Patrick Lieser, Nils Richerzhagen, Tim Feuerbach, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz. *Collaborative Decentralized Resource Reservation for Emergency Communication Networks (accepted for publication)*. In Proc. Conference on Local Computer Networks (LCN), IEEE, 2018. ISBN:

Collaborative Decentralized Resource Reservation for Emergency Communication Networks

Patrick Lieser, Nils Richerzhagen, Tim Feuerbach, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
{patrick.lieser|nils.richerzhagen|tim.feuerbach|tobias.meuser|bjoern.richerzhagen|ralf.steinmetz}@kom.tu-darmstadt.de

***Abstract*—Direct ad hoc connections among mobile devices can be utilized for emergency communication in times when communication infrastructure is unavailable due to blackouts or natural disasters. However, the quality of the resulting mobile ad hoc network strongly depends on the number of devices involved. Thus, to sustain a fully functional emergency communication network, devices need to recharge using external resources (e.g., battery packs, solar panels). Access to these resources needs to be coordinated such that the overall network lifetime is increased.**

We propose an auction-based resource reservation protocol for a decentralized resource allocation service. Through an extensive simulation study we show that our approach (i) efficiently coordinates the competition for resources, leading to a network lifetime comparable to a global allocation approach (98.8%) and (ii) delays the time until the first node runs out of energy by a factor of 1.4 compared to related work. This increases the value of the emergency communication network significantly, aiding more people over extended periods of time.

***Index Terms*—Delay-tolerant Networks, Disaster Communication, Resource Allocation, Reservation Protocol**

I. Introduction

Exceptional events such as large-scale blackouts [19] or natural disasters (e.g., hurricane Maria in 2017 [8]) increase the demand for communication and coordination among those affected by the event significantly. Unfortunately, communication infrastructure is often heavily impaired or even completely unavailable during and after such events [14]. In such cases, ad hoc communication among mobile devices such as smartphones can be utilized to establish an independent communication network [13], [16]. This network can then be utilized to provide applications such as emergency calls to cope with the aftermath of a disaster as discussed in [1], [26].

However, these applications can only offer their service as long as enough devices are participating in the network to maintain sufficient connectivity. Given the energy consumption of smartphones when working in ad hoc mode, it is not reasonable to assume that such a network can be sustained for longer periods of time without the utilization of external energy sources. In our work, we consider external infrastructure independent energy resources, such as battery packs, car batteries, solar panels, or generators to enhance the runtime of the communication devices and, consequently, of the overall network. We propose a core service for ad hoc networks that enables the coordinated assignment of resources to mobile devices such that the overall lifetime of the network is increased significantly. The resource allocation can be achieved without the need for any central coordination service, by sharing information of discovered energy resources among mobile nodes. In our previous work [17] we already demonstrated that the network lifetime can be increased by an independent resource allocation strategy based solely on local knowledge of individual nodes. However, as energy resources are limited, this local strategy leads to an "over-competition" for resources among nearby nodes.

In this paper we propose a decentralized reservation protocol called *Ad hoc On-demand Reservation Vector Auction* (*AORVA*). *AORVA* utilizes a set of ad hoc and delay tolerant networking strategies to achieve an appropriate distribution of knowledge within the network. By communicating the outcome of individual nodes' decision procedures (i.e., whether or not to compete for a certain resource) other nodes can react to these decisions by altering their own decisions, thereby avoiding over-competition and unsuccessful attempts at consuming a resource. Coordinating the allocation of energy sources with *AORVA* increases the lifetime of the overall network and, consequently, its potential utilization during and after disasters.

To evaluate *AORVA*, we model and investigate a representative scenario in which participants of an infrastructure-less network compete for limited resources. Our evaluation-based study shows that *AORVA* is especially beneficial in dense scenarios with an otherwise high probability for over-competition among nodes. Here, all the available resources could be allocated and the lifetime of the overall network is increased to 98.9% compared to the centralized approach. Even in scenarios with very limited competition among nodes, *AORVA* increases the time until the first node runs out of energy by a factor of 1.42 compared to related works.

The remainder of this paper is structured as follows: The scenario is described in more detail in Section II, followed by an explanation of the envisioned decentralized resource allocation service in Section III. The core contribution of this paper, *AORVA*, is presented in Section IV, followed by an in-depth evaluation in Section V. We discuss relevant related work in Section VI before concluding the paper.

II. Scenario

As briefly introduced in the previous section, we consider a disaster or post-disaster scenario where access to communication infrastructure is not available. We focus on an urban environment populated with mobile nodes that are equipped with energy-constrained communication devices such as smartphones. Utilizing, for example, 802.11 Wi-Fi in ad hoc mode, the respective devices are able to communicate directly with each other within a given range, forming a Mobile Ad Hoc Network (MANET). We assume that each device is equipped with one or multiple applications that utilize the MANET to provide emergency services such as the possibility to search for family and friends, exchange situation-dependent information with authorities, or call for help [16].

Devices are equipped with a battery and consume power based on their current state (e.g., is GPS enabled for navigation or are messages sent and received). If the battery is drained, the respective device is no longer able to participate in the network. Given that the aforementioned applications should provide their service for as long as possible to *everyone*, they heavily rely on a dense and fully functional network. Therefore, energy as an external resource can be utilized to extend the runtime of a device. Within our scenario, these resources (e.g., battery packs) can be discovered and consumed by a device. Whether or not a discovered resource is consumed by a device is determined by our decentralized resource allocation service, as described in more detail in Section III.

We assume that the amount of energy by a resource is limited, leading to a potential depletion of the resource over time. In our scenario, resources broadcast their availability and their remaining amount of energy to nearby devices using so-called Resource Demand Beacons (RDB). Also, nodes discovering a resource can generate those RDBs. When receiving an RDB, devices with energy demand can move to the respective resource and consume all or a fraction of the provided energy.

