# Context-aware Self-adaptive Routing for Delay Tolerant Network in disaster scenarios

Erika Rosas[a], Felipe Garay[b], Nicolas Hidalgo[c]

[a]*Departamento de Informática*
*Universidad Técnica Federico Santa María*
[b]*Departamento de Ingeniería Informática*
*Universidad de Santiago de Chile*
[c]*Escuela de Informática y Telecommunicaciones, Facultad de Ingeniería*
*Universidad Diego Portales*

**Abstract**

In the generalized loss of communication as a result of extensive damage to the communication infrastructure in disaster or emergency scenarios, mobile ad-hoc networks can play an important role in improving communication in the affected areas and working around network disruptions. Communication protocols face a highly dynamic and congested network topology that require highly adaptable solutions. This article proposes a context-aware self-adaptive routing protocol for Delay Tolerant Networks that is able to adapt to different scenarios, allowing the participants of the network to select a DTN protocol according to past performance of routing protocols under the current scenario. We define metrics to implement this approach in a disaster scenario and evaluate our proposal with simulation over two different mobility models. The proposed protocol shows a significant reduction of energy consumption with a good tradeoff with the delivery rate.

*Keywords:* Mobile Ad-hoc Communication, Delay Tolerant Networks, Disaster Scenario

## 1. Introduction

Every year, millions of people are affected by natural disasters such as earthquakes, tsunamis, volcano eruptions, hurricanes, tornadoes, and floods, and governments around the world spend large amounts of resources on the reconstruction and the preparation for such calamities. In 2017 alone, there were 335 natural disasters worldwide, affecting 95.6 million people and produced economic damage of US$335 billion [1]. Moreover, man-made disasters may lead to huge destruction in cases of terrorist attacks or war-related events.

These events can severely damage public and private infrastructure, and can dramatically compromise the welfare of people. Furthermore, they also hit communication infrastructure such as Internet and cell towers, making it difficult for the affected people and decision makers to communicate and have access to information. Communication is crucial for supporting emergency prevention, response, and recovery efforts. During and after a disaster event, communication allows for the process of victims' location, situation awareness, and emergency notifications to be provided. People's mobile devices can be used during an emergency to exchange information through text messages, videos, photos, and audio. Several disaster response applications, such as Google People Finder [2] and the Red Cross App [3] assume a scenario with Internet access and fully reliable power supplies. Such assumptions are valid only for users located outside the disaster area. Nowadays, most of the people in the affected areas have one or more mobile devices such as smartphones and tablets with WiFi or Bluetooth technology, but they have restricted or intermittent access to Internet and limited power supplies. Moreover, the mobile industry reached a major milestone in 2017, surpassing 5 billion people connected to mobile services, and the expectation is that it will reach 5.9 billion in 2025, an equivalent to 71% of the world's population [4]. Mobile devices are especially helpful in such scenarios because they have fair processing power, storage capacity, and battery life, enabling people to send relevant information [5] [6].

In the generalized loss of communication as a result of extensive damage to the communication infrastructure, mobile ad-hoc networks can play an important role in improving communication in the affected areas and working around network disruptions [7][8][9][10]. In these types of networks, each device cooperates by forwarding messages in order to reach one or more destinations. This technology feature can be naturally combined with people's behavior and daily life events and fits the context of emergency scenarios because of their minimal configuration need and deployment. Mobile adhoc networks are able to extend the operation of applications for emergency management to the affected areas, allowing decision makers from government institutions, for example, to have accurate and timely information in such events.

We have studied the broad body of research devoted to improving protocols and algorithms for mobile ad-hoc networks that perform well under challenging disaster scenarios in [11] and conclude that a main alternative to cope with failures and partitioned networks are Delay Tolerant Networks (DTNs). DTNs have been developed for challenged networks to handle the absence of end-to-end connectivity between devices, e.g., interplanetary transmissions, rural communities, military connections, and disaster scenarios. These protocols benefit from the store-carry-and-forward paradigm to resolve intermittent connectivity. They attempt to send messages through other available devices that may have a better probability of reaching the destination. We have studied DTN performance in [12], considering traditional DTN protocols: Epidemic [13], MaxProp [14], Spray-and-Wait [15], Spray-and-Focus [16] and PRoPHET [17]. In traditional DTN protocols, there is a trade-off between delivery rates and overhead of messages: good delivery rates are generally related to good message latency and a high overhead induces high energy costs for mobile devices. Both metrics are particularly important when we apply these communication protocols to disaster environments. Traditional opportunistic protocols are unable to capture the specificities of a disaster scenario that require solutions that are aware of energy costs and other characteristics, such as proximity to the disaster area, or mobility of the participant nodes.

