Average latencies from 100 to 150 milliseconds were observed between the LVC System running at NASA Ames and each of the participating NASA Centers under a light-to-moderate (fifty aircraft) traffic sample.

Nomenclature

ADRS	=	Aeronautical Data Link and Radar Simulator
ATC	=	Air Traffic Control
HLA	=	High Level Architecture (simulation middleware)
LVC	=	Live, Virtual, Constructive describing the simulation environment
MACS	=	Multi-Aircraft Control System (a pilot and air traffic emulator)
MPI	=	Multi-Purpose Interface
NAS	=	National Airspace System
RUMS	=	Remote User Monitoring System
SAA	=	Sense and Avoid
SAAProc	=	Software container for the Sense and Avoid algorithms
UAS	=	Unmanned Aircraft System
VSCS	=	Vigilant Spirit Control Station
XML	=	Extensible Markup Language (programming language)

---

[*] Project Engineer, Aviation Systems Division, MS 243-1, and AIAA Associate Fellow.
[†] Software Manager, SAIC/NASA Ames Research Center MS 243-6, 94035.
[‡] Test Director, SAIC/NASA Ames Research Center MS 243-6, 94035.

I. Introduction

THE Unmanned Aircraft Systems (UAS) Integration in the National Airspace System (NAS) Project is investigating and integrating technologies that are intended to reduce technical barriers related to the safety and operational challenges associated with enabling routine UAS access to the NAS. To support this goal, the project is developing a distributed live, virtual, and constructive (LVC) test environment to enable human-in-the-loop simulation and flight test activities. LVC test environments are not a new concept; they are widely used by the Department of Defense and throughout the aviation industry to provide a safe and relevant test and training environment.[1,2,3] The LVC test environment for the UAS in the NAS Project is comprised of air traffic control (ATC) workstations, constructive and virtual aircraft simulators, and UAS ground control stations (GCS) that, operating together, provide researchers with a relevant NAS environment to test unmanned systems. In order to maximize the use of available resources, the LVC test environment is designed for distribution, enabling technologies developed by researchers and external partners to be integrated into a simulation or flight environment.

The distributed nature of this environment adds to its complexity, not only with the logistics of running simulation and flight tests across several facilities, but also with respect to the synchronization of input data streams. The latencies associated with messages passed between LVC components may impact the ability to achieve a simulation environment that meets or exceeds the operational environment it is intended to emulate. Whether that environment is the five second requirement for display of ADS-B data to a pilot, or three seconds for the display of En-route data to a controller.[4,5,6] In addition, to properly synchronize LVC data, it is critical to understand the latency inherent among distributed components to determine whether latency differences need to be mitigated in order to properly synchronize the data sources. For this purpose, a prototype of the LVC environment was developed to support the analysis of observed latencies among LVC test components. This prototype includes instrumented software that measures the message latency between LVC components distributed across multiple NASA facilities, including delays added by any required network firewalls and security encryption. Results from initial latency and system testing further influence the overall design of the LVC infrastructure, leading to system improvements that meet the UAS in the NAS Project's research requirements and the operational air traffic display requirements the system is designed to emulate.

This paper documents the LVC test environment and software components used by the UAS in the NAS Project for its planned simulations and flight tests. It provides a detailed description of a gateway process developed to support the connection to and routing of messages between the LVC components. Lastly, it documents the observed latencies among existing and candidate simulation architectures for use in future project simulations and flight tests.

II. LVC Description

The UAS in the NAS Project is conducting a series of integrated human-in-the-loop simulations and flight tests. These tests will investigate UAS ground control station display features, UAS communication solutions, as well as evaluate pilot and controller acceptance of the usability and timeliness of self-separation advisories. To support these planned simulations and flight tests, the Project is developing a distributed LVC test environment. The LVC portion of the test environment refers to the components of the test that can be regarded as "live", "virtual", or "constructive". A "constructive" simulation generally has no interactive human involvement in simulated conditions. Instead, scenarios unfold using rule-based decisions that control the interactions between simulated actors. "Virtual" simulations involve human participants operating simulated systems (e.g. a pilot flying a flight simulator). A "live" test environment involves human particpants operating real systems.[1] While the live, virtual, and constructive components of a test environment only comprise a portion of what is required to run a simulation or flight test, the test environment is widely known as an LVC. It should be noted that categorizing components of a simulation as live, virtual, or constructive can be problematic. Since the degree of human participation in a simulation is widely variable, as is the degree of equipment realism, there is no clear division between these categories.

Figure 1 provides a high level concept of operations for the LVC test environment developed for the UAS in the NAS Project simulations and flight tests. From the beginning of project planning, it was recognized that due to the nature of flying of UAS aircraft and the restrictive nature of the FAA Certification of Authorization process, use of restricted airspace would be desirable. The overall purpose of the LVC environment is to utilize existing ATC workstations and simulation infrastructure resident at NASA Ames Research Center with the flight of aircraft in the restricted airspace at Edwards Air Force Base (where NASA Armstrong Flight Research Center resides). The LVC environment provides the connectivity between the distributed facilities and the mechanism for integrating the data from the disparate systems together into a single system. The LVC infrastructure abstracts the original source of the data, seamlessly integrating live UAS and manned surrogate UAS aircraft with constructive or virtual aircraft, enabling realistic ATC and pilot displays and data ingestion into sense and avoid (SAA) algorithms. Since the source and client processes that comprise the LVC environment may be distributed across laboratories at a single facility, or across the country, it is critical to measure latencies associated with sending messages between the LVC facilities.