The decentralized resource allocation service ensures that devices that discovered an RDB share availability information with other surrounding devices. This decentralized shared knowledge about available resources enables mobile users out of direct reach of the resources to decide, whether or not they want to spend time and energy to approach a given resource by diverting from their current path.

To model the explorative behavior of humans, the target location is either chosen from a pre-defined set of *attraction points* representing open spaces where people tend to gather in emergency situations or a random location to model people that are actively searching for relatives, first-aiders, or resources. To reach a target location, users follow streets and walkways resulting in some parts of the considered area being visited less frequently than others. This mobility model, as further discussed in Section V, leads to natural movement patterns of users between different places on a map [21].

III. Decentralized Resource Allocation Service

The proposed scenario results in two main challenges: *learning about new resources* and, consequently, *deciding whether or not to approach and consume* a resource.

Regarding the first challenge, we assume that the location and arrival time of new resources cannot be predicted by the nodes. While moving around, nodes may discover new resources but may not be interested in them for the time being. To share this information with nodes in need of resources, a node discovering an RDB generates a resource advertisement. Advertisements created by a node or received from others are placed in the node's advertisement store. For each RDB corresponding to a resource, only the most recent advertisement is stored. To reduce the amount of outdated information, advertisements are removed from the store when their generation timestamp exceeds a configured *memory span*. To share advertisements among nodes, each time a node updates its store, it broadcasts a copy of the advertisement with an increased hop count. A receiving node does not rebroadcast a message if the maximum hop count is reached (*TTL*) or if it has more current information of the RDB.

Since a low-density network faces frequent disconnectivity, only a few nodes can be reached by advertisement flooding [5]. Thus, we employ a variant of the epidemic routing protocol SPIN-1 [10] as a *Store-Carry-Forward* approach. The protocol uses a three-way handshake to exchange information.

Assuming discovery and advertising of new resources, the second challenge is deciding whether or not it is worthwhile to approach and consume certain resources at a given point in time. Due to the scarcity of resources, it may not always be the best decision to pursue all known resources. Others may already have taken a resource before the node arrives, resulting in wasted time and undesired detours.

The *Decentralized Resource Allocation Service* decides autonomously whether a node should compete for a resource. The service consists of three components: a *demand evaluator*, a *cost mapper*, and a *selection strategy* as detailed in the following. The *demand evaluator* determines whether a node is currently requiring resources, and if so, initiates the resource selection process. Demand can be derived, for example, from the battery charge state of a node. On each node a *cost mapper* computes the individual costs for known resources stored in the advertisement store. The calculation takes into account the resource amount, the distance, and the estimated energy cost to obtain the selected resource. The *selection strategy* uses the obtained cost for each resource as input to select the resource that the node should approach next. Therefore, it may draw on additional information as later discussed for *AORVA*. Once a resource is selected by the aforementioned procedure, the user is alerted and guided to the resource. Incoming advertisements are constantly monitored for better options and depleted resources to adjust the selection accordingly.

A. Node States and Energy Consumption

Guiding a user towards a resource leads to higher energy consumption, given that GPS is utilized and the screen needs to be active now and then to display directions. Therefore, we distinguish three energy consumption states: Roaming, Heading, and Offline. Per default, nodes are in Roaming state and follow their personal movement policy. In this state, the node's current energy level e_c is reduced by E_r per second. If e_c is zero, the node stops communicating and changes to Offline, from which it cannot recover. When selecting and approaching a resource, it enters the Heading state consuming E_h resources instead. We require $E_h > E_r$ to reflect the additional energy required by the phone's screen and GPS component [17]. Nodes return to ROAMING either after arriving at the resource or if the selection strategy decides that pursuing the target is no longer worthwhile.

For simplicity, charging at a resource is assumed to happen instantly. Nodes try to maximize their own profit by transferring as much energy as possible from a resource, up to their maximum capacity e_{max}. If the available amount at an RDB has changed, it increases its *Beacon Sequence Number* (BSN) when announcing its presence with the updated information. A threshold is used to determine whether a node currently has demand for a resource. In this work, we consider the cost mapper *MinDistance*, which assigns each resource a cost of $-1/d$, with d being the distance between node and resource. Based on the node's velocity v and the expected energy consumption in HEADING, the cost mapper determines the set of unreachable or unprofitable resources.

B. Resource Selection Strategies

We proposed a set of basic selection strategies in our previous work [17], which we use as a baseline for the evaluation of the *AORVA* protocol proposed in this paper. In the following, we briefly introduce these basic strategies before discussing *AORVA* in detail. The *Greedy Selection* strategy always chooses the resource with the least cost regarding energy and time required to approach it, regardless of the overall demand in the network. Still, nodes exchange information about available resources by forwarding the respective advertisements. With the *En Passant* strategy, nodes do not exchange advertisements with each other and only take resources if they have demand and are currently within the discovery range of a resource. Resource locations are forgotten as soon as that range is left. The *En Passant* strategy constitutes the lowest baseline in that no cooperative mechanisms are employed. Lastly, a centralized *Reservation Oracle* is used as the upper baseline. This strategy is aware of the location and amount of all resources. Nodes reserve resources at the *Oracle* in a first-come, first-serve manner. Like *Greedy Selection*, the *Oracle* assigns resources by minimizing the costs for each user; however, it takes all existing reservations into account and, thus, prevents over-competition.

IV. AD-HOC ON-DEMAND RESERVATION VECTOR AUCTION PROTOCOL

AORVA is based on the concept of a shared auction: by expressing and communicating interest in a given resource together with a *bid* based on the distance to the resource, other nodes can decide beforehand, whether it is worthwhile to approach an RDB. In addition to their interest in general, nodes include the amount they intend to take from the given resource, enabling other nodes to benefit from knowledge about spare capacity even if they did not initially win the auction. We model such an auction in a decentralized fashion by proposing the *Ad Hoc On-demand Reservation Vector Auction* (*AORVA*) protocol. The protocol design is inspired by the Ad Hoc On-demand Distance Vector Routing [18] protocol and the Chaos protocol for consensus in all-to-all data sharing [15].

