Federation Based on MQTT for Urban Humanitarian Assistance and Disaster Recovery Operations

Manas Pradhan

Abstract

Today's age of information and communications technologies (ICTs) in urban areas revolves around the application of the Internet of Things (IoT) and application of IoT in smart city constructs. IoT has enabled cheap but reliable ubiquitous computing for modern day ICT needs. As a result, the military community is actively looking into application of IoT for its operational needs. Federation and interoperability become complex for IoT implementation in the huge jungle of protocols and technologies available for IoT. This problem becomes critical in humanitarian assistance and disaster recovery operations where multiple agencies need to collaborate to bring quick and effective relief to disaster struck areas. Message Query Telemetry Transport (MQTT) is such an IoT-based protocol that is widely adopted in the industry for lightweight but reliable messaging. This article tries to provide insight into federation based on MQTT with a prototype implementation between military and civilian ICT systems. This federation concept would enable lightweight, vendor-agnostic, and interoperable message exchange while using existing information sources and preventing stove-piped systems.

Introduction

Humanitarian assistance and disaster recovery (HADR) operations in the modern-day context require multiple agencies from the military and civilian domains to act together [1]. Especially in urban scenarios, where the concentration of humans makes it a very complex and challenging environment, a single emergency responder agency often does not suffice to have the required recovery impact [2]. As the modern human demographics turn toward cities, the cities are equipping themselves with modern information and communications technologies (ICTs) to serve this high influx and concentration of population. The smart city concept is a step in this direction, aiming to provide better governance, participation, economic possibilities, sustainable development, and so on for its citizens. The Internet of Things (IoT) has been one of the biggest enablers of the smart city concept by providing the necessary ubiquitous and participative computing for the urban ICT needs.

IoT-based innovations in sensors, actuators, computing platforms, and more have led to their mass acceptance in the civilian and industry domains. Concepts in industry such as Industry 4.0 has shown the wide proliferation of IoT systems on the shop floor, distribution and logistics, manufacturing, computing, and analytics. In the consumer industry, smartphones, smart homes, personal computing, analytics, and so on have shown the path ahead for the IoT-driven market. As a result, over the years the technologies have matured for large-scale acceptance coupled with cheap production while being reliable for long-term usage. In urban settlements, networks such as 4G, 5G, wireless local area networks (WLANs), personal area networks (PANs), and low-power wide area networks (LPWANs) have provided the back-end network support to realize this idea of a connected world and thus the potential of IoT in everyday lives.

For HADR operations, such IoT-based innovations can be leveraged to assist emergency responders and complement their recovery efforts. Especially considering the smart city domain, where the ICT technologies tend to be more connected and organized, the ICT assets of HADR agencies can reuse the existing cities' capabilities. The quick reaction times required from responders leads to missing capabilities, which requires collaboration with the assets available on the ground. In the North Atlantic Treaty Organization (NATO) context, civil-military cooperation (CIMIC) is such a concept, which tries to address this gap of bridging the civilian ground (both human and technology) assets with the military assets [3]. Furthermore, the specialized NATO response force (NRF) and Very High Readiness Joint Task Force (VJTF) concepts further strengthen the use case for urban environments. These specialized operational forces need to be deployed in just a few days, on short notice, to respond to adversarial situations affecting the periphery of the Alliance [4].

The IST-147 Research Task Group on Military Applications of IoT, in this corresponding direction of adoption and integration of IoT in the military domain, was formed in 2016. It investigated the emerging IoT technologies and found favorable use of commercial off-the-shelf (COTS) IoT assets for complementing the military ICT assets. In a joint experiment in 2018 with multiple NATO nations, the group showed how IoT and ICT assets from multiple nations and civilian domain can be integrated in an urban HADR operation

The author tries to provide insight into federation based on MQTT with a prototype implementation between military and civilian ICT systems. This federation concept would enable lightweight, vendor-agnostic, and interoperable message exchange while using existing information sources and preventing stove-piped systems.

The author is with the University of Oslo.

Digital Object Identifier: 10.1109/MCOM.001.2000937

MQTT is a lightweight publish/subscribe messaging transport protocol that was developed keeping the needs of IoT applications in context. It suits the needs of low-power and resource-constrained remote IoT devices and applications due to its minimal code footprint and network bandwidth utilization.

