LoRAgent: A DTN-based Location-aware Communication System using LoRa

Lars Baumgärtner*, Patrick Lieser†, Julian Zobel†, Bastian Bloessl*, Ralf Steinmetz†, Mira Mezini*

Technical University of Darmstadt, D-64289 Darmstadt, Germany
*FB 20, E-mail: {baumgaertner, bloessl, mezini}@cs.tu-darmstadt.de
†FB 18, E-mail: {patrick.lieser, julian.zobel, ralf.steinmetz}@kom.tu-darmstadt.de

***Abstract*—Modern information and communication technology (ICT) is often very vulnerable to disruptions through disasters. Yet, the ability to communicate and distribute messages is vital for efficient disaster response. Furthermore, ad hoc deployment of flexible, robust, and affordable communication systems in a disaster area are often necessary. Therefore, we propose a disruption-tolerant networking bundle agent that uses LoRa radio technology to provide decentralized basic means of communication. To address the hardware's technological limitations as well as the uncertainty of the user locations and their movement behavior, we propose a geospatial routing mechanism for efficient message forwarding. In conjunction with the communication and routing solutions presented, we also designed specific pager-like hardware for intuitive message reception and bridging of smartphones into LoRa networks. We evaluated our solutions in various simulations as well as through real-world implementations on different hardware platforms.**

I. INTRODUCTION

During catastrophic events the ability to exchange information amongst people is vital. Especially for professional first responders, communication is key for the successful coordination of emergency response efforts. Due to failures and disruption of the information and communication infrastructure (ICT) caused by man-made or natural disasters or the pure lack of ICT deployment itself in rural areas, communication is often not possible. In the past, the lack of ICT was seen, for example, at the Great East Japan earthquake in 2011 [1], Hurricane Maria 2017 [2], or the Australian Black Saturday Bushfires 2009 [3].

Ad hoc communication setups based on mobile cell towers or TETRA often require severe technical training, are expensive and need excessive time and planning to be deployed [4]. Therefore, official organizations, such as local police and fire departments, are not prepared to roll out necessary communication infrastructure when ICT is unavailable. Here, low-cost, low-energy, and low-maintenance throw boxes are proposed to enhance communication possibilities [5]. Especially professional responders are used to principles such as radio discipline to reduce unnecessary communication load on the medium. This behavior favors the use of alternative links, which are not capable of transferring massive amounts of data or real-time communication, but providing properties such as long-range and energy-efficient communication. Disruption- or delay-tolerant networks (DTNs) are often proposed to provide communication in highly stressed environments. They can cope with intermittent connectivity, mobile network participants and variable communication delay but are mostly realized in conjunction with high bandwidth links such as ad hoc WiFi. DTNs usually operate on individual bundles for data dissemination in contrast to stream or connection-oriented communication protocols.

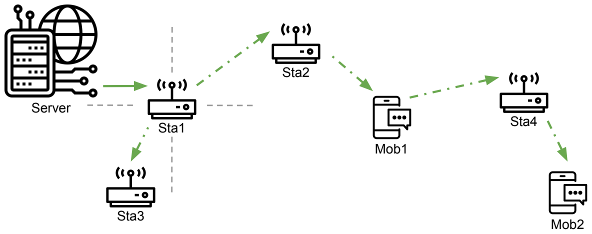

Fig. 1. DTN message distribution with static relays and mobile pager devices.

The DTN bundle protocol overhead, which is necessary for the store-carry-and-forward communication principle, makes it challenging to use DTNs with low-bandwidth communication links. To make DTNs suitable for basic emergency text messaging and pager-like services, new routing and distribution mechanisms are needed that adapt to communication technologies such as LoRa. These mechanisms are responsible for prioritization and subsequent distribution of messages, for example based on the message content or context information, such as the broadcasting node's location. The infrastructure required for the DTN communication can either be rolled out when needed, or is already available for another purpose before a disaster occurs. For example, DTN infrastructure is currently used for environmental monitoring in remote forests [6] and will be an essential component in future resilient digital cities [7].

In this paper, we present a novel DTN-based location-aware communication system for first responders consisting of low-cost *DTN relay nodes* and *DTN pager devices*. As depicted in Figure 1, the DTN relay nodes are operated in the form of static throw-boxes at dedicated locations and the DTN pager devices are mobile companion devices carried by the first responders. The primary purpose of the system is to push messages from a command center to roaming mobile users.

As it should be applicable worldwide, we propose the use of

LoRa transceivers for communication as there are unlicensed frequencies that can be used without permission anywhere on earth. Since no fully meshed network can be expected in the area of an emergency and the communication participants are assumed to be mobile, our proposed communication system is based upon DTN technology.