A node maintains a single *reservation vector* for each known RDB with information about the IDs and desired amounts of resources of reserving nodes. Nodes reserve resources by sending out *reservation requests* containing the reservation vector for the target RDB, updated to now include themselves, to other nodes. This *reservation requests* are resend periodically defined by the *Reservation Repetition Interval*. Receivers combine this vector with their local copy. If the requester's reservation is still included, they forward the request to their neighbors; otherwise, they reply with a *reservation response*. Due to mobility and potential disruptions of connectivity, the distributed reservation information can be inconsistent from a global point of view. However, *AORVA* ensures that combining two reservation vectors is deterministic and is therefore able to come to a locally consistent assignment of nodes to RDBs.

Each reservation is associated with a *bid*, and reservations by nodes are granted in order of their bid, from highest to lowest. To reduce the effect of outdated reservations in the network, *reservation vectors* will not be considered for new reservations, if they extend the *Reservation Vector Lifetime*.

A. Reservation Vectors and the Auction Merge Operation

When a node learns about an RDB, it creates a new reservation vector containing the RDB's *BeaconID*, its last known *Beacon Sequence Number* (BSN), the advertised amount of resources e_{max}, and an initially empty set of reservations R. The vector is kept separately from advertisements. If the node receives an advertisement for the same RDB with a different e_{max}, the vector is replaced with a new empty vector; otherwise, only the BSN is updated. A reservation takes the form of (*NodeID*, *RSN*, e_{res}, *bid*, *t*), where *NodeID* is the node's globally unique identifier; *RSN* the node's *Request Sequence Number*, which is incremented on every request; e_{res} the share of resources reserved at the RDB; *bid* the node's bid used during the merge auction; and *t* the timestamp of the reservation's creation. It holds that $e_{max} \geq \sum_{r \in R} r.e_{res}$.

The RSN is used to remove obsolete reservations by a node and to ensure that it can only reserve resources for at most one RDB simultaneously. Nodes keep track of the highest known RSN for every other node c, and remove reservations by c with lower RSNs from their local resource vectors before performing an auction merge. Nodes do not modify the contents of reservations made by others; especially, they do not decrease the amount of resources reserved e_{res}. If during a merge auction the resources not yet assigned to a reservation do not support the requested amount completely, the reservation is not granted and removed from the vector. The creation time *t* serves two purposes. Firstly, reservations expire after some time, which is necessary since the respective node could have gone offline or changed its interest in the meantime, but in lack of a path between the two nodes, this information did not arrive at the node that stored the obsolete reservation. Secondly, *t* is used during auction merge to project bids. For example, if a node's distance to the RDB is used as the bid type, it is necessary to estimate how much the node decreased the distance to its target since it issued the reservation.

For an auction merge operation at time point t', we use a combined distance- and energy-based bid type based on the concept of charging wireless sensor nodes in [29]. We use the operator $\oplus$ for the auction merge operation. The auction merge operation is (i) symmetric and (ii) unambiguous: For (i), given vectors A, B, it fulfills $A \oplus B = B \oplus A$. This is true due to the sorting of auctions by their bid. Thus, exchanging A and B will have no effect. Only if the bids are equal, this sorting would have no effect, but due to the continuity of the distance this never happened in our simulations. For (ii), $A \oplus B = C \Rightarrow A \oplus C = C \wedge B \oplus C = C$ holds true, as we preserve meta-information in the merge operation to prevent redundant merges.

Nodes are assumed to be in an emergency state if their resources are below a certain threshold (we used 10% in our experiments). Nodes use a bid of $1/d$ in emergency state and $-d$ otherwise, with d being the distance between node and RDB at reservation creation time. In emergency state, bids are projected as $(bid^{-1} + (t' - t)v)^{-1})$, otherwise distance-based projection is used with $\max(bid + (t' - t)v, 0)$ using a velocity estimate v.

Auction merge as outlined in Algorithm 1 is performed by a node each time it receives (or creates) a reservation vector contradicting its local copy, and results in a new vector combining the information. Vectors with obsolete BSNs are automatically rejected since the reservations are based on old resource availability information. Next, we remove expired reservations and reservations with RSNs lower than the last known RSN for that node. Using the projected bid values, the resources available at the RDB are distributed, starting with the reservation with the highest bid. If the requested amount exceeds the amount of resources left, the reservation is dropped from the vector, and reservations with lower bids have a chance of being granted. Afterwards, the node that held the merge auction replaces its local reservation vector for the respective RDB with the updated version.

Algorithm 1 Auction merge algorithm.

```
procedure AUCTIONMERGE(A, B)
    if A.BSN < B.BSN ∧ A.e_max ≠ B.e_max then
        ↑ B
    end if
    if B.BSN < A.BSN ∧ B.e_max ≠ A.e_max then
        ↑ A
    end if
    Union[] ← ∅
    for r ← A.R ∪ B.R do
        if ¬expired(r) ∧ ¬obsolete(r.RSN) then
            if r.NodeID ∈ Union then
                if Union[r.ConsumerId].RSN < r.RSN then
                    Union[r.NodeID] = r
                end if
            else
                Union[r.NodeID] = r
            end if
        end if
    end for
    e_left ← A.e_max
    R ← {}
    for r ← reservations in Union sorted in descending order of bid value at current time t' do
        if e_left ≥ r.e_res then
            e_left ← e_left − r.e_res
            R ← R ∪ {r}
        end if
    end for
    ↑ new ReservationVector(A.BeaconID, max(A.BSN, B.BSN), A.e_max, R)
end procedure
```

B. Generation of Reservation Requests

Like in *Greedy Selection*, the selection strategy that uses *AORVA* chooses the RDB with the least cost as its target. It only considers RDBs for which there are advertisements in the store. Instead of assuming the advertised amount of resources, it calculates the maximum amount of available resources considering existing reservations in its local vector and the node's current bid, even if this would result in other nodes being removed from the vector. If a suitable RDB has been found, the node increases its RSN and creates a reservation vector. This vector contains a single reservation by the node itself with the maximum amount of available resources or the amount necessary to charge back to 100% (whichever is lower), the current time, RSN and bid, and merges it with the corresponding local vector. Next, the node generates a reservation request *RReq*(NodeID, RSN, Adv, R, HSeq, TTL) with its unique identifier NodeID, its current RSN, the RDBs advertisement, the local reservation vector R, a maximum hop count TTL, and the sequence of hops passed HSeq initially only containing the node itself. The protocol is optimistic as there is no form of reservation acknowledgments by other nodes. After broadcasting the RReq, the node starts HEADING towards the RDB immediately. It is only informed if its reservation has been denied, in which case a reservation response RResp is generated.

