Naval Research Laboratory
Washington, DC 20375-5320

NRL/7340/MR—2023/4

Message Delivery by Geo-spatially Routed Autonomous Assets

BEN STRINGER

Center for Geospatial Sciences Branch
Ocean Sciences Division

September 18, 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
18-09-2023	NRL Memorandum Report	10-01-2022 – 10-01-2023

4. TITLE AND SUBTITLE

Message Delivery by Geo-spatially Routed Autonomous Assets

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ben Stringer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1Y90

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

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/7340/MR--2023/4

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

Office of Naval Research
One Liberty Center
875 N. Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Network routing protocols require a fully connected path from source to destination, however in situations without existing infrastructure, this requirement may not be met. Future battlespaces equipped with increasingly autonomous systems can capitalize on the mobility of these platforms to physically transport network traffic. We provide a motivating example and discuss two messaging implementations. These techniques have high latency, dependent upon the travel speed of the autonomous platform, however they have high throughput when compared with constrained options such as signal flares or human couriers. These techniques allow the autonomous platforms to serve dual purposes, couriering network traffic as well as performing their primary duty, e.g., intelligence gathering. We describe the application layer implementation details that can be incorporated into existing network architectures, using existing autonomous platforms.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Ben Stringer
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	U	19	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(228) 688-5880

This page intentionally left blank.

CONTENTS

This page intentionally left blank

EXECUTIVE SUMMARY

Network routing protocols require a fully connected path from source to destination, however in situations without existing infrastructure, this requirement may not be met. Future battlespaces equipped with increasingly autonomous systems can capitalize on the mobility of these platforms to physically transport network traffic. We provide a motivating example and discuss two messaging implementations. These techniques have high latency, dependent upon the travel speed of the autonomous platform, however they have high throughput when compared with constrained options such as signal flares or human couriers. These techniques allow the autonomous platforms to serve dual purposes, couriering network traffic as well as performing their primary duty, e.g., intelligence gathering. We describe the application layer implementation details that can be incorporated into existing network architectures, using existing autonomous platforms.

This page intentionally left blank

MESSAGE DELIVERY BY GEO-SPATIALLY ROUTED AUTONOMOUS ASSETS

1. INTRODUCTION

Networking and routing protocols for a majority of use cases are mature, particularly for the modern internet. Still evolving are protocols for networking at the edge, for example, mobile ad-hoc sensor networks deployed for research purposes such as volcanic seismic monitoring, measuring ocean conditions, or space exploration. These edge-network applications, although dynamic, do not have the same urgency as battlefield communications. The loss of communications in a swarm of ocean sensors results in a reduction in data resolution. It represents a problem that can be debugged and fixed, worked around, or in the worst case, overcome by deploying a new sensing system. The loss of communications on a battlefield can result in loss of life.

Current operational networks rely on traditional networking protocols. This is reasonable, since they are known to work, and off-the-shelf equipment can operate with military assets seamlessly. So long as the network remains fully connected, all existing technologies, including data transport, validation, authentication, and encryption will work fine. The battlespace, however, is dynamic, and reliable persistent connections cannot be guaranteed. Possible disruptions could include active jamming by an adversary or the need to remain undetected from an Electronic Warfare (EW) perspective.

Alternative communication options exist for these situations, each with their own drawbacks. For example, point-to-point communications are hard to jam and hard to detect, however they do not work well for mobile units and are required to maintain a Line-of-Sight (LoS). The force must be within range of a cell tower to use cellular networks, and we can not guarantee access to a foreign entity's network. Likewise, satellite communication is a viable option that may be owned by a party with conflicting interests. This paper attempts to add additional tools to the arsenal, so that even in the most restrictive environment, information flow can still be achieved. In Section 2 we discuss related work in the research literature, describing three approaches that could be used. In Section 3 we describe a motivating scenario. In Section 4 we provide implementation details for two approaches. In Section 5 we discuss the practical challenges and describe an experimental approach that could be demonstrated at an Office of Naval Research (ONR) field integration event with minimal disruption to existing performers.

2. BACKGROUND

The most prevalent routing protocol, the backbone of the internet, is the Distance-Vector (DV) algorithm [2]. A given node in the network constructs a routing table by communicating with neighbors and minimizing the number of hops between the node and all others. However, the nodes in the internet are static, that is, they change very infrequently usually as a result of hardware failure or upgrade. Although leaf nodes such as laptops and cellphones are expected to move, they are not typically addressed directly, so this is not an issue.

The DV algorithm does not work when the network consists of mobile nodes, including those comprising a swarm. Owing to the mobility of platforms in Mobile Ad hoc Networks (MANETs), links between

Manuscript approved September 18, 2023.

communicating neighbors can undergo frequent changes due to factors such as obstructions or nodes moving in opposite directions. We could naively extend the DV algorithm by updating the routing table more frequently, however this would pose an unreasonable burden on nodes. Network traffic would devolve into routing table update messages with little bandwidth remaining for actual network traffic.

Several algorithms have been developed for MANETs including the Ad hoc On-Demand Distance Vector (AODV) algorithm [3]. This algorithm determines the shortest path between two nodes *on demand*, reducing the need to maintain up-to-date routing tables in the face of high mobility. This approach is expanded upon in [4], but at its heart, still requires a path to exist at the time a message is sent. If the destination cannot be reached, the message is not sent. In the scenario described in this paper, the source and destination nodes are guaranteed to exist, however a fully connected path between them may not exist at the time a message is available to send.

Internet protocols and their inherent assumptions are not appropriate for all kinds of network configurations. In [5], the authors describe a series of so called "challenged networks" including military ad hoc networks such as the one described in this paper. These networks may be connected to the internet at the fringes with periodic, predictable connectivity. They may have limited, sporadic, and unpredictable connections between nodes in the network. Networks that are resilient to communication delays are referred to as Delay Tolerant Networks (DTNs) or Intermittently Connected Mobile Networks (ICMNs). Since a fully connected path to the destination does not exist, these network types utilize a form of "custody transfer", wherein the next hop in the network assumes responsibility for the message, providing some guarantee of reliability. We can compare this to the postal service, where custody of a package is transferred along with the package, and delivery is guaranteed unless the package is destroyed in transit.

