

Emergency Communication in Challenged Environments via Unmanned Ground and Aerial Vehicles

Lars Baumgärtner*, Stefan Kohlbrecher†, Juliane Euler†, Tobias Ritter†, Milan Stute†, Christian Meurisch†, Max Mühlhäuser†, Matthias Hollick†, Oskar von Stryk†, Bernd Freisleben*†

*Philipps-Universität Marburg, D-35032 Marburg, Germany

E-mail: {lbaumgaertner, freisleb}@informatik.uni-marburg.de

† Technische Universität Darmstadt, D-64289 Darmstadt, Germany

E-mail: {kohlbrecher, euler, ritter, stryk}@sim.tu-darmstadt.de; {mstute, mhollick}@seemoo.tu-darmstadt.de;

{meurisch, max}@tk.tu-darmstadt.de; bernd.freisleben@maki.tu-darmstadt.de

Abstract—Unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) are promising assets to support rescue operations in natural or man-made disasters. Most UGVs and UAVs deployed in the field today depend on human operators and reliable network connections to the vehicles. However, network connections in challenged environments are often lost, thus control can no longer be exercised. In this paper, we present a novel approach to emergency communication where semi-autonomous UGVs and UAVs cooperate with humans to dynamically form communication islands and establish communication bridges between these islands. Humans typically form an island with their mobile devices if they are in physical proximity; UGVs and UAVs extend an island’s range by carrying data to a neighboring island. The proposed approach uses delay/disruption-tolerant networking for non-time critical tasks and direct mesh connections for prioritized tasks that require real-time feedback. The developed communication platform runs on rescue robots, commodity mobile devices, and various drones, and supports our operations and control center software for disaster management.

I. INTRODUCTION

Rescue operations during and after disasters often expose rescue teams to high risks. Therefore, more and more unmanned vehicles (UVs) are used on the ground, in the air or in the water to support rescue operations. Typically, such UVs operate either in a semi-autonomous manner or are completely controlled by remote human operators. For example, in the ruins of Fukushima Daiichi remotely controlled robots were sent to highly contaminated areas. Remote control requires a reliable communication infrastructure to coordinate UVs and to increase their operation radius significantly. However, during the first hours of a disaster event, the existing communication infrastructure might be severely damaged, disrupted or overloaded due to network congestion. Thus, re-establishing basic ways of communication during a disaster despite fragmented IP networks and totally or temporarily disrupted network links is a key step in successful disaster management and rescue operations.

An emergency communication system should not only support UVs, but also human rescuers and civilians who are still in the disaster area. Since connectivity cannot be easily established in the entire affected area, it is more likely that small islands of devices connected to each other will evolve, with limited bridges between these islands. These islands can be formed by team members operating in the field, people trapped in houses or waiting in temporary shelters, and clusters of cooperating UVs. By using store-carry-forward technologies, humans or UVs can act as carrier pigeons and deliver data between islands, thus spreading information in an epidemic fashion. This approach can be used for person-to-person communication, in a way similar to SMS and various messengers, or for sensor data such as images from smartphones or temperature and Geiger counter readings from UVs that are shared with the public. Gathering this information and processing it in the rescuers’ operations and control center is helpful for situation analysis and coordination of rescue endeavors. In general, long distance, real-time unicast communication is not possible in such a scenario. The chances of establishing a successful multi-hop connection to a specific node are increased by altering the objectives of UVs, such as drones, to incorporate air bridges as part of their mission. In this case, a secure mesh routing algorithm is needed for real-time communication in the emergency communication system.

In this paper, we present a novel emergency communication system that relies on delay/disruption-tolerant networking (DTN) for non-time critical tasks and direct mesh connections for prioritized tasks that need real-time feedback. It is used for the distribution of sensor data, human-to-human communication, as well as direct and indirect control of unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV). To demonstrate the real-world applicability of the developed emergency communication system, called *UV4EC* (Unmanned Vehicles for Emergency Communication), we run it on *RoboCup Rescue* proven [1] robots as well commodity mobile devices, and various drones, and support

our operations and control center software for disaster management. Experimental results indicate that a combined DTN and mesh networking approach is a good compromise for providing emergency communication based on semi-automatic and remotely controlled autonomous UVs. In particular, we make the following contributions:

- We present a novel hybrid DTN and mesh networking communication middleware for disaster scenarios.
- We present a novel approach to emergency communication where a semi-autonomous UGV establishes UAV-based communication bridges between dynamically forming communication islands of mobile devices.
- We present an operations and control center for commanders to get a more complete situation overview of the disaster scenario and to control the rescue mission by sending high-level commands to mobile rescuers or UVs.

The paper is organized as follows. Section II discusses related work. In Section III, we present the design and implementation of our emergency communication system. Section IV shows experimental results from the test lab and the real world. Section V concludes the paper and outlines areas of future work.

II. RELATED WORK

Several approaches that address emergency communications needs have been presented in the literature [2]. Some of them require special hardware (e.g., radio-link technologies or satellites), are engineered for specific tasks, or are not usable on consumer-grade hardware with commodity operating systems [3], [4], [5], [6], [7], [8], [9], [10]. Since our focus is to enable as many people as possible to communicate in an emergency network, our approach attempts to leverage commodity hardware and software, such as WiFi-enabled mobile end-user devices, where possible.

Other research has shown that DTNs and mesh networks are viable solutions in emergency scenarios [11], [12], [13], [14]. By leveraging ubiquitous mobile devices, such as smartphones or tablets, novel communication solutions have been proposed [15], [16], [17], [18], [19] and evaluated [20]. Since both DTNs and mesh networks have distinct capabilities, we incorporate a dual networking stack in *UV4EC*, which uses mesh networking for real-time communication and DTN for data sharing and messaging.

While the use of both UAV and UGV systems (and their combination) for search and rescue applications has received considerable attention from the research community [21], the combination of robots with both DTN and mesh networking has not been studied extensively. However, based on the experiences with real-world deployments [22], [23] and the associated communication difficulties, this appears to be a promising direction of research. While mesh-based approaches have been used successfully with UVs before [24], the body of research on the additional use of DTN with UVs is limited to mainly theoretical results [25], [26], [27].