A. LVC System

Figure 2 depicts the high level connectivity among the software components, designed for a UAS in the NAS Project flight test. The flight test incorporates live aircraft merged with constructive background traffic integrated into a virtual air traffic control environment. The primary LVC infrastructure components are shown as ovals, while simulation client processes are shown in boxes. The LVC Lab at NASA Ames contains the High Level

Figure 1. LVC Environment Concept of Operations. *An LVC environment promotes the integration of multiple live and virtual data sources.*

Architecture (HLA) and toolbox processes that provide the message routing among the distributed facilities. The LVC Gateway process, shown running at the NASA Armstrong LVC Lab, routes data within the facility among several local LVC software clients. The Multi-Aircraft Control System (MACS) provides air traffic control and constructive aircraft pilot displays run at the NASA Ames ATC and Pilot Labs. The Vigilant Spirit Control Station (VSCS) provides the control of the surrogate UAS aircraft as well as the ground control station test environment and is run out of NASA Armstrong. The sense and avoid algorithm container (SAAProc) provides an interface to the self-separation algorithm and is connected to the LVC Gateway. The surrogate UAS aircraft, as well as any manned intruder aircraft, are flown out of the Edwards Air Force Base restricted airspace. Data from the surrogate aircraft are sent to a ground station via the prototype UAS communication system and on to the VSCS and LVC Gateway. The LVC environment also provides near real-time access and visualization of the data as it is being recorded through the Remote User Monitoring System (RUMS). A description of each of these components is provided in the following section.
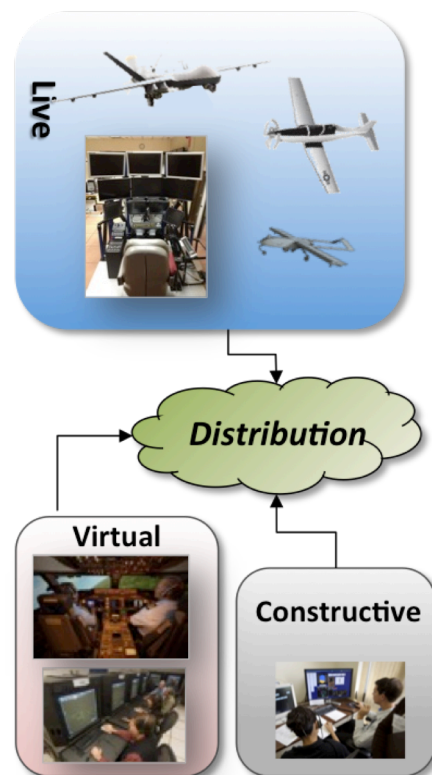
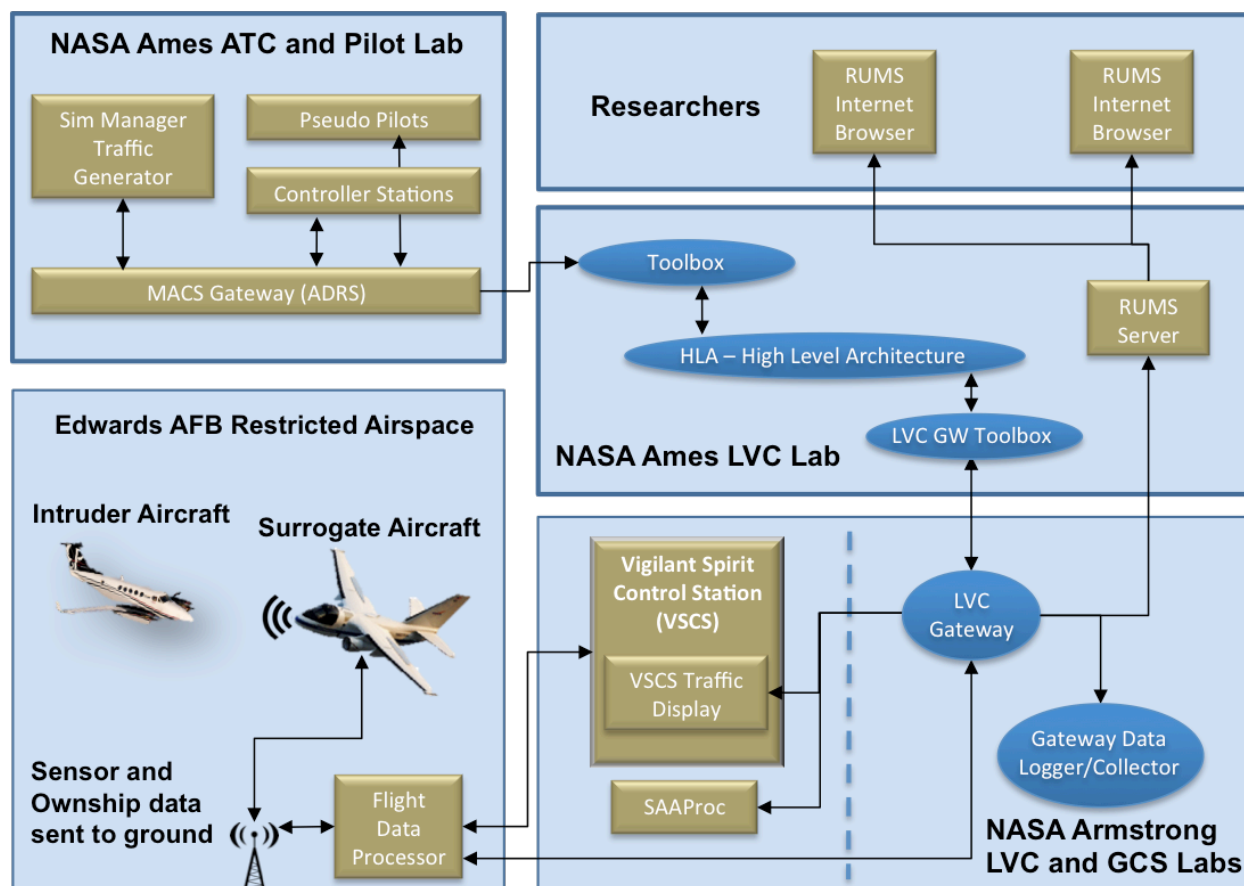American Institute of Aeronautics and Astronautics

