Saving Resources in Discovery Protocol on Delay-Sensitive Rescue Mobile Networks

Janine Kniess, Luciana Arantes, Pierre Sens, Célio Albuquerque

To cite this version:

Janine Kniess, Luciana Arantes, Pierre Sens, Célio Albuquerque. Saving Resources in Discovery Protocol on Delay-Sensitive Rescue Mobile Networks. The 31st IEEE International Conference on. Advanced Information Networking and Applications (AINA-2017), Mar 2017, Tapei, Taiwan. hal-01515369

HAL Id: hal-01515369

https://hal.inria.fr/hal-01515369

Submitted on 27 Apr 2017

Saving Resources in Discovery Protocol on Delay-Sensitive Rescue Mobile Networks

Janine Kniess*, Luciana Arantes †, Pierre Sens†, Célio V. N. Albuquerque‡
* State University of Santa Catarina (UDESC), Joinville - SC, Brazil
†Sorbonne Universités, UPMC, LIP6, CNRS, Univ Paris 06, INRIA, France
‡Fluminense Federal University (UFF), Niterói - RJ, Brazil
E-mail:janine.kniess@udesc.br, luciana.arantes@lip6.fr, pierre.sens@lip6.fr, and celio@ic.uff.br

***Abstract*—The search for service providers (e.g., ambulance, fire truck, etc.) after a disaster, must take place within a short time. Therefore, service discovery protocol which looks for providers that can attend victims, respecting time constraints, is crucial. In such a situation, a commonly solution for ensuring network connectivity between victims and providers is ad hoc networks (MANET), composed by battery-operated mobile nodes of persons (victims or not). However, an efficient service discovery protocol must care about energy consumption of mobile nodes and also prevent useless movement of providers. These are the aims of the Resource Reservation Protocol (ΔRRP), presented in this paper. Applying both Gauss-Markov [1] and Mission Critical Mobility [2] models to characterize human mobility, performance evaluation results on the Network Simulator NS-2 confirm the effectiveness of ΔRRP protocol when compared to other protocols.**

I. INTRODUCTION

A disaster is an emergency situation that requires short response time to attend victims. The search for service providers (e.g., ambulance, fireman, etc.) must take place within a short time after the occurrence of the disaster. In such a context, a resource discovery and reservation protocol which allows victims to effectively localize and allocates service providers is extremely important. However, when a disaster occurs, regular communication is usually disrupted due to infrastructure damage and power outages.

Through a MANET composed by battery-operated nodes [3], victims can send messages reporting information about their status and asking for rescue or aid. However, such a situation will increase network traffic as well as redundant discovery messages for service providers. Flooded over the network, these messages will also induce large consumption of energy of mobile devices. Since the communication network relies on battery-operated wireless devices, it is thus essential to minimize energy consumption of these nodes in order to prolong their lifetime until the communication infrastructure is restored [4].

Many discovery protocols have been proposed for disaster situations [5], [3], [6], [7], [8]. Nevertheless, the majority of them do not consider the problem of battery energy consumption of those mobile nodes responsible for broadcasting discovery messages over the disaster area, ensuring connectivity between victims and service providers. Hence, aiming at reducing mobile devices energy consumption during the service discovery phase, we propose the ΔRRP protocol, an extension of the *Resource Reservation Protocol* (RRP), presented in [9]. The RRP protocol is divided into the *Service Discovery and Selection* phase and *the Invocation* phase.

After a disaster, a victim (client) that needs a service provider (e.g., an ambulance) would like that the latter arrives at the location within some maximum delay of time. To this end, the client will start the *Service Discovery and Selection* phase of the RRP protocol by broadcasting a discovery message to the network searching for the providers able to satisfy her/his requesting time constraints. The message will be broadcast over the network by intermediate nodes. Upon receiving answers from one or more provider nodes, the client will call one of them, starting then the *Invocation* phase. Notice that other clients can also send discovery and invocation messages concurrently looking for the same kind of providers.

According to [10], in disasters scenarios, victims keep close to each other and present similar behavior in the discovery phase, i.e., they send a lot of messages seeking for assistance. Thus, aiming at reducing the number of messages over the network and, consequently, saving battery energy spent by those nodes that ensure network connectivity, we propose to apply an aggregation message mechanism to the $RRP\ Discovery$ phase. A second contribution that we added to RRP concerns the reduction of useless movement of providers. In our case, ΔRRP prevents providers from moving towards a client whenever his/her request will in fact be satisfied by a second provider. In view of this requirement, we added an acknowledge mechanism to the *Invocation* phase of the original version of RRP.

We should point out that, since there is a great difference in the order of magnitude between the maximum time (in minutes) that a client waits for a provider and the time that this client waits for provider replies to his/her discovery request (in milliseconds), or the time that the invocation protocol takes (in milliseconds), the above two improvements do not entail much degradation to the RRP protocol. Hence, in order to confirm the gain in performance of ΔRRP, we conducted an extensive set of comparative experiments on top of the simulator NS-2. Contrarily to [9] whose simulations consider only Gauss-Markov mobility model [1], we used in our simulations the latter but specially the Mission Critical Mobility Model (MCM) [2], a mobility human model for obstacle-constrained ad hoc networks, tailored for disaster scenarios. By applying different metrics, we compare and discuss the performance of ΔRRP with RRP as well with flooding and gossip protocols.

The road map of the paper is organized as follows. Section II discusses some related work. Section III summarizes the RRP protocol while the extension that we propose to RRP are described in IV. Performance evaluation results are presented in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

In this section, we discuss some works related to service discovery protocols over MANETs as well as some obstacle-constrained mobility models where nodes, after the disaster, must get around obstacles.