Due to limitations of using LoRa technology, such as low-bandwidth and duty-cycle restrictions, we propose a novel geospatial quadrant-based routing algorithm and a time slot-based communication scheme. To simplify the configuration and rapid deployment of our communication system, optional GPS modules at the DTN relay nodes provide accurate geolocations and auto-configuration. Furthermore, a direct 1-hop communication channel for maintenance and configuration coordination between relay node operators is provided via LoRa. The newly designed bundle protocol-based messaging protocol supports text compression as well as indicators for potential payload encryption and support for message signing.

For the intuitive and straightforward operation of the communication system, we settled on a robust design resembling 90s-style pager companion devices, to display text to the receiving user directly. Because of the lack of complex user-interaction-patterns and the reduced number of necessary components for such DTN pager devices, the chances of technical or usability failures are minimized. The easy- and fast-to-use characteristic of the system is particularly important in stressful situations in which first responders often find themselves. The DTN pager devices can optionally be paired with a smartphone to act as a LoRa modem for other applications and also to forward the received messages directly to the users' phone.

We conducted an extensive evaluation of the two individual components of our communication system, the static DTN relay nodes and the mobile DTN pager devices. We investigated the application of different compression algorithms, the proposed time-slot scheduling for transmissions, and the expected energy consumption of system components.

In this paper, we present the following contributions:

- We present a novel infrastructure-less communication system for emergency response.
- We designed a novel communication protocol for short messages delivered via the LoRa technology.
- We present a novel DTN routing and air-time scheduling algorithm for high range, low-bandwidth links.
- We present a novel and cost-efficient solution for pager-like communication devices that can be paired with state of the art smartphones.
- We provide a thorough evaluation of the most common compression algorithms in the context of DTN and bandwidth constraint technologies such as LoRa.

The paper is organized as follows: Section II discusses related work. In Section III, we present requirements and designs decisions. Section IV discusses implementation issues. Section V presents experimental results. Section VI concludes the paper and outlines areas of future work.

II. RELATED WORK

Infrastructureless ad hoc communication in Delay-tolerant Networks (DTNs) has received increasing attention in the last years, as it can provide robust and flexible communication in disaster areas. Usually, a flooding-based store-carry-forward approach is used to cope with the highly dynamic network. But this results in a significant increase in message duplicates, and thus, communication overhead. However, more sophisticated approaches like geographical or geospatial routing mechanisms can be applied to limit communication effort only towards a specific region of the network, but usually require knowledge of the current and the target location. Directional Flooding [8] reduces the overhead compared to pure flooding by flooding messages only within a specific corridor towards the target location. If this location is not known, predictions of the destination location can be made, for example, based on a history of encounters or local node mobility [9], [10]. A well-known example is *Probabilistic Routing Protocol using History of Encounters and Transitivity* (PRoPHET) [9], which assumes that nodes with more encounters in the past are more likely to have more encounters in the future, e.g., due to increased mobility or recurring mobility patterns. In combination with a history of encounters, probabilities are used to determine nodes that will encounter the receiver itself or other nodes that are more likely to encounter the receiver. Messages are then only forwarded to the nodes with the highest probability. Additionally, the probability can be calculated, for example, based on the distance to the destination [10] or based on a fair share of each node's duty cycle, energy consumption, or workload [11].

For emergency communication systems, often only text-based services are provided, as these keep the data load in the network low, and the support of multimedia content often offers no added value [12]. A real-world field test has shown that a simple but highly robust epidemic routing approach is sufficient for distributing only text-based messages using the WiFi communication interface [13]. The most significant impairment of communication during the mentioned field test was not the overload of the network but the limited communication range of WiFi in combination with the lack of node mobility over long distances. As presented in [14], by introducing controllable high-mobility nodes such as Unmanned Aerial Vehicles (UAVs) to the network, the long-range communication performance of otherwise short-range ad hoc networks can be significantly increased. Instead of additional and controllable data carriers, individual range extension devices can connect intermittent network partitions over long distances. For example, a range extender used by ServalMesh [15] operating on UHF Radio on 915 MHz ISM bands can provide device-to-device single-hop communication of around 200m inside buildings up to 3km in open line-of-sight environments.

Originally designed for the Internet of Things (IoT) as *Low-Power Long-Range Wide Area Networks* (LPWANs), the physical layer protocol LoRa (for Long Range) provides low-power long-range communication. It operates within unlicensed frequency bands worldwide, as the 433 MHz ISM band or 868 MHz SRD band in the EU and the 915 MHz ISM band in North and South America, using a robust Chirp Spread Spectrum (CSS) modulation [16], [17]. LoRa may provide communication ranges of more than 16 kilometers in LoS environments [16] or several kilometers in non-LoS environments [18], depending on the LoRa modulation settings such as Spreading Factor, Bandwidth, and Coding Rate. However, long-range and low-power advantages come with the drawback of small data rates and possibly long signal air times. Tough duty cycle requirements in most SRD or ISM bands, e.g., 1% in the EU's 868 MHz band, however, require a trade-off between throughput and range [19], [20], [17].