Collecting, analyzing, and visualizing emergency data at a central point (e.g., in a control center) to provide a situation

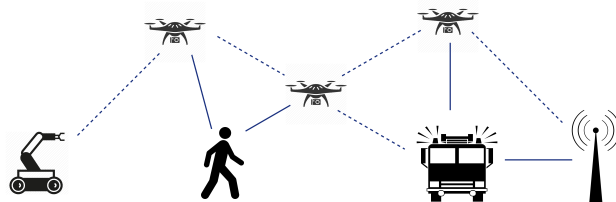


Fig. 1: Emergency communication scenario

overview for rescuers is crucial in disaster management [28]. However, many proposed control centers are stationary and rely on centralized approaches, assuming Internet connectivity to exchange data or to control UGVs [29]. Only a few control centers have been designed for mobile use and decentralized communication [30], [31]. For instance, the SENEKA project presents an adaptable and scalable ground control station integrated in a van for gathering sensor data from stationary sensors, mobile ad hoc networks and mobile sensor platforms [31]. In addition to the data channel, it establishes a control channel to control UVs in the field. Inspired by this work, we present a lightweight offline control center, running on a customary laptop for mobile use and relying on mesh networking and DTN for decentralized communication while providing a similar range of functions as state-of-the-art approaches, namely data gathering, fusion and processing, providing visual situation overviews, as well as realtime and delay-tolerant control of semi-autonomous UAVs and UGVs.

III. UV4EC

The design of *UV4EC* supports unmanned vehicles and professional responders as well as civilians in the affected area. All communication links can be volatile and connectivity is expected to be lost. Given an opportunity to exchange data, each peer will communicate with all of its neighbors for maximum data distribution, as indicated in Fig. 1.

A. Unmanned Ground Vehicles

The UGVs used in *UV4EC* employ ROS as a robotics middleware. ROS has become the de-facto standard in the robotics community in recent years. Based on a modular software design, components for different tasks like navigation and perception can be exchanged and adjusted for use with different vehicle platforms. For *UV4EC*, we use a tracked mobile robot based on the Taurob Tracker platform. To support multimodal sensing in disaster environments, the robot is equipped with a comprehensive sensor suite consisting of (among others) a spinning 3D LIDAR, a thermal camera, a 30x optical zoom camera, and a depth camera. The camera sensors are mounted on a sensor arm to provide flexibility in sensor positioning. The robot can perform autonomous exploration of unknown environments, using a Simultaneous Localization and Mapping (SLAM) approach to learn the environment and map and simultaneously localize objects within the environment [32]. The robot operator can select between teleoperation or autonomous operation at any time.

B. Unmanned Aerial Vehicles

Since we concentrate on communication aspects in this paper and to avoid the (administrative and organizational) overhead of deploying real UAVs, we use multiple simulated UAVs based on the Gazebo simulator. It allows us to simulate multiple UAVs with dynamics. The UAVs establish communication bridges between UGVs and our operations and command center (OCC) from time to time, i.e., the UAVs should position themselves near specific prescribed locations between UGVs and the OCC to enable communication. The task of allocating and getting to these positions is solved cooperatively and decentrally; necessary control inputs for each UAV are computed locally on board of the UAVs by applying a feedback and optimization-based vehicle controller [33], [34]. This vehicle controller uses a mixed logical dynamical (MLD) approach to model the multi-vehicle system [35] consisting of every UAV within the communication radius. Based on the MLD formulation, an optimal control problem is set up, in which the number of and distance to unoccupied bridge locations is minimized subject to vehicle dynamics, allocation logic and constraints regarding collision avoidance. This optimal control problem is solved over a limited time horizon in a model predictive control fashion so that a mixed integer linear program has to be solved in every time step. The first elements of the resulting control input sequence are applied by each individual UAV.

C. Emergency Communication

In disaster scenarios, a number of communication services and applications are required to ensure effective disaster response. These applications include human-to-human (message-based) communication, sensor data sharing for situational reporting, and also “real-time” control channels for operating UGVs and UAVs from a remote location. Unfortunately, during a disaster, the local communication infrastructure, such as cell towers and Internet-connected WiFi hotspots that would normally be used to support these applications, might be unavailable either due to poor coverage at the disaster site or because it has been destroyed by the disaster.

Thus, we rely on decentralized ad-hoc communication technologies to provide connectivity in such extreme environments. To enable a wide range of applications with different communication requirements, we employ a dual networking stack consisting of (i) a mobile cloud based on delay/disruption-tolerant networking that can be used for messaging and information sharing applications, and (ii) a highly adaptive end-to-end communication protocol based on wireless multi-hop routing that quickly finds and exploits communication bridges and can be used for control applications with “real-time” feedback. We present the two components in detail below.

1) *Mobile Cloud*: In a network consisting of many mobile and fast-moving nodes including UGVs and UAVs, The contact times between two nodes are typically short. When communication links are only available for short periods and in the near vicinity, classical multi-hop routed network

communication is hardly possible. As an alternative, store-carry-forward based DTN has been proposed especially for challenged environments such as space communication and emergency communication. In DTN, nodes are considered as “data mules”: they carry their own messages as well as messages from others. When two nodes meet, they exchange all messages they are carrying by replicating them. Using this approach, messages are spread in an epidemic fashion throughout the network, which increases the chances that a message eventually arrives at its destination. In DTN, delivery performance highly depends on node mobility. This makes it attractive for a disaster scenarios where mobile devices of civilians and professional disaster response staff as well as UGVs and UAVs are present and physically move around. Due to its best-effort service, DTN is suitable for applications such as text messaging, collecting sensor data, and transmitting geo-location updates.

We rely on the Serval Project [36] to realize a DTN protocol for *UV4EC*. Serval provides basic messaging with built-in end-to-end encryption and a portable C code base for further extensions. To verify the viability of Serval in our disaster scenario, we conducted an in-depth performance analysis of the existing protocol and software [37]. Based on our results, we reduced the protocol overhead by dynamically adjusting the neighbor announcement interval [38] and implemented transparent computation task offloading to preserve local resources [39]. To allow the integration of Serval in *UV4EC*, we implemented pre- and post-receive message filters as well as content hooks that are triggered when manipulating the data store¹². Furthermore, we support more advanced features such as append-only sensor logs and file updates for rapid prototyping and deployment in the field.³ For non-mobile users, custom ways of interfacing with the Serval application are currently in development, including a web interface⁴, standalone desktop application⁵, and a full-screen console application⁶. The final system enables any WiFi-enabled UNIX-based system, such as smartphones running Linux, Android, or MacOS, to communicate with each other. This includes one-to-one encrypted text messaging and file transfers, and publicly shared information such as images or position logs. It can easily be ported and used by new specialized systems such as rescue robots, drones or static sensor nodes in addition to the ones we are already using.