We propose a new protocol called CSAR, that stands for Context-aware Self-Adaptive Routing, that switches between routing protocols considering past performance under the current characteristics of the environment and the device. We study the use of a multi-criteria algorithm, with concepts that are of particular importance in disaster scenarios and provide each node the ability to take uncoordinated decisions about message routing. In order to select the appropriate routing protocol, we consider the performance of the protocols in the recent past, avoiding the use of pre-defined rules about when to use a specific routing protocol with this approach. Given the complexities of testing the proposal under large scale real-life settings, we use simulated environments. We evaluate the proposed solution in the simulator The ONE [18], providing mobility and energy models, which are specially tailored for disaster scenarios [19][20].

The remainder of this article is organized as follows: Section 2 introduces the main background concepts and related work. Section 3 details the proposed protocol, and Section 4 describes the case study of a disaster scenario. Section 5 presents and discusses the simulation results, describing the communication and mobility models considered for the evaluation. Finally, our concluding remarks and future work are presented in Section 6.

## 2. Literature review

Context-aware protocols are able to adapt to the current network situation and differ from traditional protocols that use encounter probability to decide whether to forward a message to a neighbor; instead, context-aware protocols use information such as node density, buffer capacity, or speed of movement. These systems are multi-criteria, in the sense that the routing decision has to be made considering different options and metrics. The CAOR protocol [21], for example, proposes a dynamic forwarding delay considering link quality, progress of the message, and energy and link validity. With an analytic hierarchy process, the protocol selects the next hop for a Mobile Ad-Hoc Network (MANET). Khan Kasi et al. proposed a context-aware routing protocol for Wireless Sensor Networks (WSNs) [22], a system that focuses on saving energy for sensors considering the energy level and heterogeneity of devices.

In the context of opportunistic networks, several context-aware approaches have been proposed but without considering the challenges of a disaster scenario [23][24][25][26][27]. Boldrini et al. [23] used a utility function to decide on routing messages using user information, social information, battery lifetime, and other physical characteristics of the devices. The solution called CARTOON [27] uses network contexts to change between two modes: epidemic if there is a low density of nodes and probabilistic mode in case of high density of nodes. Cabacas et al. [25] used contexts based on the history of encounters and velocity of the nodes and were able to predict where the nodes head to, with the objective of analyzing how good they are to forward a message. This is a deterministic protocol and nodes know their position, which is costly information for energy restricted devices. The closest work to our proposal is CAR [26]. Musolesi et al. defined contexts as a set of attributes that describe different aspects of the system to make routing decisions. A context is defined as a n-tuple $(X_1, X_2, \ldots, X_n)$ where each element is an attribute, for example, the battery level or the mobility of the host. Nodes compute the utility to forward a message considering the context of the current node and its neighbors, giving different weights to each attribute in the utility function. In contrast to CAR, our proposal includes the evaluation of performance of each protocol in order to select a different routing protocol, not only selecting whether to forward to a particular node.

Energy-aware versions of traditional routing protocols have been proposed. Spray and Wait, Gao et al. [28] used information about speed and residual energy to allocate the number of copies between the corresponding pair nodes in the spray phase. For the Epidemic protocol, De Rango et al. [29] proposed a dynamic adaptation of the number of copies sent to neighbors based on energy considerations and node density. Recently, Bista et al. [30] proposed an energy-aware protocol based on PRoPHET that considers the energy of nodes and available free buffer for storing forwarded messages. In the context of disaster scenarios, Uddin et al. [31] presented a protocol whose objective is to minimize the energy expended in communication, exploiting mobility and contact patterns in the network. These proposals improve network survivability while we propose to analyze a hybrid design that directly decreases the number of messages exchanged in the network. Other approaches use game theory to minimize total routing and rate allocation cost [32] and optimization strategies based on Bayesian games [33] that capture the multi-copy replication decisions, the energy constraints of nodes, and the belief about energy of other nodes. Due to the complexity of these algorithms, it may be difficult to use them in large areas with many nodes.