Figure 2. High-level system connectivity for the upcoming UAS Flight Tests. *The UAS flight tests will incorporate multiple live aircraft merged with virtual background traffic and an ATC workstation emulation.*

B. LVC Software Components

This section provides background information on the software components used in the LVC instantiation as tested to support the configuration illustrated in Figure 2.

1. High Level Architecture Middleware

The LVC development team used a version of the IEEE 1516 standard Pitch portable Run Time Infrastructure High Level Architecture and Federation Object Model middleware to exchange information about the air traffic environment (aircraft state, flight plans, etc.) among the participants operating from distributed facilities.[7,8] The use of HLA provides an interface to well-defined air traffic data structures and promotes interoperability with simulation architectures using other middleware solutions such as AviationSimNet, Distributed Interactive Simulation, or Test and Training Enabling Architecture.[9,10,11] The HLA middleware ran at NASA Ames and served as the backbone for the LVC control capability, routing traffic information as specified in a configuration file. Simulation components were connected to the HLA via a Toolbox, which formats the messages as defined by the HLA interface.

2. LVC Gateway

The LVC Gateway provided connectivity to an external software component where connecting directly to the HLA environment was not desired. For example, the SAAProc sent advisory messages directly to the VSCS Traffic Display as shown in Figure 2. Instead of each connecting remotely to the HLA, an LVC Gateway was used to route local message traffic and provided a single connection from the remote facility to the HLA at Ames. Components connecting to the LVC Gateway published messages according to the LVC interface control document. Any number of sites can be added to the LVC environment by connecting additional LVC Gateways to the architecture. The design of the LVC Gateway is further broken out in the following section of this paper.

3. HLA Toolboxes

While the HLA has a well-defined message interface, each software component connecting to the LVC environment for a simulation may have its own, which may not be consistent. Toolboxes translated messages from software components to comply with the defined HLA interface. There were two primary reasons for the use of

4

American Institute of Aeronautics and Astronautics

Toolboxes instead of developing the interface directly in the component software: the software component may be commercial or government off-the-shelf (i.e. the development team does not control the software in order to implement the interface); and the software component may be used to connect to multiple different versions of middleware. The LVC environment utilized Toolboxes to connect to constructive target generators, the LVC Gateway, and flight simulators. The Toolboxes were designed to record and output the times messages are received, thus providing another physical location within the LVC network where data are time tagged.

4. Gateway Data Logger

The Gateway Data Logger process was developed to collect message timing and throughput data. The Gateway Data Logger connects directly to the LVC Gateway running in a distributed mode and stores the time an aircraft position update message was created and received by the LVC Gateway. The purpose for creating a separate process for data recording was to offload the data recording duties of the LVC Gateway, thus reducing run-time data processing and its software complexity. During data latency testing, the Gateway Data Logger files were converted into time-series data sets for analysis.

5. Gateway Data Collector

The Gateway Data Collector also recorded the content of messages passing through the LVC Gateway, however unlike the Gateway Data Logger, it can be used for system playback. The LVC message data were time-stamped and recorded in their native format. During playback the messages can be sent back into a connected system based on the time-stamp recording without any additional processing. The purpose was to simplify testing of modifications to LVC client components, easing the ability to perform regression-testing checks without having to run a complete LVC environment. The Gateway Data Collector also acted as a redundant data recording capability, which is critical for more complex distributed systems.

6. Remote User Monitoring System (RUMS)

In order to facilitate the monitoring of the data collection, the RUMS software processes connected to the LVC Gateway process and provided the ability to access and display live data via a web browser. The RUMS server connected to the LVC Gateway and handled the web browser data requests. RUMS was specifically developed to monitor a distributed air traffic simulation where aircraft state data originates independently in real-time from various networked subsystems. RUMS was also designed to display air traffic control target information and to provide a simple set of analysis and diagnostics tools for researchers.[12]

7. SAAProc

The SAAProc serves as a software message wrapper between the self-separation and collision avoidance algorithms and the LVC Gateway. Typically the SAA algorithms are intended to provide the UAS pilot with traffic advisories and possible maneuvers to maintain aircraft separation. Operationally the self-separation algorithm may be directly integrated into the prospective pilot display, whereas the collision avoidance algorithm would be resident on the aircraft. However in order to ease the testing of these concepts, integration of the algorithms and display was designed to communicate via the LVC Gateway.