2) *Communication Bridges*: Certain applications, such as directly controlling a UGV or UAV, require real-time communication. This cannot be realized using our mobile cloud, since it does not provide delivery guarantees or feedback. Therefore, control messages might never or only very lately reach their destinations, which would render UGV/UAV control unreliable. To solve this problem, we resort to classical

¹<https://github.com/umr-ds/serval-contentfilters>

²<https://github.com/umr-ds/serval-dna/tree/nicer-filters>

³<https://github.com/gh0st42/servalshells>

⁴<https://github.com/umr-ds/serval-web>

⁵<https://github.com/gh0st42/ServalDesktopApp>

⁶<https://github.com/gh0st42/sdnatui>

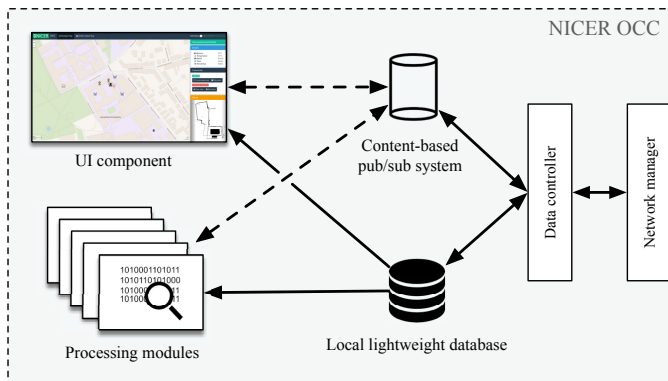


Fig. 2: Architecture of *NICER OCC*

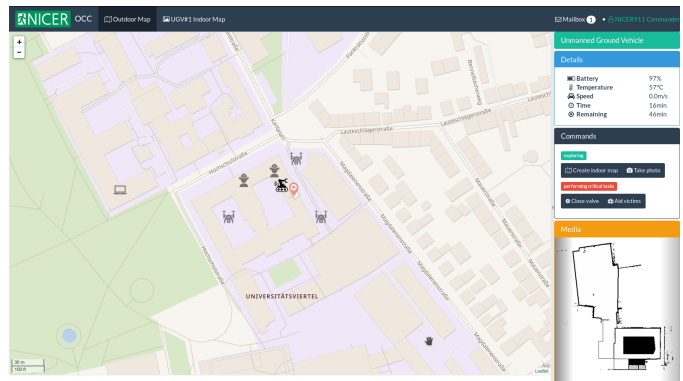
store-and-forward routing where messages are forwarded from one node to the next node until it reaches its destination. If a destination is not reachable, for example, because the network is partitioned or the node has moved out of range, messages should be dropped: in this case, there is no need to buffer packets, since we are not interested in late deliveries. However, the system should be able to notify the user in a timely manner that communication was unsuccessful or that an ongoing communication was interrupted.

For *UV4EC*, we employ SEMUD [40] that provides an end-to-end communication protocol. It supports on-demand route discovery and constantly adjusts its active routing paths. This is important for a highly mobile network where topology changes are frequent. In other words, SEMUD maintains a stable route while there is active communication between two nodes, for example, between the operator's device and a UGV; and does not generate any traffic while end-to-end communication is inactive. This is in contrast to classical proactive routing protocols for mesh networks such as OLSR [41]. SEMUD achieves these goals by a combination of per-packet feedback, the integration of routing with actual data transmission, the use of reliability as a distance metric, and lightweight cryptographic mechanisms. These features allow SEMUD to be highly adaptive to topology changes and resilient to a wide range of common attacks on wireless routing protocols even in the case that a device is compromised or captured by an adversary. Furthermore, a prototypical and portable C++ implementation of the protocol is freely available⁷.

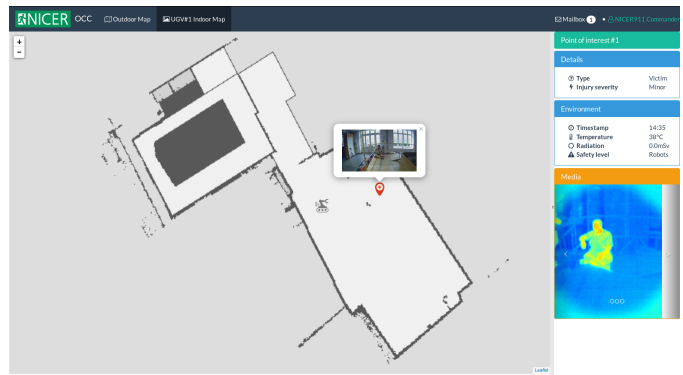
D. Operations and Control Center

To provide a more complete situation overview and coordinate rescue endeavors, we have built a user-friendly operations and control center (termed *NICER OCC*), that is commonly located outside of the affected region. It comprises (1) *data gathering, fusion and processing*, (2) *visualization*, and (3) *coordination* of mobile rescuers or semi-autonomous unmanned vehicles (UVs) with specific focus on high-level commands, namely *exploring* (e.g., mapping, taking photos) or *performing critical tasks* (e.g., closing a valve). To show the applicability

⁷<https://seemoo.de/semud>



(a) Situation overview (*outdoor map*)



(b) Detailed view within a building (*indoor map*)

Fig. 3: UI component of *NICER OCC*

for the outlined disaster scenario, we implemented a proof-of-concept prototype that relies on the previously described communication network and can run on a customary laptop.

1) *Data Gathering, Fusion and Processing*: The *NICER OCC* strives to collect a comprehensive picture about the situation and the interactions of mobile rescuers and UVs in the field. Pursuing a loosely coupled and event-driven architecture (cf. Fig. 2), the system or the *data controller* first gathers and fuses sensor data of multiple nodes from the decentralized ad-hoc communication network; and second, (a) persistently stores them in a *local lightweight database* for historical views, as well as (b) publishes them to a *content-based pub/sub system* for push updates to the subscribed modules, such as *processing modules* or the *UI component*. Subscribed processing modules can then asynchronously analyze the data (e.g., image editing, object recognition) and publish the results back to the pub/sub system, where other subscribers (e.g., the UI component) can use them. The computational workload (i.e., processing modules) can also be moved to the network by integrating in-network processing approaches [42], which is especially designed for such infrastructure-less scenarios. In urban environments, the system can also utilize upgraded home routers as DTN communication bridges and computing nodes [43].

