ARL-TR-8960 ● May 2020

Experimental Evaluation of Mockets Communication Protocol for Integrated Sensor Architecture (ISA)

by Jade Freeman, Timothy Gregory, Michael Lee, James Michaelis, and Theron Trout

NOTICES

Disclaimers

ARL-TR-8960 ● May 2020

Experimental Evaluation of Mockets Communication Protocol for Integrated Sensor Architecture (ISA)

Jade Freeman, Timothy Gregory, Michael Lee, and James Michaelis
Computational Information Sciences Directorate, CCDC Army Research Laboratory

Theron Trout
Stormfish Scientific Corporation

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
May 2020	Technical Report	April 2019–March 2020

4. TITLE AND SUBTITLE

Experimental Evaluation of Mockets Communication Protocol for Integrated Sensor Architecture (ISA)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jade Freeman, Timothy Gregory, Michael Lee, James Michaelis, and Theron Trout

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CCDC Army Research Laboratory
ATTN: FCDD-RLC-IB
2800 Powder Mill Rd
Adelphi, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-8960

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

ORCID IDs: Jade Freeman, 0000-0003-4085-2235; James Michaelis, 0000-0003-3732-2145

14. ABSTRACT

Successful convergence of cross-domain capabilities is dependent on rapid and continuous interoperability and integration across joint military services, interagency, and multinational partners. However, tactical networking environments are often characterized by limited bandwidth and unreliable connectivity due to their wireless ad hoc nature where the performance issues include frequent disconnections, congestions, packet losses, and data backlog in the application and network queues, resulting in increased latency in the delivery of the data. In particular, the applications using the traditional Transmission Control Protocol/Internet Protocol suite designed for wired networks would suffer from inadequate communication performance. The potential consequence can be a missed opportunity or an action based on outdated information. Mockets (Mobile Sockets) communications protocol is designed to ensure the timely and reliable delivery of data in tactical wireless environments. This report provides qualitative analyses of Mockets-enabled communications under various network environments. The experimentation focuses on the Mockets transport protocol that supports transmission and reception of the data between US Army's Integrated Sensor Architecture servers and clients on timeliness and robustness of data transport. Our results show that the Mockets protocol improves message transport performance under the constrained network environment.

15. SUBJECT TERMS

Mockets, TCP, ISA, wireless, network, tactical, communication, performance

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Jade Freeman
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified	UU	24	(301) 394-4962

Contents

List of Figures

List of Tables

1. Introduction

One of the underpinning capabilities in US Army's Multi-Domain Operation (MDO) is rapid convergence of cross-domain capabilities. *The U.S. Army in Multi-Domain Operations, 2028*[1] recognizes the potentials for Soldiers and leaders entering operations are enhanced and maximized by man-machine interface, enabled by artificial intelligence and high-speed data processing, and human–decision making is improved in both speed and accuracy. Further, interoperability across joint military services, interagency, and multinational partners is a key element to executing MDO. Future operations against a near-peer threat require the joint force to conduct continuous and rapid integration of multi-domain capabilities to gain cross-domain overmatch at decisive spaces.[1] Multi-domain interoperability with joint forces and coalition partners require data collection and rapid dissemination within minutes for analysis to exploit brief window of enemy deterrence, defeat, and destruction while minimizing the cost to the friendly forces.

Despite such requirements, tactical networking environments are often characterized by limited bandwidth and unreliable connectivity due to their wireless ad hoc nature.[2] Systems and applications linking sensors in support of joint force operations can produce high-volume data, and such dissemination of information across domains and echelons is constrained by the networking environment. A potential consequence can be a missed opportunity or an action based on outdated information.

Mockets (Mobile Sockets) communications libraries are designed to ensure the timely and reliable delivery of data in tactical wireless environments. The performance issues in wireless networks include frequent disconnections, congestions, packet losses, and data often accumulates in the application and/or network queues, resulting in increased latency in the delivery of the data. Thus, applications using the traditional Transmission Control Protocol (TCP)/IP suite designed for wired networks would suffer from inadequate communication performance. The Integrated Sensor Architecture (ISA) controller is one of such TCP-based applications.

The primary objective of this study is to evaluate the performance of the ISA controller with Mockets-enabled communications under various experimentation environments. The results were analyzed using three different metrics: completeness, latency, and throughput. This report builds on the previous work,[3] which compared the performance of communications protocols for dissemination of position information and focused on the dissemination of sensor data. Mockets protocol is evaluated and compared to TCP protocol in a traditional wired networking environment (Ethernet local area network [LAN]), which provides the baseline results, and then in the Extensible Mobile Ad hoc Networking Emulator (EMANE). Further, the protocols were evaluated in wireless settings in the context of tactical edge networks with significant constraint in bandwidth affecting timeliness and reliability.

The main contribution of the experimentation and analyses is assessing the performance improvement gained on ISA transmission and reception of data between ISA-enabled servers and clients by using the Mockets transport protocol under a variety of network environment settings.