Hybrid models have also been proposed. Martín-Campillo and Martí [34] proposed a combination of MaxProx and TTR in order to improve the delivery ratio during the first steps in routing while maintaining low overhead for the rest of the routing steps. Lakkakorpi et al. [35] proposed a switch between an opportunistic protocol and a MANET protocol considering the density of the nodes, which is also the approach followed in [36] and [37]. Miyakawa and Koyama [38] discussed the use of PRoPHET and Spray and Wait at the same time to improve delivery rate using the probability of delivering a message used by PRoPHET to decide which protocol to use at each forwarding step. They stated that the Spray and Wait protocol has good performance under high mobility, so they propose to adjust the number of copies of a message using the information provided by PRoPHET. These approaches used static pre-defined rules that determine when to use a particular protocol. We propose a dynamic adjustment of the decisions to the current performance of each routing protocol enabling the flexibility to change a protocol that is not performing well under a particular scenario.

## 3. Context-aware self-adaptive model for DTNs

In this article we propose CSAR, a Context-aware Self-Adaptive Routing model for DTNs that allow the creation of a family of hybrid protocols that integrate DTN protocols considering different priorities of the network. The protocol takes into account the past performance of the protocols in different states of the nodes and environments, automatically deciding which is the best routing protocol for the current situation and defined priority.

In the CSAR model, a *node* refers to a mobile or static device that has storage, processing capacity, and wireless network connectivity, which is compatible with other nodes in the scenario, for example, WiFi and Bluetooth. We assume that the nodes can crash and stop participating in the network, that they are able to send messages to any node in the network and that they do not disseminate false information. Despite the last assumption, this work can be complemented with protocols that ensure trustworthines or secure communication in a DTN protocol as the one proposed in [39].

### *3.1. General Model*

The CSAR model is based on two concepts: *metrics* and *contexts*. Consider a set of available routing protocols $S = \{DTN_1, DTN_2, \ldots, DTN_d\}$:

- A *context* is defined as a n-tuple $(c_1, c_2, \ldots, c_n)$ of numeric values that represent the characteristics of a node and its environment. Each component represents one independent characteristic. Examples of the *characteristics* that can build a context are: mobility of the environment, density of the environment, and energy availability.

- A *metric* $m_j$ is the performance quantification of a protocol for a context. CSAR defines a set of $l$ metrics $(j = 1 \ldots l)$ for a given context and a protocol in $S$. Examples of the metrics associated may be the overhead of messages, delivery rate, latency of messages, and average time of messages in the queues.

Each node in the protocol builds knowledge based on metrics and contexts for the available protocols considering local and global information that other nodes disseminate in the network. The algorithm is described in Algorithm 1. First, a random DTN protocol from set $S$ is selected, and then metrics and context characteristics are initialized (line 1-3 Algorithm 1). We require a random protocol at first to accumulate evidence about all the protocols' performance. Context characteristics may be extracted directly from the device in real-time or computed during a longer time during the execution. For example, the energy available or the speed of the node can be directly obtained from the device in real-time. However, other context characteristics, such as the density of the environment, have to be approximately computed from the interactions with other nodes and updated from time to time.

During the execution, every time there is contact with a neighboring node the current selected $DTN_i$ protocol selects the messages to exchange according to its particular routing rules (line 6-7 Algorithm 1). Each node has a defined value of $DTN_{selected}$ according to the Algorithm's decisions, and there is no issue to have different protocols between the nodes that encounter, since they are building the messages to send to the other node locally. If any message has to be exchanged, the node includes local information about metrics and contexts to disseminate the behavior of used DTN protocols (line 8 Algorithm 1). Then, the exchange is performed, also receiving information from the neighboring node. With this information the node updates metric and context characteristics (line 9-12 Algorithm 1). Finally, every $\Delta t$ the CSAR protocol decides about continuing or changing the current DTN protocol based on the collected information (line 14-20 Algorithm 1). In the following we will describe details about this general protocol.