[5]. But the experiment left some open questions, such as:

- How can the multiple command and control (C2) systems federate with each other?
- How can an IoT protocol be leveraged for federation between the coalition partners?
- Which C2 systems and which interfaces from the C2 systems can be used for federation?
- How can the civilian data be accessed and at what granularity?

As a response to these questions, the IST-176 group was formulated to focus on federated interoperability of military C2 and IoT systems. This article tries to provide an insight to enable this federation between military C2 systems while interfacing civilian ICT systems with the focus on the IoT domain for future HADR operations.

The rest of the article is organized as follows. Message Query Telemetry Transport (MQTT) protocol and the state-of-the-art related research is presented. Based on features of MQTT and the latest advances in its research, a federation mechanism based on MQTT to enable multiple C2 systems' interoperability is described. The architecture and prototypical implementation to showcase the federation concept is then detailed to show how MQTT is used in a real-life use case. Finally, the conclusion and the lessons learned are presented, based on which the corresponding future work follows.

MQTT

MQTT is a lightweight publish/subscribe messaging transport protocol that was developed keeping the needs of IoT applications in context. It suits the needs of low-power and resource-constrained remote IoT devices and applications due to its minimal code footprint and network bandwidth utilization. It fits the unreliable nature of remote and overcrowded networks operating at the edge by allowing persistent sessions and varied quality of service (QoS) settings. Various flavors of MQTT available, supported by its security features, portability, and extensibility, have allowed MQTT applications to be deployed in the cloud, containers, and enterprise environments [6]. MQTT, in its core, operates with a client-server mechanism; that is, a centralized server is responsible for coordinating and mediating the exchange of messages between the endpoint clients. All communication and exchange of messages is based on topics with the clients publishing or subscribing to messages. Earlier versions of MQTT 3.1.x used the term "broker" for the central entity responsible for message exchange between the publishers and subscribers. From MQTT 5.0 onward, the term "broker" has been replaced with "server" due to the new features added and the new nature of interactions. Similarly, the terms "publishers" and "subscribers" are replaced with "clients."

The NATO IST-147, IST-150, and IST-161 groups have extensively evaluated MQTT for its application in tactical and coalition operation environments [5, 7, 8]. The experiments have shown favorable results for application of MQTT for federated distributed environment settings. The following subsection describes the state-of-the-art findings and the scope for further research.

State of the Art

A comparison of web services (WS) Notification with MQTT was presented in [9]. WS-Notification is used as the NATO Messaging Core Service, but it is not well suited to low-capacity tactical networks due to its overhead. Tactical network features very closely resemble IoT application environments, so protocols with lower overheads are always preferred in such settings. MQTT was found to offer similar functionalities as WS-Notification but with less overhead for disconnected, intermittent connectivity, and limited bandwidth (DIL) scenarios. But the article showed that the centralized broker architecture which exists for WS-Notification and MQTT is not a favorable setup due to a single point of failure.

In the joint experiment done by the NATO IST-147 group [5], such a centralized broker MQTT setup was demonstrated while combining multiple coalition IoT assets and the available city's ICT assets. It used a simple ontology with messages in Java Script Object Notation (JSON) format to achieve interoperability between the various data sources and the C2 applications. However, it lacked distributed inter-broker communication, which would remove this single point of failure in case the central broker crashed.

In [7], MQTT with Blue Force Tracking (BTF) was presented with a federated multi-broker approach. However, the implementation again used a central broker, which was bridged to two other brokers. It meant that if the central broker broke down, the two bridged brokers could not communicate anymore. Furthering the idea of multi-brokers, [10] showed the evaluation of a federation mechanism using multiple brokers. This article leverages the concept, extending it to application in urban HADR scenarios and exploiting some of the features provided by the federation mechanism for interoperability and flow control for situational awareness (SA) exchange between the HADR agencies.

MQTT Version 3.x vs. Version 5.0

The implementation and prototype described in this article leverages the new MQTT version 5.0, which replaces versions 3.1 and 3.0 [11]. The new version 5.0 has certain new elements for both the server and client sides. These are described below; they enable the deployment of a distributed masterless architecture and reducing traffic overheads.