Although LoRa networks are primarily applied in a star topology for static sensor networks, it is feasible to build a LoRa mesh or use LoRa nodes as relays with an appropriate LoRa MAC protocol [21], [19]. However, these approaches build static network routes for collecting sensor data in a central sink. For more human-like communication in mobile environments, we presented a smartphone application in previous work [18] that interacts with a low-cost ESP32 device as a LoRa transceiver to achieve long-range chat communication between smartphone users. Although the design is somewhat similar to that approach by ServalMesh [15], our approach allows point-to-multipoint communication and also showcases the possibility of LoRa as a DTN communication technology.

III. DESIGN

In this section, we present our design for a novel, lightweight LoRa DTN communication system, comprised of statically deployed relay nodes and mobile pager devices. To reduce the overall network load we *i)* limit the system to small text-based messages, similar to the *short message system* (SMS), *ii)* bundle messages together to use transmit-windows more efficiently, and *iii)* use a quadrant-based geographical forwarding approach to limit message spread to certain directions, which are described in the following.

A. Network Maintenance

Currently, the communication system has three main types of packets which are all encoded as *Concise Binary Object Representation*[1] (CBOR) arrays for transmission. The same encoding is also used in the specification of the DTN Bundle Protocol RFC draft[2], as it is space-efficient and a common standard in IoT protocols.

1) Announcements: Announcements are spread periodically as beacons and can be received by any neighboring DTN relay node or DTN pager device within the communication range. They contain: *i)* the node's name, *ii)* the node's current location, *iii)* a hash of the already received and stored bundles of a node, and *iv)* the current timestamp.

2) Direct 1-hop Neighborhood Communication: To support fast and reliable local communication between DTN relay stations and DTN pager devices, our system provides a text-based messaging service similar to the one described in [18]. This is mainly intended for network operators to coordinate locally when deploying or debugging nodes without flooding other DTN nodes. The service groups text messages by channel and limits the propagation of the text message to the 1-hop distance of the originator. This way, close-range real-time (non-DTN) communication can be implemented. The term *close-range* hereby strongly depends on the utilized communication technology. Optionally, each message can be compressed to preserve bandwidth.

3) Bundle: A bundle combines the data and control information of a message in a standardized way. Bundles can be delivered asynchronously from source to destination via several intermediate nodes [22]. Forwarding nodes add their node identifier to the packet containing the bundle to keep track of which nodes are already aware of a specific bundle. Optionally, the payload of a bundle can be compressed.

B. Bundle Distribution

Forwarding and distribution of bundles occur in two different cases: 1) new received bundles are forwarded directly to neighboring nodes, in a location aware-manner, to ensure the distribution of new information in the network, and 2) already forwarded bundles are stored, periodically checked and, if needed, redistributed. Both cases are elaborated in more detail in the following.

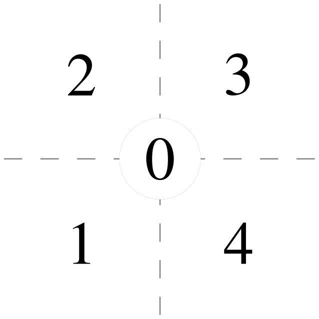

Fig. 2. Quadrant numbers for priority calculation in distance function.

1) Immediate Retransmission: As LoRa can theoretically achieve a communication range of up to 16 km, a received bundle from one direction might not have reached other nodes in the opposite direction of the receiving node. Therefore, immediate retransmission is helpful to spread new content to distant areas of the network quickly. Since bandwidth is limited, nodes individually calculate a forward-priority for each new received bundle. The priority is calculated based on *i)* the direction from which the bundle was received, *ii)* the node's recent neighbors, *iii)* the neighbors' bundle storage, *iv)* and the neighbors' locations. We propose a quadrant-based numbering scheme for locations, as shown in Figure 2. Any neighbor with an unknown position is assigned to quadrant

---

[1] https://tools.ietf.org/html/rfc7049

[2] https://tools.ietf.org/html/draft-ietf-dtn-bpbis-24

number 0 to make handling in the distance function clearer. Distances are then weighted as either 1 for directly adjacent quadrants and unknown positions or weighted 2 for directly opposite directions, as here the delivery success from the initial broadcast is the lowest.

Algorithm 1 Bundle Retransmission

```
1:  procedure RETRANSMIT_PRIORITY($rx\_hdr$, $bndl$)
2:      $q \leftarrow$ GET_QUADRANT($rx\_hdr.gps$)
3:      $priority \leftarrow 0$
4:      for $i \leftarrow$ GET_RECENT_NEIGHBORS() do
5:          if $bndl.id \in i.received\_bids$ then
6:              continue                ▷ Node already has bundle
7:          end if
8:          if $q \neq i.q$ then               ▷ Different quadrant
9:              $priority \leftarrow priority +$ DIST($q, i.q$)
10:         end if
11:     end for
12:     return $priority$
13: end procedure
```

The actual score calculation is shown in Algorithm 1. First, the quadrant has to be derived from the GPS coordinate of a newly received bundle (line 2). Then for each recently seen neighbor, it is checked whether it has itself already forwarded this specific bundle at least once (line 5). If this is not the case, the priority is increased by the distance from the received bundle quadrant to the neighbor's quadrant (line 9). The sum of these checks determines to the total score. If the score is greater than 0, the bundle and its priority are sorted into the send buffer for the next send slot of the node.