2) *Visualization*: The *UI component* of the system provides a visual overview of the situation by displaying the collected and processed sensor data (cf. Fig. 3). We use modern lightweight web technologies to accelerate UI development, to provide an easy to learn and use application, as well as to easily distribute and port our system (*cross-platform compatibility*) in infrastructure-less disaster scenarios.

Figure 3a shows the offline (*outdoor map*), which is supposed to provide an entire situation overview for the commanders. It displays and updates all *units* in the field (e.g., UVs, mobile rescuers) who are at least equipped with communication technology and location sensor, *points of interest* (POIs), and overlaid *indoor maps*. By clicking on a unit, a detail view on the right side of the map appears, representing all relevant collected information (e.g., health status) and possible high-level commands (e.g., take a photo). The same is true for a POI, which geographically marks a situation that the commanders may find interesting, e.g., detected victims or damages. Receiving new data (e.g., status or location updates) by listening on the pub/sub system, the UI component only updates the visual representation of the referenced unit or object without refreshing the whole map.

To provide a more detailed view for inaccessible buildings (e.g., a nuclear reactor) to the commanders, the *NICER OCC* can display and update indoor maps created via laser scanning by on-site UGVs (cf. Fig. 3b). The indoor map view is based on the user interface of the outdoor map to apply same interaction concepts.

3) *Coordination*: The *NICER OCC* also enables commanders to coordinate the units in the field by sending high-level commands. We define two types of commands to control semi-autonomous UAVs and UGVs, namely (i) *exploring* (e.g., mapping, taking photos), and (ii) *performing critical tasks* (e.g., closing a valve; handing special equipment, emergency rations or medical kits over to injured people). Internally, these commands are encoded and sent via messages over the available network. For a high-priority command or a direct control of an UV, the *NICER OCC* can easily request a so-called *communication bridge* from the units in the field (cf. Section III-C2). The *NICER OCC* can also be extended by integrating civilians who use our mobile application for smartphones (cf. [44]).

IV. EXPERIMENTAL EVALUATION

We evaluate *UV4EC* by performing real-world tests (Section IV-A) and performing comprehensive simulation and emulation of various nodes in realistic setups (Section IV-B) based on the generated data.

A. Real World Setup

To provide a test scenario that can be transferred to real applications, we emulate real-world disaster scenarios by relying on well-established approaches for evaluating disaster response robots. This includes NIST standard test methods [45] that are used in the RoboCup Rescue competition. They are designed to provide reproducible test setups representative

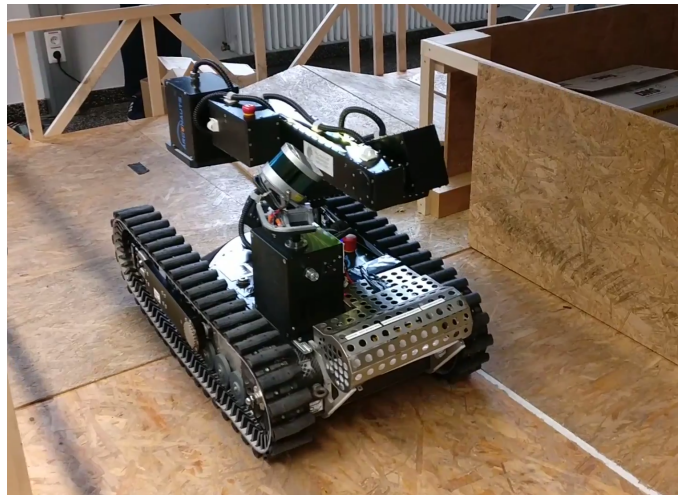
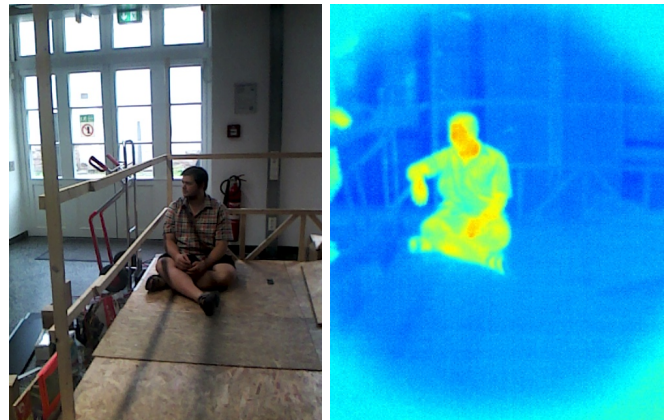


Fig. 4: Hector Tracker robot operating in a simulated disaster scenario.



(a) RGB Image

(b) Thermal Image

Fig. 5: Robot sensor data for a "victim found" event

for the challenges encountered in real disasters. Figure 4 shows and image of our robot operating in the scenario.

To simulate the use of a robot for victim search in a disaster scenario, multiple persons are placed in a simulated disaster zone, simulating trapped victims. The Hector Tracker vehicle is then used to explore the environment and search for victims. The robot can be controlled by teleoperation or can operate fully autonomously, and the used control paradigm can be changed at any time. Using onboard sensors, it generates a map of the environment including geometric (point cloud) data and found victims or objects of interest.

Communication requirements for robot operation depend on the operation mode. For fully autonomous control, neither up- or downlinks are required, but often desirable to monitor the vehicle. For pure teleoperation, connectivity between robot and operator station must be established. The downlink (robot to operator) direction then requires the transmission of low latency image data.