8. Multi-Aircraft Control System, Aeronautical Data Link and Radar Simulator

The Multi-Aircraft Control System (MACS) provided multiple functionalities used for simulation, namely the ATC display environment, simulation traffic coordination, and constructive aircraft traffic position updates. The MACS Simulation Manager (SimMgr) receives the flight path, flight intent, and starting position for a set of aircraft from a simulation file. It then generated flight trajectories for these aircraft and provided the LVC environment with position updates (the MACS pilot displays are shown in Figure 3). For this series of tests, the SimMgr was run as a constructive aircraft data source, providing simulated aircraft data without pilot input. On the ATC side, MACS was configured to run the En Route Automation Modernization environment. The Aeronautical Data Link and Radar Simulator (ADRS) was a companion program to MACS. It translated, filtered, and transmitted messages to and from instances of both the MACS SimMgr and MACS ATC.[13,14,15] It should be noted that the MACS program did not directly interface with the LVC environment, but through the associated ADRS gateway (as shown in Figure 2). MACS and ADRS were developed at NASA Ames and have been used for many years for ATC simulation.[13,16]

Figure 3. MACS Pseudo Pilot Displays. *The Pseudo Pilot displays handle multiple aircraft and allow for basic maneuvering.*

9. Vigilant Spirit Control Station and Traffic Display

The Vigilant Spirit Control Station (VSCS) UAS simulator is a suite of software programs developed by the Air Force Research Laboratory to provide a virtual model of a UAS aircraft and its GCS. The VSCS simulator connected to the LVC and the rest of the simulation environment via the LVC Gateway, providing the intended flight path of the aircraft and ownship position update reports. The VSCS also contained a Traffic Display, providing the pilot with information intended to support three general categories: 1) situation awareness, 2) conflict detection, and 3) conflict resolution. The VSCS Traffic Display (shown in Figure 4) can also show resolution maneuvers and support "vector-planning". Vector-planning allows the pilot to test various horizontal or vertical vectors to help determine appropriate trajectories to avoid potential conflicts. Maneuver resolutions and vector-planning functionalities are facilitated by the SAA advisories supplied by SAAProc.
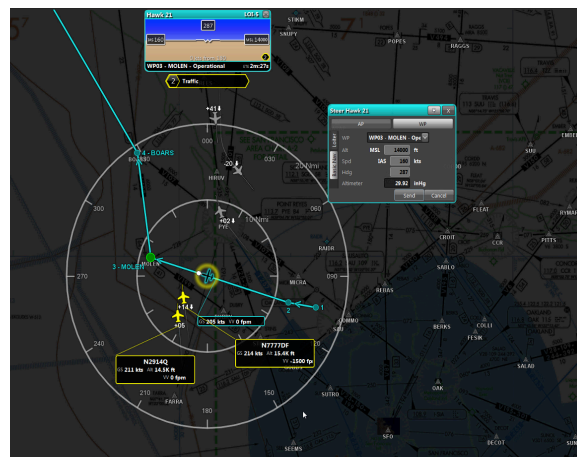

Figure 4. VSCS Traffic Display. *The display shows a centered ownship, with proximal traffic and advisories.*

C. LVC Gateway Design

HLA Middleware and the LVC Gateway constitute the core components of the LVC environment. The HLA LVC Gateway Toolbox interfaces with the LVC Gateway as a client. Test participants and simulation components connect as clients to the LVC Gateway's socket server. The LVC Gateway interface control document provides data structures that are a subset of the Multi-Purpose Interface (MPI) originally developed to facilitate communication between ADRS and external clients. The interface includes self-separation alert and resolution messages as well as trial planning messages that meet UAS research requirements for the traffic display. Upon initial connection, each client sends a handshake message to the LVC Gateway defining the message types to which the client would publish and subscribe.

The LVC clients connect to the LVC Gateway using MPI interface type on the server's Transmission Control Protocol (TCP) socket. The client is required to map its native data structure to the MPI interface. The LVC Gateway sends a periodic heartbeat message to each client to check the health of the connection. If the socket connection does not reply in a timely manner due to a client process crash, process disconnect (voluntary or involuntary), or socket failure, the LVC Gateway will send a delete message to the LVC clients for tracks associated with that client.

1. LVC Gateway Languages

The LVC Gateway software development was written using the C++ programming language and Object Oriented Design methodology enabled by the language paradigm. The Object Oriented abstraction concept was utilized for communication classes to capture specifics of different clients' interfaces as well as for configuring socket types as a client or server. The design can accommodate other protocols such as User Datagram Protocol, Multicast, or Loopback sockets.

2. LVC Gateway Class Diagram

The Unified Modeling Language class diagram of the LVC Gateway design is shown in Figure 5. The figure depicts the software architecture representing the relationship between the classes of the LVC Gateway. An Extensible Markup Language (XML) input configuration file is used to define the LVC Gateway configuration in term of interface types, interface socket types, and data structures that are transmitted by the LVC Gateway. The GW Client Manager class is responsible for the instantiation of all LVC Gateway components in the initialization phase of the code based on the XML configuration file. The GW Client Manager contains a collection of GW Client objects. In addition, it utilizes the observer/observable pattern where the *observable* is associated with clients that publish messages defined in the interface while the *observer* is associated with clients that subscribe to those messages.