2. Integrated Sensor Architecture

The ISA, a US Army Service-Oriented Architecture, is an interoperability solution that allows for the sharing of information between sensors and systems in a dynamic tactical environment, enabling information exchange between sensors and other systems. ISA enables Army sensors and systems to readily integrate into an existing network and dynamically share information and capabilities to improve situational awareness in a battlefield environment.[4] ISA provides information on available devices using a publish/subscribe mechanism in which a subscriber can register its interest in receiving information of multiple generic types. When devices are available, a configuration file describing the device capabilities is published to the interested subscribers.[4]

An ISA is not a physical layout, but a network construction. It allows sensors to communicate needs or capabilities to other ISA-enabled devices and dynamically locate other ISA-enabled devices within the area of operation. Dynamic discovery programming lets devices identify common protocols while communication, creating a net-centric integration of systems, opens pathways between devices regardless of service or platform.[5]

3. Mockets

Mockets is a novel communications library specifically designed for wireless networking scenarios to improve communications in Mobile Ad Hoc Network environments. The design and implementation of Mockets was motivated by the needs of tactical military information networks, which are typically wireless and ad hoc with low bandwidth, intermittent connectivity, and variable latency. The initial implementation of Mockets was completed for use by the US Army Combat Capabilities Development Command (CCDC) Army Research Laboratory (ARL) as part of the Warrior's Edge initiative of the Horizontal Fusion Portfolio's Quantum Leap demonstrations.[6]

All the messages exchanged by the Mockets communication protocol are encapsulated in User Diagram Protocol (UDP) packets.[7] Reliability and stream abstractions are provided by Mockets on top of the unreliable UDP packet delivery service. Communications between two unicast datagram Mockets are established by connecting an active Mockets to a passive one listening on a peer that, apart from the case of explicit endpoint migration commands issued at the application level, will not change during the entire communication. On the server side, the communication is not established on the same port on which the server application listens for an incoming connection, but on a new, system-assigned port.

Implementing Mockets protocol provides several advantages, including phased integration and portability. Mockets operates on any platform supporting a TCP/IP stack, regardless of the underlying hardware and operating system. Mockets is implemented in C++, with bindings for Java and C#. Mockets also provides a TCP-compatible application programming interface that allows commercial off-the-shelf (COTS) applications to be gradually ported to the tactical environment, supporting and facilitating a phased transition process.[8]

4. Previous Work

Several research efforts on Mockets protocols are aimed at evaluating efficiency and robustness of communications in tactical networks and other highly dynamic wireless networking environments. The Air Force Research Laboratory experimented using an ARC-231 radio communicating with a ground-based PSC-5 tactical radio link.[8] During the experiment, the aircraft transmitted a series of Joint Tactical Information Distribution System (JTIDS) messages and servicing queries from the ground client. In total, there were 113 JTIDS messages and 293 query results. As the data transfer was highly asymmetrical, DAMA, or demand-assigned multiple access, assigned most of the communication channel to the ground station, effectively preempting channel use from the aircraft. This caused poor performance using COTS TCP implementations because of timeouts in acknowledgment transmissions from the aircraft to the ground station. However, adapting Mockets resulted in a performance improvement 8 times more than achieved with a half-duplex link.

Subsequently, CCDC Army Research Laboratory replicated this scenario in a controlled laboratory environment using a hardware radio emulator (i.e., EMANE). The results collected in the laboratory, consistent with those achieved in the field, showed that Mockets performed better than TCP. Most other transport protocols used acknowledgment-based clocking and congestion control mechanisms similar to TCP and therefore showed similar results.[9]

5. Experimentation Design and Setup

Our experiment builds on the previous scenarios in a controlled environment as well as outdoor environments. During the experiment, the data were captured while they were transmitted by ISA applications between two systems (Fig. 1). We used the Open Standard for Unattended Sensors (OSUS) messages from unattended ground sensors (UGS), collected at the Storm Force '18 exercise, Bogue Field, North Carolina. The data contain 6,134 event messages from the UGS, including Tactical Remote Sensor Systems, Battlefield Anti-Intrusion System, and Expendable UGS over a period of 24 h. The data format is the OSUS, which is a Java- and Open Service Gateway Initiative (OSGi)-based software interoperability architecture for UGS controllers. The OSUS messages from these sensors log were played back, sent to a system with ISA server that would act as a sender and transmit those messages to a client, a receiving system with ISA. The network traffic between the systems was captured via TCPDUMP application.

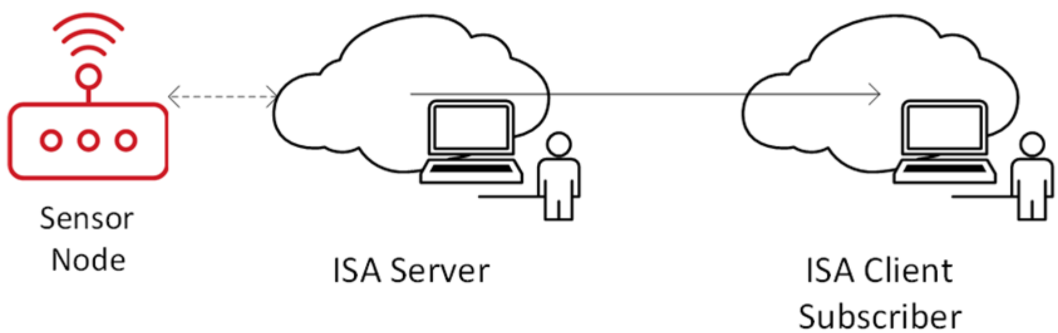

Fig. 1 Setup of experimentation

The tests were performed using the most recent release of Mockets as of October 2019. The experiments were set up using two machines running Intel Core i7 Ubuntu OS 18.04.2 LTS. The machines were connected through Ethernet connection with EMANE, simulating unreliable connectivity and TrellisWare (TW-400 CUB) tactical radios of TSM waveform, with specification of 8 Mbps IP throughput per channel with a network coverage of 26-mile line of sight per network hop.