The node stops HEADING if (i) after merging a reservation vector from a RReq or RResp for the target RDB, the node's reservation is no longer granted, or (ii) the node receives a new advertisement and reevaluates its decision. By sending a new RReq with an increased RSN, the node implicitly withdraws its previous reservation. However, the node may also choose to no longer pursue any RDB. In this case, the node cancels the reservation by sending a RReq for the last targeted RDB with the increased RSN and a reservation of 0 resources.

Due to mobility, the neighbor set and reachable nodes change frequently. To discover new competitors, the node periodically (Reservation Repetition Interval) sends out a new RReq with an updated bid, creation time, and RSN.

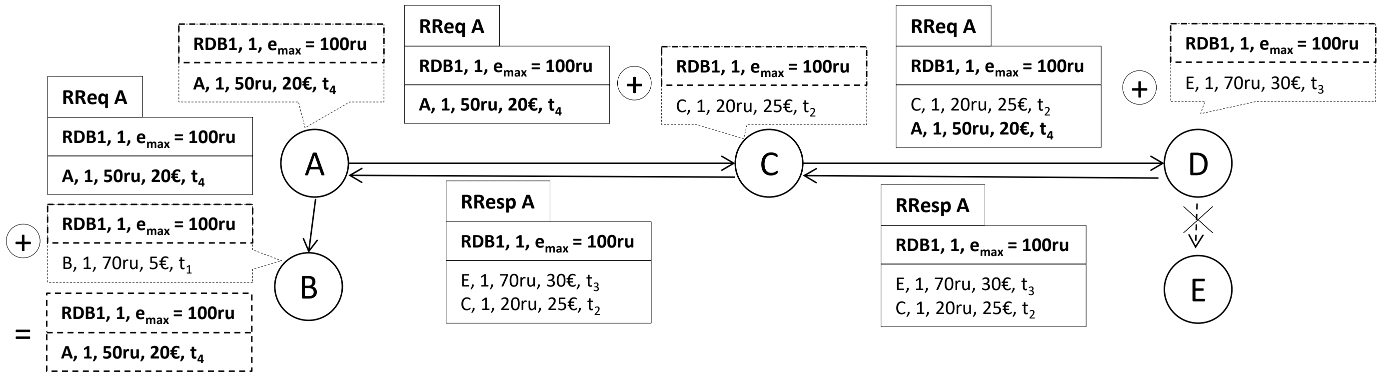

Figure 1: Forwarding a RReq by node A in *AORVA* with reservations in the form of (NodeID, RSN, reserved resources, bid, creation time). Local vectors at nodes are marked with dashed lines. The merging of vectors from the response is omitted.

C. Forwarding Requests and Sending Responses

The RReq is flooded up to a number of TTL hops. On receiving a RReq, a node first compares the request's RSN with the value for the given NodeID in its local table. If the same or a higher RSN has already been seen, the RReq is dropped. Next, it updates its local advertisement store with the advertisement contained in the request. If this advertisement is obsolete, that is, the receiving node has an advertisement with a higher BSN and a different amount of resources available than specified in the RReq's reservation vector, it replies with a RResp containing the more current advertisement. This allows the originator of the request to adjust its reservation to the new situation, for example increasing the amount of resources reserved. The receiver auction merges the vector from the request R_{req} with the node's local version R_{loc} while considering the projected bid at the current time, and replaces both R_{req} and R_{loc} with the updated result. If the requester is still a member of the vector, the node decreases the TTL by 1, adds itself to the hop sequence, and broadcasts the RReq containing the updated vector. Otherwise, it generates a reservation response RResp with the updated vector and routes it via the reverse path HSeq. Nodes along the RResp's route perform auction merge to update their local state with reservation information from nodes that forwarded the RReq.

Figure 1 shows an example of routing a RReq and corresponding RResp. For simplicity, we ignore projections by using money as a time-independent bid. Initially, Node A wants to reserve 50 of the 100 available resource units (ru) at RDB 1. It adds itself to the reservation vector for RDB 1 and broadcasts a RReq. Node B already reserved 70 ru at the same RDB; however, after receiving A's RReq and merging the vector with its local version, B finds that its bid is smaller, and thus A has precedence during the auction. Since the remaining 50 ru do not support the reservation of 70 ru by B, it is removed from the vector. Afterwards, B rebroadcasts the RReq.

The reservation made by C at RDB 1 has a higher bid than A's. However, both reservations can be fulfilled by the RDB. C merges both reservation vectors and broadcasts A's request with the updated version. D itself has not reserved any resources at RDB 1, though its local vector contains a reservation by node E. After performing auction merge, the reservation by A is removed as E is considered first and the remaining 30 ru do not satisfy A's reservation. Since A is no longer part of the reservation vector from the RReq, D does not further forward the RReq. Instead, it replies with a RResp informing A that its request has been denied. If A still desires to obtain resources from RDB 1, it either has to lower the amount of resources to reserve or increase the bid.