Shared Subscriptions: With standard MQTT subscriptions (i.e., prior to version 5), each subscribing client received a copy of the message to which it subscribed. Hence, if a subscription node failed, published messages were lost (QoS 0) or accumulated in the server (QoS 1, 2). The solution to this issue was to increase the subscribing nodes, which resulted in large numbers of duplicate messages and thus lots of extra traffic. With shared subscriptions, clients that share the same subscription within a subscription group receive messages in an alternating fashion. This feature enables client load balancing since the load of the same subscribed topic is distributed among all subscribing clients. In contested HADR and DIL environments, this mechanism allows for reducing traffic and distribution overheads.

Clean Start Mode: In earlier versions of MQTT, the Clean Session mechanism was used by MQTT clients to have temporary connections to brokers or not subscribe to messages at all. The idea was to support offline or persistent sessions to handle connection interrupts. But the mechanism did not support the expiry of a persistent session (i.e., the session never expired or was deleted). With the Clean Start mechanism, a session starts without using an existing session. This results in simplified state management: the session data for a client is discarded only when all the messages are exchanged and not because of a network failure. The expiry times for the session states can be set with Clean Start mode, which allows the session state to be deleted by the server if the client does not connect within a certain time. This forces the client to reconnect to the server just to clean up session state. This reduces the server overload in DIL networks when multiple clients keep on appearing and disappearing from the network.

Flow Control: In a real-time environment for MQTT usage, clients and servers with different processing and connectivity levels interact, and thus they have different tolerance levels for managing in-flight messages. A client can connect to multiple MQTT servers having different restrictions and management properties on the number of in-flight messages. With flow control, all involved parties do not need to negotiate in-flight windows beforehand. Dynamic message flow adjustment is used that involves heterogeneous systems and devices. It ensures that neither the server nor the client overwhelm each other with message processing.

Bridging of Servers: Bridging of multiple servers for a distributed environment is indirectly supported in MQTT 5.0. It calls for providing subscription options to allow for message bridge applications. It also includes an option to not send messages originating on local clients and options for retaining subscribed messages. The bridge implementations in earlier versions supported a single server to be configured as a bridge, and all other servers acted as client servers connecting to the bridge, as was used in [7]. With the new MQTT version, now multiple servers on the same hierarchy can be configured to have a bridge between them.

Non-Retransmission of MQTT Messages: Earlier versions of MQTT allowed for retransmission of MQTT messages with QoS 1 and 2 in case the TCP connection broke down. If the MQTT clients are overloaded with MQTT message processing, further duplicate or new MQTT messages deprecate its performance. With MQTT 5.0, servers and clients are not allowed to retransmit messages, but instead re-send unacknowledged packets when the TCP connection is closed.

Use of Zero-Length String: For cases when data is published to a single topic, clients and servers can set a zero-length string in the publish message for the topic. It basically informs the client or server to use the previous topic instead of explicitly sending out the topic name. It further reduces the overhead in message exchange on the MQTT bus.

Federated HADR Operations

As described in [5], an HADR operation requires capabilities of multiple agencies to be federated. Apart from the agencies' assets, the existing assets from cities such as the ICT assets and on-ground humans can be used to complement, bringing effectiveness and precision to the operations. Concepts such as edge computing, crowdsourcing, and crowdsensing further enable IoT ICT operations [12]. Earlier, an architecture is described to leverage these capabilities, and the related implementation is described later.

Architecture

Figure 1 shows the architecture envisioned for an urban HADR operation where the federated MQTT servers provide SA data exchange between different parties. Here, federation refers to the standardized and agreed ontology, and thus the data exchange between the parties involved. The various components involved in the architecture are the following.

HADR Agencies' C2 Applications: HADR agencies such as the military use C2 applications to have the SA pictures of their assets, ground reports, task assignment, and so on. C2 applications from various agencies are designed for their use-case-specific requirements. These applications support different types of operations and SA behavior. Application programming interfaces (APIs) exposed by the applications are used to leverage their functionalities. Based on their interaction and use cases, they can use either REST or SOAP-based APIs to interact with their assets and external partners. Most of the HADR agencies use private APIs with API gateways as defined in the CIMIC doctrine of NATO. For the case at hand, we target applications based on IoT and in order to interact with IoT as well as legacy assets. These APIs are bound to the MQTT clients, which exchange data on behalf of the C2 application. These clients connect to the federated MQTT servers, which can be associated on the same platform where the C2 application is running or at the headquarters (HQ) level.