**Algorithm 1:** CSAR general algorithm in a node

```
DTN_selected = random DTN_i ∈ S;
initialize local metrics;
initialize local context characteristics;
t_select = t_current + Δt;
while in execution do
    if contact a neighbor node n_j then
        buffer ← select msgs to exchange according to DTN_selected;
        attach local metrics and contexts to the messages to exchange
          in the local buffer;
        exchange messages in buffer with n_j ;
        extract neighbor's metrics and contexts from the exchanged
          messages;
        update metrics;
        update context characteristics;
        update t_current;
        if t_current > t_select then
            if enough data then
                compute CSAR objective function for all DTN_i ∈ S
                  for the current context C;
                DTN_selected = DTN_i with the best objective function;
                t_select = t_current + Δt;
            else
                DTN_selected = random DTN_i ∈ S;
```

### *3.2. Metric and Context Information*

Messages exchanged between nodes contain information added at each hop by the node that has forwarded the message: the DTN protocol used, the metrics computed locally, the information for computing global metrics, and the context in which it has executed the DTN selected protocol. By extracting this information, the nodes are able to store a set of tuples $(c_i, m_j)$ for each DTN protocol in $S$, where $c_i$ represents a context and $m_j$ represents a metric, since the behavior represented as a metric value depends on the context in which the protocol is being executed. A way of representing this data is the graph shown in Figure 1 where for each DTN protocol there are points that represent the average value for a metric in a context. To simplify computation, we discretize the range for the context and compute average points for each context we have information for. This data allows for the estimation of the future performance of a DTN protocol in a particular context considering simple computation values: compute the line between the two closer points to the requested context so as to interpolate or extrapolate the value. The efficiency of this approach largely depends on the amount of data available for the set {DTN protocol, metric, context}: larger amounts may produce a better estimation instead of producing imprecise values when there are small amounts of data available for the particular protocol. Using this simple technique saves computation time and saves energy for the mobile device. If this is not a restricted resource, we could use complex approaches such as computing a linear regression with the available information.

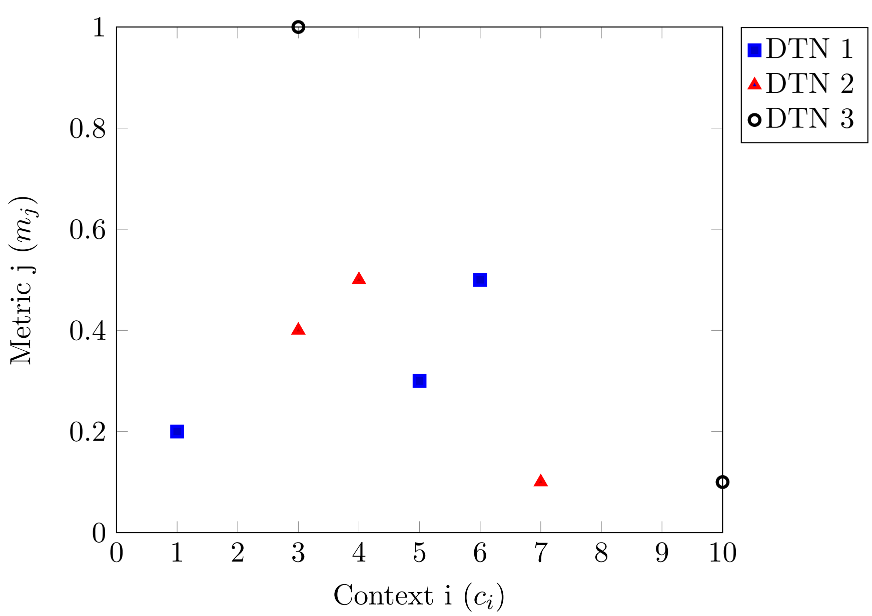


Figure 1: Metric and context information stored in CSAR

Additionally, we use a sliding window to consider only the lasts tuples $(c_i, m_j)$ stored in a node. The size of the sliding window impacts the consideration of the past history of the results.

### *3.3. Protocol Selection*

The value $\Delta t$ is the time interval used by the protocol to gather information about metrics before a re-evaluation of the DTN protocol. The protocol selection uses an objective function $f(m_1, \ldots, m_j)$, where $m_j$ represents the metrics considered to optimize the protocol. For each DTN candidate, we use the local current context of the mobile device to obtain an estimated value for each metric of the protocol. The protocol selected will be the one that optimizes $f$.

*3.4. CSAR communication, storage, and processing costs*

The main communication costs of CSAR compared to the use of only one DTN protocol are:

- An increase in the size of the messages exchanged: Metrics and contexts dissemination is performed using the same messages exchanged by the users in the network while trying to reach their destination. This increases the size of the messages exchanged but without incrementing the number of messages or connections that is the event that uses larger amounts of energy.
- Some DTN protocols exchange information about past encounters with neighbors. CSAR requires that this information is exchanged even if the selected DTN protocol does not use this information so as to be prepared to change to any DTN protocol in $S$. However, some protocols do not require any large data exchange, and others can use the same information to make different routing decisions.