DTNs generally fall into one of two categories: periodically connected networks or non-linear sporadically connected networks. We introduce a third type, dispatch routed networks, where the connection between two nodes requires a physical device to move from one node to the other carrying the message payload. Periodically connected networks are discussed in Section 2.1 and dispatch routed networks are discussed in Section 2.2. Sporadically connected networks are much more challenging. We discuss this network type in Section 2.3, however we do not attempt to apply it to the scenario described herein.

2.1 Periodically Connected Networks

Periodically connected networks are characterized by a predictable mobility of some components resulting in a periodic (dis)connection or partitioning of the network. The mobile component could be either the sensing component which periodically comes within range of a static base station or a dedicated "data ferry" device which visits static sensing nodes along a pre-defined route. Also possible, both the sensing node and data transport nodes could be mobile, however at least one of the node's route must be predictable so that a rendezvous can be planned.

The simplest system to implement is the static base station servicing mobile sensors. One example is an orbiting telescope which can only transmit to the ground station as it passes over a particular region of the earth. Another example is an Autonomous Underwater Vehicle (AUV) or a swarm of AUVs tasked with exploring a large area. They collect high resolution sensor data, but due to underwater communication limitations, they must navigate to a base station to transfer this data. These mobile nodes will buffer their outgoing messages until they are able to communicate. In addition, the base station may transmit messages to the nodes, for example, mission commands or knowledge state updates from other nodes.

Because the connectivity is periodic and predictable, it is fairly easy to build this type of system. The application layer of the OSI model is usually equipped with the logic for message buffering, including any rules for purging low priority messages if the buffer becomes too full. These types of networks typically have full connectivity while the link between nodes exists. Mobile nodes can connect to anywhere on the internet, provided the base station is connected. By positioning the base station in an optimal location, application developers can think of this less as a networking problem and more as a data management problem. We are able to transmit as much data as the communication window and bandwidth of the system allows, although in some cases these constraints prohibit sending everything. For example, the sensor's resolution could be too high to send all collected data during the communication window. This is the case for [6], an interferometric telescope whose minimum resolution would require 9.5 days to transmit one day's worth of readings. To handle this, the satellite performs onboard processing of the collected data and only transmits the results.

At the opposite end is a static, stationary sensor network. As an example, consider an aircraft-deployed remote monitoring system. The nodes are widely spaced, preventing the formation of a mesh network. In this case, operators could send a fixed-wing drone along a path such that each node will be within communication range of the drone for a short period. The drone would be loaded with messages to deliver to the sensors before takeoff. When the drone is in range of a given node, it would deliver these messages and receive the sensor's buffered readings. This route could be scheduled to be run hourly, daily, weekly, or so on, depending on how close to real-time the data is needed.

Unlike with mobile nodes, the fixed-position sensors are unable to connect directly to the internet. Messages *must* be buffered, and the sending node should not expect a reply to any message during the current communication window. If a node is entering a failure state and needs human intervention to recover, it must wait for the next periodic visit from the data ferry. Architecting this system is likewise straightforward. If all sensors are sending a fixed amount of data, and the system contains a fixed number of sensors, then engineers can determine the amount of storage required for a round-trip. If the sensors can generate variable size data, the data ferry must be prepared to handle the worst case, that each sensor collects the maximum amount of data. The data ferry must have sufficient buffer space for all messages, or it must have logic for aggregating or selectively dropping messages,

If the sensors are close enough to form a mesh network, or if they are able to form partitioned clusters, data ferrying can still be a valuable tool. In a typical mesh network, the nodes closer to the base station will route a larger number of messages than those at the fringes, depleting their power supply faster. This burden can be relaxed while still collecting high-resolution data from the network. Instead of the ferry visiting each node, it will visit *some* nodes in the network. Sensors would route messages to the nodes that are visited by the ferry, reducing the overall network traffic. If the network is not partitioned, high-priority messages can still be routed through the mesh network using any existing routing protocol.

Finally, the periodic connection approach works for fully mobile networks as well. This is described in [7]. Mobile nodes are aware of the ferry's route and will navigate themselves so that they are within range of the ferry as it passes along its route. This is particularly useful for mobile nodes that are not able to travel fast or far, or that need to stay within some area of interest. A specialization of this approach allows for a node to use low-bandwidth, long-range communication to request a visit from the data ferry.

2.2 Dispatch Routed Networks

As an April Fool's publication, the Internet Engineering Task Force (IETF) released [8], an Request for Comments (RFC) that describes a networking approach using avian carriers. Although it was intended as a

humorous break from typical RFCs, a real-world implementation [9] demonstrates that, for certain parts of the world, physically transferring the data rather than transmitting it is actually faster and more reliable, even when infrastructure already exists.

Before modern networks, battlefield commanders communicated by sending runners with messages. A modern commander will not rely solely on computers, but it is hard to argue with the volume of data that can be represented and transferred digitally. In the event of a breakdown in communication so catastrophic that data must be hand delivered, it would still be better to hand carry digital data that can be processed and used by autonomous systems and humans alike. Information such as asset deployment, enemy detections, and updated battle plans can be too complex to be delivered on paper. One option is to copy the relevant data to a thumb drive, to be couriered and injected on the remote machines.

If we compare delivering data on paper with delivering it digitally, both have similar latencies. Both approaches will require non-negligible time to transport the physical medium, although the digital option may have higher throughput depending on how much data is transferred. The first option will suffer a delay while the new knowledge state is manually input into the system so that the autonomous squads can make use of it. The second option will suffer a delay while an operator aggregates the data in a format that can be written to disk. Neither is ideal primarily because neither option is as seamless as regular network traffic. Our applications already have a means of transmitting data between each other. Serializing them into files and then later deserializing and ingesting them does not fit into the normal workflow and therefore is error prone.