We measured the performance of the dissemination and delivery of messages via ISA between TCP and Mockets protocols. The default communication protocol is TCP, and ISA needed to be commanded to use Mockets protocol. The experiments were conducted in four network environments: Ethernet bandwidth channel, Ethernet channel with noise, and a wireless channel with and without obstruction. Table 1 lists the design details of experiment. The EMANE effect was set at latency of 40 ms, packet loss rate of 10%, and a jitter of 10 ms. Figure 2 shows the environment settings for wireless network.

Table 1 Design of experimentation

Experiment	Location	Protocol
Control (Ethernet)	Laboratory	TCP
Control (Ethernet)	Laboratory	Mockets
EMANE (Ethernet)	Laboratory	TCP
EMANE (Ethernet)	Laboratory	Mockets
Wireless (radio)	Woods	TCP
Wireless (radio)	Woods	Mockets
Wireless (radio)	Open road	TCP
Wireless (radio)	Open road	Mockets

A. **B.**

Fig. 2 Wireless experimental setting: a) woods and b) open road; distance of 0.25 km

6. Analysis and Results

The laboratory experiments conducted on Ethernet wired setting showed that TCP and Mockets protocols were nearly identical in performances, as shown in Table 2. Under the stable wired network connection, neither protocol provided any better advantage in transfer speed or throughput capacity. When the EMANE effect was applied to the network, the performance of message transfer suffered for both protocols compared with the earlier results without any EMANE effect (Table 3). Both protocols suffered a few losses of messages, but the count is negligible (0.39% and 0.05% for TCP and Mockets, respectively). The Mockets protocol seemed to perform slightly better than TCP for having fewer extreme latencies and throughput values but not in any notable magnitude.

Table 2 Wired connection experiment in laboratory setting

Protocol		Ethernet		EMANE	
		TCP	Mockets	TCP	Mockets
N Messages		6,134	6,134	6,134	6,131
% Dropped messages		0%	0%	0.39% (n = 24)	0.05% (n = 3[a])
Latency (ms)	Minimum	0.66	0.55	31.19	31.05
	25th Percentile	1.80	1.77	38.96	38.41
	Median	2.79	2.78	45.13	44.01
	75th Percentile	3.82	3.79	50.74	49.66
	Maximum	4.99	7.51	21,650.38	1,813.60
	Mean	2.81	2.77	201.57	66.37
	Standard deviation	1.16	1.17	1,072.85	86.45
Throughput (bpms)	Minimum	87.84	57.95	0.10	0.24
	25th Percentile	114.99	115.58	8.64	8.84
	Median	157.04	158.20	9.73	9.96
	75th Percentile	243.20	248.57	8.92	9.56
	Maximum	668.13	810.04	11.27	11.41
	Mean	195.29	201.56	14.178	14.17
	Standard deviation	106.00	116.13	3.68	2.95

[a] Two messages unlogged in ISA and one message unsent

Table 3 Wireless connection experiment

Protocol		Woods		Road	
		TCP	Mockets	TCP	Mockets
N Messages		6,134	6,084[a]	6,134	6,134[b]
% Dropped messages		0	0.815%[a]	0%	0%
Latency (ms)	Minimum	38.31	24.83	10.32	0.025
	25th Percentile	69.11	55.90	34.18	23.21
	Median	76.73	63.44	43.08	32.88
	75th Percentile	98.00	82.69	61.91	50.34
	Maximum	48,427.59	5,067.65	236.29	311.45
	Mean	120.85	83.45	50.81	45.58
	Standard deviation	1117.47	140.15	24.13	37.08
Throughput (bpms)	Minimum	0.01	0.09	1.84	1.42
	25th Percentile	4.48	5.31	7.12	8.70
	Median	5.72	6.91	10.18	13.35
	75th Percentile	6.35	7.85	12.87	18.85
	Maximum	11.56	17.88	42.16	17,735.91
	Mean	5.45	6.62	10.43	22.46
	Standard deviation	1.28	2.33	4.46	244.69

[a] 5 unsent+ 45 dropped
[b] n = 82 (1.34%) Negative Latency; Average Latency = –5.6 ms

The results from the wireless experiments confirm the overall degraded performance on the network transport performance in comparison with Ethernet transport for both protocols. For the experiments in the woods setting, the lack of clear line of sights affected the reliability of the transport between the server and client that resulted in lower performance compared to the wired setting, as expected. However, the Mockets protocol suffered from dropped messages during the protocol. This fraction includes, out of a total 6,134 messages logged in ISA, 5 messages that initially failed to send to the client and 45 messages dropped during the transport and not received by the client. However, the proportion of dropped messages was less than 1% of the total messages, indicating the impact was negligible.