HADR Agencies' ICT Applications: HADR agencies such as the military, police, and fire fighters have their ICT assets deployed on the ground for operations. These assets can be legacy assets including tactical radios from the military and police, drones used for search and reconnaissance, sensors deployed at strategic areas, and so on. Due to the adoption of IoT, agencies are leaning toward use of IoT assets in their operations. Thus, the ICT applications are being redesigned or refactored to provide support for IoT asset integration. These ICT applications can either use APIs directly to expose their services or bind their API functionalities to the MQTT clients. These MQTT clients can then interface the existing APIs to their topic structures and in turn connect to the federated MQTT servers provided by the agencies.

City ICT Applications: Cities have their own ICT infrastructures for management and governance. Their assets also include legacy assets such as CCTV cameras, traffic sensors, network and cloud infrastructures, and others. With the concept of smart cities, fast advancing cities are

With flow control, all involved parties do not need to negotiate in-flight windows beforehand. Dynamic message flow adjustment is used that involves heterogeneous systems and devices. It ensures that neither the server nor the client overwhelm each other with message processing.

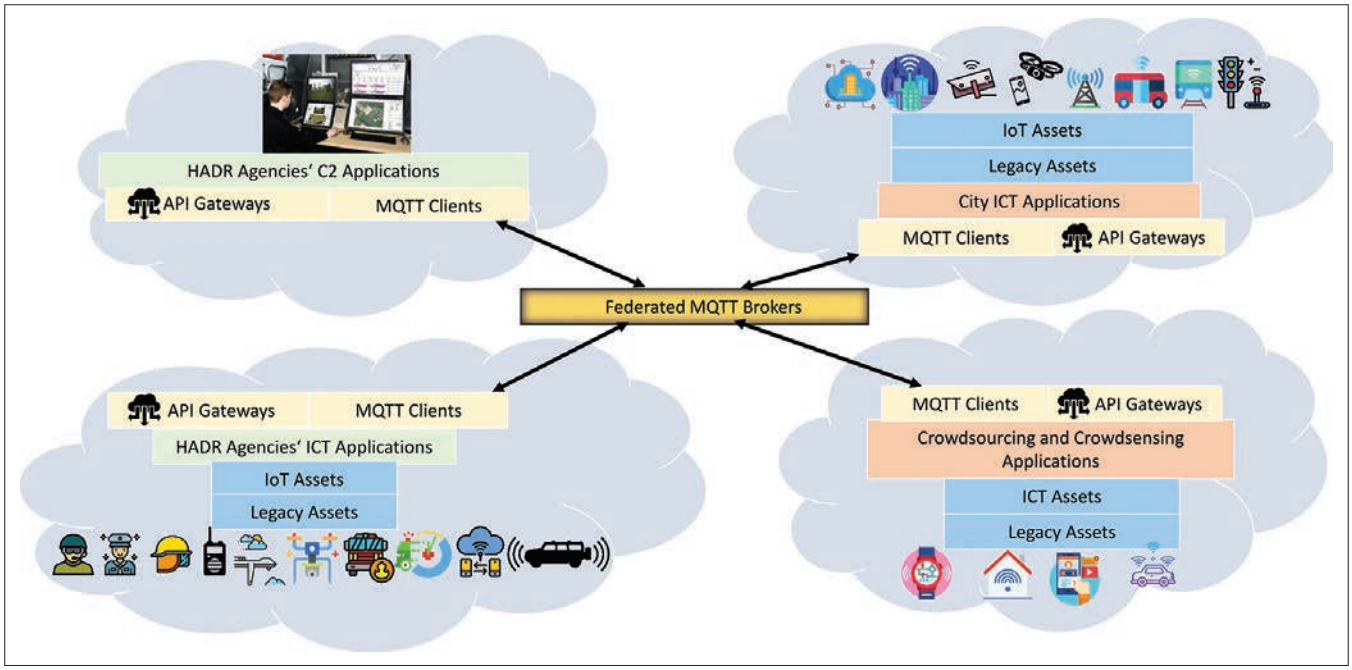

FIGURE 1. Architecture for federated MQTT-based SA exchange for HADR operations.