Dispatch routed networks are characterized by communication between two points involving deliberate movement of some infrastructure component for point-to-point transfers. An autonomous platform in range of the source receives and stores one or more messages and navigates into range of the destination and transmits those messages. While this appears to be a specialized case of data ferrying (Section 2.1), it remains distinct. With ferrying, the data ferry is traveling along a pre-planned route. With dispatch routing, the mobile nodes are sent on demand, directly to the intended destination. This type of communication is ideal for high priority messages as it has the least latency of the challenged networking options.

Aside from the light-hearted RFCs, very little literature covers this type of communication. The prevailing thought is that solutions should involve infrastructure improvements. For most situations, improving infrastructure makes sense. The scenario we describe in this paper, however, is only one of many realistic cases where infrastructure improvements are not an option. This includes establishing infrastructure in the form of a periodic data ferry. Dispatching drones to deliver messages is not particularly challenging; we are already dispatching drones for other purposes. The challenge is making it seamless. Messages need to route through the network as expected if a path to the destination is available. If not, it will need to be transmitted on a physically dispatched device. Since this protocol will allocate limited physical resources, some level of human interaction will be necessary, however it should not be the extreme end: converting messages to files, writing them to disk, and loading the disk on an autonomous platform. A human should be included in the loop to confirm when the platform should be dispatched and to define or augment the route that the platform will attempt to navigate. Actually transferring the data to and from the platform should happen automatically.

2.3 Non-linear Sporadically Connected Networks

The third type of DTN is characterized by frequent link breakage and unpredictable partitioning of node clusters. These disconnections follow a non-linear pattern, making prediction impossible. Examining the

pair-wise interactions of nodes *over time*, we can find a multi-hop route through the challenged network, however trying to anticipate this route is non-trivial. This behavior is typical of a swarm whose members are performing a random walk, for example, trying to search a large space.

This type of routing has received significant interest in the academic community. For example, in [10], the authors propose a brute-force solution, the *epidemic* routing scheme. As the name implies, this type of message routing works by flooding the network with copies of the message. This approach guarantees that if a message is able to reach the destination, it will do so in the shortest time possible. On the other hand, it has terrible storage requirements. Every node must store a copy of the message, even if the message has already been delivered. While its requirements are unreasonable, this algorithm serves as a good baseline against which other options can be compared.

The *PROPHET* routing protocol proposed in [11] attempts to improve this by determining the probability that a given node will be on the route to the destination based on a number of factors including the time since a considered node last encountered the destination. The *spray-and-wait* algorithm presented in [12] further improves by fixing the number of messages allowed to be sent. In the *spray* phase, the sender sends half of the remaining messages each time a potential route is encountered. When only one message remains, the sender enters the *wait* phase, retaining the message until it encounters the destination. While this algorithm has much more manageable storage requirements, its delivery times are much harder to characterize, particularly since the choice of which nodes to communicate with can affect these results.

The *HYMAD* algorithm, described in [13], is a hybrid of DTN algorithms and typical MANET routing protocols. Here, groups of nodes form clusters. This approach is useful when certain nodes are expected to work together, for example a ground based systems paired with aerial scouts. The clusters communicate internally using typical MANET routing protocols, capitalizing on the fact that cluster members are not likely to change frequently. Transmitting outside of the cluster can be achieved using a variation of the spray-and-wait algorithm.

These messaging schemes are typically used within a swarm or multi-agent system. They are not typically used to route external messages *through* a swarm. In the highly dynamic battlespace environment, every option should be on the table. If a swarm of this type is operating between the source and destination, a message could be routed *through-the-swarm*. Getting this to work would require cooperation of multiple disparate systems. It would require unrelated systems to agree on a communication standard including radio type and messaging algorithm, and it would require smaller platforms reserve significant resources for buffering messages that may never be delivered by the drone in question. On the other hand, this type of message routing may go unnoticed by the adversary. Any device could be participating in the message traffic. What appears to be two reconnaissance drones temporarily operating within range of each other could be valuable intelligence hopping across drones to a platoon with kinetic forces.

3. MOTIVATION

For motivation, we consider the following scenario. A forward blue force team will land on San Clemente Island in the Pacific, shown in Figure 1, and the red force will located on the mainland in Los Angeles. Blue force does not intend to establish a base on the island, however they will remain a short time, on the order of hours. They will perform some task, for example, installing surveillance or kinetic equipment, after-which they will withdraw. They wish to remain undetected so that red force does not become unaware of their activities on the island.

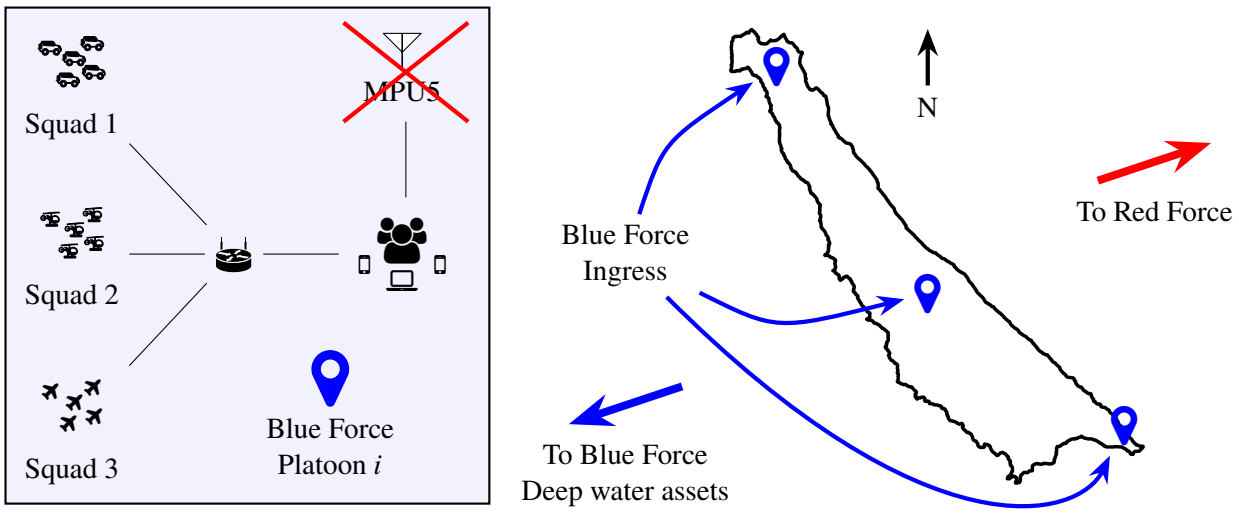


Fig. 1— The island of San Clemente, used as the basis for the scenario described in this paper. Three platoons are shown. One is a base of operations, deployed on the north of the island. Another is a mobile operations center on the south of the island. A third is executing mission objectives along the center of the island. Each platoon is composed of several squads of autonomous systems. Typical scenarios would connect these platoons via MPU5 radios. The need for stealth prevents the use of these long-range radios, and the terrain and need for mobility prevents Line-of-Sight (LoS) communication.