Algorithm 2 Bundle Spreading

```
procedure SPREAD_PRIORITY($bndl$)
    $priority \leftarrow 0$
    for $i \leftarrow$ GET_RECENT_NEIGHBORS() do
        if $bndl.id \in i.received\_bids$ then
            continue                ▷ Node already has bundle
        end if
        $priority \leftarrow priority + 1$
    end for
    if not SENT_IN_NEIGHBORHOOD($bndl$) then
        $priority \leftarrow priority + 1$       ▷ Send at least once
    end if
    return $priority$
end procedure
```

2) Periodic Spreading: As new nodes might enter transmission range or send windows might have been too small to transmit every relevant bundle, a periodic spreading of bundles is necessary. To preserve resources, this procedure should be aborted if there are no neighbors currently in range. For each bundle, a node possesses a spreading priority is calculated as shown in Algorithm 2. It is checked which of the recently seen neighbors has already spread the bundle with the corresponding ID. Each node missing the bundle increases the priority by one, as it is not a direct retransmission, the quadrants do not influence the scoring. Furthermore, even if all neighbors have already received a bundle, we want to transmit it at least once per neighborhood to compensate for collisions and faulty transmissions.

These scores are sorted into the send queue where also the retransmission bundles are placed. Due to the quadrant scoring, these always have a higher probability to be sent first. Depending on the bundle sizes and the time frame for each nodes transmissions, not all bundles in the queue might get sent in one send slot.

C. Short Message System Protocol

The communication protocol is based upon the recent Bundle Protocol version 7 draft for its encapsulating and a custom payload with the higher-level protocol information. To make the transition from existing communication infrastructure easier and provide unique peer names, each endpoint is identified by its telephone number. The bundle itself should only consist of the primary block, optionally a hop count block and the payload block. Within the primary block, the timestamp must be set correctly to also have a message send time. Furthermore, the payload protocol should support optional encryption and compression of the message text as well as the possibility to sign the message for authenticity.

D. Physical Communication System Components

As our communication system consists of two device categories, low-cost *DTN relay nodes* and *DTN pager devices*, we give a quick overview of the design considerations regarding them in the following.

1) DTN Relay Node: Each relay node is designed to be statically deployed without expert knowledge required for setup. As it might need to host several different services, a platform capable of running Linux is necessary. Furthermore, a LoRa transceiver that can be connected to various different antennas is needed. To automatically configure the system with the correct time and deployment location, a GPS receiver is also necessary for the relay.

2) DTN Pager Device: Users in the field need a simple and robust way to receive messages during a disaster. Therefore, a lightweight, battery-powered device is needed, which can receive LoRa packets and inform users about new messages. It should at least provide a simple display with an optional attention-grabbing mechanism such as a buzzer or vibration motor. To minimize misuse or confusion on user side no major ways of configuration or interaction should be required. As an advanced user feature pairing with existing smartphones is a desired feature to incorporate companion apps.

IV. IMPLEMENTATION

Due to the resource constraints and security criticality of the services running on the DTN relay nodes, everything has been implemented in the Rust programming language, which provides memory safety, concurrency and resource efficiency.

For a basic DTN agent implementation and the handling of bundle protocol itself the *bp7*[3] and *dtn7*[4] libraries were used.

A. Short Message System Protocol

Each peer is identified by a unique number such as a telephone number, which can also be used for identity-based cryptography. These numbers are encoded in the IPN address scheme with the node name being the telephone number and a fixed service number of 767 (T9 code for SMS) for source and destination fields in the primary block of a bundle.

The actual message protocol uses CBOR-encoded hashmaps within the payload block of the bundle. It includes flags for compression and encryption, indicating how the message string has to be processed. The used compression algorithm is fixed to *smaz* [23] as it provides superior short text compression in comparison to other well-known compression algorithms (c.f. Tab. I). Alternatives like *stat3* [24] exist, but even though they may provide a higher compression rate, they also require significantly more time for compression and decompression. Furthermore, they lack native rust implementations. Rewriting these algorithms is not feasible due to large code bases and increased complexity, and binding to such large legacy C code might introduce security or stability issues. The signature of the message is optional and might be omitted.

B. Hardware

In the following, we provide a brief overview of the implementation details of the relay node and pager device hardware.

1) DTN LoRa Relay Node: Due to its broad availability and relatively low-power requirements, we choose Raspberry Pis as the basis of our relays. For stations running more services the Pi 3 is a good choice while the most energy-efficient minimal solution is based upon a Pi Zero W. Both Single Board Computers (SBCs) also provide WiFi for either mesh networking or a local access point. If no further sensors are needed on the board the most energy-conserving solution for LoRa communication and GPS was the Dragino LoRa+GPS Hat[5], which directly attaches to the GPIO pins of the Pi. Alternatively, we also had prototypes running using a TTGO T-Beam LoRa+GPS[6] board connected via USB. The downside here is that the board has an ESP32-based CPU and USB connection which both increase power consumption a bit.