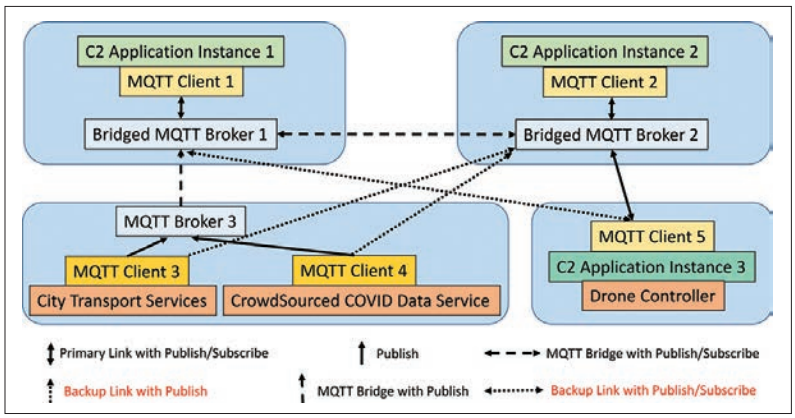

FIGURE 2. Federation of C2 systems based on MQTT-bridge for SA exchange

adopting IoT usage for their infrastructures. Concepts such as smart buildings, which provide automated sensing and building management, smart traffic, which provide dynamic traffic management, and so on are being revolutionized in cities. In turn, these assets expose their functionalities to the cities though their APIs. For our use case:

- The MQTT servers can be deployed directly by the cities, which can exchange information with other federated MQTT servers from the agencies.
- The cities use MQTT clients to connect to the federated servers provided by the agencies.
- The agencies use MQTT applications that bind to the city APIs and exchange data with the servers.

Crowdsourcing and Crowdsensing Applications: End users and private entities are one of the biggest sources of innovation in IoT. Concepts such as smart homes, smartphones, smart watches, and smart vehicles have reached out to all corners of modern society. These crowdsourced and crowdsensed data would be the eyes of ears of the future HADR operations [12]. Citizens on the ground with their smart IoT objects can report ground SA faster and more effectively since agencies' assets are mostly overwhelmed in such operations. Many of these applications use MQTT clients to exchange data with their corresponding edge and cloud applications. This MQTT data can be directly wrapped with the interfacing MQTT topics to provide direct access to the end users' device reports. Furthermore, the APIs from the service providers would be interfaced to the MQTT clients run by the agencies, which in turn connect to the federated servers.

Implementation: Federated MQTT Brokers

The proof-of-concept (PoC) implementation for a federated MQTT server-based interaction is presented in Fig. 2. The term "broker" is used instead of "server" since the implementations used in the PoC used broker in their documentation. The various components involved are outlined below.

C2 Application Instances 1 and 2: These C2 instances run the Sitaware Frontline C2 application. Custom wrappers transform the data exchanged through the MQTT topics determined for interoperable data exchange. The MQTT clients are used here as Mosquitto Version 5.0 clients that connect to VerneMQ brokers as in [10]. Brokers 1 and 2 are bridged to connect to each other to demonstrate the federation mechanism. In turn, broker 1 is bridged to broker 3.

C2 Application Instance 3: This C2 instance runs the custom developed C2 application as presented in [13]. The C2 application is based on the open source Vaadin framework and uses Openstreetmaps for showing SA data. The C2 application is bundled with an IoT-based drone controller to enable drone actuation operations. This C2 application is further interfaced with a HiveMQ Version 5.0 client and broker. MQTT client 5 in turn connects to broker 1 as the primary broker link and to broker 2 as the fallback broker in case broker 1 disconnects.

City Data Endpoint: This endpoint represents the city data services that are used for the HADR demonstration operations presented earlier. The city transport services API provides the latest SA data, including the various types of events on the street network of Germany used here. Detailed traffic information regarding traffic jams and related incidents are pushed through this API. The minute details show attributes such as the location, road names, length, significance and type of delay, and distance. To represent the crowdsourcing component in an urban HADR scenario, the ongoing COVID pandemic is used. Risklayer provides aggregated and verified crowdsourced cumulative datasets about the COVID situation in Germany [14]. It parses through official and individual crowdsourced details such as numbers of new infections per day/week/month, and deaths per state, city, district, and community. These details are updated daily and cross verified across multiple COVID data providers. Both city transport and COVID data sources expose their datasets through the determined MQTT topics and MQTT Version 5.0 publishing clients (3 and 4), which publish their data through broker 3.