The island is roughly 20 mi lengthwise, with volcanic mountains reaching nearly 2000 ft. It is roughly 40 mi from the mainland, and is part of a chain of islands. The island is sparsely populated, and no existing communication infrastructure exists, i.e., the island does not have cellular coverage. Because they are trying to remain undetected, typical long-range communication options such as MPU5 radios are unavailable. Other options, such as a directional radio will require infrastructure to address the terrain interference. For the purposes of this scenario, the force's tasking will not allow time to establish this infrastructure.

Blue force is small, composed of eight to twelve soldiers, divided into several platoons of two to four soldiers. Each platoon is responsible for several squads of autonomous platforms including aerial, ground, and sea-surface vehicles. Some squads are configured as robotic swarms, where a swarm is composed of many simple autonomous devices which act as a component of the larger system, the "swarm". Human operators will interface with the swarm rather than any individual devices, in the same way that one would give high-level commands to a complex robot rather than control the robot's individual actuators. Squads have internal communications as necessary, for example WiFi or ZigBee radios. Squads communicate directly to the platoon only. At times, they may move out of short-range communications range. During this time, the squad will not be able to communicate, however any messages will be buffered until a link can be reestablished. Network traffic between squads is accomplished through the platoon, which can determine what information each squad needs and at what resolution.

In this scenario, the platoons are distributed across the island, with a base of operations at both the north and south end of the island. Upon arrival, some portion of the autonomous systems are deployed. An aerial reconnaissance swarm, for example, may be used to image the island for increased situational awareness, to detect adversary forces in the surrounding areas, or to identify safe routes for autonomous ground vehicles. In

addition, one platoon will be responsible for deploying and maintaining the sea-surface swarms, which will patrol the perimeter of the island, listening for approaching submerged vehicles. Communication between platoons would simplify the mission, however instantaneous communication is not required. Reach-back capability to a deep-water, over-the-horizon asset is also desirable, but not necessary for mission success.

Red-force is able to detect long-range communication, and they are able to monitor the island remotely. Because San Clemente Island is one of many islands, red force is not focusing specifically on it. Blue force operations will go unnoticed provided they do not transmit on any loud spectrums or deploy large aerial platforms. If red force does detect them, they will engage jamming tools, which would prevent the use of long-range communications and would interfere with a coordinated withdrawal.

For the scenario described, platoons are effectively disconnected. Communication between nodes is only possible if infrastructure components are established, for example, LoRa repeater nodes strategically placed. These could be established by autonomous systems, however for this scenario, the necessary platforms capable of this task are not available, and all platforms are needed for specific purposes. This paper will describe communication alternatives in the face of these challenges, piggybacking on hardware already deployed to support the mission.

4. IMPLEMENTATION

There is no reason why battlefield networking capabilities should be limited to the constraints of the internet. Failure modes should support reasonable failover states without resorting to primitive solutions: flags, flares, and runners. With the technology available today, it is possible to implement nearly seamless degraded communication options. This section will detail two possibilities within the scenario described in Section 3.

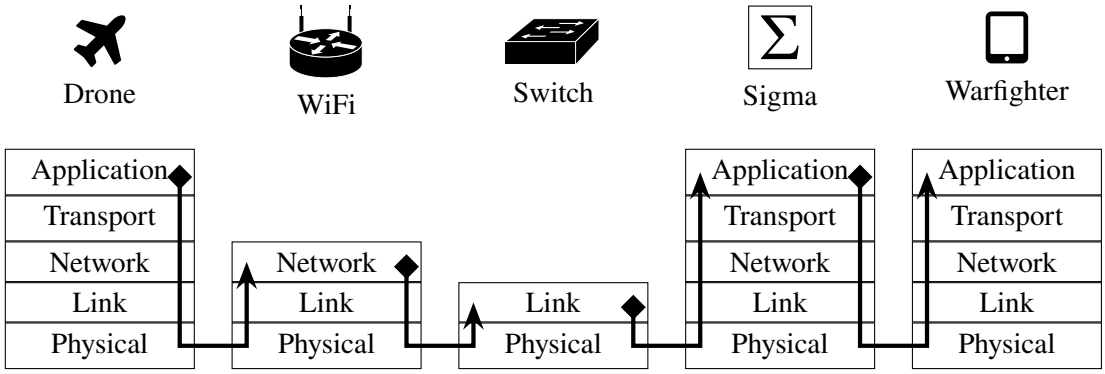


Fig. 2— The Internet Protocol Stack. Messages traverse various levels in this stack as they move through the network.

Our goal is to support networking in environments without infrastructure and without establishing our own *fixed* infrastructure. This does not mean the options we propose have no infrastructure requirements, however, we can show that these requirements can largely be met before the operation begins, so that during the mission, seamless degraded communications require little to no additional effort on the part of the warfighter. We want solutions that will work with the systems already in use. In Section 2.1 we examine a periodically connected solution, and in Section 2.2 we examine a dispatch routed solution. We do not discuss implementation details for non-linear sporadically connected networks because this option is non-trivial.

Consider Figure 2. This depicts message traffic between a squad member, for example, an aerial reconnaissance drone or a ground penetrating radar, to the platoon for review. Most importantly, note that this network is fully connected. The drone is able to find a route to the platoon so that a human can review the sensor data. This Figure includes the 5-layer Internet Protocol (IP) stack, which we will use for our DTN solution. Notice that the traffic does not need to travel up the stack at each hop in the route to the platoon. This is because a switch operates at the link-layer, and a router operates at the network layer. Our solutions will operate at the application layer. Arguments could be made for an implementation lower in the IP stack, but this would require significant concessions for this edge case without providing any benefit to the standard network operation.