The results on latency and throughput using wireless network in the woods showed that the transport via Mockets performed better than TCP overall. As shown in Table 3, the shorter latency and higher throughput were observed with Mockets in comparison to TCP protocol. The median and mean latencies from Mockets were 63.44 and 83.45 ms, respectively, whereas those from TCP median and mean latencies were 76.73 and 120.85 ms, respectively. Also, the median and mean throughputs from Mockets were 6.91 and 6.62 bpms, respectively, whereas those from TCP were 5.52 and 5.45 bpms, respectively. These results indicate shorter delivery time and higher throughput in message transport between sender and client on Mockets protocol. The largest latency was observed at 48,427.59 ms for TCP while the largest latency was observed at 5,067.65 ms for Mockets. Similarly, the largest throughput was observed at 11.56 bpms for TCP while the largest latency was observed at 17.88 bpms for Mockets, indicating higher throughput performance on Mockets.

Wireless experiments on the road also indicated better latency performance on Mockets. The median and mean latencies on Mockets were 32.88 and 45.58 ms, respectively, compared to the median and mean latencies on TCP of 43.08 and 50.81 ms, respectively. Further, the throughputs were higher on Mockets where the median and mean were 13.35 and 22.46 bpms, respectively, and those for TCP were 10.18 and 10.43 bpms, respectively.

In this analysis, some results were removed due to the issues where the message sent timestamps were generated an earlier time than the corresponding message received timestamps. Even though time synchronization was implemented prior to running each experiment, a few message transfers were randomly affected by slippage of Network Time Protocol (NPT) time synch. For an unknown reason, the wireless experiment on the road had a negligible fraction (~1%) of results affected by this issue.

7. Probability Densities and Statistical Testing

From the visual inspections of the probability and cumulative density plots in Figs. 3–6, one can see the consistent results in latencies and throughput between TCP and Mockets. The central location of the probability densities is indicated at the median value of the measurements in dotted lines in the probability density graphs. The maximum distance between two cumulative distributions is indicated in dotted lines in the Empirical Cumulative Distribution Function (ECDF) graphs.

While the density and ECDF overlap in the Ethernet experiment (Fig. 3), EMANE experiments (Fig. 4) exhibited slight shifts of Mockets from TCP, to the left for latencies and to the right for throughput in densities. The larger shifts are seen in the wireless experiments (Figs. 5 and 6), especially from the woods setting experiment (Fig. 5). Therefore, latency measurements are seen to be distributed lower for Mockets than TCP from the EMANE experiment and the wireless experiments. Throughput measurements are seen to be distributed higher for Mockets than TCP from the EMANE experiment and the wireless experiments.

To test the difference in the quantified performances between TCP and Mockets, the Kolmogorov–Smirnov and Mann–Whitney U-tests were used. Both methods are nonparametric tests that assess if two samples have been obtained from a common distribution. Kolmogorov–Smirnov tests whether two underlying one-dimensional probability distributions differ by measuring the difference in location and shape of cumulative distributions of two samples. The presence of ties in the data can affect the accuracy of p-value, a large sample such as in our datasets can provide a good approximation. ECDF provides the proportion of scores that are less than or equal to each score.

Kolmogorov–Smirnov does not test for the location shift. However, under the similar shapes in distributions, the Mann–Whitney U test can be used to determine the location shift in distribution. In other words, the alternative hypothesis is that there is higher probability to obtain larger or smaller values in one sample compared to the other. The Mann–Whitney U-test measures the discrepancy between the mean ranks of the groups.

Table 4 provides the summary of the distribution tests on latency and throughput data from TCP and Mockets. For the latency and throughput measurements, all the experiments except for the Ethernet setting showed strong support against TCP and Mockets having the same underlying distribution. In particular, the wireless experiment from the woods has the largest Kolmogorov–Smirnov's D statistics, 0.4242 and 0.4251 for the latency and the throughput, respectively, indicating the largest magnitude of difference between the two distributions. Similarly, Mann–Whitney's U tests showed strong support against TCP and Mockets having the same underlying distribution for all except for the Ethernet setting.