2) DTN LoRa Pager Device: Our device is based upon the TTGO ESP32 LoRa OLED modules[7], which incorporate a LoRa transceiver, a 0.96" OLED display, as well as WiFi/Bluetooth connectivity. They can be powered directly by a LiPo battery or by a powerbank over USB. The pager software is written in C/C++ and based upon the *rf95modem* firmware[8][25], which also allows us to pair the pager with mobile devices using Bluetooth Low Energy. Additionally, a buzzer and vibration motor for acoustic and haptic notifications, respectively, can be connected over GPIO pins.

V. EXPERIMENTAL EVALUATION

In the following, we present an experimental evaluation of some of the key components of our approach. First, various compression algorithms are evaluated for potential use in our communication system. Second, we analyze our proposed scheduled transmission scheme to show its usefulness. Third, we provide some numbers regarding real-world power consumption of the implemented solutions. Finally, a brief overview of the costs of our devices is given.

A. Bundle Compression for LoRa Transmission

As the bandwidth of LoRa is very limited and the maximum transmission unit (MTU) size is less than 255 bytes, it is crucial to encode bundles prior to transmission efficiently. There are many different compression algorithms, often optimized for specific payload types, that can be used to reduce the number of bytes needing transmission. Even a minimal DTN bundle without extra canonical blocks or long endpoint IDs is already nearly 80 bytes in size. In the following, we evaluated the effects of different compression algorithms on the bundle protocol itself as well as various payload types such as text, images, or other files. For all algorithms implementations or bindings using the Rust programming language were used. The experimental setup and results can be found online[9]. The compression libraries used can be roughly grouped into classic ones such a *gzip*, *zlib*, *gzip* or *bz2*, rather new ones like *brotli* and *snap* and data type optimized ones such as *smaz* which is designed for the compression of short texts. We applied these to several representative data sets in 7 categories. The first two targets consist of plain bundles, one minimal without extra canonical blocks except for a minimal payload block and the other one with all the canonical blocks specified in the IETF draft of Bundle Protocol version 7. The next two data sets use randomly generated "lorem ipsum" text in two different lengths, representing short messages (15 words) and long text files (2000 words). The next target is a combination of short text and a minimum bundle configuration. This is basically what the bundles for our short message protocol look like. Finally, two binary sets were evaluated consisting of image (∼ 21 KB) and pdf (∼ 200 KB) data.

The results of this evaluation can be seen in Table I. All compression ratios are colored whether the size has significantly increased (red), stayed the same (yellow) or decreased (green). As it is expected, the pure bundle protocol is already encoded pretty efficiently using CBOR. Thus, most compression algorithms introduce more overhead and even increase the overall size. Only *snap* was able to achieve some compression for larger bundle configurations. The biggest compressable factors are the used endpoint identifiers in the primary block or an optional previous node block as these might be long strings that get used in multiple places. Except for a few exceptions

---

[3] https://crates.io/crates/bp7

[4] https://crates.io/crates/dtn7

[5] https://wiki.dragino.com/index.php?title=Lora/GPS_HAT

[6] https://github.com/LilyGO/TTGO-T-Beam

[7] https://github.com/LilyGO/TTGO-LORA32

[8] https://github.com/gh0st42/rf95modem

[9] https://github.com/stg-tud/ghtc2020_eval

TABLE I
COMPRESSION ALGORITHMS APPLIED TO DIFFERENT DATA SETS.

Compression Lib	**min bundle**	**max bundle**	**short text**	**long text**	**text bundle**	**png bundle**	**pdf bundle**
brotli	1.05	1.04	0.81	0.36	0.98	1.00	0.78
bz2	1.82	1.40	1.18	0.37	1.19	1.02	0.82
libflate_deflate	1.25	1.06	0.84	0.45	0.94	1.00	0.83
libflate_gzip	1.49	1.22	1.05	0.45	1.05	1.00	0.83
libflate_zlib	1.33	1.11	0.91	0.45	0.98	1.00	0.83
miniz	1.07	1.05	0.84	0.39	0.94	1.00	0.82
raw	1.00	1.00	1.00	1.00	1.00	1.00	1.00
smaz	1.11	1.11	0.66	0.65	0.80	1.20	1.17
snap	1.01	0.90	1.01	0.63	1.01	1.00	0.94
xz2	1.81	1.47	1.69	0.39	1.27	1.00	0.79

most algorithms were able to achieve good compression ratios for short and large strings. Never the less, *smaz* clearly won for short texts over the others with a compression ratio of 0.66 vs. 0.81 for the next best contender *brotli*. As the results for text and pure bundles were very different the results of the combined approach with a minimal bundle header and some short payload text are the most interesting for our proposed communication system. The *smaz* compression keeps its edge over the others and clearly provides the best compression for the use-case with a ratio of 0.80 while the next best libraries (*libflate_deflat* and *miniz*) achieve only 0.94. Even though not really necessary for our use-case, we include tests regarding binary data for the sake of completeness. Due to the fact that image data such as png files are usually are already compressed, we did not improve the overall transmission size by applying the algorithms. In case of *smaz*, it made it even worse by increasing the size compared to the original image. Since *smaz* was specifically designed for short human texts, it does not come as a surprise that binary data is not handled well. For pdf files most evaluated libraries were able to achieve some slight compression except for, again, *smaz*.