Service Discovery Protocols: Konark Gossip presents in [6] a push/pull mechanism that allows clients and servers to discover and advertise services. However, its operation is tied to the existence of a multicast routing support. The FTA approach proposed by [7] is based on the theory of electrostatic fields. Requests to an instance of a given service type are routed selectively in the direction of the provider that generated the highest field gradient. However, this approach does not scale well when different types of services are available.

Narayanan and Ibe [3] apply algorithms to collect and distribute information during disaster. The purpose of these algorithms is to enable survivors to report their locations to a Command Center and, then, the rescue team could be forwarded to those locations. The authors analyze battery life metrics, but they do not propose any mechanism to reduce battery node's energy consumption. Gadallah and Serhani present in [8] a discovery protocol for disaster situations over MANET where service providers send announces to a central node which performs the selection of providers. In [5], Chenji et. al present a complex ad hoc system whose aim is to reduce the response time to detect victims in large urban area. We should point out that all the above protocols do not cope with energy consumption issues of battery-operated nodes.

Obstacle-constrained Mobility Models: Several mobile models have been proposed for disaster scenarios such as the Gauss-Markov mobility model [1]. In this model, for every constant time period, a node calculates the speed and direction of movement based on the speed and direction of the previous time period, along with a certain degree of randomness incorporated in the calculation. The node is assumed to move with the calculated speed and direction during the time period. However, Gauss-Markov mobility model does not take into account the presence of physical obstacles.

We are particularly interested in obstacle-aware mobility models since they can characterize movement constraints of people due to after disaster obstructions (e.g., fall of buildings). Two surveys [11] [12] summarize some of existing obstacle-aware mobility models. In Aschenbruck et al. [11], obstacles are modeled by a graph where its vertices correspond to the obstacle's vertices (polygons). Based on this graph, vehicles deduce the shortest movement path to avoid the obstacles. Similarly, in [13], nodes move using the shortest-safe path, which is the shortest distance path that avoids all static obstacles. Pomportes et al. [14] propose a solution based on Voronoi diagram. The latter is created with obstacle corners, defining safe paths for rescue team.

In the Mission Critical Mobility Model (MCM) [2], nodes move around obstacles in a way similar to how humans do. The MCM model offers two activity modes: the *emergency workers* and the *medical staff*. The former comprises groups of firemen, policemen or soldiers that, after answering to an event, immediately attend another event while the latter characterizes behavior of providers such as ambulances that, after completing their mission related to an event, return to a certain base point before attending a new one.

III. DISASTER SCENARIO AND RRP PROTOCOL

In this section we briefly describe the *Resource Reservation Protocol* (RRP) proposed in [9]. More details about the protocol can be found in the article. We consider disaster scenarios where communication infrastructure is not available neither for the rescue team nor for the citizen (victims or not). Rescue workers (providers) and citizens carry out mobile devices interconnected by some wireless technology (e.g., IEEE 802.11 MAC). In the disaster scenarios, we distinguish the following participants: Clients (victims), those that request a service or aid; Providers, offer a service (e.g.,ambulance); and Intermediate nodes, those that can retransmit and aggregate messages. They can also be clients or providers.

We assume that each node in the network is aware of its geographic position by means of a localization system. Intermediate and provider nodes are mobile and the latter should arrive at the point where the service is requested within a maximum delay. Client nodes are also mobile but remain static after sent a discovery message and while waiting for a service provider. The *Resource Reservation Protocol* (RRP) is divided into the *Service Discovery and Selection* phase and *the Invocation* phase.

A. The Service Discovery and Selection phase

The following constants are known by all nodes.

- $speed_{max}$: the maximum speed that a node can have;
- α: the forth and back delay of a one-hop message;
- β: the maximum time that an intermediate node, closer to the client, will store one response before forwarding it;
- $range$: a node transmission message range;

In order to discover available providers, client i broadcasts a discovery message, because the location of providers is unknown. The latter has the following information:

- id: i's identifier;
- $\#req$: identification of $i's$ request;
- XYZ: i's geographical coordinates;
- s: type of service;
- Δ_{tmax}: maximum delay that i will wait for service s.

Based on i's geographical coordinates and some of the above constants, both the *Service Discovery and Selection* protocol and client i can estimate the diameter R_i that defines the range within which providers should be looked for:

$R_i \leftarrow speed_{max} \times \Delta_{tmax}$

The discovery service mechanism limits the search diameter R_i, on the basis of the maximum speed that a node can reach, $speed_{max}$ (each type of resource knows this value), and the maximum response time for one request, Δ_{tmax}. Upon sending a discovery message, client i starts a timer initialized with $\Delta_{\textit{timerclient}}$ which corresponds to the delay that client i will wait for an answer to his/her discovery request. Such a delay is composed by Δ_{req}, which is proportional to R_i, plus a safety margin Δ_{margin}. If i does not receive responses within this $\Delta_{\textit{timerclient}}$, i will send a new discovery message. Δ_{req} and $\Delta_{timerclient}$ are, respectively, given by:

$$\Delta_{req} \leftarrow (\alpha + \beta) \times (R_i/range)$$
$$\Delta_{timerclient_i} \leftarrow \Delta_{req} + \Delta_{margin}$$

As a result of the discovery process, client i can receive answers from multiple providers which responded to its service request. Thus, RRP protocol provides a response selection mechanism, executed by intermediate nodes during the reply transmission, which aims at reducing the number of reply messages over the network. In order to select the providers answers, the mechanism takes into account the requester node's geographic location, the maximum response time to attend one request, and the speed that the service provider moves. Upon receiving a first reply from a provider or from another intermediate node, an intermediate node starts a timer which is proportional to β and inversely proportional to the distance between the client and the intermediate node that sends the reply. Hence, during this interval time, the intermediate node gathers responses from different providers. Upon expiration of the timer, it aggregates those replies into a single message and sends it, reducing, therefore, the number of replies retransmissions. For more details about RRP selection mechanism, see [9].