The storage costs of CSAR go hand in hand with the communication costs:

- The tuples $(c_i, m_j)$ should be stored for each DTN protocol in $S$ while they fit in the sliding time window. These are small numeric values.
- The information to compute the contexts should be stored when they fit in the context time windows. These are small numeric values.
- The information required for all DTN protocols in $S$ should be stored in each node. This does not differ from what is stored in each node if we use DTN protocols such as Prophet or MaxProp protocols that require the history of past encounters.

Finally, processing costs largely depends on $\Delta t$ since this is the period of time for evaluating the objective function for each DTN protocol in S. For each $\Delta t$, the CSAR protocol computes average values for the contexts and predicts the metrics related to such contexts so as to obtain the comparative value to decide which is the best protocol to use.

## 4. Case Study in Disaster scenario

We use the general CSAR model proposed in the previous section to fit the requirements of a disaster scenario where the goals are to maximize the delivery rate while minimizing energy costs. We call this solution CSAR-DS. We assume that in a disaster scenario, people connected to mobile devices want to communicate via these devices but have restricted power supply in order to increase network lifetime.

### *4.1. Contexts selected for a disaster scenario*

Based on the study previously conducted in [12], we have found that the mobility of the nodes impacts how the DTN protocols perform. In order to characterize this behavior, we define the density of the neighboring nodes and the average contact time of the two context metrics to include in this proposal. Both contexts depend on the technology of communication between the nodes.

#### *4.1.1. Density neighbor nodes*

There are some DTN protocols that perform well in high density environments while others perform well in sparse environments. The density of nodes around a particular node is computed counting the number of encounters ($\#e$) a node has in a certain time window, called *context time window* (CTW). The CTW determines the set of encounters $e$ that will be considered in the computation of the *current* context and is expressed in seconds. The use of CTW allows to erase all the information of past encounters by storing the timestamp of each stored encounter in a queue.

The following equation describes the first context:

$$Density = \frac{\#e \in CTW}{CTW} \tag{1}$$

#### *4.1.2. Average contact time (ACT)*

The average contact time $ACT$ measures the speed of the nodes in an indirect way, which allows to characterize the mobility of the nodes in the network. When a node has high speed, it generally implies a low contact time between the node and their neighbors. The opposite occurs when the nodes have low speed, and consequently a higher average contact time with the nodes they encounter. Each time there is an encounter, a timestamp is stored in a local queue, and each time a connection is closed, another timestamp is stored.

The following equation describes the second context, where $n$ is the number of encounters in $CTW$, and $CTE_i$ is the Contact Time of the Encounter $i$:

$$ACT = \frac{\sum_i^n CTE_i}{n} \tag{2}$$

### 4.2. Metrics selected for a disaster scenario

CSAR-DS minimizes the energy spent and maximizes the delivery rate at the same time. The goal is to avoid the case where a protocol with a high delivery rate is selected using high amounts of energy, or the opposite situation, where a protocol does not make any copies of the messages, poorly performing in terms of delivery rate.

As a representation of the energy spent, we have used the number of copies ($nc$) the protocol creates for each message since it is a value that is related to the overhead of the protocol and to the energy consumption according to an equation presented in [34], where the authors relate the energy with the number of transmitted messages, the average contact time, and the average number of nodes contacted.

As in the case of contexts, we use a time window for the metrics information in order to erase the information that is too old to be considered. However, intuitively, this should be equal or longer than CTW, so it is possible to have the history for the protocols.

#### 4.2.1. Copies of messages (Overhead)

The metric that indicates the copies of messages ($nc$) is updated each time the node creates new copies of a message to pass to another node. The system stores one $nc$ value for each of the protocol $DTN_i$ in $S$, considering the past history of the protocols. The metric for each protocol is computed as:

$$Overhead = \frac{nc}{overhead_{max}} \tag{3}$$

It is important to notice that the *Overhead*, as well as the other metrics to be considered in the objective function, should be normalized to transform the obtained value into a common frame of reference with other metrics.

Therefore, in order to obtain a value between 0 and 1, we use simulation to estimate a maximum value of the overhead that we called $overhead_{max}$. In real life, this maximum value can be piggybacked on the messages to provide a lazy update of the maximum starting value.