We conclude that while *snap* might be suited for compression of complex bundles with many different block types included, it does make sense to apply use-case specific compression within the payload. Therefore, we support optional *smaz* compression within our short message system protocol.

B. Transmission Scheduling and Collision Avoidance

To evaluate the applicability of our approach regarding transmission collisions and packet air-time, we conducted large-scale simulations with a varying number of nodes, focusing on successful messages deliveries and occurring collisions. The simulations were conducted in the discrete event simulator OMNeT++[10]. All nodes are randomly placed in an area of 1500 meters by 1500 meters and the communication range of the simulated LoRa link maxes out at 500 meters plus double that for interference range. These defaults are quite a bit more pessimistic compared to the often announced LoRa properties, but previous research [18] has shown that our values are more in line with to be expected real-world performance. The simulated half-duplex link provides a maximum of 3 kbps and all transmissions are limited to 10 seconds of air time for delivering the 100 bytes of payload data for the announcement. We simulated a sparsely populated area by placing only 10 nodes in the scenario, medium ones with 50 to 100 nodes and densely populated areas with 250 and even 500 nodes. As our target is to provide communication for small groups of people in larger badly connected areas, these numbers are already very high and provide plenty of evidence for scalability in our scenario. As a baseline we have a pure collision scenario where all transmissions are guaranteed to happen at once. Therefore, no packets should be received except for on the sending node itself. By randomizing the start interval, most of these collisions can obviously be avoided. We use a fixed send interval of two minutes, in which we can handle 60 nodes without collisions if we assume a maximum air time of 2 s of perfectly distributed senders. Similar fixed interval settings are most commonly found in network services. Unfortunately, if nodes are fixed on the same transmission slot collisions will occur permanently until a node moves out of distance or changes its schedule. Finally, we have our uniformly distributed starting slots plus a slight variation in the interval to cope with otherwise reoccurring collisions. All simulations are run for one hour to get realistic readings and are repeated 10 times with different random seeds.

The results of these experiments can be seen in Figure 3. It can clearly be seen that with an increasing number of nodes, our approach, shown in the green bars, provides significantly fewer collisions. The more nodes are involved the bigger the delivery rate gap between classic fixed interval transmissions and our implementation gets. Scalability-wise we can observe that with 500 nodes the number of hosts reached drops significantly as the chances for collisions of these long-range transmissions are increasing with so many nodes on such a limited area. Never the less, we still are able to achieve almost the same delivery rate as with only 50 hosts in the same area. Classic fixed interval schedules did not even reach the delivery rates of a network with only 10 nodes.

Due to our announcement protocol design, we have even more possibilities to detect and avoid collisions without adding more packets and protocols to the transmissions. Using the

---

[10] https://omnetpp.org/

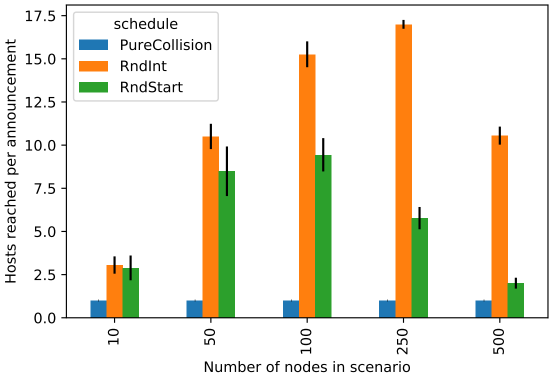

Fig. 3. Transmission scheduling policies compared in various settings.

Fig. 4. Opened DTN pager device with a Raspberry Pi for size reference.

timestamps in announcement packets and the known names of neighboring nodes we can infer when their next transmission will be scheduled and reschedule our own or tweak our sending interval to avoid collisions.