B. The Invocations phase

After the service discovery and selection phase, the client sends a service invocation message to the provider it has chosen among those that answered to his/her request. The selected provider replies with a service confirmation message in which the provider informs that it is available to go to the place where the service is required. It then begins to move towards the requesting client.

IV. DELAYED RRP PROTOCOL

In this section we present two proposals for improving RRP performance: (1) an aggregation message approach that reduces the number of request messages sent in the discovery phase of the RRP protocol, and (2) an invocation protocol that avoids useless provider movements.

Both improvements are possible due to the great difference in the order of magnitude between Δ_{tmax} (in minutes), the maximum time that a client will wait for a provider and $\Delta_{\textit{timerclient}}$, the time that this clients will wait for a provider answer to his/her request (in milliseconds), or the time that the invocation protocol takes (in milliseconds). Hence, if an extra Δ of time is added to $\Delta_{\textit{timerclient}}$ or an acknowledge mechanism is added to the invocation protocol, the incurred delay will not degrade the quality os service of RRP but will in fact improve it by either reducing number of message retransmissions, or avoiding unnecessary movement of providers. We denote this new version of the protocol ΔRRP.

A. Aggregating messages in the Discovery phase

In disasters scenarios, recharging of node's battery can be difficult or even impossible. Thus, the reduction of message request and reply retransmissions over the network will induce less energy consumption by the nodes. We then propose to decrease the number of message transmissions in the discovery phase of the RRP protocol by aggregating client request messages on intermediate nodes, whenever possible. To this end, the $\Delta_{timerclient_i}$ of client i will be incremented by Δ_{Agg_i}:

$$\Delta_{timerclient_i} \leftarrow \Delta_{req} + \Delta_{margin} + \Delta_{Agg_i}$$

Δ_{Agg_i} is a parameter of the DISCOVER message sent by client i. Fig. 1 shows the relation between the above Δs.

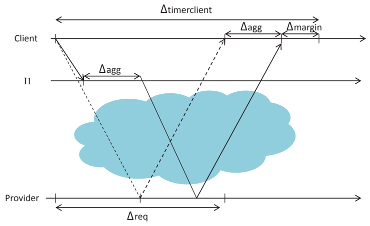


Fig. 1: Relation between the Δs

When receiving a discovery message from client i, an intermediate node $I1$ (Fig. 1), which is a one hop neighbor of i, starts a timer set to Δ_{Agg_i} in order to wait for discovery messages from other one-hop client neighbors. Upon expiration of the timer, node $I1$ aggregates the requests within a single message and broadcasts it. The intermediate node, aggregates discovery messages coming from different clients since it receives a discovery message from client nodes which are one hop from it. The intermediate nodes two hops away do not aggregate discovery messages from these clients but only resend the discovery messages.

Our choice for considering that only one-hop neighbors of clients aggregate discovery messages is accordance to [10], which states that disaster victims keep close and present similar behavior during a disaster situation. Therefore, intermediate nodes, which are neighbors of victims, tend to receive much more simultaneous requests for the same service than nodes that are far from these clients.

Algorithm 1 describes our discover message aggregation approach. It distinguishes two discovery messages: DISCOVERC, broadcast by requesting clients, and DISCOVERF, sent by intermediate nodes. Both messages keep the same parameters that the discovery messages of the original RPP protocol. Furthermore, Δ_{Agg} and R are included to the parameters of DISCOVERC and DISCOVERF messages respectively.

Algorithm 1 Aggregation of DISCOVER messages

```
 1: task Initialization
 2:     $timerAgg \leftarrow 0$
 3:     $\#req = 1$
 4: end task

 5: task T1 [DISCOVERY request]                                  ▷ Client $i$
 6:     $R_i \leftarrow speed_{max} \times \Delta_{tmax}$
 7:     $\Delta_{req} \leftarrow (\alpha + \beta) \times R_i/range$
 8:     $\Delta_{timerclient_i} \leftarrow \Delta_{req} + \Delta_{margin} + \Delta_{Agg_i}$
 9:     Broadcast DISCOVERC$(i, \#req, XYZ_i, s, \Delta_{tmax}, \Delta_{Agg_i})$
10:     Set $timerClient$ to $\Delta_{timerclient_i}$
11: end task

12: task T2 [DISCOVERC Reception]
13:     Upon reception of DISCOVERC
        $(j, \#req, XYZ_j, s, \Delta_{tmax}, \Delta_{Agg j})$
14:     $R_j \leftarrow speed_{max} \times \Delta_{tmax}$
15:     $msg \leftarrow \langle j, \#req, XYZ_j, s, \Delta_{tmax}, R_j \rangle$
16:     if $(timerAgg = 0)$ or $(\Delta_{Agg_j} < timerAgg)$ then
17:         Set $timerAgg$ to $\Delta_{Agg j}$
18:     end if
19:     $AggMsg \leftarrow AggMsg \cup msg$
20: end task

21: task T3 [timer expiration]
22:     Upon expiration of $timer$
23:     if (timer = timerClient) and (no replies from providers) then
24:         $\#req = \#req + 1$
25:         Broadcast DISCOVERC$(i, \#req, XYZ_i, s, \Delta_{tmax}, \Delta_{Agg_i})$
26:         Set $timerClient$ to $\Delta_{timerclient_i}$
27:     else
28:         FowardMsg()
29:     end if
30: end task

31: task T4 [DISCOVERF Reception]
32:     Upon reception of DISCOVERF$(AggMsg)$
33:     FowardMsg()
34: end task

35: function $ForwardMsg()$
36:     for all $m \in AggMsg$ do
37:         if $XYZ_i \notin m.R$ then
38:             $AggMsg \leftarrow AggMsg \setminus m$
39:         end if
40:     end for
41:     if $AggMsg \neq \emptyset$ then
42:         Broadcast DISCOVERF$(AggMsg)$
43:         $AggMsg \leftarrow \emptyset$
44:         $timerAgg \leftarrow 0$
45:     end if
46: end function
```