*4.2.2. Delivery rate*

The second metric for our case study is the delivery rate, that can be computed according to the following equation:

$$\text{DR} = \frac{d}{c} \tag{4}$$

where $d$ is the quantity of delivered messages when the protocol $DTN_i$ was used in forwarding, and $c$ is the number of messages that have used the protocol $DTN_i$ in forwarding. This metric has to be computed globally to be precise, which is unfeasible in a DTN scenario. For this case, we choose a local estimation in the nodes.

We obtain the value of $c$ as the number of messages the node has received from others and the ones that have produced themselves using the protocol $DTN_i$ for forwarding. If the node is able to send all those messages to their destinations, it would have a delivery rate of 1. If none of them arrive to the destination, then it would have a delivery rate of 0.

The value of $d$ is estimated with a notification from the neighbors that have received the messages, so the nodes can update it. The direct solution would be to create a message notification and send it as a broadcast in the network, or use a epidemic protocol. However, this would generare energy costs that we want to avoid. The alternative we have selected is to piggyback these notifications in message passing.

When a message arrives to the destination, the identifier of the message together with its routing information is distributed using piggybacking. The additional information is in the worst case $m * h$, where $m$ is the number of messages piggybacked and $h$ is the maximum number of hops of the messages.

It is possible to reduce this overhead using alternative support data structures such as Bloom filters, which are a probabilistic data structure that contains false positives, which could cause an overestimation of the delivery rate.

It is important to notice that the value of delivery rate is changing overtime as the node is aware of new messages arriving that have used the protocols selected for use, and the failure or success of the decision is evident.

### *4.3. Protocol selection*

The set of DTN protocols available for our case study are $S = \{$S&W, S&F, Epidemic, MaxProp, Prophet$\}$, which are the main state of the art protocols.

For this study we will consider low overheads and high delivery rates as the main requirements for the life of a DTN in a disaster scenario. These metrics are normalized as values between 0 and 1. With this in mind, we build the following objective function that is computed for the protocols in $S$:

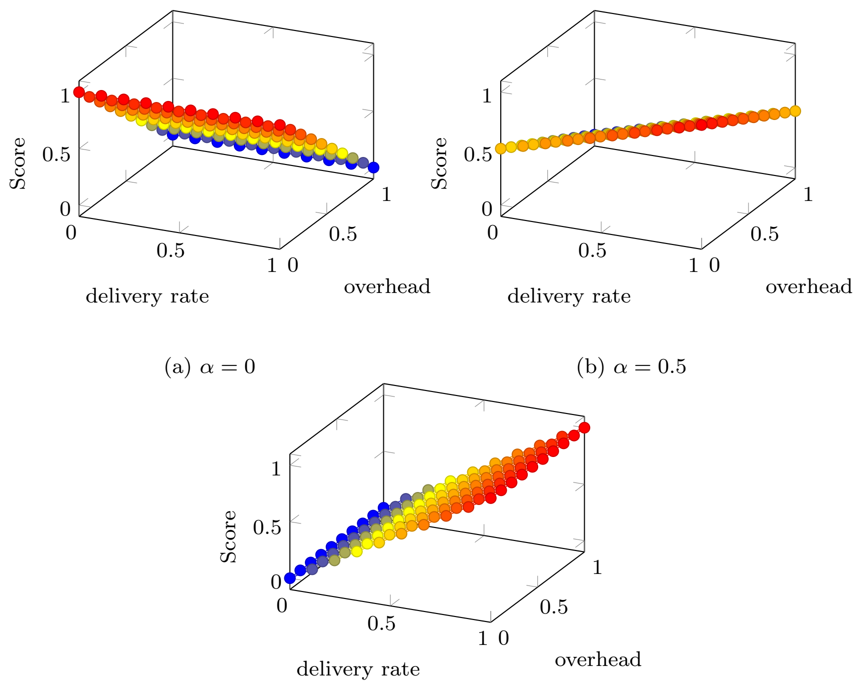
$$f(context) = \alpha * DR(context) + (1 - \alpha) * (1 - Overhead(context)) \qquad (5)$$

The objective function is evaluated for each protocol $DTN_i \in S$, considering the current context. The function DR(context) returns the interpolated value of the delivery rate for the corresponding context (density or average contact time) and the function Overhead(context) returns the predicted value of the overhead for the corresponding context. Additionally, an $\alpha \in [0, 1]$ value is included to change the importance that is given to each of the two metrics. Figure 2 shows how Equation 5 changes for three different values of $\alpha$. For $\alpha = 0$, the resulting value is high for protocols with low overhead without the influence of the delivery rate while the opposite happens for $\alpha = 1$. In the case of $\alpha = 0.5$, the function is influenced by both metrics equally.