Table 4 Kolmogorov–Smirnov and Mann–Whitney distribution tests

Experiment		Latency		Throughput	
		Kolmogorov–Smirnov	**Mann–Whitney**	**Kolmogorov–Smirnov**	**Mann–Whitney**
Wired	Ethernet	D = 0.0189 p-value = 0.2227	W = 19164719 p-value = 0.0729	D = 0.0184 p-value = 0.249	W = 18461973 p-value = 0.07352
	EMANE	D = 0.1383 p-value < 2.2e-16	W = 20277247 p-value = 2.4e-15	D = 0.1383 p-value < 2.2e-16	W = 17181163 p-value = 2.3e-15
Wireless	Woods	D = 0.4242 p-value < 2.2e-16	W = 26384053 p-value < 2.2e-16	D = 0.4251 p-value < 2.2e-16	W = 10955895 p-value < 2.2e-16
	Road	D = 0.2908 p-value < 2.2e-16	W = 24049394 p-value < 2.2e-16	D = 0.2919 p-value < 2.2e-16	W = 13069318 p-value < 2.2e-16

In Figs. 3–6, the TCP and Mockets distributions have the similar shapes as seen in probability density plots. This indicates the location shift in the distribution can be inferred from Mann–Whitney's U-test in determining whether the measurements are distributed at the higher level in comparisons between TCP and Mockets. TCP and Mockets measurements from the Ethernet experiment were close in distributions as shown in Fig. 3, as Mann–Whitney's test did not present strong support against distribution differences. However, in the EMANE and wireless experiments, the distributions of latency measurements from Mockets were lower than those from TCP, while the distributions of throughput measurements from Mockets were higher than those from TCP.