V. Evaluation

The goal of the evaluation is to assess the impact of our resource allocation service and the *AORVA* protocol on the lifetime of a disaster communication network. We simulate our service using the event-based Simonstrator Framework [22] which comprises the IEEE 802.11g standard from the ns-3 simulator [11] to model ad hoc Wi-Fi communication among mobile nodes. The scenario models and the setup of our simulations are presented in the following sections, followed by a detailed discussion of the obtained results.

A. Scenario Models

The evaluation is based on a sophisticated model of the scenario presented in [17] and summarized in the following. Given the likelihood of over-competition in a post-disaster scenario, we propose a second scenario in this paper to explicitly study the system behavior under over-competition.

1) Scenario: Long Term Behavior (s1): In scenario s1, the long term behavior of the network is studied to assess the lifetime of the network with different configurations of the proposed resource allocation service. Nodes are placed randomly on streets and places on the simulated map, starting with different initial energy levels. One resource per node is placed randomly on the map, providing an energy capacity of 200% (i.e., enabling two full recharges). This scenario reflects the beginning of an emergency situation where nodes discover resources over time and try to consume them, if they run out of energy. Although the number of resources is limited, the number of nodes which simultaneously have a demand for these resources varies greatly depending on the node density in an area and the availability of other resources in proximity.

Table I: Scenario and Simulation Setup for the Long Term Scenario (s1) and the Over-Competition Scenario (s2).

Simulated Area $[m \times m]$	2000 $\times$ 2000
Max. WiFi Comm. Range $[m]$	100
WiFi Standard	802.11g
Movement Speed $[m/s]$	$1.5 - 2.5$
Movement	13 attraction points with 20 % random waypoint probability
Density $[\text{nodes}/km^2]$	s1: 25, s2: 12.5
Max. Battery Capacity	14 400 ru (Resource Units)
Start Energy	s1: normal distributed, $\mu = 67\,\%$ s2: fixed = 30 %
Initial Node Placement	s1: random, s2: central
Number of Attraction Points	13 in parks
RDB Generation Interval [*min*]	s1: 2, s2: immediately
Energy Amount per RDB [*ru*]	s1: 2 $\times$, s2: 0.5 $\times$ max. Bat. Cap.
Overall Energy [*ru*]	s1: 2 $\times$, s2: 0.5 $\times$ #Nodes $\times$ max. Bat. Cap.
Roaming Cost $[ru/s]$	1.0
Heading Cost $[ru/s]$	3.11
Heading Threshold	
– Reservation Oracle	.1, .2, .3, <u>.4</u>, .5, .6, .7, .8, .9
– En Passant	.1, .2, .3, .4, .5, .6, <u>.7</u>, .8, .9
– Greedy Selection	.1, <u>.2</u>, .3, .4, .5, .6, .7, .8, .9
– Reservation Vector	.1, .2, <u>.3</u>, .4, .5, .6, .7, .8, .9
Resource Announcement Timer [*s*]	<u>5-10</u>, 10-20, 20-40, 40-60
Reservation Vector Lifetime [*min*]	1, 2, 4, 5, 10, 20, 30, <u>40</u>, 80, 120
Reservation Repetition Interval [*s*]	1, 2, 3, 4, <u>5</u>, 7, 10, 12, 15, 20, 30
Memory Span [*min*]	1, 5, 10, 20, <u>40</u>, 60, 80, 100
TTL [*hops*]	1, 2, 3, 4, 5, 6, <u>7</u>, 8, 9

This scenario was used in [17], enabling us to compare *AORVA* against the respective baseline strategies.

2) Scenario: Over-Competition (s2): The second scenario s2 models a situation with high competition among nodes and a general scarcity of resources. In this scenario, all nodes start at the same location and have the same knowledge about available resources. Consequently, they compete for a limited amount of resources (only 50% of the demand can be satisfied). Such situations occur in the event of a disaster, for example, when organizations drop additional resources in a disaster area [24]. The scenario helps to assess the benefit of coordination among nodes with *AORVA*, as we expect a lower number of unsuccessful attempts to consume a resource.

B. Evaluation Setup

Table I summarizes all simulation parameters for both scenarios. Underlined parameters represent the optimal setting for the given scenarios as a result of an extensive parameter evaluation which is not presented in this paper. We configured the damping factor of the Wi-Fi model such that the maximum communication range of a broadcast is 100 m, with the effective communication range in dense scenarios being lower as determined by the 802.11 MAC model [11]. The simulated area of 2x2 km^2 uses real-world map data of a residential district from OpenStreetMap. The nodes' personal movement policy is based on attraction points. Nodes move with a speed between 1.5 and 2.5 m/s and randomly select one of the 13 locations marked as `amenity=park` in OpenStreetMap as their next target. Since resources are placed randomly on the map and, therefore, may not lie on a node's route, nodes may also select a random point as their next target instead of an attraction point. This behavior is controlled with an *exploration factor*, set to 0.2 in our simulations. A node pauses for 15-20 min before selecting its next target.

Nodes have a maximum battery capacity of 14 400 resource units (ru). Together with a consumption rate of $E_r = 1\,ru/s$, this allows nodes to communicate for 4 h in ROAMING state with a full charge. The consumption in HEADING is $E_h = 3.11\,ru/s$ considering the phones screen energy consumption with brightness set to 50 %. The estimated consumption in ROAMING and HEADING state is based on a power usage study of an HTC Dream smartphone conducted by Zhang et al. [31]. In s1, the initial energy of nodes is normally distributed with a mean μ of 67 % (9648 ru), which is a typical average battery charge of a user's smartphone [7]. For s2, we set the initial energy to 30% on all nodes to create instant demand for resources. The demand for energy is defined by the *Heading Threshold*. If a nodes energy level is below that threshold it tries to switch to HEADING state in order to collect energy resources. This *Heading Threshold* is optimized individually for the different allocation strategies. Some strategies need to allocate resources sooner than others, for example the *En Passant* strategy that only can allocate resources if the node has a demand and is nearby an energy resource. For *Over-Competition* scenario s2, we increase the *Heading Threshold* for the *Greedy Selection* strategy to from 0.2 to 0.3 to have the effect of instant energy demand for all used strategies. New resources are generated at uniformly distributed random places on the map and broadcast their availability and their remaining amount periodically by a given *Resource Announcement Timer*.