The GW Client class is an abstract class that morphs into the MPI or ADRS interface types as defined by the XML configuration file. Each GW Client has multiple client and server type sockets. Each interface on the LVC Gateway side receives data messages such as Flight State, Flight Plan, Trajectory Intent, SAA and Trial Planning data from respective participants in the MPI data format. Those messages are time stamped, buffered, and then passed to the subscribing clients.
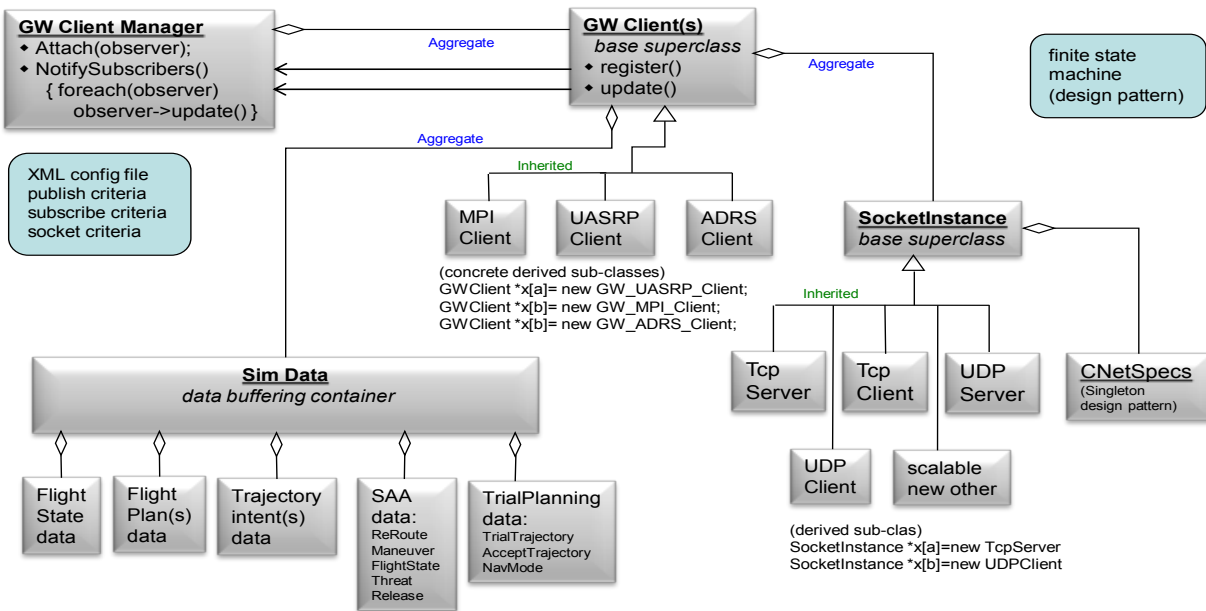
Figure 5. LVC Gateway Class Diagram. *The LVC Gateway is designed to support multiple types of inputs, allowing for scalability.*

III. Latency Testing

A. Test Objectives

One of the goals of the LVC environment was to provide a simulation infrastructure that emulates an operational air traffic control environment able to mix live and simulated air traffic data. Operationally, the maximum allowable latency is based on a combination of the surveillance source timing and the required time for processing and display at the facility. ATC Terminal facilities have a 1.0 second processing requirement, while En-route facilities have a 1.6 second processing requirement.[17] When combined with the radar sensor and communication timing this allows detection-to-display times of 2.2 seconds and 3.0 seconds for Terminal and En-route facilities, respectively.[6] The maximum acceptable generation and transmission time for ADS-B data is 2.5 seconds allowing for a total of 5.0 seconds for display in the cockpit.[4,5] These values provide the threshold for the LVC latency testing to measure against.

In order to inform the development of the future UAS flight test environment, the initial LVC environment characterization tests had two primary objectives:

1) Measure the latency of sending aircraft position updates from the source to the LVC Gateway
2) Measure the latency of sending aircraft position updates between LVC networked facilities

The first objective was designed to provide a comparison of the latency times to publish aircraft position updates among several potential data sources. The second provides the data to understand how long it takes to send an aircraft message to remote systems. In this way partial latency contributions between intermediate components can be used to build a unique LVC environment instantiation for a given set of requirements. These objectives, when applied together along different points and between different LVC environment configurations, provide a general understanding of the system in terms of its ability to transmit the appropriate data in a timely manner. Due to the anticipated need to synchronize data from each of the live, virtual, and constructive aircraft sources during testing, precise measurement of the latencies for these different air traffic inputs is critical.

B. Test Methodology and Materials

1. Test Design

The tests were designed to mimic potential simulation and flight test scenario system loads and system architectures. Aircraft data provided by virtual and constructive sources at NASA Ames (i.e. local sources) are distributed to remote facilities via the associated HLA Toolbox through a virtual private network (VPN) and to an LVC Gateway at the remote facility. Remote facilities provided either virtual flight simulator or live aircraft data to NASA Ames through the LVC Gateway, LVC Gateway Toolbox, and HLA. Refer to Figure 6 for a high-level diagram of the data flow. This is a subset of the full LVC environment that will be used in upcoming UAS flight-testing and shown in Figure 2.

Latency measurements were divided into two categories: 1) time required to send data from the source to a local LVC gateway, and 2) time required to send data between facilities. This allows designers of future LVC infrastructures to piece together simulation components over various facilities with some confidence of the latencies that can be expected. In order to collect data for both categories, the MACS Sim Manager at NASA Ames was run with a traffic scenario file that contained 50 aircraft. MACS was set to update the position of each aircraft once every second to mimic the data rate of ADS-B. This was intended to supply a nominal amount of background traffic typical for the planned UAS simulations. The scenario file was run for five minutes (after an initial ramp-up period, during which the flight plans for the aircraft were loaded into the MACS system). These times were written to a file for post processing by the Gateway Data Logger.

Figure 6. High-level view of the system under test for message latency characterization. *The Distributed Test Environment included LVC infrastructure that enabled connection of multiple data sources.*

To measure the latencies of sending aircraft data from a source to the local LVC environment, aircraft position message generation time was recorded by the source, and message receipt time was recorded by the local LVC process. For the MACS target generator and B747 flight simulator operating at NASA Ames, the time of aircraft state message receipt was logged by the LVC Gateway Toolbox. For live aircraft data operating at NASA Glenn and the Ikhana flight simulator at NASA Armstrong, the time the aircraft state message receipt was logged by the LVC Gateway (refer to Figure 6). To measure the latencies between facilities, the time the ADRS Toolbox at NASA Ames received the aircraft position messages from the MACS target generator and the time it was received by the LVC Gateway running at the remote facility were compared. In both cases, these times were collected and written to a file by the Gateway Data Logger.