Client i broadcasts DISCOVERC messages and starts a timeout equals to $\Delta_{timerclient_i}$ (lines 9-10). Task T2 is executed whenever a first intermediate node ($I1$) receives a DISCOVERC message. The range of the client's request is computed (line 14) and the message is aggregated to other DISCOVERC messages that this node may have received from other client neighbors. If i's request is the first one, $I1$ will set $timerAgg$ to Δ_{Agg_i}. Furthermore if, in the meanwhile, $I1$ receives a DISCOVERC message from another client j and Δ_{Agg_j} is smaller than the current value of $timerAgg$, this one will be restarted to Δ_{Agg_j} (lines 16-17). Upon expiration of the timer, node $I1$ broadcasts the aggregated message to nodes within its range by calling the *ForwardMsg* function (line 28, task T3). When an intermediate node IN receives an aggregate message (task T4), it also calls the *ForwardMsg* function (line 33). This function removes from the set of aggregated messages those whose area, defined by the respective R, does not include IN, i.e., the geographical coordinates of IN is out of the R area (lines 36–40). Then, the node broadcasts the remaining messages (line 42). If the timer of client i, $timerClient$, expires (task T3) and client i did not receive an answer from any provider, i rebroadcasts its request (line 25). We should point out that, since $I1$ is close to its clients, we consider that the computed ranges centered on $I1$ satisfy the clients scope constraints.

Fig. 2 shows an example of the principle of the algorithm with two clients c_1 and c_2 that issue a request for a provider (messages m_1 and m_2 respectively) which will be received by the same intermediate node $I1$. We also consider that c_1 needs the service in a shorter delay than c_2, i.e., $\Delta_{tmax_1} < \Delta_{tmax_2}$. Upon receiving the DISCOVERC messages m_1 and m_2, $I1$ computes the range of both c_1's and c_2's requests, i.e., R_1 and R_2, and then aggregates both messages into a single one. This message will be forwarded until it is out of the border of the area limited by R_1. Then, at this time, m_1 will be removed from the aggregated message and, similarly, m_2 will be forwarded till received by nodes that are out of R_2 area.

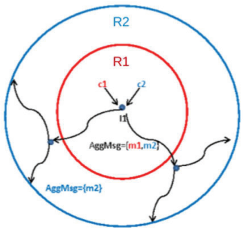


Fig. 2: ΔRRP Aggregation approach

B. Invocation Protocol

In the RRP protocol briefly described in Section III, a selected provider could initiate its movement towards the client without having a confirmation that this client still needs provider's assistance.

In the service invocation phase, clients and providers check the viability of attendance and the providers move to the place where the service is required. If *node P*, a selected provider by client i, receives a service invocation message informing that it was chosen by i, P sends a service confirmation message (*msgConfirmation*) to i, in which P informs that it is available. If i does not receive the service confirmation message from P within Δ_{inv}, client i will resend the discovery message. Note that $\Delta_{inv} = \alpha \times (R_i/range) + \Delta_{Agg}$.

On the other hand, we should remember that in the *Selection* phase, which takes place before the Invocation protocol, P might have received several requests messages related to other clients than i and have answered to all of them. However, at the invocation phase, P can just attend one client at a given time. To address this issue, we have introduced in the ΔRRP invocation protocol a mechanism to deal with such limitation.

This mechanism works as follow. Upon receiving the *msgConfirmation* from P, i will send it an ACK message in order to confirm that it selects P. If the latter does not receive the acknowledgment confirmation message within Δ_{inv}, P will discard i's request. Otherwise, P moves to the point where the service is required and maintains the i's request information until it finishes the attendance. At the same time, if P has other pending requests from other clients to which it sent a *msgConfirmation* message, it sends a NACK message to them to inform that it has been allocated to i. In other words, among those clients which have selected P in the *Discovery and Selection* phase, P will be reserved to the client from which it received the first ACK message. We should point out that the adding of this extra phase in the invocation protocol avoids that two or more providers move to the requesting point for satisfying the same client request, which would waste the provider's time that could, instead, attend requests from other clients.

If a provider P is in transit towards the incident location receives a discovery message from the same client i that fired the provider, it will consider a new request and restart the discovery process with this client. Only after the provider receives an acknowledgment confirmation message from this client it will be reserved again.

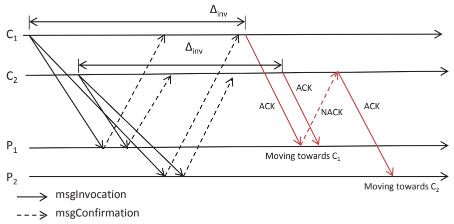


Fig. 3: ΔRRP Invocation protocol

Fig. 3 shows the ΔRRP Invocation protocol. We consider two clients C_1 and C_2 which, after the *Discovery and Selected* phase, have replies from P_1 and P_2 providers, i.e., both P_1 and P_2 can satisfy the clients respective requests.

Thus, at the invocation phase, C_1 and C_2 send an invocation message to both P_1 and P_2. When receiving such messages, P_1 and P_2 reply with a *msgConfirmation* messages to C_1 and C_2. Let's then suppose that both clients select provider P_1 by sending it an ACK message. However, as P_1 received C_1's ACK message before C_2's ACK message, it will send a NACK message to the latter and will starts moving towards C_1. When receiving the NACK message from P_1, C_2 will send an ACK message to P_2 which will start moving to C_2 when it receives the ACK.