Independent of the operation mode, the robot's mission

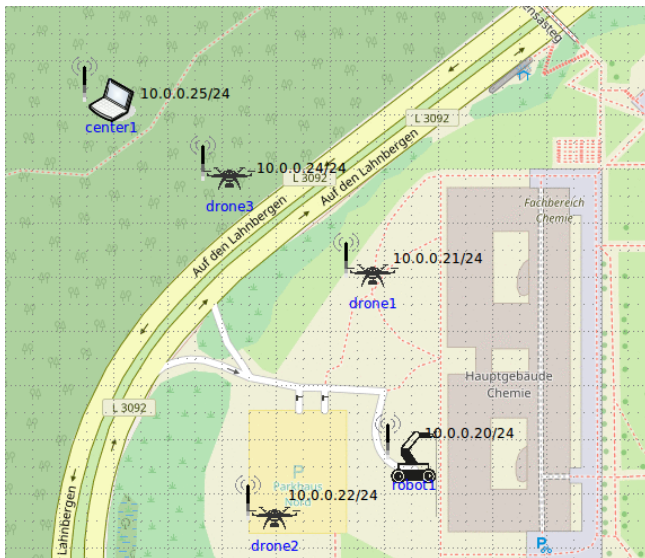


Fig. 6: Simulation setup with three drones, one ground robot and a command center operating on our university campus

state needs to be communicated as the main deliverable for responders. It mainly consists of robot pose, 3D environment map as well as event-based tracking of objects of interest, such as finding a trapped victim. Since all related computation tasks take place onboard the robot, all mission state related information can be communicated in a delay-tolerant manner. This motivates the use of image data for evaluating communication performance in Section IV-B. Figure 5 shows example camera imagery that is associated with a "victim found" event onboard the robot when a victim has been automatically detected.

B. Lab Test Environment

To get a basic understanding of the performance of *UV4EC*, we performed several experiments in a controlled simulated and emulated environment derived from scenarios faced in Sec. IV-A. The following results can be directly transferred to real world deployments, due to our focus on emulation of systems instead of simulated algorithms.

1) *Test Environment*: All tests were performed on an i7-4771 CPU @ 3.50GHz, supporting 8 threads, with 32 GB of RAM. The nodes were simulated using a combination of the Common Open Research Emulator (CORE)⁸, Gazebo⁹ robot simulator and ROS¹⁰. The system was designed with a static operations and command center node *center1*, a robot slowly moving in one direction until it gets stuck at a building entrance *robot1* and three highly mobile drones *drone1-3* with various objectives. These drones either have fixed points of interest for their mission or circle around specific positions. Furthermore, they periodically seek contact to *center1*, and *robot1*. The setup is shown in Figure 6 where an accident in the chemistry building of our university is simulated. In

⁸<https://www.nrl.navy.mil/itd/ncs/products/core>

⁹<http://gazebo.org/>

¹⁰<http://www.ros.org/>

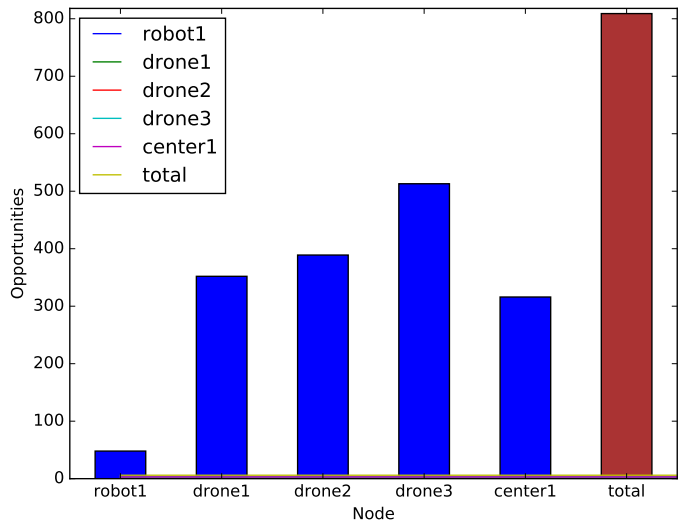


Fig. 7: Opportunities for data exchange

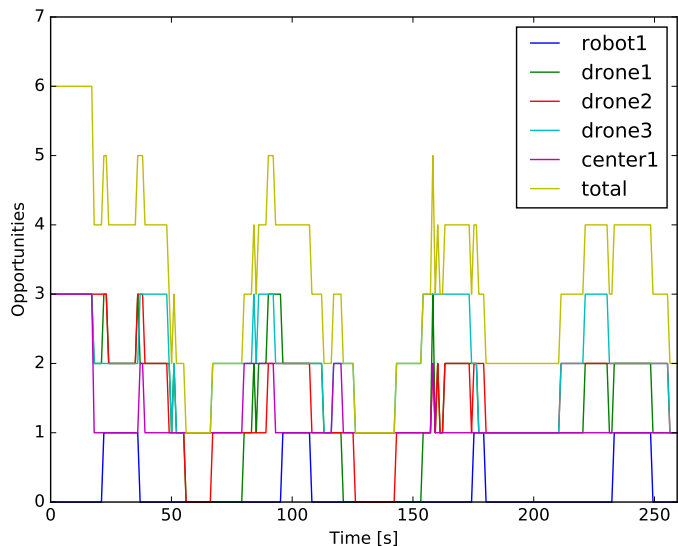


Fig. 8: Opportunities for data exchange over time

our simulations, the area covered by all involved entities is about $250,000 \text{ m}^2$. Each node has a 802.11g WiFi interface in ad-hoc mode for direct mesh communication. Therefore, the bandwidth cannot exceed 54 Mbit/s and has a maximum simulated range of 80 meters. Each experiment was repeated 10 times and ran for about 260 seconds.

2) *Communication Opportunities*: For delay-tolerant communication, it is important how often peers get a chance to exchange their data with others. In our simulation, we determined, using one second intervals, how many peers are currently in communication range.

Overall, there were over 800 total contact opportunities in the entire simulation across all nodes. The individual number of contacts per node is shown in Figure 7. The number of opportunities over the runtime per node is depicted in Figure 8. It shows that within this area of action and without any specific targets for the autonomous drones regarding communication,

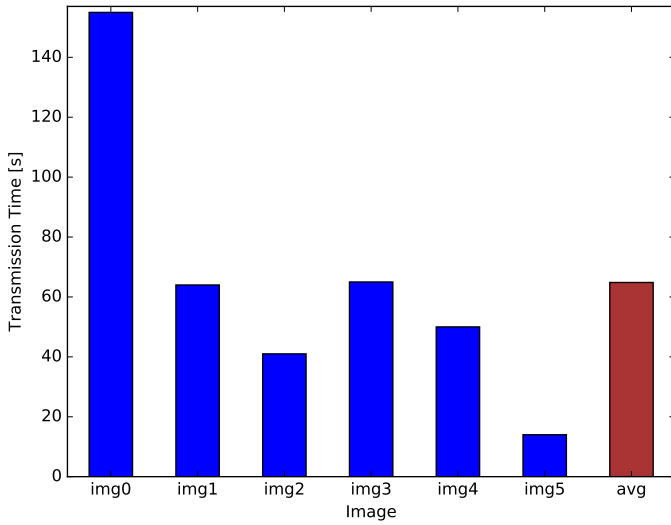


Fig. 9: File distribution times

there are plenty of opportunities to exchange data, even though the mobile drones have more opportunities than the rather static or remote nodes, such as *center1* and *robot1*.