It should be noted that latency measurements are only as accurate as the system time of the computer logging the data. At each of the facilities, the time server providing the current time to each of the computers used for testing was connected directly to a Global Positioning System device for truth time. Any variability in the a specific computer was measured by comparing time clocks between each computer and the computer running the LVC Gateway using a set of software scripts that determine the offset in the same manner Network Time Protocol offsets are calculated.[18] These offsets were factored into the calculated latency times.

2. Method and Materials

NASA Glenn's S-3B Viking aircraft served as the live asset for the characterization test. In order to transmit telemetry (and other data) from the aircraft to the ground, the UAS project developed the Aircraft Ground Station that served as a bridge between data from a live aircraft and the LVC environment. It received formatted telemetry data from a live aircraft at a 1 Hz rate via the prototype UAS datalink radio and relayed the data to the LVC
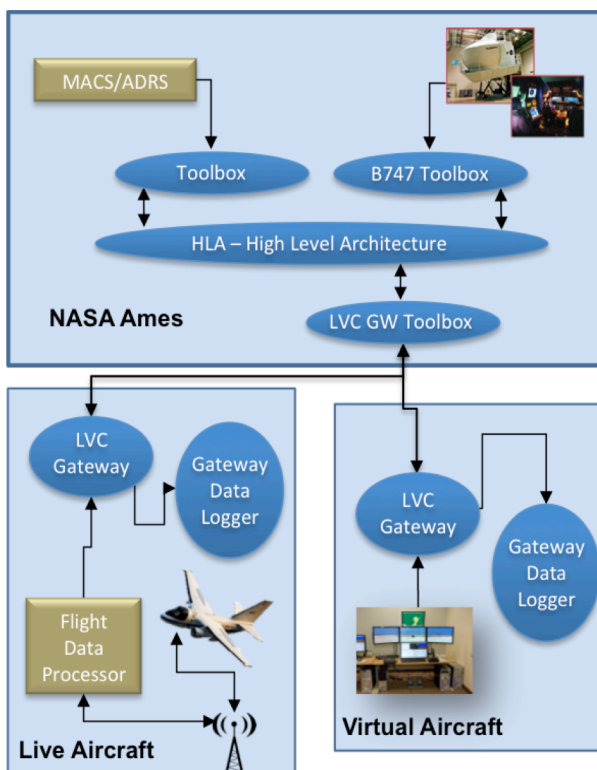
Gateway. The prototype radio was an early generation of the proposed unmanned aircraft Control and Non-Payload Communication System. Shalkhauser, et. al. provide a detailed description of the prototype radio and communication technologies used during communication flight testing.[19] Two programs were run to acquire and transmit telemetry data from the aircraft:

1. The Flight Data Aggregator program composed flight state messages from data monitored on the aircraft's MIL-STD 1553 and ARINC 429 data busses.
2. The LVC Interface program sent the messages to the Data Link radio for transmission to the Ground Station. In addition, the data messages from the aircraft were transmitted from the Ground Station to the Internet (and the LVC) via a 3G Cellular signal. This is not the operational concept for transmission, but the available solution during the prototype testing.

The Ikhana UAS Simulator at NASA Armstrong provided an interface to a Predator B (MQ-9) aircraft model. The fully functional simulator flew a virtual aircraft according to a pre-programmed flight path, allowing for dynamic maneuvering via the joystick interface and simulated instrument panel (and out the window view). The position reports of the virtual aircraft were sent to the LVC Gateway at a 5 Hz rate.

The B747 flight simulator at NASA Ames provided virtual aircraft state data during specific test configurations. The B747 is an FAA-certified "Level D" simulator[§] modeling all modes of airplane operation. The B747 transmitted the virtual aircraft position to the LVC at a 5 Hz rate via the B747 Toolbox and HLA.

MACS, as described above, provided an additional source of aircraft data. The MACS Sim Manager capability reads traffic scenario files that contain the initial position, speed, and altitude conditions of each aircraft specified in the file and flies them according to a prescribed flight path. The position of each aircraft was sent to the LVC via ADRS, the ADRS Toolbox and HLA at a 1 Hz rate.

Table 1 provides a list of the specific test runs.

Table 1. LVC Characterization Test Configurations

Type of Test	Test Title	Test Description
Aircraft Source Latency	MACS to LVC	During a standalone test, the MACS system at NASA Ames was connected to the LVC via the ADRS Toolbox and HLA. The fifty aircraft scenario file was run and the aircraft position updates were sent over to the LVC Gateway running at NASA Ames. The ADRS Toolbox recorded the timing data.
Aircraft Source Latency	B747 Flight Simulator to LVC	The B747 flight simulator was flown using a canned flight, connected to the LVC via the B747 Toolbox and HLA. During the flight, the aircraft position was sent to the LVC and recorded at the LVC Gateway Toolbox.
Aircraft Source Latency	Ikhana Simulator to LVC	The Ikhana flight simulator was flown using a canned flight, connected to the LVC via the LVC Gateway. During the flight, the aircraft position was sent to the LVC and recorded by the Gateway Data Logger.
Aircraft Source Latency	S-3B Viking to LVC	The S-3B was flown on June 24th 2013, over Northern Ohio. During the flight, the aircraft position was transmitted from the aircraft to the ground and the LVC network via a portable ground station that converted the signal from the aircraft into a Cellular signal. The cellular signal was then transmitted to the LVC over a 3G network with the time the position update reached the LVC Gateway recorded.
Distributed Network Latency	NASA Ames to NASA Armstrong	During the Ikhana simulator data collection, the MACS system at NASA Ames was connected to the LVC via the ADRS Toolbox and HLA. The fifty aircraft scenario file was run and the aircraft position updates were sent over to the LVC Gateway running at NASA Armstrong, where the timing data were recorded.