We measure the total number of nodes that are still alive and the available resources on the map over time (with a sampling period of one minute). Additionally, we measure the percentage of a node's lifetime spent in HEADING state, the average time spent with unsuccessful attempts to recharge, and the average first/half/last nodes dead metrics [9]. Each setup is repeated with ten different seeds for all sources of randomness, i.e., affecting node mobility and the placement of resources.

C. Behavior in the Long Term Scenario (s1)

The goal of the resource allocation service is to maintain a high node density as long a possible to support emergency communication for any node in the network. Figure 2a shows the number of online nodes over time for the duration of the simulation. As expected, both the *Greedy Selection* and *AORVA* perform in between the lower baseline (*En Passant*) and the upper baseline (*Oracle*), while allocating almost all resources available during the simulation (Figure 2b). While the overall behavior of a *Greedy Selection* and *AORVA* is comparable, the negotiations in *AORVA* have one significant benefit: *AORVA* further delays the point in time where the first nodes start to run out of energy. The first node dead metric is raised to 7:00h for *AORVA*, compared to 2:47h using *En Passant* and 4:55h using *Greedy Selection*. Using the global

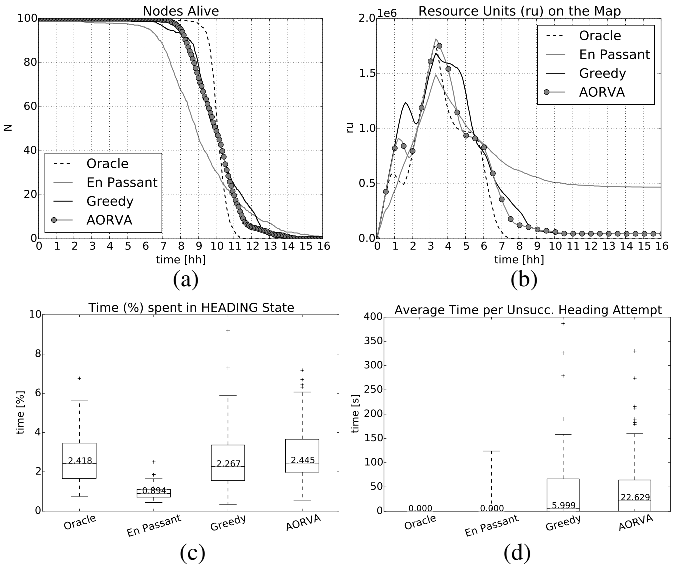

Figure 2: Evaluation results of Scenario 1

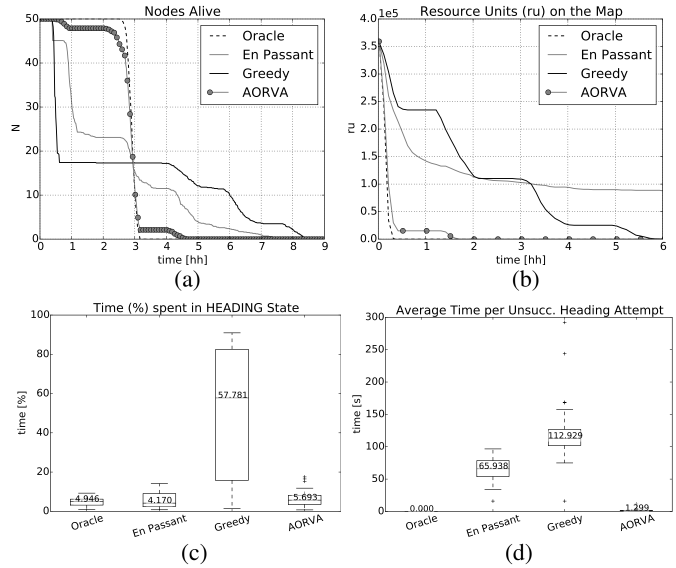

Figure 3: Evaluation results of Scenario 2

knowledge approach (*Oracle*), the first node runs out of energy after 7:55h. Maintaining a dense network and enabling all nodes to benefit from the applications running on top of that network for as long as possible is a crucial requirement. In our previous work [17] we have extensively evaluated the communication characteristics using the emergency communication network in scenario s1, revealing that a successful data exchange is no longer feasible when the total number of nodes drops below 50%. The fact that a fraction of nodes is supplied with energy for a long time afterwards does not lead to any positive effects for the overall network.

Figure 2c displays the percentage of their total lifetime nodes spent in HEADING (successful and unsuccessful attempts). Due to the fact, that nodes in HEADING have a 3.11 times higher energy consumption compared to ROAMING, time spent in this state should be minimized while still enabling a successful allocation and consumption of resources. Except *En Passant*, all approaches have a similar share of HEADING (2.267% – 2.445%) that is necessary to allocate the resources among the nodes. As expected, time spent in HEADING when using *En Passant* is low since nodes only try to consume resources in their direct reach, resulting in low chances of concurrent competitors. Even if there are competitors, the time spent in an unsuccessful HEADING attempt (Figure 2d) is negligibly small due to the proximity of known resources. As *AORVA* increases the overall lifetime of the full network significantly (Figure 2a), this higher node density towards the end of the lifetime results in higher competition for the remaining resources. Consequently, nodes in *AORVA* spend a larger percentage of their time in HEADING (Figure 2c) and have more unsuccessful HEADING attempts towards the end of the lifetime of the network (Figure 2d). This is due to partitions in the network, leading to nodes heading to the same resource from different directions without being able to exchange their reservation vectors beforehand. The impact of increased competition on the performance of *AORVA* is analyzed in more detail within scenario s2, discussed in the following section.