C. Energy Consumption of Different Deployments

To get a sense of the power requirements of our communication system we evaluated various different configurations and components. The mobile pager device is based on an off-the-shelf hardware platform that contains an ESP32 with a LoRa transceiver and a small OLED display. While we use an internal LiPo battery via 3.3 V pins, the energy measurements were conducted through the 5 V USB charging port. Therefore, the power consumption is expected to be a bit higher but is also more comparable as USB powerbanks also power the other configurations. As a basis for the relay nodes, both Raspberry Pi 3 B (not B+) and Raspberry Pi Zero W were evaluated. In case the relay node might need to perform extra duties besides handling the LoRa messages, the increased processing power of the Pi 3 might come in handy. For LoRa and GPS functionality we considered two different solution: *i)* the Dragino LoRa+GPS hat connected directly to the GPIO headers and *ii)* the TTGO t-beam LoRa+GPS MCU that can be accessed via USB. Both systems can have external antennas attached for LoRa as well as GPS. While the Dragino hat only provides the raw transceiver/receiver chips, the t-beam also ships a complete ESP32 CPU. For the latter, we flashed the *rf95modem* firmware[11] onto it to make it easily accessible. To make both solutions accessible in a similar way we wrote a compatibility wrapper for the *rf95modem* firmware on Raspbian called *rf95dragino*[12].

For the mobile companion device the idle power consumption is around 70 mA including the activated OLED display. When used as a two-way radio, the transmitting power still stays below 200 mA. In the future, this could be further reduced if pairing with smartphones is not required and the device is only used as a receiving pager by going into a deep sleep and reacting to an interrupt when new packets arrive.

On both GPS-capable platforms, the cost for activating GPS functionality is less than 50 mA. To preserve energy this functionality can be toggled on and off via software.

The Pi 3B uses significantly more power even when idle (250 mA) compared to the Pi Zero W (70 mA). For the basic functionality of LoRAgent the smaller device is enough, having more cores and higher CPU clock speed mainly helps with a very high number of messages, far exceeding of what LoRa is capable of handling, or when other services such as Wikipedia mirrors, on-device sensor-data processing or local collaboration software should be run. Without the power consumed by the GPS the LoRa functionality on the Dragino hat is not measurable when only receiving data. If data is transmitted the power consumption is increased by up to 150 mA. The alternative to the LoRa+GPS hat is to use one of the TTGO t-beam boards as an extension on the Raspberry Pi. This requires around 60 mA when receiving data via LoRa with a deactivated GPS module. When sending data and having GPS active this can peak to over 200 mA.

In summary, one can expect the DTN pager device to require about 356 mW on average. Depending on the configuration, a DTN relay node consumes about 357 mW to 1.12 W for a Pi Zero W with a Dragino hat and 1.73 W to 2.29 W for a Pi 3B with attached t-beam and activated GPS. Currently, the *rf95modem* firmware for the t-beam is more mature than the one available for the Dragino hats. Additionally, the price per unit is lower, thus, if the 306 mW consumed by the ESP32 are tolerable, this is our preferred setup.

D. Estimated Cost per Unit

In this section we give a rough estimated how much the different units cost. All prices are for purchases in small quantities from world-wide available stores such as AliExpress and Amazon. The DTN LoRa Pager modules are available from Heltec as well as TTGO and cost around \$25. Small LiPo batteries, depending on size, cost from \$2 to \$10 and the case is 3D printed with a few cents worth of filament. This, eventually, results in a total cost of less than \$40 for each device. Figure 4 shows the opened device with all components next to a Raspberry Pi.

Depending on the used Raspberry Pi model for the DTN Relay Nodes, the base unit costs between \$20 and \$35 each.

---

[11] https://github.com/gh0st42/rf95modem

[12] https://github.com/gh0st42/rf95dragino

For LoRa and GPS functionality, we can choose between the Dragino LoRa+GPS hat for about $45 or the TTGO t-beam for about $25. The costs for cables and wires are neglectable, but additional antennas for less than $20 per piece may be useful. The SD card size may vary depending on extra functionalities the node should provide, but usually 16 GB can be found for around $5. As casing, we used modified off-the-shelf food storage boxes as they are readily available, reasonably cheap, and easy to modify. The last component required is a battery as a power supply, that also can be enhanced by a solar panel. Here, prices vary heavily depending on the setup. Thus, they are omitted from our calculation. In some cases nodes might also be connected to the power grid and not need anything else. With all this considered, a DTN Relay Node can be built for something between $60 and $120 depending on the exact configuration.

VI. CONCLUSION

In this paper, we presented a novel communication infrastructure based upon DTN software and LoRa radio transceivers. Our approach features a novel geospatial DTN bundle routing algorithm specifically designed for low-bandwidth, long-range radio links with an appropriate new high level messaging protocol. Furthermore, we implemented a complete communication system to automatically configure and coordinate fixed Raspberry Pi-based base stations and mobile pager-like companion devices. Through various experiments we have shown the effect of our design decisions regarding compression algorithms, advanced transmissions slot scheduling, and overall energy consumption.

In the future, we plan to expand on different other radio link technologies and build directional antenna support into the routing algorithm for further optimizations. Moreover, adaption to different other payloads besides text messaging poses a challenge under the restrictions such as air time and MTU size in LoRa networks.