Broker and Client Configuration:

- Broker 1 and 2 are bridged to each other for all the topics for both publishing and subscription at their endpoints. This means that all the clients connected to brokers 1 and 2 can publish and subscribe to the shared topics.
- Broker 1 is bridged to broker 3 for the topics, but only for subscription. Clients 3 and 4 publish topics to broker 3, which just forwards them to broker 1. But no topic data is sent back to the broker 3 or clients 3 and 4.
- MQTT clients 3 and 4 have a backup connection to broker 2, which would be used in case the link to broker 3 is lost. They can then publish their data to the bridged broker network.
- MQTT client 5 is connected to broker 2, which publishes and subscribes to the topics determined in the network.
- All brokers require user IDs and passwords for authorization.
- The brokers are configured to handle idle client connections and disconnect unresponsive ones in case they do not respond within a time interval.
- The clients probe the brokers to check if the brokers are reachable and can publish/subscribe to their messages. If such a check fails for a connection to a broker, they automatically switch over to the backup link to the next broker.
- Asynchronous messaging APIs are used in case of intermittent network disconnections. This enables the broker to store the messages destined for the clients for a predetermined time and deliver them if the client comes back online within the expiry timeframe.
- Topic filters are set at the bridged brokers to limit and regulate the topic data being sent out on the bridged network.

MQTT Topics: The MQTT topics form the base of interoperability and thus the basis for federation between the parties involved. In [5], we demonstrated the use of simple JSON-based topic structures to exchange information between the coalition partners. Furthering that topic structure and using the ontology concepts defined for IoT-lite and IoT-O [15], the following describes an example topic structure and the JSON payload:

- The topic structure in Fig. 3 shows that a device of type actuator and sub-type drone is being held by Germany under the organization header of NATO. The topic specifies that the topic intends to message the type "information," which in this case is the location of the drone.
- The JSON message in Fig. 4 contains the id of drone DEUDRONE1 at timestamp 2020-11-15 16:10:30.277125 and at location (50.618062, 7.12863).

As mentioned earlier, topic filters are used to limit the traffic on the brokers and thus on the clients. Hence, as presented previously, the drone controller is connected to MQTT client 5, which in turn connects to bridged MQTT broker 2. The topic data is directly sent over to all the subscribing parties. Otherwise, filters are used to restrict which brokers and thus clients are authorized to receive the location of the drone. This further helps to reduce unnecessary traffic overhead.

Working Use Case: The following scenario is envisioned as an example use case to demonstrate this federation:

- There is a COVID associated disaster situation in the city, and the local agencies have summoned the HADR agencies to come and assist on short notice.
- The host country provides the city transport and COVID data service through its IoT-enabled edge computing applications and IoT radio networks.
- The host country's NATO counterpart connects to the city infrastructure to receive the essential HADR-related data provided by the city. MQTT broker 3 connects to broker 1 for this purpose. The SA data is presented on C2 application instance 1 running on an armored personnel carrier (APC) vehicle.
- The invited coalition partner comes in with its assets and views the SA data on C2 application instance 2 running on a military ambulance vehicle. It establishes a bridge to the host country's MQTT broker 1, and uses pre-established topics and topic filters to exchange data. The city's MQTT broker gets the invited partner's broker details through the host country's broker list published using MQTT topics and configures a backup link to broker 2.

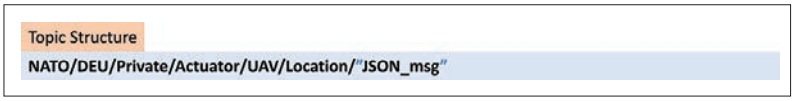

FIGURE 3. Example of MQTT topic format.

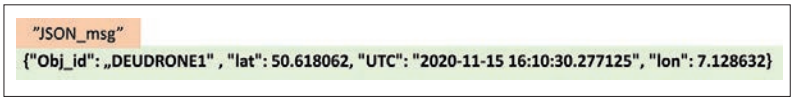

FIGURE 4. Example of JSON message.