3) *Delivery Times*: Knowing that many opportunities exist for data exchange in our scenario, we must still determine how fast responses to individual commands can be expected in such circumstances.

Two different tests were performed to analyze the delivery times. The first test injects image data at *robot1* and measures how long it takes for this data to arrive in the operations and control center (*center1*). In the second test, *center1* wants a direct connection to *robot1* and therefore sends the commands for an air-bridge to all three drones. We measured how long it takes for all drones to receive the command. Both experiments were repeated with different starting times and therefore different geo-spatial distributions of nodes.

File distribution depends on the file size. In our experiments, we sent 1080p images as they are recorded by our physical robot. The transmission times are shown in Figure 9. Average file transmission time is about 60 seconds with the worst time of about 155 seconds. Without this image transmission, the average time goes down to about 46 seconds. This can be explained with bad timing, no peers in range or moving out of range during transmission, and Serval needing some time to recover from failed transmissions. The transmission always needs at least 2-3 hops to travel from *robot1* to *center1*, considering that the area of action covered by an average transmission time of 60 seconds is more than acceptable.

When it comes to message delivery to the drones, the picture is similar. If messages are sent faster than they can be delivered, they are transmitted together, arriving at the same time even although they have been sent at different points in time. The arrival times are shown in Figure 10. Since text messages are much smaller than images, the chances of a successful transmission even with only short contact periods is much higher, resulting in an average transmission time

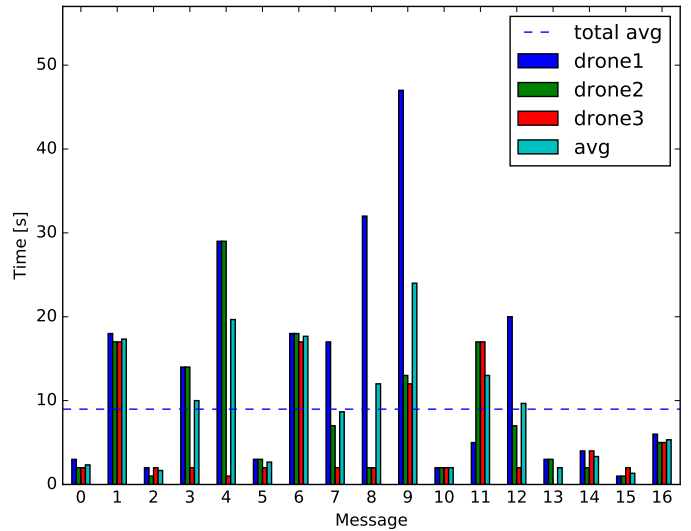


Fig. 10: Message distribution times

of about 9 seconds in our scenario. Considering the worst delivery rates recorded in our experiments, a message still reaches its destination in under one minute. Thus, even with the rather low number of participants in our simulation, direct control can be triggered within a quite short period of time.

All these numbers are highly scenario specific, and by having humans with network devices also in the affected area, even better delivery rates can be achieved. Having the drones programmed with multiple objectives, one seeking opportunities for data exchange, gives a powerful tool to dynamically adapt to the environment and the tasks at hand.

4) *Connection Establishment*: We also evaluated the responsiveness of our communication system to newly established communication bridges, i.e., when drones have moved into positions to allow a multi-hop connection from the control center to a robot. For the experiment, we let *center1* generate UDP/IP traffic to *robot1* using the iPerf tool.¹¹ We recorded the incoming packets at *robot1* over time using *tcpdump* and plotted their arrival times in Fig. 11. The figure also marks the point in time when the drones have aligned and the communication bridge has been established.

We can see that SEMUD almost instantaneously is able to deliver messages to the destination after the communication bridge has become active. This is due to SEMUD's design of integrated route discovery and data transmission: there is no explicit protocol that needs to be run prior to actual data transmission, but data is piggybacked to every message that is sent, which keeps delay at a minimum. In addition, SEMUD essentially follows a trial-and-error approach: every message is flooded through the network until feedback from the destination is received, in which case SEMUD switches to unicast transmissions to reduce network load. In contrast, OLSR is significantly slower in exploiting the communication bridge: it needs approximately 17 seconds to find a path. This

¹¹iPerf website: <https://iperf.fr>

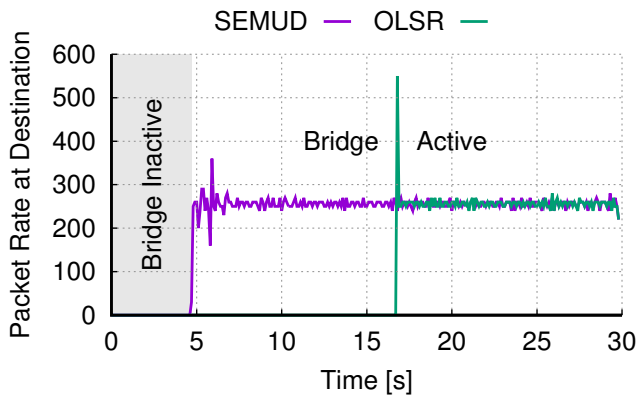


Fig. 11: Reaction to a newly established communication bridge of SEMUD and OLSR

is due to the fixed interval at which OLSR exchanges routing information. Note that this additional delay is not a one-time cost, but has to be paid whenever the topology changes. This is especially problematic when long-term continuous control of a moving robot is required and more drones are added to maintain the bridge.

5) *Message Processing*: Besides the delivery time of messages through the network, it is also important that the system has low message processing times at the network devices to ensure fast responses. Therefore, we evaluated our NICER OCC by system performance tests with respect to the entire message lifecycle between the network and the UI, i.e., from the arrival of the message at the OCC through to the visualization of the contained sensor data in the UI or on the map, so that the commanders can perceive them.

We considered both directions of message processing, (a) reading *sensor data* from the network (including deserialization) and updating the UI, as well as (b) sending *commands* from the UI to the network (including serialization). As performance measures, we use *latency* and *throughput*. Latency is the time required to process one message. Throughput is the number of such messages processed per unit of time. It is important to note that one message can contain multiple sensor data such as location updates. However, to analyze the different message types and make the tests comparable, we limit the number of sensor data or commands to one per message for our lab test.