We define two application layer stand-ins, which will allow us to generalize this problem. We will refer to Σ as a data management application running at each platoon. Σ serves three functions. First, it ensures that all data within the platoon's network is routed to the appropriate platoon-level application, for example, image detections might be routed to a human operator or to an image classification algorithm running on a tactical server. Second, it applies a utility function to each piece of data with respect to other platoons. Items that exceed some threshold are routed, while most traffic, only useful locally, is not, thereby reducing unnecessary traffic on the critical "backbone" link. Finally, it guarantees delivery without the source needing to integrate directly with each consumer. It achieves this by retaining all messages in an internal buffer until an acknowledgement message is returned.

Similarly, we define σ, a variation of Σ with two notable differences. First, σ does not apply utility functions. If it receives a message, it will ensure that message is delivered. Second, σ is geospatio-temporally aware. Unlike typical networks which only concern themselves with the *connections* between nodes, σ will know where each node is physically located and will have some path planning capabilities to augment an autonomous system's built in navigation. It will be able to process relevant messages in order to know when a platoon has or will move.

In practice, Σ and σ may not be any single application pair. Σ could be a collection of messaging pipelines, databases, and rule sets, and σ may represent some custom code imbedded in the Autonomous Aerial Vehicle (AAV) code. We use these symbols as a representation of functionality implemented at the application layer of the IP stack. By handling all of this at the application layer, we remove the need for building a custom protocol stack for a literal and figurative edge case.

4.1 Periodically Connected Networks

We will begin by examining a periodically connected option, also referred to as "data-ferrying". Figure 3 depicts a periodic connected network and shows the communication occurring at the operations centers. The operations centers are located at the north and south end of the island, and they will be responsible for maintaining this network's infrastructure. The AAVs, described in Section 1, will be configured before the mission begins. Each base will launch a device towards the other, imaging a portion of the island in the process. The devices will intersect their paths with the platoons operating on the island. When the device comes within range of a platoon, it connects to that platoon's local wireless network and authenticates. It delivers any messages for that platoon including the collected imagery, and it buffers any outgoing messages from the platoon. When the vehicles reach opposite ends of the island, they perform the same networking activities and land. Included in the network traffic is the imagery collected. This process repeats itself; either a second drone is launched while the first recharges, or the battery is changed and the same drone is launched.