D. Behavior for Over-Competition (s2)

In scenario s2, we can evaluate how strategies deal with an over-competition in case of a scarcity of resources and whether consensus among nodes is reached. As previously discussed, nodes start at the same location with the same demand and knowledge about all resources in the scenario. As *En Passant* does not store information about resources outside of the direct communication range, knowledge is limited to these resources in that case. Figure 3a shows the resulting lifetime of nodes for different strategies. The reservation vectors (*AORVA*) have a significant positive impact, with the difference to *Greedy Selection* being also significantly larger than for scenario s1. Similar to the *Oracle* approach, *AORVA* can allocate almost all available resources immediately (Figure 3b) because all the nodes are in communication range when they start exchanging reservation vectors. This is also reflected in the similar share of time spent in HEADING state (Figure 3c).

The *Greedy Selection* performs even worse than the *En Passant* approach (which does not exchange or store any knowledge at all). This is an indicator that more knowledge without any coordination or consensus can have a significant negative impact on the overall wellbeing of the network. Figure 3c shows that *Greedy Selection* results in a massive increase in the time spent in HEADING (58%) and on average 113 seconds spent on unsuccessful attempts (Figure 3d). Reaching consensus using *AORVA* helps in avoiding such unsuccessful attempts, as nodes can derive that a resource will be depleted before their arrival. A closer look on the data used for Figure 3a shows, that *AORVA* can extend the time until the first node runs out of energy to 44:28 min. compared to 24:50 min. using *Greedy Selection*. The *Oracle* approach can further extend this time to a maximum of 2:38h, with all nodes going offline almost simultaneously afterwards. This results in a near-optimal network characteristic for the emergency communication. As all nodes try to charge completely at a resource and reserve the corresponding amount of energy in *AORVA*, the overall scarcity leads to a fraction of nodes that are unable to reserve and access any resource before their battery drains, as shown in Figure 3a. Still, the average time until half of the nodes went offline can be extended to 2:52h, almost reaching the 2:54h achieved with *Oracle* and outperforming the other strategies significantly. As mentioned in Section V-C, this behavior is important for a successful data transfer in the emergency network and, consequently, for the utility of the respective applications.

VI. Related Work

The problem of nodes running out of energy is usually addressed by reducing consumption, for example by using energy-aware routing schemes [27] or by applying data aggregation schemes [30]. While those approaches can extend the lifetime of the network, taking additional physical resources into account can increase the runtime of nodes significantly. To the best of our knowledge, there is no previous work on distributing vital resources among nodes in a MANET. However, the underlying concept of resource allocation and distributed consensus has been studied in related areas. Stavrakakis et al. [25] examined a scenario where players choose between a set of limited, low-cost resources and an unlimited resource with high costs. They found that providing players with knowledge, e.g., the number of competitors, may result in a higher social cost than in the case without additional information. Ayala et al. [3] formulated the problem of allocating limited parking spots to drivers as a finite assignment game where each driver selects a parking spot. The closest car will get the spot while the other competitors pay an additional cost for the unsuccessful attempt. Each instance of the parking spot assignment is assumed to be independent of all others, whereas in our case, the additional energy consumption in HEADING state leads to higher demand in all future instances. Parking spots can also be defined as sources of gravitational force [4]. Cars move in the direction of the strongest force, i.e., the area with the most parking spaces instead of just picking the closest spot. This approach reduces the number of competitors in an area with limited resources. Other approaches use a central coordination unit [20] or choose a dedicated coordinator [6] to allocate parking spots or charging stations [23] for electronic vehicles. A reverse situation of the resource distribution game is studied with the problem of recharging static wireless sensor nodes, using vehicles to recharge the nodes [29]. The use of auction-based resource reservation schemes has been studied for network resources like bandwidth or storage [12]. The economic sector uses decentralized market protocols for allocating tasks among agents that compete for scarce resources [28]. In these scenarios, agents trade tasks and resources at prices determined by an auction protocol.

VII. Conclusions and Future Work

In this paper we introduced a *decentralized reservation protocol* called *Ad hoc On-demand Reservation Vector Auction* (*AORVA*) to distribute scarce resources among nodes in a post-disaster scenario. The goal of the resource distribution is to extend the lifetime of an emergency ad hoc network and to enable as many nodes as possible to benefit from the respective services for as long as possible. We evaluated *AORVA* in two representative scenarios taking into account high over-competition of available resources. *AORVA* is especially beneficial in dense scenarios with an otherwise high probability for over-competition among nodes as shown in our evaluation. Here, all available resources are allocated and the lifetime of the overall network is increased to 98.9% compared to a centralized approach. Even if there is little competition among nodes, *AORVA* increases the time until the first node runs out of energy by a factor of 1.42 compared to related works.

We are currently investigating the impact of less greedy recharging strategies, where nodes state a reduced demand and recharge more frequently instead. Thereby, we expect a more evenly distribution of resources across the area, which should help to maintain a connected network while nodes are recharging. In this work, we are considering charging at a resource to be instant, like a node taking an energy pack and moving away. In future work we like to focus on a more realistic charging phase considering large non mobile energy resources, such as a car battery. This will probably have an influence on the nodes movement and will result in a higher node density around energy resources. The impact of this changed behavior on the overall resource allocation and reservation still needs to be investigated. Also considering energy resources that can be carried by a node and distributed afterwards, opens new interesting research questions.

We further conducted a large field trial with 125 people participating in an emergency communication network. Based on the analysis of the participants behavior and interactions [2] during the field trial, we plan to extend our strategies to account for real-world effects of human behavior and additional interaction outside of the application.

Acknowledgment

This work has been co-funded by the following projects: the LOEWE initiative (Hesse, Germany) within the NICER[1] project, the German Federal Ministry of Education and Research within the SMARTER[2], the Software Campus project "DisasterSense" (Support Code: 13N13406 & 01IS12054) and the German Research Foundation (DFG) as part of project B1 within the Collaborative Research Centre (CRC) 1053 – MAKI[3].