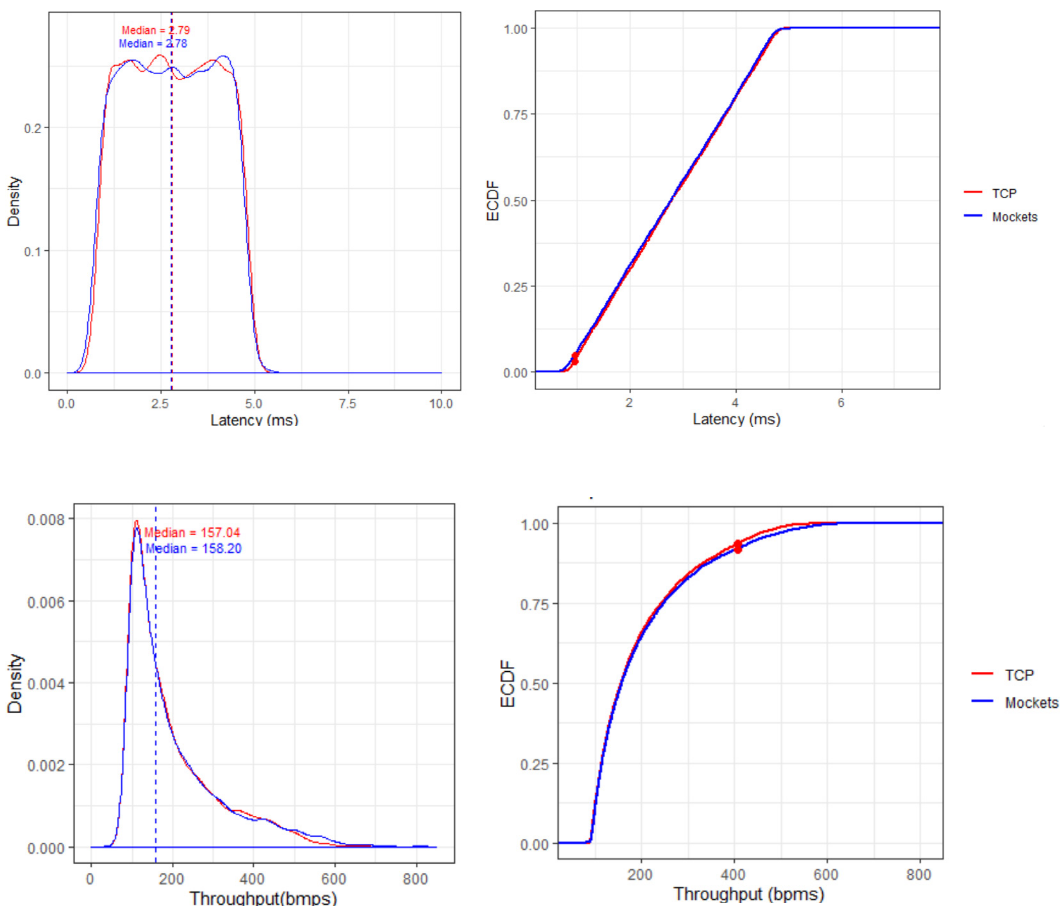

Fig. 3 Ethernet distribution functions

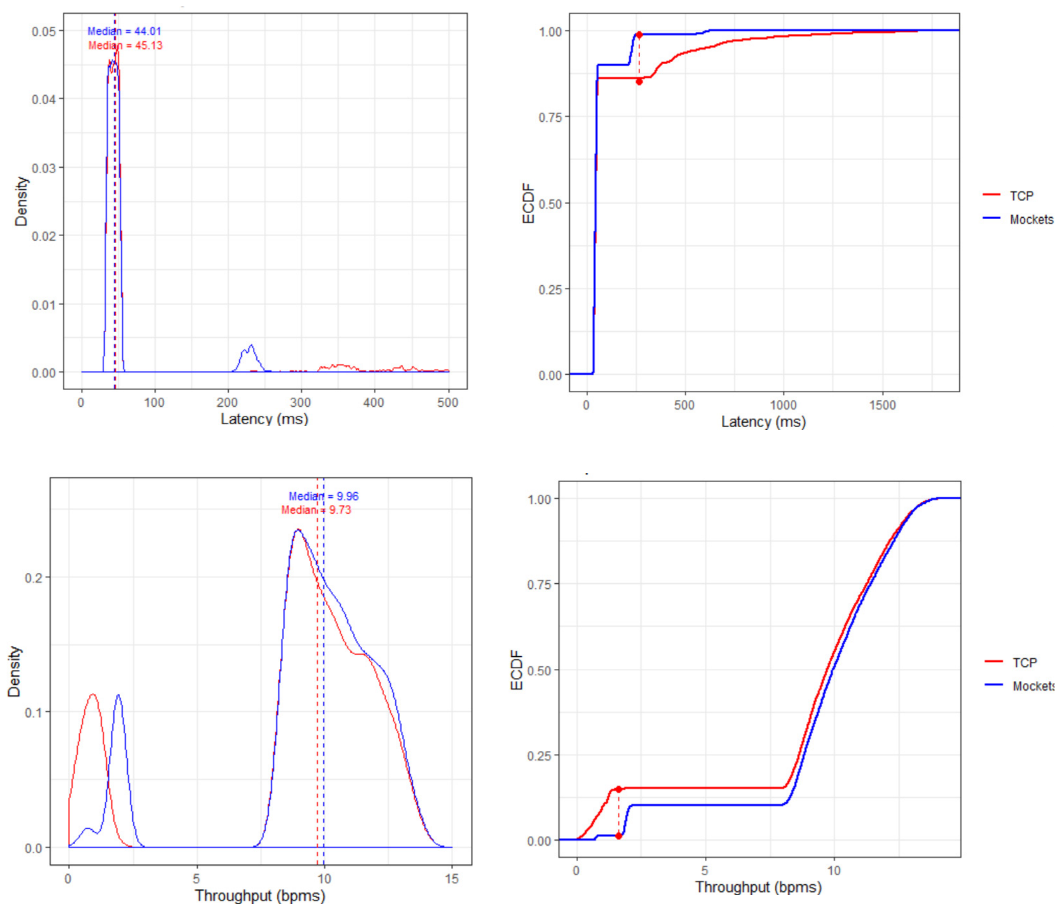

Fig. 4 EMANE distribution functions

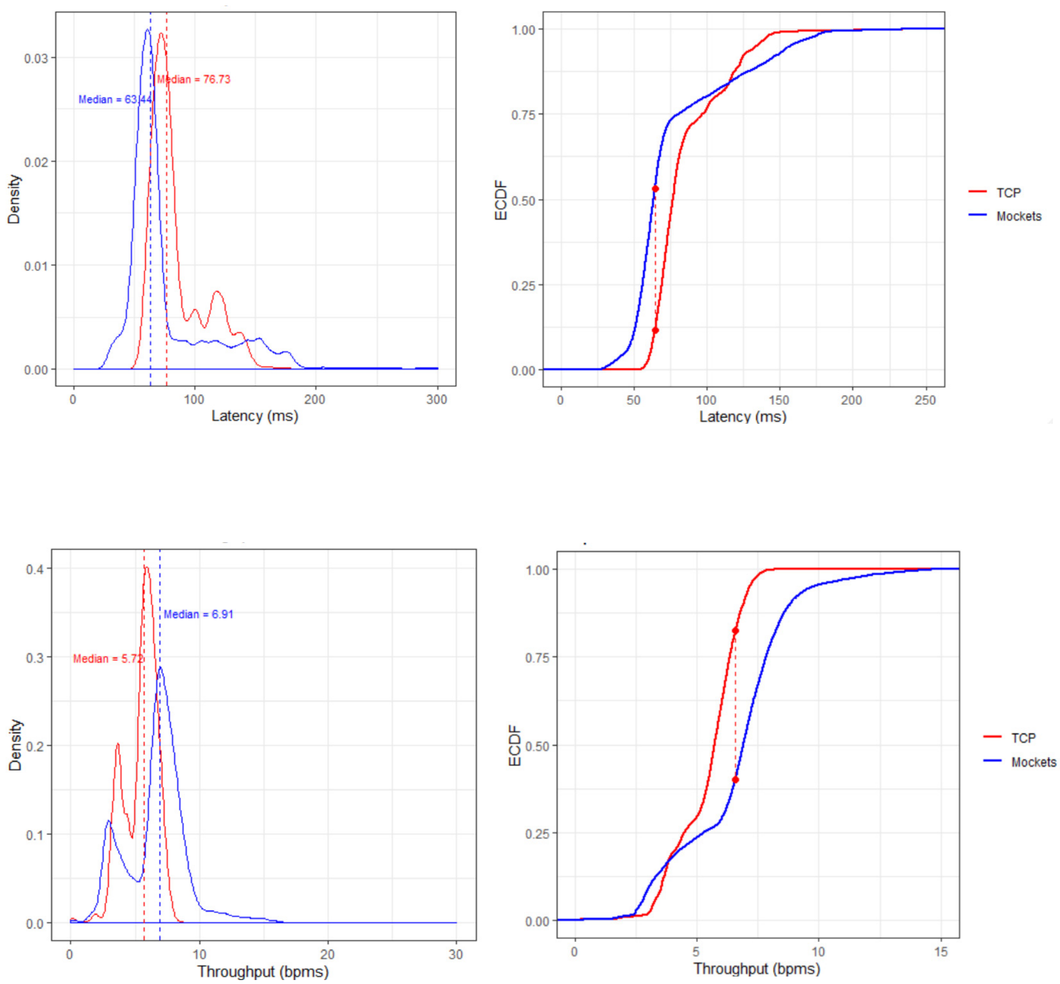

Fig. 5 Wireless woods distribution functions

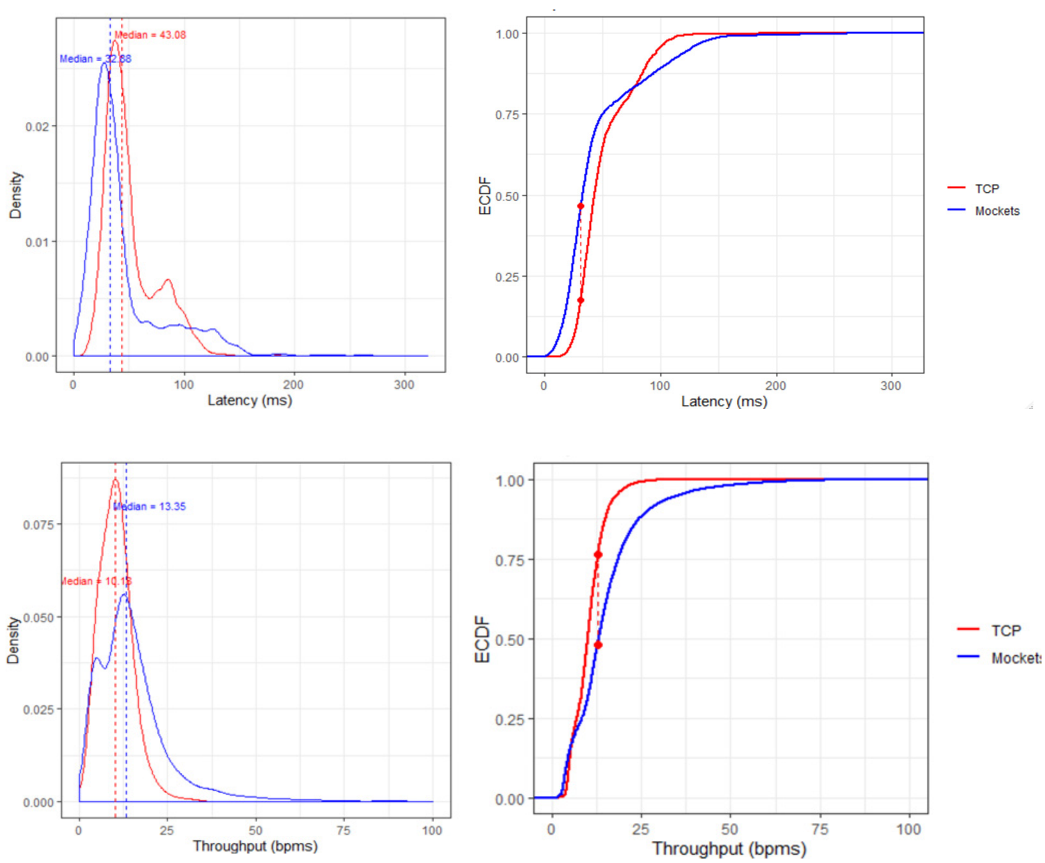

Fig. 6 Wireless road distribution functions

8. Additional Functionalities in Mockets

As mentioned previously, Mockets addresses specific challenges including the need to operate on a mobile ad hoc network (where TCP does not perform optimally). Mockets provides a mechanism to detect connection loss, allows applications to monitor network performance, provides flexible buffering, and supports policy-based control over application bandwidth utilization.

Mockets can measure the network condition in terms of available bandwidth along the communication path, round trip time, packet loss rate, and peer reachability.[8] This allows an application using Mockets to adapt to changes in the network environment by tuning several communication parameters. Prioritization functionality can dynamically change the priority and maximum lifetime of messages. By tagging messages, applications can separate message flows and perform group operations on a specific type of messages, allowing applications to use the functionalities of message cancellation and replacement. As an example, an application that faces a congested network situation or limited bandwidth can decide to cancel all the messages of a specific type. A call to cancel will cause all the messages marked with the specified tag to be deleted from the transmit queues. Further, when a new message is available, the tagged messages in the specific flow are cancelled from the transmit queues and the new message is enqueued.