V. PERFORMANCE EVALUATION

In this section, we present results of experiments conducted on top of the Network Simulator (NS-2) [15]. Our goal is to fairly compare the performance of the protocol ΔRRP with the original one, the RRP [9], but also with the $Flooding$ and $Gossip$ protocols. Contrarily to ΔRRP and RRP that restrict the search to the area defined by the diameters R (see Section III), the $Flooding$ and $Gossip$ protocols consider the whole area of the disaster. In the latter, every intermediate node broadcasts discovery messages with a certain probability p within [0,1]. For instance, if p is equal to 0.5 for an intermediate node, it broadcasts a message to 50% of its neighbors and if p=1, the protocol behaves like $Flooding$.

We consider two mobility models: Mission Critical Mobility Model (MCM) [2] and Gauss-Markov Mobility Model [1] (see Section II). For the MCM, we applied both the *emergency workers* and the *medical staff* activity modes. For sake of evaluation comparison, we also carried out simulations without NS-2, denoted *Exact Approach*, with mobility models. The aim of the *Exact Approach* is to evaluate the mobility models in the optimal case, i.e., to define which is, at a given time, the most appropriate providers that fit a client request. The output data of this method is compared with the results obtained by simulations over the network with NS-2.

A. Experimental Setup and Metrics

We consider two areas: 2000m x 2000m (4km^2) and 5000m x 5000m (25km^2). The distribution of the nodes over this area is performed by the mobility model. Each provider offers one type of resource, i.e., ambulances, firetruck, etc.. Resources are randomly distributed among the providers. We carried out a set of experiments varying the number of providers: 15% up to 30% of the nodes in the network. Each client asks for one resource. The number of clients also varies from 1 up to 10 as well as the maximum response time (Δ_{tmax}) which is set to 1.5 to 15.0 min. In order to prevent the search diameter R to reach dimensions close to the area size, which characterizes a flooding, we set the search diameter R to 1350m. To this end, when the speed is 1.5 m/s (resp., 15.0 m/s), Δ_{tmax} is set to 15.0 min (resp., 1.5 min).

The value of α is equal to 100 ms, and $\beta \leq \Delta_{timerclient_i}$ at all times. The confidence interval presented in the results is 95%. Table I summarizes some of the other parameters used in the experiments.

TABLE I: System Parameters

Parameter	Value
Number of nodes	100 up to 300
Number of obstacles	10 up to 20
Simulation time	28800s
Transmission range	250m
Service discovery package size	120 bytes
Response and Invocation package size	136 bytes
Initial energy on nodes	1000 J

Each node is equipped with the default wireless network energy module provided by NS-2 and according to IEEE 802.11 radio consumption model, we adjusted the transmission power to 54mW and the receiving power to 46mW. The initial energy of all the nodes is set to 1000 joules. Nodes present bounded energy supply and client nodes issue requests during the 28800 secs., the duration of the simulation.

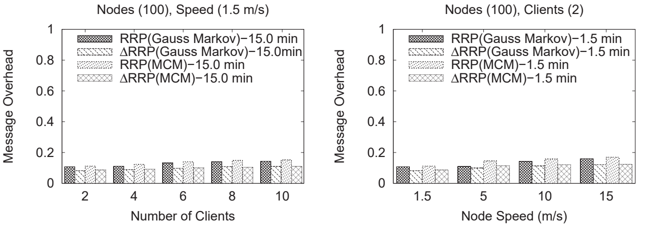
The following metrics are evaluated by all protocols:

- SD (*discovery success rate*): number of responses received per number of sent requests;
- EC (*node energy consumption*): total energy consumption consumed per node's initial energy. Let y = average_energy_consumption - initial_energy. $EC(\%)$ = (initial_energy * EC)/(y * 100);
- ID (*invocation success rate*): number of providers that answered to the service invocation phase per number of requests sent in the discovery phase;

- $TQoS$ (*time response quality of service*): average time to attend a request per total number of received responses.
- PE (*percentage of success*): number of replies per number of replies obtained with the *Exact Approach*;
- LK (*packet loss*): number of lost packets per total number of messages generated by the protocol;

B. Discovery phase evaluation

In the current experiments, we evaluate just the *discovery phase*. We have considered the MCM mobility model with the *emergency workers* activity mode as well as Gauss-Markov mobility model. We assume that clients need just one provider. A number of *obstacles* were distributed over the area, based on a graphical user interface provide by MCM mobility model [2].



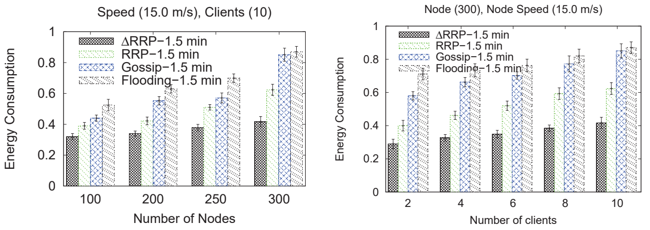
(a) Message overhead × clients (b) Message overhead × speed

Fig. 4: ΔRRP and RPP Message Overhead

Aiming at evaluating the impact of ΔRRP aggregation message approach, Fig. 4 summarizes the results related to message overhead of both the ΔRRP and RRP protocols in the discovery phase. Since the selection and invocation phase were not taken into account in these experiments, we set the value of β equal to 0 (see Section III). Fig 4(a) shows the discovery overhead for different number of clients and 100 nodes. Δ_{Agg_i} varies from 100ms up to 200ms. When 10 clients send requests simultaneously, ΔRRP overhead is 11% smaller for both MCM and Gauss-Markov mobility model than the other protocols. The overhead of RRP is 15.20% with MCM mobility model and 14.35% with Gauss-Markov. Fig 4(b) shows the overhead for different node speeds. The number of clients is 2 and $\Delta_{Agg_i} =$ 100ms and 200ms.