[§] A level "D" flight simulator provides a motion platform with six degrees of freedom, at least 150 by 40 degrees of a collimated out the window view, and realistic sounds and other visual and motion effects. At level "D" it can be logged as flight time.

Distributed Network Latency	NASA Ames to NASA Glenn	During a flight of the S-3B, the MACS system at NASA Ames was connected to the LVC via the ADRS Toolbox and HLA. The fifty aircraft scenario file was run and the aircraft position updates were sent over to the LVC Gateway running at NASA Glenn, where the timing data were recorded.

IV. Latency Results

A. Aircraft Source Latency

Table 2 lists the mean times and standard deviations of sending data from the various aircraft position data sources to the LVC environment. The two true flight simulators (the B747 and Ikhana simulators) had very similar and low latency times. This is expected since they were written to output data as very precise time intervals, with processing spread out over multiple computers. The MACS target generator had a larger average latency. MACS had a multi-purpose design, however, performing many functions simultaneously. The MACS software is being reviewed to determine the source of the message latency. The live aircraft position data from the S-3B has the greatest latency values. As described above, the data messages were transmitted from the aircraft to the ground via the prototype telemetry link and sent to the LVC environment network via a 3G cellular signal. In addition, the position data generation time was truncated, introducing up to one second of error. As a result, the average latency and variability were much greater than the virtual sources.

Table 2. Latency Times between data source and LVC.

Test Name	Average Latency to LVC (sec)	Standard Deviation (sec)	Max Latency (sec)
MACS to LVC	0.812	0.021	0.851
B747 to LVC	0.028	0.011	0.062
Ikhana Sim to LVC	0.022	0.002	0.097
S-3B to LVC	2.450	0.091	3.234

B. Distributed Network Latency

The observed latency times required to publish data between the distributed test facilities were consistent across both test runs. The values provided in Table 3 indicate slight increase in latency times sending data between NASA Ames and NASA Glenn over NASA Armstrong. There are two possible reasons for this; first the overall physical distance to NASA Glenn was greater than to NASA Armstrong. Second, the connection to NASA Armstrong was routed via the NASA Integrated Services Network, which handled all traffic within NASA. However, the NASA Glenn UAS Communication Lab had portable test equipment and therefore had an external connection to the Internet. Though the overall latency between the distributed facilities was quite low, the connection between the NASA Ames and NASA Glenn saw a few instances of large latency.

Table 3. Distributed Network Latency Times.

Test Name	Average Latency to Ames (sec)	Standard Deviation (sec)	Max Latency (sec)
NASA Armstrong to NASA Ames	0.102	0.034	0.177
NASA Glenn to NASA Ames	0.142	0.075	0.838

C. Latency Comparison

The original purpose for studying the data source latencies were to understand whether the data were usable for ATC emulation and whether data from different sources needed to be synchronized to better simulate coming from the same source. Figure 7 shows the average cumulative latencies calculated from each state data source to the LVC network at NASA Ames (the location of the ATC workstations used for simulation). The maximum allowable En-route and terminal latency thresholds are shown for reference (2.2 and 3.0 seconds respectively). The values represent the time the data is available for display to an ATC workstation, not the time the data were actually displayed. This is due to the mechanism of how the MACS ATC workstation selects targets to be drawn. So the display value can range from 0 to 0.5 seconds. The first bar represents the time latency for live aircraft.

American Institute of Aeronautics and Astronautics

As expected the publishing time accounts for the majority of the latency and goes beyond the available latency to be usable for Terminal radar emulation. In addition it is only 0.18 seconds less than the En-route threshold; so when added to the possible display latency based on using MACS, it is unsuitable for realistic simulation. It should be noted that the 3G Cellular transmission method for the live aircraft is not the candidate air to ground delivery mechanism for the UAS in the NAS project, but was tested to understand the existing capabilities. The candidate air to ground system will be tested once the system is developed. The second and third bars represent the Ikhana and B747 latencies. Both are well below the Terminal and Host latency thresholds, even with the display latency added. The

Figure 7. Cumulative Latency from Source to LVC at NASA Ames. *A relative comparison of the total time to send an aircraft target from the source to the LVC lab at NASA Ames. This is plotted against the required en route and Terminal delay times.*

last bar represents the MACS SimMgr latencies. The MACS publishing time had greater than expected latencies. The LVC development team is investigating the root cause at this time. Even with this extra delay and accounting for the MACS DSR display time, these data were well below the required latency thresholds and well suited to support realistic simulation.

Comparing the MACS target generator against the Ikhana and B747 flight simulators, the difference in the overall latency is enough to warrant some mitigation to better synchronize the tracks updates, if required. In particular for the UAS in the NAS project, if the data are being used for high fidelity SAA research between aircraft, the time taken to send the data from the simulator to the algorithm may need to be increased so the time and location of the tracks correspond to the data from MACS. These mitigations need to be evaluated and applied on a case-by-case basis.

V. Conclusions and Next Steps

The LVC message latency tests demonstrated that the virtual aircraft flight simulators and target generators are able to supply data to the LVC system within the time required by operational En-route and Terminal systems. As tested with a cellular connection, the latency in the connection between the live aircraft and the LVC system was greater than what is required for ATC simulation. This is not a concern, however, as the 3G Cellular system used for this connection is not the final proposed solution, but a prototype provided to achieve minimum early functionality for testing. It provided valuable insight into the design of a live data connection for future test iterations.