Message type	Latency [ms]	Throughput [1/s]
Sensor data (location)	26.1 ± 1.3	53.8 ± 4.0
Sensor data (image)	76.6 ± 2.5	12.5 ± 0.5
Commands	69.1 ± 1.8	19.6 ± 0.3

TABLE I: Performance test of OCC w.r.t message processing

Table I shows the results of our performance tests. We can observe that messages with different kinds of sensor data are processed differently: a message with location data needs about 26 ms to be processed in our OCC while messages with image content need roughly three times more (~ 76 ms). This is also reflected in the throughput measurements. Considering

the other direction, commands can be sent in about 70 ms from the UI to the network, which is more than sufficient to be able to smoothly control the rescue mission.

All in all, the message processing of our operations and control center is negligible compared to the delivery times of the messages through the network. Thus, there is no lack of performance for the OCC; quite the contrary, the OCC supports situation overviews with high-frequency updates in the magnitude of the human eye or screen refresh rate.

V. CONCLUSION

In this paper, we have presented *UV4EC*, a novel emergency communication system involving unmanned vehicles. By combining a DTN-based mobile cloud infrastructure for non-time-critical tasks with reactive mesh routing for real-time interaction, novel ways of operating UGVs and UAVs are provided that also support humans in the affected area. UGVs and UAVs are used to bridge communication gaps that otherwise would significantly reduce their radius of operation. We also have developed a lightweight operations and control center that complements *UV4EC* to provide all relevant functionalities (i.e., data gathering and processing, providing visual situation overviews, and sending high-level commands) to commanders for coordinating rescue missions. Furthermore, we have demonstrated the viability of our approach by experimental evaluation and real world deployment.

There are several areas of future work. For example, additional work should be invested in developing further DTN-aware control mechanisms for various robot operations. Furthermore, the coordination of the UAVs can be fine tuned and optimized for specific scenarios. Depending on the tasks at hand, data prioritization should be considered at the UAV/UGV, DTN and/or routing levels for improved network performance. Finally, for low-priority tasks (e.g., analyzing local sensor data), we plan to move the computational workload from the control center to the network, i.e., the data should already be analyzed by the nodes in the network.

ACKNOWLEDGMENT

This work is funded by the LOEWE initiative (Hessen, Germany) within the NICER project and the Deutsche Forschungsgemeinschaft (DFG, SFB 1053 - MAKI).

REFERENCES

- [1] H. L. Akin, N. Ito, A. Jacoff, A. Kleiner, J. Pellenz, and A. Visser, "Robocup rescue robot and simulation leagues," *AI magazine*, vol. 34, no. 1, p. 78, 2012.
- [2] P. D. Pradeep, B. A. Kumar *et al.*, "A survey of emergency communication network architectures," *International Journal of u-and e-Service, Science and Technology*, vol. 8, no. 4, pp. 61–68, 2015.
- [3] D. Fan, B. Li, and M. Chen, "Design of global emergency mobile communication system based on tdrss," in *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, June 2016, pp. 322–325.
- [4] K. Igarashi, K. Umeno, M. Okada, and M. Kikuchi, "Study on emergency message communication system for ensuring safety in antarctica under extremely severe environments," in *2016 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, Oct 2016, pp. 116–119.