As future work, we are setting up a federated testbed where the coalition partners can test out this federation mechanism and carry out further measurements for application of MQTT brokers in tactical environments. The new spiral for FMN is also trying to underline the concept of CIMIC by incorporating civilian standards for data exchange for future civil-military collaboration.

- A third-party HADR agency such as a non-government organization (NGO) also joins the effort by providing an IoT-based drone. It has an open-source-based C2 application that in turn is also a drone controller running on a command vehicle. It connects directly to broker 2 and configures a backup link to broker 1.

The City Transport API used for this implementation reports fine-grained details of the incidents and their locations. Also, the COVID API reports the precise numbers for minute geographical territories. Based on this data, an incident is reported at a street periphery that needs immediate assistance:

- MQTT client 3 publishes this incident through broker 3 to broker 1. C2 application instance 1 shows the incident requiring immediate evacuation.
- Since broker 1 is bridged to broker 2, they share the topics, and thus the incident is pushed to C2 application instance 2.
- The vehicle from the coalition partner is closer to the location of the incident and thus moves to the incident location reporting its status through its shared location topic to C2 application instance 1.
- The coalition partner triggers drone-based surveillance of the location through the drone controller topics to provide imagery of the affected location.
- The drone's location and imagery data (over http link [13]) is streamed to C2 application instances 1 and 2 along with the location of the coalition partner's vehicle. Figure 5 shows the location of the involved parties on the Sitware C2 application using the Military Standard 2525b and 2525c symbols.

This working use case thus demonstrates the federation mechanism using MQTT bridging and the shared filtered topics in an emulated HADR operation.

Conclusion and Lessons Learned

This article describes an MQTT-based bridging mechanism for federation between multiple MQTT brokers based on established ontology. It discusses the new features of MQTT that make it even more favorable for military IoT exchange than the last versions of MQTT. It underlines the involvement of civilian and city reports as a supplement to the military information systems for getting ground reports for effective HADR operation in urban environments. It also shows how multiple C2 systems using the bridging concept, shared topics, and topic filters can achieve federation and interoperability.

For the prototypical setting, the scenario plays fine, but in actual scenarios, the approach might come with drawbacks. It was observed that whenever an MQTT 5.0 server tried to connect to an MQTT 3.x server using the MQTT 5.0 API, it returned an error saying that the protocol version is not supported. This would definitely be an issue considering that many MQTT users might still be running older MQTT broker versions even though the federation through standard ontologies might exist. MQTT in its core uses Transmission Control Protocol (TCP)-based packets for MQTT data exchange.

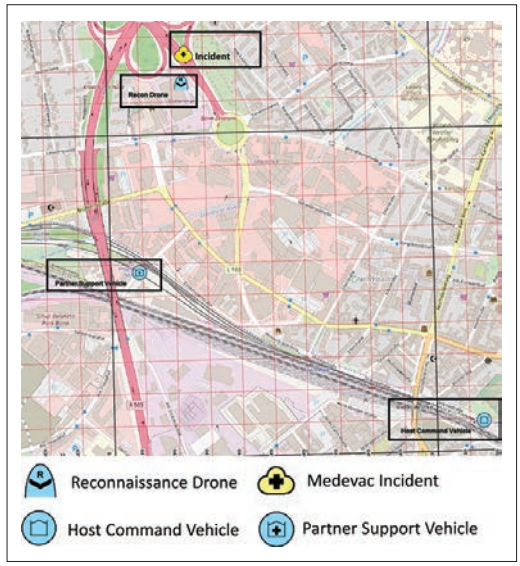

FIGURE 5. C2 status.

Although TCP ensures reliability, it always fares worse in terms of latency and throughput when transmitting large and continuous data packets. Further, the bridging mechanism will always be a bottleneck w.r.t scalability of a federated system. Clustering of distributed and logically connected servers would be needed for an HADR scenario, which might involve large numbers of connected clients creating huge amounts of network traffic.