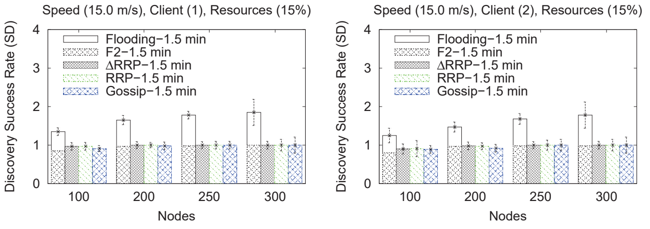
In the figure, ΔRRP mechanism outperforms RRP. However, for both protocols, the overhead grows with speed. For instance, with 15.0 m/s, the overhead is 12.35% and 16.99% for ΔRRP and RRP respectively. The reason for such increase is that the higher the speed, the greater the diameter R and, therefore, the higher the probability of finding providers. The overhead with Gauss-Markov is slightly higher than MCM's one. Such results reveal that some characteristics of MCM contribute to reduce the discovery overhead when compared to Gauss-Markov (see Section II).

Fig. 5 evaluates energy consumption EC of the four protocols: ΔRRP, RRP, $Flooding$, and $Gossip$. Fig. 5(b) shows EC in relation to the number of clients when Δ_{tmax} is equal to 1.5 min. We can observe that ΔRRP protocol presents the lowest energy consumption thanks to the discovery messages aggregation mechanism. As the number of clients increases, $Flooding$ and $Gossip$ protocol waste a lot of energy on the discovery phase. Similarly, in Fig. 5(a), as the number of nodes increases, ΔRRP outperforms all other protocols. Such results confirm that, by reducing the number of message transmissions, our aggregation approach improves the performance of the discovery phase. For instance, when the number of clients is 10, EC is 41.69% for ΔRRP, 62.35% for RRP, 85.10% for $Gossip$, and 87.02% for $Flooding$ after 288800 secs. of simulation. It is worth remarking that ΔRRP protocol saves energy without degrading the discovery phase.



(a) Energy consumption × nodes (b) Energy consumption × clients

Fig. 5: ΔRRP Energy Consumption



(a) SD Client1 × nodes(4 km^2) (b) SD Client2 × nodes(4 km^2)

Fig. 6: Discovery Success Rate (SD) in relation to nodes

The discovery success rate, SD, in regard with the number of nodes when two clients issue requests simultaneously, is shown in Fig. 6. In the experiments, 15% of the nodes offer the resources, the nodes present limited energy supply, and the number of request providers is 1. Fig. 6(a) and 6(b) show SD for *client 1* and *client 2* respectively. When N=100 and $speed_{max} =$ 15.0 m/s, ΔRRP, RRP, $Flooding$, and $Gossip$ protocols present SD below 1. The reason for such rate is that as the search diameter R is 450m, any provider is capable of sending responses within the period defined by $\Delta_{timerclient_i}$.

Fig. 7 shows the SD when the speed of nodes varies, considering different area sizes. The aim is to evaluate network connectivity of the protocols. We set N= 200 and 20 obstacles were deployed over the area. In Fig. 7a, when $speed_{max} =$ 15.0 m/s and the area size is 4 km^2, the SD for $client1$ and $client2$ with $Flooding$ protocol is 1.58 and 1.45 respectively. In this case, all providers that have the requested resource reply to the client node, including those which are not fast enough to reach the incident point in time. To overcome this problem, we extracted from all the responses received by clients i those that respect i's time constraints. We named this mechanism as $F2$ as shown in the figures, Fig. 7(a), Fig. 7(b), and Fig. 7(c). Consequently, with $F2$, SD becomes equal to 0.78 and 0.68.

Such a reduction can be explained because in the $Flooding$ protocol the load of discovery messages as well as the number of collisions and dropped discovery messages are higher than in ΔRRP, RRP, and $Gossip$ protocols. In Fig. 7(c) (25km^2 area), ΔRRP has reached the goal of 1 (one) response at both 10.0 m/s and 15.0 m/s. On scenarios where nodes move at 1.5 m/s and 5 m/s, the SD is 0.67 and 0.95 respectively. This behavior is due to the low number of providers within the radius. RRP, $Flooding$, and $Gossip$ protocols follow the results presented by ΔRRP. Over 5 m/s node mobility, the SD under the $F2$ is about 0.55 to 0.95 for RRP, and 0.94 for $Gossip$.

C. Evaluation of all phases of the protocols

Discovery, selection, and invocation phases have been considered in the current experiments.

The service invocation success, ID, expresses the percentage of satisfied service requests. Fig. 8 shows the ID successful rate for each protocol when the number of nodes varies. Based on the MCM mobility model, 100 up to 300 nodes were distributed over the 4km^2 and 25km^2 disaster areas. Two clients simultaneously require providers, 15% of the nodes had the resource, 10 and 20 *obstacles* were distributed respectively to the area size and the value of β is 50ms. The *emergency workers* activity mode is applied in Fig. 8(a), Fig. 8(b), and Fig. 8(c) while the *medical staff* in Fig. 8(d). We observe that in all figures, ΔRRP ID does not degrade when the number of network nodes increases, i.e., message overhead is reduced being responsible for keeping the invocation success rate constant.