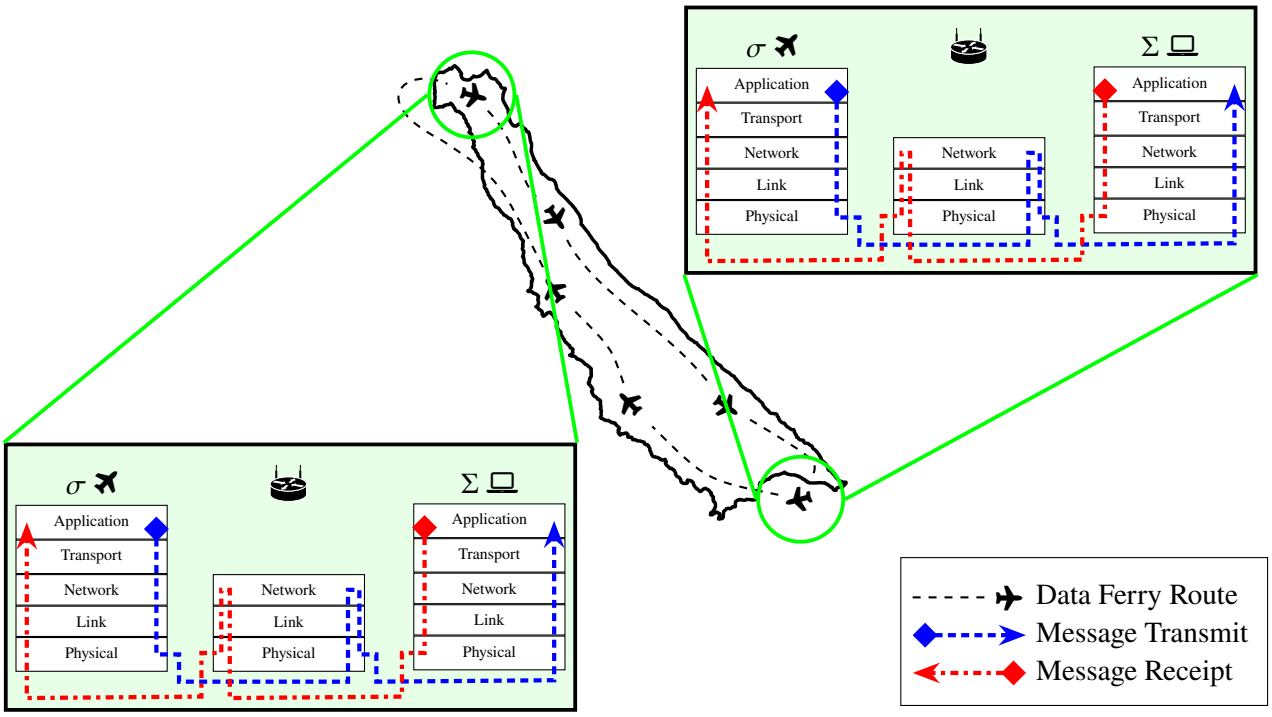


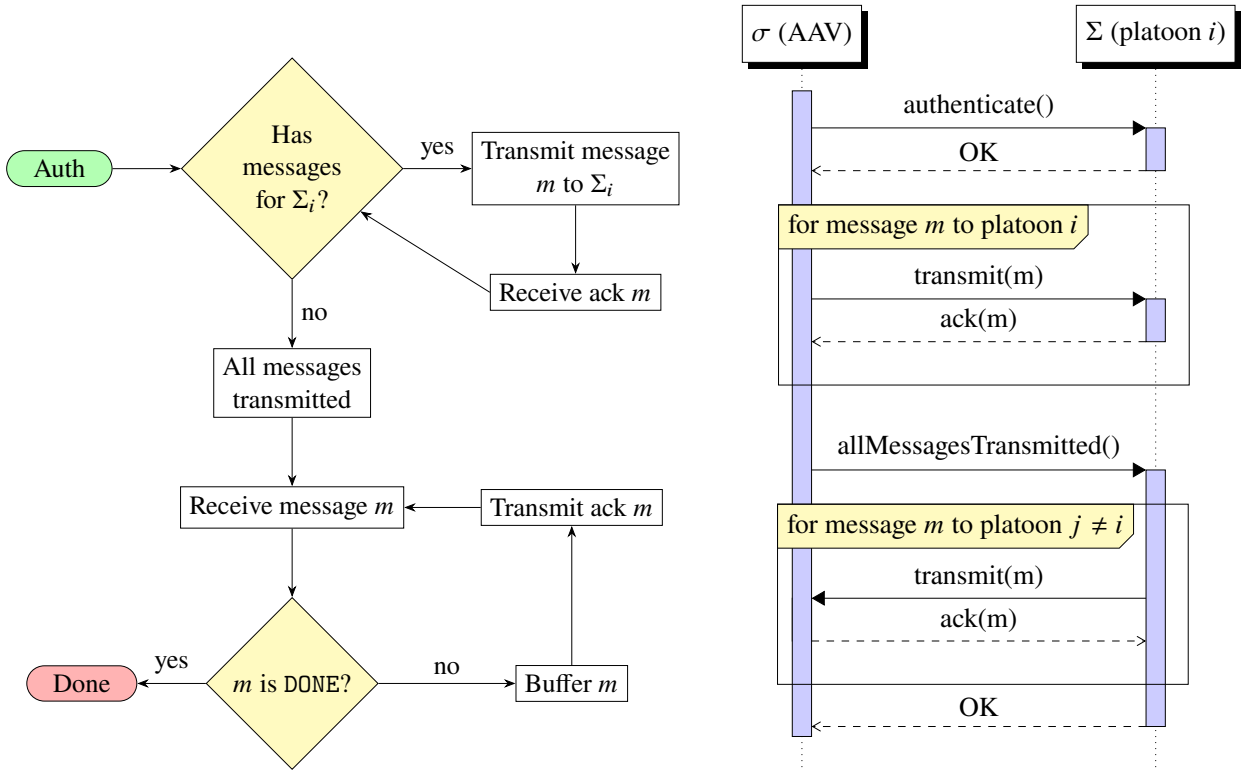
Fig. 3— A periodically connected network using data ferrying. A drone traverses the island, imaging it in the process. When it arrives within range of a platoon, it transmits any messages intended for the platoon and receives messages intended for other platoons. Messages are routed through some local wireless communication such as WiFi.

As we noted, this does constitute infrastructure. We need at least two drones capable of traversing the island longways. We need batteries and recharging capabilities. Most importantly, we need human capital to launch and recharge. While this is not trivial, it provides a number of improvements over other options. Contrast this with a fully-connected option. While it is possible for autonomous systems to establish a series of repeaters within line-of-sight of each other from one end of the island to the other, this option is non-trivial and much more prone to failures. Worse, the equipment is likely single-purpose and single-use. It does not provide additional intelligence gathering, and in the event of a hasty withdrawal, the hardware will likely be left behind. Once established, it does enable standard networking, but this option may take longer to establish than necessary. Most importantly, this option requires the warfighter carry additional hardware *solely* for communications infrastructure.

When long-range communications are in operation, Σ will transmit the local messages which are deemed useful to the larger force. When the long-range network is disabled, Σ will buffer messages until access is returned. In our scenario, long-range options will not be possible, first due to the need to remain undetected, and then due to active jamming if blue force is discovered.

With the periodic routing option available, Σ will still identify messages to be transmitted. These include status updates, planned force movements, and updated mission plans. Since a long-range option is not available, these messages will be buffered. When a σ node establishes connection with the platoon, it will communicate these messages as with typical long-range communication. However, instead of receiving an

immediate acknowledgement, Σ must wait for the platform to make the trip to the destination and return with the acknowledgement message.



(a) Flow chart (b) Sequence diagram

Fig. 4— UML diagrams depicting the communication between an AAV running σ and a platoon running Σ to support periodically connected networking.

Figure 4 depicts the interactions between σ and Σ. After authentication, σ will transmit any buffered messages intended for platoon i, including the imagery generated by the AAV. Σ will receive these and propagate them through the local network. From the perspective of any component in the local network, these messages will arrive as if they had been transmitted over the long-range communications network. Once it has received all incoming messages, Σ will transmit all outgoing messages, and σ will copy these to its internal buffer. Once complete, the AAV will move on, following its route to the next unit.

The latency of this approach is significant, although predictable. In the worst case, the message is not delivered due to hardware failure. This should be anticipated, and any message traffic should be treated as undelivered until acknowledged otherwise. If the carrier does not arrive as expected, the loss can be detected. Otherwise, the latency is the time taken for the AAV to make the one-way trip plus the time taken to transfer the messages into and out of σ's buffer. The delay for receiving the acknowledgement is similarly the time taken for the vehicle to make the return trip. Depending on the ferry's route, this may not be the same duration.

A significant security implication of this approach is that an adversary could track the path of the AAV to determine the location of one or more units. Since this is an imaging platform, some forces may go

undetected as the device is simply performing its normal operation, but if the enemy is able to track the vehicle as it lands, they can narrow down the force's location. One mitigation may be to have the landing and deploy locations separate from the force, however this will introduce additional overhead. The message ferrying option works when the adversary does not know the blue force is present, but once detected, any point along the vehicle's route should be considered compromised.

4.2 Dispatch Routed Networks

Dispatch routing has many implementation details in common with data ferrying. However, instead of waiting for the transport platform to come within range so that it can carry the messages to their destination potentially via a non-optimal route, the messages are loaded onto an autonomous vehicle already on hand. This vehicle is then sent directly to the destination. As noted in Section 2.2, the primary concern is reducing the operator workload. To this end, we utilize Σ as before, which will be responsible for transferring the data to the delivery platform; and σ, which is responsible for geospatially routing to the destination.