- [5] M. Berioli, N. Courville, and M. Werner, "Emergency communications over satellite: the WISECOM approach," in *16Th IST Mobile and Wireless Communications Summit*. IEEE, 2007, pp. 1–5.
- [6] W. Wang, W. Gao, X. Bai, T. Peng, G. Chuai, and W. Wang, "A framework of wireless emergency communications based on relaying and cognitive radio," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2007, pp. 1–5.
- [7] K. Heimerl and T. S. Parikh, "How users understand cellular infrastructure," EECs Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2012-42, Apr 2012. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-42.html>
- [8] K. Heimerl, K. Ali, J. Blumenstock, B. Gawalt, and E. Brewer, "Expanding rural cellular networks with virtual coverage," in *10th USENIX Symp. on Netw. Systems Design & Implementation*, 2013, pp. 283–296.
- [9] T. Pecorella, L. S. Ronga, F. Chiti, S. Jayousi, and L. Franck, "Emergency satellite communications: research and standardization activities," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 170–177, 2015.
- [10] M. Manic, D. Wijayasekara, K. Amarasinghe, J. Hewlett, K. Handy, C. Becker, B. Patterson, and R. Peterson, "Next generation emergency communication systems via software defined networks," in *Third GENI Research and Educational Experiment Workshop*. IEEE, 2014, pp. 1–8.
- [11] A. S. Cacciapuoti, F. Calabrese, M. Caleffi, G. Di Lorenzo, and L. Paura, "Human-mobility enabled wireless networks for emergency communications during special events," *Pervasive and Mobile Computing*, vol. 9, no. 4, pp. 472–483, 2013.
- [12] H. Chenji and R. Stoleru, "Delay-tolerant networks (dtns) for emergency communications," *Advances in Delay-tolerant Networks (DTNs): Architecture and Enhanced Performance*, p. 105, 2014.
- [13] C. Köbel, W. B. Garcia, and J. Habermann, "A survey on wireless mesh network applications in rural areas and emerging countries," in *IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 2013, pp. 389–394.
- [14] D. Reina, J. Coca, M. Askalani, S. Toral, F. Barrero, E. Asimakopoulou, S. Sotiriadis, and N. Bessis, "A survey on ad hoc networks for disaster scenarios," in *International Conference on Intelligent Networking and Collaborative Systems (INCoS)*. IEEE, 2014, pp. 433–438.
- [15] H. Ntareme, M. Zennaro, and B. Pehrson, "Delay tolerant network on smartphones: applications for communication challenged areas," in *Proc. of the 3rd Extreme Conf. on Communication*. ACM, 2011, pp. 14–21.
- [16] J. Thomas, J. Robble, and N. Modly, "Off-grid communications with Android meshing the mobile world," in *2012 IEEE Conference on Technologies for Homeland Security*, 2012, pp. 401–405.
- [17] "Briar," <https://briarproject.org/index.html>, (Accessed on 06/25/2016).
- [18] O. Garden, "Firechat," <http://opengarden.com/about-firechat>, (Accessed on 06/25/2016).
- [19] Z. Lu, G. Cao, and T. La Porta, "Networking smartphones for disaster recovery," in *International Conference on Pervasive Computing and Communications (PerCom'16)*. IEEE, 2016, pp. 1–9.
- [20] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communication Magazine*, vol. 52, no. 4, pp. 56–65, Apr. 2014.
- [21] R. Murphy, S. Tadokoro, D. Nardi, A. Joacoff, P. Fiorini, H. Choset, and A. Erkmén, "Search and rescue robotics, Fundamental Problems and Open Issues in Handbook of Robotics, eds. Siciliano, Bruno; Khatib, Oussama," 2008.
- [22] R. R. Murphy, "Trial by fire [rescue robots]," *IEEE Robotics & Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.
- [23] T. Yoshida, K. Nagatani, S. Tadokoro, T. Nishimura, and E. Koyanagi, "Improvements to the rescue robot quince toward future indoor surveillance missions in the fukushima daiichi nuclear power plant," in *Field and Service Robotics*. Springer, 2014, pp. 19–32.
- [24] A. Hart, N. Pezeshkian, and H. Nguyen, "Mesh networking optimized for robotic teleoperation," Space and Naval Warfare Systems Center San Diego, CA, Unmanned Systems Branch, Tech. Rep., 2012.
- [25] C. Luo, P. Ward, S. Cameron, G. Parr, and S. McClean, "Communication provision for a team of remotely searching uavs: A mobile relay approach," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 1544–1549.
- [26] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "Connectivity-aware planning of search and rescue missions," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2013, pp. 1–8.
- [27] J. A. Dias, J. N. Isento, V. N. Soares, and J. J. Rodrigues, "Impact of scheduling and dropping policies on the performance of vehicular delay-tolerant networks," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.
- [28] M. Frassl, M. Lichtenstern, M. Khider, and M. Angermann, "Developing a system for information management in disaster relief-methodology and requirements," in *7th International Conference on Information Systems for Crisis Response And Management (ISCRAM'10)*, vol. 1, 2010.
- [29] A. M. Khaleghi, D. Xu, S. Minaeian, M. Li, Y. Yuan, J. Liu, Y.-J. Son, C. Vo, A. Mousavian, and J.-M. Lien, "A comparative study of control architectures in uav/ugv-based surveillance system," in *IEE Annual Conference. Proceedings*. Institute of Industrial Engineers-Publisher, 2014, p. 3455.
- [30] H.-B. Kuntze, C. W. Frey, I. Tchouchenkov, B. Staehle, E. Rome, K. Pfeiffer, A. Wenzel, and J. Wöllenstein, "Seneka-sensor network with mobile robots for disaster management," in *IEEE Conference on Technologies for Homeland Security (HST'12)*. IEEE, 2012, pp. 406–410.
- [31] H.-B. Kuntze, C. Frey, T. Emter, J. Peterleit, I. Tchouchenkov, T. Mueller, M. Tittel, R. Worst, K. Pfeiffer, M. Walter *et al.*, "Situation responsive networking of mobile robots for disaster management," in *Proceedings of ISR/Robotik 2014: 41st International Symposium on Robotics*. VDE, 2014, pp. 1–8.
- [32] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, O. von Stryk, and U. Klingauf, "Hector open source modules for autonomous mapping and navigation with rescue robots," in *Proceedings of 17th RoboCup International Symposium*, 2013.
- [33] J. Kuhn, C. Reinl, and O. von Stryk, "Predictive control for multi-robot observation of multiple moving targets based on discrete-continuous linear models," in *Proceedings of the 18th IFAC World Congress*, 2011, pp. 257–262.
- [34] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk, "Decentralized dynamic data-driven monitoring of atmospheric dispersion processes," *Procedia Computer Science*, vol. 80, pp. 919 – 930, 2016.
- [35] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1999.
- [36] Serval project. [Online]. Available: <http://www.servalproject.org>
- [37] L. Baumgärtner, P. Gardner-Stephen, P. Graubner, J. Lakeman, J. Höchst, P. Lampe, N. Schmidt, S. Schulz, A. Sterz, and B. Freisleben, "An experimental evaluation of delay-tolerant networking with Serval," in *IEEE Global Humanitarian Technology Conference (GHTC '16)*. IEEE, 2016, pp. 70–79.
- [38] L. Baumgärtner, P. Graubner, J. Höchst, A. Klein, and B. Freisleben, "Speak less, hear enough: On dynamic announcement intervals in wireless on-demand networks," in *13th Annual Conference on Wireless On-demand Network Systems and Services (WONS'17)*. IEEE, 2017, pp. 33–40.
- [39] A. Sterz, L. Baumgärtner, R. Mogk, M. Mezini, and B. Freisleben, "DTN-RPC: remote procedure calls for disruption-tolerant networking," in *IFIP Networking Conference (Networking'17)*, 2017.
- [40] M. Schmittner, A. Asadi, and M. Hollick, "SEMUD: Secure multi-hop device-to-device communication for 5G public safety networks," in *IFIP Networking Conference (Networking'17)*, 2017.
- [41] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The optimized link state routing protocol version 2," IETF RFC 7181, Apr. 2014.
- [42] T. A. B. Nguyen, C. Meurisch, S. Niemczyk, D. Böhnstedt, K. Geihs, M. Mühlhäuser, and R. Steinmetz, "Adaptive task-oriented message template for in-network processing," in *International Conference on Networked Systems (NetSys'17)*. IEEE, 2017, pp. 1–8.
- [43] C. Meurisch, T. A. B. Nguyen, J. Gedeon, F. Kohnhäuser, M. Schmittner, S. Niemczyk, S. Wullkotte, and M. Mühlhäuser, "Upgrading wireless home routers as emergency cloudlet and secure dtn communication bridge," in *26th International Conference on Computer Communications and Networks (ICCCN'17): Posters (to appear)*. IEEE, 2017.
- [44] C. Meurisch, T. A. B. Nguyen, S. Wullkotte, S. Niemczyk, Kohnhäuser, and M. Mühlhäuser, "NICER911: Ad-hoc Communication and Emergency Services Using Networking Smartphones and Wireless Home Routers," in *18th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'17): Poster*. ACM, 2017.
- [45] A. Jacoff, R. Sheh, A.-M. Virts, T. Kimura, J. Pellenz, S. Schwertfeger, and J. Suthakorn, "Using competitions to advance the development of standard test methods for response robots," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM, 2012, pp. 182–189.