In Fig. 8(a), ΔRRP has reached the goal of 1 (one) response for all number of nodes with *client1*. However, when N=100 (see Fig. 8(b)), *client2*'s ID is equal to 0.93. Compared to *client2*, we can conclude that, in this scenario, *client1* has benefit from its geographical position in relation to the providers and obstacles geographical positions. Overall, the ID for $F2$ protocol has a lower ID than the one of ΔRRP protocol. Such a difference happens because the $Flooding$ protocol generates a larger number of responses which increases the number of collisions. In Fig. 8(c) (25 km^2 area), contrarily to the other protocols, the ID of ΔRRP meets the goal of 1 (one) response, except when N=100. Notably, such a result ensures that the acknowledge message mechanism included in the invocation protocol of ΔRRP prevents useless movement of providers, avoiding that two or more providers move towards the same requesting client, reducing, therefore, the number of providers useless response. Fig. 8(d), we note that both protocols have similar behaviors under this mobility model. Consequently, our extension to RRP *Invocation* protocol performs efficiently even in the presence of different mobility patterns and obstacles.

Fig. 9a summarizes nodes energy consumption EC considering all phases of the protocols. When the number of clients is 4, EC is 37.6% for ΔRRP, 78.40% for $Flooding$, 44.80% for RRP, and 58% for $Gossip$. With 10 clients, ΔRRP can save around 45% of energy when compared to $Flooding$, and around 20% when compared to RRP. We also observe that ΔRRP performs better when the number of clients increase.

Packet loss rate, LK, is shown in Fig. 9b. $Flooding$ and $Gossip$ lose a significant number of messages compared to ΔRRP and RPP. When N= 300 and 15% of the providers nodes, the LK is 30.69% for $Flooding$, and 12.93% for ΔRRP. Once more, the results confirm that our aggregaton mechanism contributes for the decreasing of message collisions and packet loss.

Fig. 9c shows the response quality provider attendance ($TQoS$) for the requesting nodes, i.e., the average attendance time. The $TQoS$ is better for ΔRRP compared to $Gossip$ in all evaluated scenarios. Such a behavior can be explained since, during the reply transmissions (selection phase), ΔRRP selects the responses of the best providers, discarding those with higher time delay. The figure also includes the $TQoS$ of the *Exact Approach*. As we can observe, the $TQoS$ of RRP and ΔRRP are very close to the $TQoS$ of the *Exact Approach*. Results suggest that the aggregation mechanism has no negative impact in the $TQoS$. Furthermore, ΔRRP performs better in scenarios with higher number of resources. For instance, when $N = 300$, the $TQoS$ for ΔRRP (resp., *Exact Approach*) is equal to 50.12s (resp., 49.80s). On the other hand, when $N = 100$, the $TQoS$ for ΔRRP (resp., *Exact Approach*) is equal to 54.80s (resp., 52.45s).

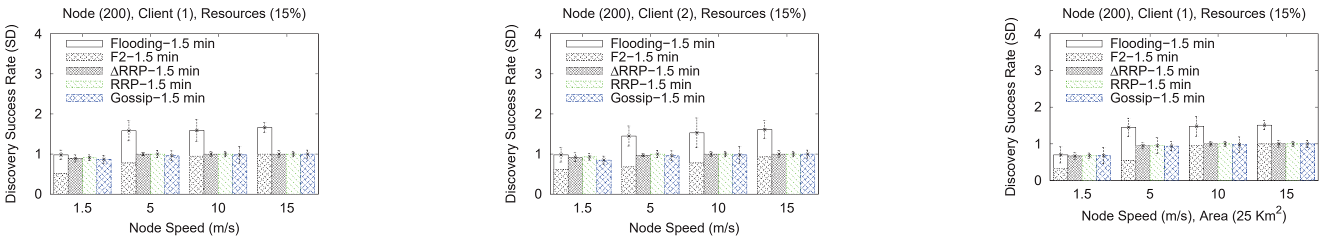
TABLE II: ΔRRP Percentage of Success (%) × Nodes

Node	ΔRRP	RRP	$Gossip$
Resources (15%), speed$_{max}$ = (15.0 m/s), Δ_{tmax} (1.5)			
100	72.2 (CF:1.7)	71.9 (CF:1.5)	73.4 (CF:2.9)
300	78.35 (CF:1.45)	77.24 (CF:2.4)	72.5 (CF:2.6)
Resources (30%) speed$_{max}$ = (15.0 m/s), Δ_{tmax} (1.5)			
300	81.0 (CF:1.5)	78.4 (CF:2.3)	72.38 (CF:2.5)
Resources (15%), speed$_{max}$ = (5.0 m/s), Δ_{tmax} (6.0)			
100	74.0 (CF:1.2)	72.46 (CF:1.4)	73.9 (CF:1.9)

Table II summarizes the percentage of success, PE, in regard with the *Exact Approach*. CF(%) corresponds to the confidence interval. As we can observe, ΔRRP protocol presents the higher PE when compared to RPP and $Gossip$ protocols. In addition, the PE of both RPP and ΔRRP increases when the number of resources increases, which is not the case for the $Gossip$ protocol due to the great number of dropped messages and the absence of selection mechanism. Such differences strengthen the advantages of the message aggregation approach applied by ΔRRP.