Σ transmits messages and routing information to σ, but aside from standard TCP responses, σ will communicate back. The platform will have been in storage or performing some local task, otherwise oblivious to the disconnected portion of the network, and therefore has no messages to deliver to the source. After navigating to the destination, it will connect to the network and transmit the messages. In many cases, this will be a one-way trip. The destination will take ownership of the vehicle, which it may or may not use to send a reply. Due to power constraints, the platform may not be capable of making the return trip, and an immediate reply may necessitate a second vehicle.

Human interaction is more involved than with the data ferrying option. Because this method will allocate resources—an autonomous vehicle—a human operator should confirm that the messages need to be dispatched to the destination. This requires an additional user-facing component not available in Σ. The operator should be able to review the messages Σ has selected for dispatch to ensure all relevant information is included as well as confirming the transmission.

Figure 5 depicts the network connections involved in a dispatch routed network. At the source, messages are transmitted from Σ to σ via the local network and are buffered in the mobile platform's internal storage. The device is launched. It navigates to the destination, and upon arrival, transmits all the messages in the buffer. Although this Figure demonstrates routing to a deep-water asset, it could be used to send immediate messages to another platoon. For example, if a platoon needs to immediately relocate so that it will no longer be on the data ferry's route, and the platoon does not have time to wait for the ferry, it may send a direct message to one of the other platoons, which will ensure the ferry eventually becomes aware of the new position. Additionally, this Figure shows an aerial platform, however nothing prevents the use of ground-based, ocean surface, or underwater vehicles. One could even imagine a combination of these approaches, for example using a ground vehicle to navigate away from the platoon which launches an aerial vehicle, to obscure the location of the source.

Another variation does not require that the platform navigate to the destination. Instead, we send a drone outside the range of jamming devices or to a remote location in order to obfuscate the troop locations. Here, we can switch to long-range communication, possibly sending and receiving data from any internet resource. Having sent its messages and buffered the replies, the device would return to the contested space to deliver its payload.

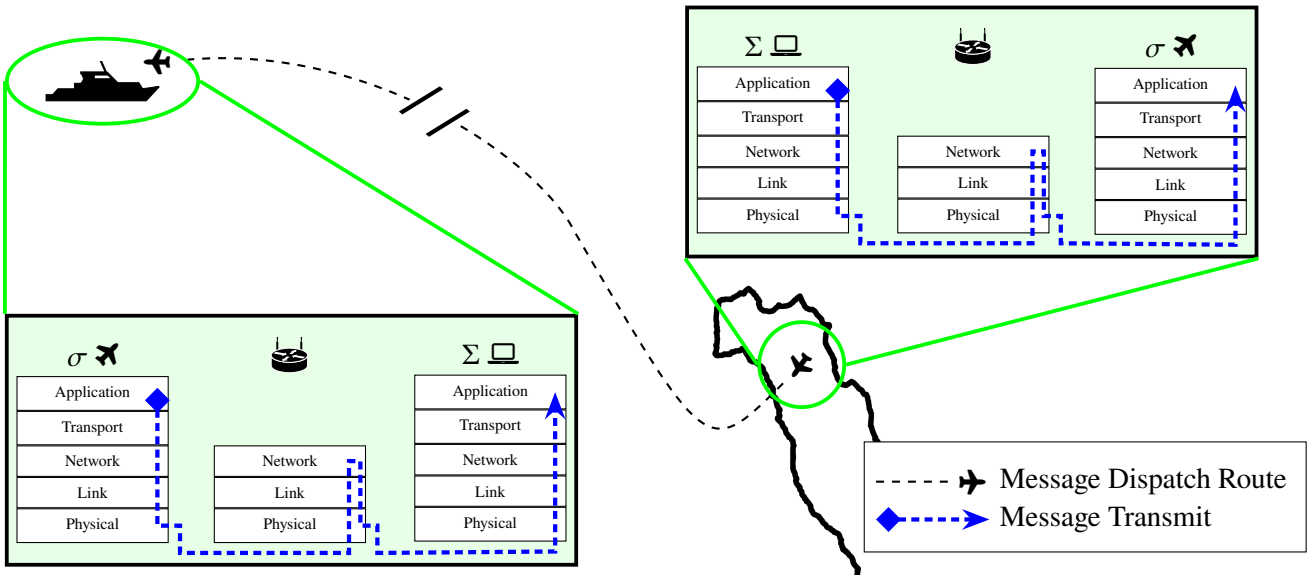


Fig. 5— Dispatching a message from a platoon on the island to a ship over-the-horizon. Message traffic goes one-way only. If the ship wants to send a reply, it must dispatch a second device.

Data ferrying supports some receipt acknowledgement capability. When a message is delivered to a platoon, that platoon's Σ would respond with an `ACK` message, which can be returned to the source. Although there is high latency involved, it is possible to track lost messages and resend, if the message is important enough. Dispatch routing does not support this capability since in most cases, the network traffic is one-way. If an acknowledgement is needed, the destination must have the ability to reply. However, dispatch routing is not necessary for the reply. The constraint is that the *source* remain undetected. The over-the-horizon asset can transmit using long-range communications since spectrum analysis will show that they are in international waters. Both red and blue force would be able to hear the response. Blue force would be able to interpret the message, however red force may not ascribe any significance to it, especially if the deep-water asset is transmitting a lot of messages.

5. DISCUSSION

The biggest hurdle in utilizing these approaches is implementation and integration. Battlefield networking is currently implemented with off-the-shelf hardware using well established protocols. There is very little impetus to modify the network stack to include support for challenged networks, instead relying on infrastructure improvements. Yet these literal edge cases where infrastructure improvements are not possible are the very ones that would most benefit from these solutions. This is why we have chosen to integrate with the application layer. We assume that some software is already in place for managing data flows between the various systems in the deployed force. This does not need to be a purpose built application. It could be a combination of messaging protocols such as RabbitMQ or Web Application Messaging Protocol (WAMP) along with a database for persistence. Instead of connecting the pipeline to the long-range radio, one could connect it to a simple application that can interface with an autonomous platform, necessitating a minor update to the messaging stack.