As LVC environment development continues, analyses of other test conditions are planned. These include expanding the number of client data sources and increasing the overall traffic levels to determine where message throughput is impacted. Testing of the candidate UAS Communication radio hardware will also be tested to evaluate its relative performance and capture latency data. In addition, each simulation or flight test environment is unique with respect to its systems, network, and data flow. The overall LVC system, therefore, will be characterized prior to testing in order to understand how changes to the system impact performance.

References

[1] DoD: "Modeling and Simulation Master Plan", DoD 5000.59P, Oct 1995

[2] Henninger, Amy E., Cutts, Dannie, Loper, Margaret, et al, "Live Virtual Constructive Architecture Roadmap (LVCAR) Final Report", Institute for Defense Analysis, Sept, 2008

[3] Bezdek, W. J., Maleport, J., Olshon, R., "Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations," AIAA 2008-7090, AIAA Modeling and Simulation Technologies Conference and Exhibit, August 2008
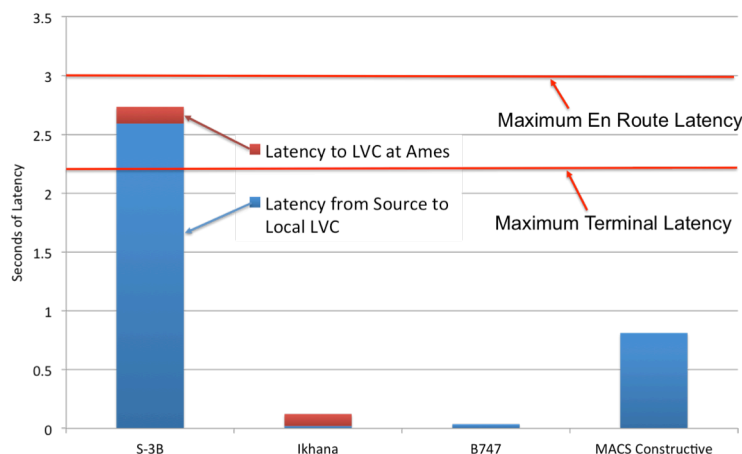
[4] RTCA: "Minimum Operational Performance Standards (MOPS) for Aircraft Surveillance Applications (ASA) System," RTCA DO-338, Dec 13, 2011

[5] FAA: "Automatic Dependent Surveillance – Broadcast (ADS-B) Out Performance Requirements to Support Air Traffic Control (ATC) Service"; Final Rule, Date: 5/28/2010, Federal Register Volume 75, Number 103, Document ID: FAA-2007-29305-0289, Washington, DC.

[6] FAA: NAS-SR-1000, National Airspace System (NAS) System Requirements, Department of Transportation Federal Aviation Administration, January 2005

[7] Software website: http://www.pitch.se/prti, August 2013

[8] Lutz, R., LeSueur, K., Fast, P., Graeff, R., Simolte, A., Rutledge, J., and Motti, R., "A Persistent LVC Simulation Environment for UAS Airspace Integration," The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), December 2010.

[9] AviationSimNet Standards Working Group, "AviationSimNet Specification", Version 2.2, MITRE Corporation, June 2010

[10] "IEEE Standard for Distributed Interactive Simulation – Application Protocols," IEEE 1278.1-2012, Dec 19 2012

[11] Edward T. Powell, Jason Lucas, Kurt Lessmann, and George J. Rumford, "The Test and Training Enabling Architecture (TENA) 2002 Overview and Meta-Model," published in the Proceedings of the Summer 2003 European Simulation Interoperability Workshop, 03E-SIW-041, found at http://www.sisostds.org/

[12] Solar, G. Jovic, S., Murphy, J., "RUMS - Realtime Visualization and Evaluation of Live, Virtual, Constructive Simulation Data," In Press, AIAA Infotech@Aerospace Conference, January 2015.

[13] Prevot, T., "Exploring the Many Perspectives of Distributed Air Traffic Management: The Multi Aircraft Control System (MACS)," International Conference on Human-Computer Interaction Aeronautics, HCI-Aero 202, 23-25 October 2002, Massachusetts Institute of Technology, Cambridge, MA.

[14] Prevot, T., Smith, N., Palmer, E., Mercer, J., Lee, P., Homola, J., Callantine, T., "The Airspace Operations Laboratory (AOL) at NASA Ames Research Center," AIAA 2006-6112, AIAA Modeling and Simulation Technologies Conference, August 2006.

[15] Prevot, T., and Mercer, J., "MACS: A Simulation Platform for Today's and Tomorrow's Air Traffic Operations," AIAA-2007-6556, AIAA Modeling and Simulation Technologies Conference, August 2007.

[16] Mercer, J., Prevot, T., Jacoby, R., Globus, A., Homola, J., "Studying NextGen Concepts with the Multi-Aircraft Control System," AIAA 2008-7026, AIAA Modeling and Simulation Technologies Conference, August 2008.

[17] FAA: "NAS System Specification, Functional and Performance Requirement for the National Airspace System, General, Volume 1," FAA, Department of Transportation, NAS-SS-1000, 15 April 1995.

[18] Software website: http://www.ntp.org, August 2014

[19] Shalkhauser, K., Griner, J., Kerczewski, R., "Flight Tests of First Generation Prototype CNPC Radio," International Civil Aviation Organization (ICAO), Aeronautical Communications Panel (ACP) 29th Meeting of Working Group F, 05-12 September, 2013