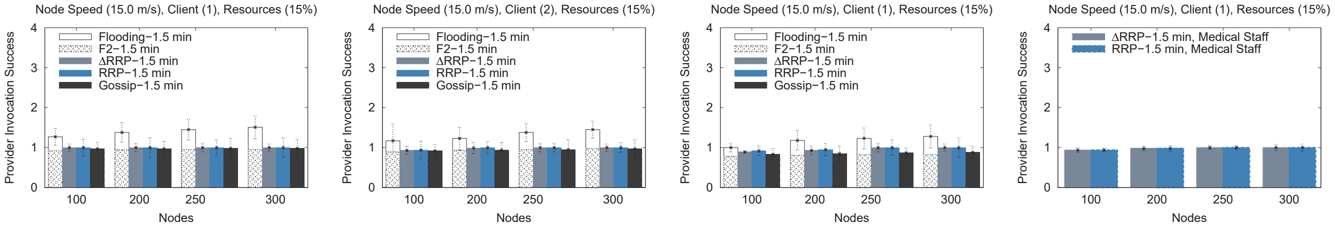
VI. CONCLUSION

We present in this paper the *Resource Reservation Protocol* ΔRRP which is an extension of the protocol RRP [9]. Aiming at reducing the number of messages over the network due to victim's requests and, therefore, node's battery consumption, we have proposed an aggregation mechanism for the discovery messages of the RRP service discovery phase. A second proposal includes the inclusion of an acknowledge mechanism to the protocol of the RRP *invocation* phase in order to avoid useless movement of providers. By applying different metrics, we have compared and discussed the performance of ΔRRP



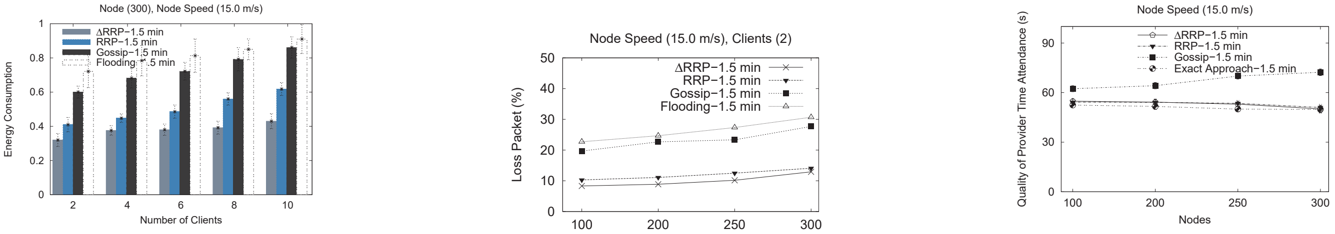
(a) SD Client1 × speed(4 km^2)
(b) SD Client2 × speed(4 km^2)
(c) SD Client1 × speed(25 km^2)

Fig. 7: Discovery Success Rate (SD) in relation to speed



(a) ID Client1 × nodes(4 km^2)
(b) ID Client2 × nodes(4 km^2)
(c) ID Client1 × nodes(25 km^2)
(d) ID Client1 × nodes(4 km^2)

Fig. 8: Invocation success rate in relation to nodes



(a) ID Energy consumption × clients
(b) ID Loss packet × nodes
(c) Quality of provider attendance × nodes

Fig. 9: ΔRRP, Energy consumption (a), Loss packet (b), Quality of Provider Attendance (c)

with three other protocols (RRP, $Gossip$, and $Flooding$) on top of NS-2 simulator. Results confirm the better performance of the ΔRRP in terms of reducing both the number of messages over the network and node's battery consumption as well as service resource saving.

REFERENCES

[1] C. Waal and M. Gerharz, "Bonnmotion: A mobility scenario generation and analysis tool," *Communication Systems group, Institute of Computer Science IV, University of Bonn, Germany.*, 2003.

[2] C. A. Papageorgiou, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "Modeling human mobility in obstacle-constrained ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 421–434, 2012.

[3] R. G. Lakshmi Narayanan and O. C. Ibe, "A Joint Network for Disaster Recovery and Search and Rescue Operations," *Computer Networks*, Jun. 2012.

[4] M. Y. S. Uddin, H. Ahmadi, T. F. Abdelzaher, and R. Kravets, "Intercontact routing for energy constrained disaster response networks," *IEEE Transaction of Mobile Computing*, vol. 12, no. 10, pp. 1986–1998, 2013.

[5] H. Chenji, W. Zhang, M. Won, R. Stoleru, and C. Arnett, "A wireless system for reducing response time in urban search & rescue," in *31st IEEE International Performance Computing and Communications Conference, IPCCC*, 2012, pp. 215–224.

[6] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark - a service discovery and delivery protocol for ad-hoc networks," in *IEEE Wireless Communications and Networking Conference (WCNC'03)*, 2003, pp. 2107–2113.

[7] V. Lenders, M. May, and B. Plattner, "Service discovery in mobile ad hoc networks: A field theoretic approach," *Pervasive and Mobile Computing*, vol. 1, no. 3, pp. 343–370, 2005.

[8] M. A. Serhani and Y. Gadallah, "A service discovery protocol for emergency response operations using mobile ad hoc networks," in *Proceedings of the 2010 Sixth Advanced International Conference on Telecommunications*, ser. AICT '10, 2010, pp. 280–285.

[9] J. Kniess, O. Loques, and C. V. N. Albuquerque, "Green service discovery protocol in mobile ad hoc networks," in *The Third International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*. Lisbon, Portugal: IARIA, 2013.

[10] E. A. D. Heide, "Common misconceptions about disasters: Panic, the disaster syndrome, and looting," 2004.

[11] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "Modeling mobility in disaster area scenarios," *Performance Evaluation*, vol. 66, no. 12, pp. 773–790, Dec. 2009.

[12] M. C. Lus Conceiao, "Modelling mobility based on obstacle-aware human behaviour in disaster areas," *Wireless Personal Communications*, vol. 82, no. 1, pp. 451–472, 2015.

[13] H. Y., H. W., N. K., and L. W., "Corps: Event-driven mobility model for first responders in incident scene," in *the IEEE Military Communications*, 2008, pp. 1–7.

[14] P. S., T. J, and V. V., "Ad hoc network in a disaster area: A composite mobility model and its evaluation," in *Advanced Technologies for Communications*, 2010, pp. 17–22.

[15] T. Issariyakul and E. Hossain, *An Introduction to Network Simulator NS2*. Springer, 2009. [Online]. Available: http://books.google.com.br/books?id=cD69He_oU60C