The physical layer of these challenged networks has a much larger footprint than typical networking schemes since the devices must move within the physical environment. This introduces a new set of security considerations. Just as a normal network must be resilient against attack, we must ensure that adversarial forces are unable to gain an advantage by accessing the physical layer. We have already discussed potential information gained by the adversary simply by observing the physical movement of the network nodes. These can be mitigated by obfuscation techniques, but this should not be relied upon as a security measure. Other issues arising from physical layer attacks include accessing the information stored on the device and device impersonation.

The first issue requires that the adversary capture one of the devices. With physical access, they can access the physical storage, potentially reading all buffered messages or the locations of units. This can be mitigated by using strong encryption for all data at rest. This will ensure that physical access to the device does not provide any additional information. The possibility for data loss exists if devices are captured or eliminated. Initiating a simple Denial of Service (DOS) attack could involve shooting down all drones. With a periodically connected network, the system will note the device not arriving as expected, which may alert the force to enemy actions. The messages will remain undelivered, however, and in the case of a dispatch routed message, there may be no indication that the message has been lost.

The second issue requires the adversary either capture and modify a device or build their own capable of impersonating one. The device would come within range of a unit and transmit bogus messages. This can be mitigated with strong authentication between the device and the platoon's network. This could be implemented using a cryptographic token stored on the device's encrypted drive. Authentication using a token would involve network traffic that is not susceptible to a playback attack, ensuring that only valid devices are able to participate in communication.

As we have noted, the forward deployed units must carry this infrastructure with them. We have assumed that the devices are already being included and that the data ferrying is a secondary task. This does not add additional burden, while providing additional functionality. Configuration can be performed beforehand as well, so that when the force deploys, they only need to launch the devices. This is not true for dispatch routing though, which requires the unit have spare devices on hand. Depending on the mission, this added cost may be acceptable, especially if the devices are small and lightweight.

Our motivating example described a stealth operation, yet these approaches can be used in active battlespaces as well. When the adversary is using active jamming, they expect that very little, if any, network communication is taking place. However, a unit could dispatch intelligence messages including high-resolution imagery of the adversary's base of operations, and the force commanders could dispatch highly detailed mission plans. One challenge is that the units may move based on the dynamic environment, rendering the location information on the mobile platform invalid. This is true in any scenario, however depending on the platform's autonomy capabilities, it may be capable of searching for and locating the unit. While this does increase the cost and complexity of the platforms, this search capability may already be incorporated to search for enemy units.

6. CONCLUSION

Great advancements have been made in battlefield communication, however there are no good fall-over options in degraded conditions. In a forward deploy location without existing infrastructure and no long-range options, the best options are to send a message via runner or using some low throughput signaling

option. With autonomous vehicles being included in the force's arsenal, additional communication options become available. We have discussed three approaches for these so-called challenged networks. We detailed two approaches that can be implemented and tested using existing technology. While many other options are also available, the ones we described have the benefit of capitalizing on existing platforms and behaviors.

ACRONYMS

AAV	Autonomous Aerial Vehicle
AODV	Ad hoc On-Demand Distance Vector
AUV	Autonomous Underwater Vehicle
DTN	Delay Tolerant Network
DOS	Denial of Service
DV	Distance-Vector
EW	Electronic Warfare
ICMN	Intermittently Connected Mobile Network
IETF	Internet Engineering Task Force
IP	Internet Protocol
LoS	Line-of-Sight
MANET	Mobile Ad hoc Network
ONR	Office of Naval Research
RFC	Request for Comments
WAMP	Web Application Messaging Protocol

REFERENCES

1. E. Copernicus Sentinel-2," https://commons.wikimedia.org/w/index.php?curid=95522500, Mar 2019. URL `https://scihub.copernicus.eu/dhus/#/home`, CC BY-SA 3.0 igo.

2. J. F. Kurose and K. W. Ross, *Computer Networking: A Top Down Approach*, Seventh ed. (Pearson Education, Inc., May 2016).

3. C. Perkins, E. B. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, RFC Editor, Jul 2003. URL `https://www.rfc-editor.org/rfc/rfc3561.txt`.

4. C. Shin and M. Lee, "Swarm-Intelligence-Centric Routing Algorithm for Wireless Sensor Networks," *Sensors* **20**(18), 5164 (2020).

5. K. Fall, "A Delay-tolerant Network Architecture for Challenged Internets," Proceedings of the Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03 (ACM Press), 2003. URL `https://lens.org/144-465-394-004-364`.

6. H. Linz, D. Bhatia, L. Buinhas, M. Lezius, E. Ferrer, R. Förstner, K. Frankl, M. Philips-Blum, M. Steen, U. Bestmann, et al., "Infrared astronomy satellite swarm interferometry (IRASSI): overview and study results," *Advances in Space Research* **65**(2), 831–849 (2020).

7. W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," Proceedings of the Proceedings of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing, 2004, pp. 187–198.

8. D. Waitzman, "A Standard for the Transmission of IP Datagrams on Avian Carriers," RFC 1149, RFC Editor, April 1990. URL `http://www.rfc-editor.org/rfc/rfc1149.txt`.

9. BBC, "SA pigeon 'faster than broadband'," `http://news.bbc.co.uk/2/hi/africa/8248056.stm`, September 2009.

10. A. Vahdat, D. Becker, et al., "Epidemic Routing for Partially Connected Ad Hoc Networks (2000).

11. A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *Mobile Computing and Communications Review* **7**(3), 19–20 (July 2003), doi:10.1145/961268.961272.

12. T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," Proceedings of the Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant Networking, 2005, pp. 252–259.

13. J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Computer Communications* **33**(13), 1483–1492 (August 2010), doi:10.1016/j.comcom.2010.03.005.

14. AeroVironment, "Quantix Mapper," Brochure, Jul 2021. URL `https://www.avinc.com/images/uploads/product_docs/Quantix_Mapper_Datasheet_07222021.pdf`.