Future Work

As future work, we are setting up a federated testbed where the coalition partners can test out this federation mechanism and carry out further measurements for application of MQTT brokers in tactical environments. The new spiral for FMN is also trying to underline the concept of CIMIC by incorporating civilian standards for data exchange for future civil-military collaboration. This proof of concept would be further developed to incorporate wrappers for military and civilian data exchange to provide real-time emulation of CIMIC. The current implementation does not involve encrypting the MQTT transmission, which will be extended in the future. Further, authentication and authorization mechanisms supported by MQTT 5.0 enhancements would be tested. The concept of creating virtual distributed clusters would be tested to address the scalability issue of the bridging mechanism. Finally, serverless solutions based on UDP would be experimented to circumvent the flaws of MQTT TCP packets.

References

[1] F. T. Johnsen *et al.*, "Application of IoT in Military Operations in a Smart City," *2018 Int'l. Conf. Military Commun. and Info. Systems*, 2018, pp. 1–8.
[2] N. Suri *et al.*, "Exploiting Smart City IoT for Disaster Recovery Operations," *2018 IEEE 4th World Forum on Internet of Things*, 2018, pp. 458–63.
[3] K. Wrona *et al.*, "Civilmilitary Collaboration in Smart Environments Under Adversarial Conditions," *1st Int'l. Wksp. Internet of Things for Adversarial Environments (in conjunction with IEEE INFOCOM 2019)*, 2019, pp. 1–6.
[4] C. Cucos, "Very High Readiness Joint Task Force — A New Concept of NATO," *Int'l. Scientific Conf. "Strategies XXI,"* vol. 2, Bucharest, Hungary, "Carol I" National Defence Univ., 2015, pp. 196–201.

[5] M. Pradhan *et al.*, "Toward an Architecture and Data Model to Enable Interoperability Between Federated Mission Networks and IoT-Enabled Smart City Environments," *IEEE Commun. Mag.*, vol. 56, no. 10, Oct. 2018, pp. 163–69.
[6] C. Pahl *et al.*, "A Container-Based Edge Cloud Paas Architecture Based on Raspberry Pi Clusters," *2016 IEEE 4th Int'l. Conf. Future Internet of Things and Cloud Wksps.*, 2016, pp. 117–24.
[7] M. Manso *et al.*, "Mobile Tactical Forces: Experiments on Multi-Broker Messaging Middleware in a Coalition Setting," *Proc. 2019 24th Int'l. Command and Control Research and Technology Symp.*, 2019.
[8] N. Suri *et al.*, "Experimental Evaluation of Group Communications Protocols for Data Dissemination at the Tactical Edge," *2019 Int'l. Conf. Military Commun. and Info. Systems*, 2019, pp. 1–8.
[9] F. T. Johnsen *et al.*, "Publish/Subscribe versus a Content-Based Approach for Information Dissemination," *IEEE MILCOM 2018*, 2018, pp. 1–9.
[10] F. T. Johnsen, M. Manso, and N. Jansen, "Evaluation of Message Broker Approaches for Information Exchange in Disadvantaged Tactical Networks in a Federated Environment," *Proc. 2020 25th Int'l. Command and Control Research and Technology Symp.*, 2020.
[11] A. Banks *et al.*, "Mqtt Version 5.0," OASIS Std., vol. 7, 2019.
[12] M. Pradhan *et al.*, "Leveraging Crowdsourcing and Crowdsensing Data for HADR Operations in a Smart City Environment," *IEEE Internet of Things Mag.*, vol. 2, no. 2, June 2019, pp. 26–31.
[13] M. Pradhan and S. Devaramani, "Enabling Interoperability for Ros-Based Robotic Devices for Smart City HADR Operations," *IEEE MILCOM*, 2019, pp. 1–6.
[14] Risklayer; http://www.risklayer-explorer.com/event/100/detail, accessed Dec. 17, 2020.
[15] F. Sivrikaya *et al.*, "Internet of Smart City Objects: A Distributed Framework for Service Discovery and Composition," *IEEE Access*, vol. 7, 2019, pp. 14,434–54.

Biography

Manas Pradhan (manas.pradhan@fkie.fraunhofer.de, manas.socgen@gmail.com) is a Ph.D. fellow in the Department of Technology Systems, University of Oslo, Norway, and the Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Germany. He received his Bachelor's degree in computer science and engineering from the Institute of Technical Education & Research, Bhubaneswar, India, in 2009. He worked as a software engineer before commencing his Master's studies in media informatics at RWTH Aachen University, Germany. His research interests lie in the area of military interoperability, Internet-of-Things technologies, and smart cities.