At the time of this report, these features are not fully available for ISA-Mockets integration. A previous Mockets evaluation on the Joint Battlespace Infosphere (JBI) system took advantage of its unique message replacement capability, under a simulated scenario involving Blue Force Tracking and found message replacement as the feature that would most benefit applications with timeliness constraints in the JBI system.[2]

Lastly, the Mockets permits a mobile agent to easily move all its network connections. When the Mockets detects the changes in the network layer address, the applications are moved without forcing the application to shut down and reopen the network connections. Mobility support in Mockets permits a communication endpoint to be moved from one node to another without interrupting the communication session.[10,11]

9. Conclusions and Future Work

The primary goal of a Mockets transport protocol is to maximize the reliability and timeliness of data delivery to the information space and clients while minimizing the overhead in providing this delivery capability.[9] This experimentation indicates performance improvement on TCP-based ISA by using Mockets communication protocol under the constrained network in both wired and wireless environment. Our results showed that the Mockets protocols outperform TCP transport protocols on wireless settings in addition to stressed-wired networking environments.

As described in the previous section, the effects of using configuration parameters and mobility support capability on the network performance of ISA remains to be evaluated. Future work will include a qualitative analysis evaluating the performance of ISA with those capabilities.

10. References

1. US Army Training and Doctrine Command. The U.S. Army in multi-domain operations 2028. Fort Eustis (VA): US Army Training and Doctrine Command; 2018 Dec 6. Pamphlet No.: 525-3-1. https://www.tradoc.army.mil/Portals/14/Documents/MDO/TP525-3-1_30Nov2018.pdf.

2. Benvegnù E, Suri N, Hanna J, Combs V, Winkler R, Kovach J. Improving timeliness and reliability of data delivery in tactical wireless environments with Mockets communications library. Proceedings of the MILCOM 2009 - 2009 IEEE Military Communications Conference. 2009 Oct 18–21; Boston, MA. p. 1–8. doi: 10.1109/MILCOM.2009.537995.

3. Suri N, Breedy MR, Marcus KM, Fronteddu R, Cramer E, Morelli A, Campioni L, Provosty M, Enders C, Tortonesi M, et al. Experimental evaluation of group communications protocols for data dissemination at the tactical edge. Proceedings of the 2019 International Conference on Military Communications and Information Systems (ICMCIS); 2019 May 14–15; Budva, Montenegro. p. 1–8.

4. Poltronieri F, Sadler L, Benincasa G, Gregory T, Harrell JM, Metu S, Moulton C. Enabling efficient and interoperable control of IoBT devices in a multi-force environment. Proceedings of the MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM); 2018 Oct 29–31; Los Angeles, CA. p. 757–762.

5. Industry Trends IoT News. Integrated sensor architecture: deploying smart sensors and improving the mobile Soldier's UX. Liberty (SC): Sealevel; 2018 Feb 27 [accessed 2020 May 6]. https://www.sealevel.com/2018/02/27/integrated-sensor-architecture-deploying-smart-sensors-and-improving-the-mobile-soldiers-ux/.

6. Suri N, Tortonesi M, Arguedas N, Breedy M, Carvalho M, Winkler R. Mockets: a comprehensive application-level communications library. Proceedings of the MILCOM 2005 - 2005 IEEE Military Communications Conference; 2005 Oct 17–20; Atlantic City, NJ.

7. ihmc/nomads. Mockets overview. San Francisco (CA): GitHub; c2020 [accessed 2020 May 6]. https://github.com/ihmc/nomads/wiki/Mockets-Overview.

8. Suri N, Benvegnù E, Tortonesi M, Stefanelli C, Kovach J, Hanna J. Communications middleware for tactical environments: observations,

experiences, and lessons learned. IEEE Communications Magazine. 2009;47(10):55–63.

9. Breedy M, Budulas P, Morelli A, Suri N. Transport protocols revisited. Proceedings of the MILCOM 2015 - 2015 IEEE Military Communications Conference; 2015 Oct 26–28; Tampa, FL. p. 1354–1360. doi: 10.1109/MILCOM.2015.7357633.

10. Tortonesi M, Stefanelli C, Suri N, Arguedas M, Breedy M. Mockets: a novel message-oriented communications middleware for the wireless Internet. Proceedings of the WINSYS 2006 - International Conference on Wireless Information Networks and Systems; 2006 Aug 7–10; Setubal, Portugal. p. 258–267.

11. Benvegnù E, Suri N, Tortonesi M, Esterrich T 3rd. Seamless network migration using the Mockets communications middleware. Proceedings of the 2010 - MILCOM 2010 Military Communications Conference; 2010 Oct 31–Nov 3; San Jose, CA. 10.1109/MILCOM.2010.5680364.

List of Symbols, Abbreviations, and Acronyms

ARL	Army Research Laboratory
CCDC	US Army Combat Capabilities Development Command
COTS	commercial off-the-shelf
ECDF	Empirical Cumulative Distribution Function
EMANE	Extendable Mobile Ad-hoc Network Emulator
IP	Internet Protocol
ISA	Integrated Sensor Architecture
JBI	Joint Battlespace Infosphere
JTIDS	Joint Tactical Information Distribution System
LAN	local area network
MDO	Multi-Domain Operation
NTP	Network Time Protocol
OS	operating system
OSGi	Open Service Gateway Initiative
OSUS	Open Standard for Unattended Sensors
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UGS	Unattended Ground Sensors

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	CCDC ARL FCDD RLD CL TECH LIB
4 (PDF)	CCDC ARL FCDD RLC IB J FREEMAN T GREGORY M LEE J MICHAELIS