ACKNOWLEDGMENT

This work has been co-funded by the LOEWE initiative (Hessen, Germany) within the *Nature 4.0 – Sensing Biodiversity* project and the *emergenCITY* centre, as well as the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 MAKI. Furthermore, it has been co-funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

REFERENCES

[1] F. Ranghieri and M. Ishiwatari, *Learning from Megadisasters: Lessons from the Great East Japan Earthquake*. World Bank Publications, 2014.

[2] M. Gallucci, “Rebuilding Puerto Rico’s Power Grid: The Inside Story,” *IEEE Spectrum*, 2018.

[3] J. Whittaker, K. Haynes, J. Handmer, and J. McLennan, “Community safety during the 2009 australian’black saturday’bushfires: An analysis of household preparedness and response,” *International Journal of Wildland Fire*, 2013.

[4] H. A. Eiselt and V. Marianov, “Mobile phone tower location for survival after natural disasters,” *European journal of operational research*, 2012.

[5] W. Zhao, Y. Chen, M. H. Ammar, M. D. Corner, B. Levine, and E. W. Zegura, “Capacity enhancement using throw-boxes in mobile delay tolerant networks,” Georgia Institute of Technology, Tech. Rep., 2006.

[6] N. Friess, J. Bendix, M. Brändle, R. Brandl, S. Dahlke, N. Farwig, B. Freisleben, H. Holzmann, H. Meyer *et al.*, “Introducing Nature 4.0: A sensor network for environmental monitoring in the Marburg Open Forest,” *Biodiversity Information Science and Standards*, 2019.

[7] M. Hollick, A. Hofmeister, J. I. Engels, B. Freisleben, G. Hornung, A. Klein, M. Knodt, I. Lorenz, P. Lieser, M. Mühlhäuser, P. Pelz *et al.*, “Emergencity: A paradigm shift towards resilient digital cities,” in *World Congress on Resilience, Reliability and Asset Management (WCRRAM)*, 2019, pp. 383–406.

[8] D. Hwang and D. Kim, “DFR: Directional flooding-based routing protocol for underwater sensor networks,” in *OCEANS*, 2008.

[9] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, 2003.

[10] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “Sector-based routing with destination location prediction for underwater mobile networks,” in *International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2009.

[11] O. Jumira, R. Wolhuter, and S. Zeadally, “Energy-efficient beaconless geographic routing in energy harvested wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, 2013.

[12] P. Lieser, F. Alvarez, P. Gardner-Stephen, M. Hollick, and D. Boehnstedt, “Architecture for responsive emergency communications networks,” in *Global Humanitarian Technology Conference (GHTC)*. IEEE, 2017.

[13] F. Álvarez, L. Almon, P. Lieser *et al.*, “Conducting a Large-scale Field Test of a Smartphone-based Communication Network for Emergency Response,” in *13th Workshop on Challenged Networks*. ACM, 2018.

[14] P. Lieser, J. Zobel, B. Richerzhagen, and R. Steinmetz, “Simulation Platform for Unmanned Aerial Systems in Emergency Ad Hoc Networks,” in *Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2019.

[15] P. Gardner-Stephen, R. Challans, J. Lakeman, A. Bettison, D. Gardner-Stephen, and M. Lloyd, “The serval mesh: A platform for resilient communications in disaster & crisis,” in *Global Humanitarian Technology Conference (GHTC)*. IEEE, 2013.

[16] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology,” in *14th Intl. Conference on ITS Telecommunications (ITST)*. IEEE, 2015.

[17] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios,” *IEEE Wireless Communications*, 2016.

[18] J. Höchst, L. Baumgärtner, F. Kuntke, A. Penning, A. Sterz, and B. Freisleben, “LoRa-based Device-to-Device Smartphone Communication for Crisis Scenarios,” in *Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2020.

[19] H.-C. Lee and K.-H. Ke, “Monitoring of large-area iot sensors using a lora wireless mesh network system: Design and evaluation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, 2018.

[20] J. Finnegan and S. Brown, “A comparative survey of LPWA networking,” *arXiv preprint arXiv:1802.04222*, 2018.

[21] M. Bor, J. E. Vidler, and U. Roedig, “LoRa for the Internet of Things,” in *Proc. of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN)*. ACM, 2016, pp. 361–366.

[22] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, “Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges,” *IEEE Communications Surveys & Tutorials*, 2011.

[23] S. Sanfilippo, “SMAZ—Compression for Very Small Strings,” 2009, https://github.com/antirez/smaz.

[24] P. Gardner-Stephen, A. Bettison, R. Challans, J. Hampton, J. Lakeman, and C. Wallis, “Improving compression of short messages,” *Int’l J. of Communications, Network and System Sciences*, vol. 2013, 2013.

[25] L. Baumgärtner, A. Penning, P. Lampe, B. Richerzhagen, R. Steinmetz, and B. Freisleben, “Environmental monitoring using low-cost hardware and infrastructureless wireless communication,” in *Global Humanitarian Technology Conference (GHTC)*. IEEE, Oct 2018, pp. 1–8.