[1]Networked Infrastructureless Cooperation for Emergency Response
[2]Smartphone-based Communication Networks for Emergency Response
[3]Multi-Mechanisms Adaptation for the Future Internet

REFERENCES

[1] A. Al-Akkad, C. Raffelsberger, A. Boden *et al.*, "Tweeting 'when online is off'? Opportunistically creating Mobile ad-hoc Networks in Response to Disrupted Infrastructure," in *ISCRAM*, 2014.

[2] F. Alvarez, L. Almon, P. Lieser, T. Meuser, D. Yannick, B. Richerzhagen, M. Hollick, and R. Steinmetz, "Conducting a Large-scale Field Test of a Smartphone-based Communication Network for Emergency Response," in *ACM CHANTS (accepted for publication)*, 2018.

[3] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, "Parking Slot Assignment Games," in *ACM SIGSPATIAL GIS*, 2011.

[4] D. Ayala, O. Wolfson, B. Xu, B. DasGupta, and J. Lin, "Parking in Competitive Settings: A Gravitational Approach," in *IEEE MDM*, 2012.

[5] M.-C. Chuah and P. Yang, "Performance Evaluation of Node Density-based Adaptive Routing Scheme for Disruption Tolerant Networks," *IJAHUC*, vol. 3, no. 3, pp. 174–184, 2008.

[6] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte, "A Cooperative Reservation Protocol for Parking Spaces in Vehicular Ad Hoc Networks," in *ACM MobiCASE*, 2009.

[7] D. Ferreira, A. K. Dey, and V. Kostakos, "Understanding Human-Smartphone Concerns: A Study of Battery Life," in *IEEE PerCom*, 2011.

[8] M. Gallucci, "Rebuilding Puerto Rico's Power Grid: The Inside Story," in *IEEE Spectrum*, 2018. [Online]. Available: https://spectrum.ieee.org/energy/policy/rebuilding-puerto-ricos-power-grid-the-inside-story

[9] M. J. Handy, M. Haase, and D. Timmermann, "Low Energy Adaptive Clustering Hierachy with Deterministic Cluster Head Selection," in *IEEE WiMob*, 2002, pp. 368–372.

[10] W. R. Heinzelmann, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *ACM/IEEE MobiCom*, 1999.

[11] T. R. Henderson *et al.*, "ns-3 Project Goals," in *ACM WNS2*, 2006.

[12] G. R. Hiertz, J. Habetha, P. May, E. Weib, R. Bagul, and S. Mangold, "A Decentralized Reservation Scheme for IEEE 802.11 ad hoc Networks," in *IEEE PIMRC*, vol. 3. IEEE, 2003, pp. 2576–2580.

[13] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, and C. Rohner, "Twitter in Disaster Mode: Opportunistic Communication and Distribution of Sensor Data in Emergencies," in *ExtremeCom*, 2011.

[14] M. Kobayashi, "Experience of Infrastructure Damage Caused by the Great East Japan Earthquake and Countermeasures against Future Disasters," *IEEE Communications Magazine*, vol. 52, no. 3, 2014.

[15] O. Landsiedel, F. Ferrari, and M. Zimmerling, "Chaos: Versatile and Efficient All-to-All Data Sharing and In-Network Processing at Scale," in *ACM Sensys*, 2013.

[16] P. Lieser, F. Alvarez, P. Gardner-Stephen, M. Hollick, and D. Boehnstedt, "Architecture for Responsive Emergency Communications Networks," in *IEEE GHTC*, 2017.

[17] P. Lieser, N. Richerzhagen, T. Feuerbach, L. Nobach, D. Böhnstedt, and R. Steinmetz, "Take it or Leave it: Decentralized Resource Allocation in Mobile Networks," in *IEEE LCN*, 2017.

[18] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, 2003.

[19] P. Pourbeik, P. S. Kundur, and C. W. Taylor, "The Anatomy of a Power Grid Blackout-Root Causes and Dynamics of recent major Blackouts," *IEEE Power and Energy Magazine*, vol. 4, no. 5, pp. 22–29, 2006.

[20] H. Qin and W. Zhang, "Charging Scheduling with Minimal Waiting in a Network of Electric Vehicles and Charging Stations," in *ACM VANET*, 2011.

[21] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-walk Nature of Human Mobility," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 630–643, 2011.

[22] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, "Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications," in *SIMUTOOLS*, 2015.

[23] D. Schürmann, J. Timpner, and L. Wolf, "Cooperative Charging in Residential Areas," in *IEEE ITS*, 2016.

[24] J.-B. Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, 2007.

[25] I. Stavrakakis and E. Kokolaki, "Managing Competition for (Public) Resources in Human-centric Networked Environments," in *International Conference on Computing, Networking and Communications*, 2015.

[26] N. Suzuki, J. Zamora, S. Kashihara, and S. Yamaguchi, "SOSCast: Location Estimation of Immobilized Persons through SOS Message Propagation," in *INCoS*, 2012.

[27] C.-K. Toh, H. Cobb, and D. A. Scott, "Performance Evaluation of Battery-life-Aware Routing Schemes for Wireless Ad Hoc Networks," in *IEEE ICC*, 2001.

[28] W. E. Walsh and M. P. Wellman, "A market protocol for decentralized task allocation," in *IEEE MAS*, 1998, pp. 325–332.

[29] C. Wang, J. Li, F. Ye, and Y. Yang, "Multi-Vehicle Coordination for Wireless Energy Replenishment in Sensor Networks," in *IEEE IPDPS*, 2013.

[30] J. L. F. Zamora, N. Suzuki *et al.*, "Battery-saving Message Collection Method for Disrupted Communication Service Areas," in *IEEE CCNC*, 2014.

[31] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, L. Zhangt, and R. P. Dickt, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," in *ACM/IEEE CODES and ISSS*, 2010.