It is important to notice that Equation (5) is a local value that each node computes locally, considering what the information is that it currently has about the behaviour of each protocol. The real values of delivery rate and overhead are different to the ones a node may perceive at a particular time. However, the goal of this equation is to balance the tradeoff between the two metrics the node has available and let the node decide at each round which is the best routing protocol according to its partial view of what is happening in the scenario.

Being able to select any of the protocols in $S$ implies storing the internal state of all the DTN protocols available. CSAR-DS stores arrays with the probability of encountering other nodes for MaxProp, Prophet, or Spray and Focus. This information is maintained, and the information is updated even if they are not the selected protocol at a particular time.

Latency may also be an important metric for a disaster scenario, when an authority requires to deliver a security message as soon as possible. In this




(a) $\alpha = 0$

(b) $\alpha = 0.5$

(c) $\alpha = 1$

Figure 2: Behavior of equation (5) for $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$ respectively.

case this metric should be included in the objective function so the protocol is able to take the latency information into account when selecting a protocol.

## 5. Simulation

### *5.1. Mobility models for disaster scenarios*

We used two mobility models in the experiments: the Post-Disaster Mobility Model (PDM) [19] and the Valparaíso Mobility Model (VMM) [20].

PDM is a mobility model specially designed for disaster scenarios that includes a pattern for the transportation network, population, and relief vehicle movement. In particular, Uddin et al. define four movements for each role: (1) a recurrent motion between centers to simulate transportation, (2) a localized random motion for rescue workers, (3) a recurrent path motion though multiple neighborhoods for police patrols, and (4) a motion switching from center to and back to a random location for ambulances. VMM is a mobility model based on PDM specially adapted for a coastline city located in a seismic zone, which is exposed to tsunamis. This model is inspired by the reality of the city of Valparaíso; however, this model fits many other coastline cities located in the Pacific Ring of Fire fault. VMM includes real evacuation routes, security points, and the mobility patterns published by the Chilean National Office for Emergency (ONEMI). As an extension of PDM, VMM includes the same mobility patterns of PDM.

PDM and VMM have a similar underlying graph and behavior. They represent an evacuation towards safe places. However, they differ in complexity. In particular, from the analysis performed in [20], the models are similar in the clustering coefficient, considering that VMM form a larger number of clusters with less people at each cluster, which implies a smaller average number of neighboring nodes and a smaller maximum number of neighboring nodes. The variance of node density is also smaller in VMM compared to PDM since the latter has a large number of nodes in some evacuation centers. In general, VMM is more challenging given the higher degree of partitioning of the network which creates a complex scenario for mobile ad-hoc communication protocols.

In comparison with other models, less dynamic models will have less to worry about the selection of the size of time windows to obtain good results while highly dynamic models will only have good performance if the system is able to gather enough information about the current context and metrics. This depends on the time the system requires to obtain metrics, which can be computed at each encounter, or after a round trip time from source to destination of a message.

### *5.2. Experiment configuration*

All our simulations were executed in a Xeon E5-2650 2.6GHz with 32 cores and 124GB of RAM. We consider that mobile devices communicate using Bluetooth 4.0 technology reaching a communication range of 100 m and a 1Mbps bandwidth. In our scenarios, the communication model assumes that any node can select any other participant node to interact with. We assume a disaster scenario where people may want to communicate with their relatives, authorities, friends, or viceversa.

Table 1 shows the default experiment configuration for our simulation environment. Global parameter values such as number of nodes, number of pedestrians, and message generation intervals were defined according to state of the art configurations. The values for parameters $\Delta t$, the size of the context, and metric time windows were obtained experimentally. Their values are presented in Table 1. Value selection was based on the trade-off observed using PDM and VMM. The parameters study is presented in the following section. All the results show the performance of the entire network, not only the results of a particular node.

| Configuration | PDM | VMM |
|---|---|---|
| Simulation Time | 80000s | 80000s |
| Number of nodes | 330 | 270 |
| Pedestrians | 200 | 200 |
| $\Delta t$ | 5000s | 5000s |
| Context time window | 5000s | 5000s |
| Metric time window | 5000s | 5000s |
| Message generation interval | 120s | 120s |

Table 1: Default parameters configuration for both mobility scenarios: PDM and VMM.

### *5.3. Parameters Selection*

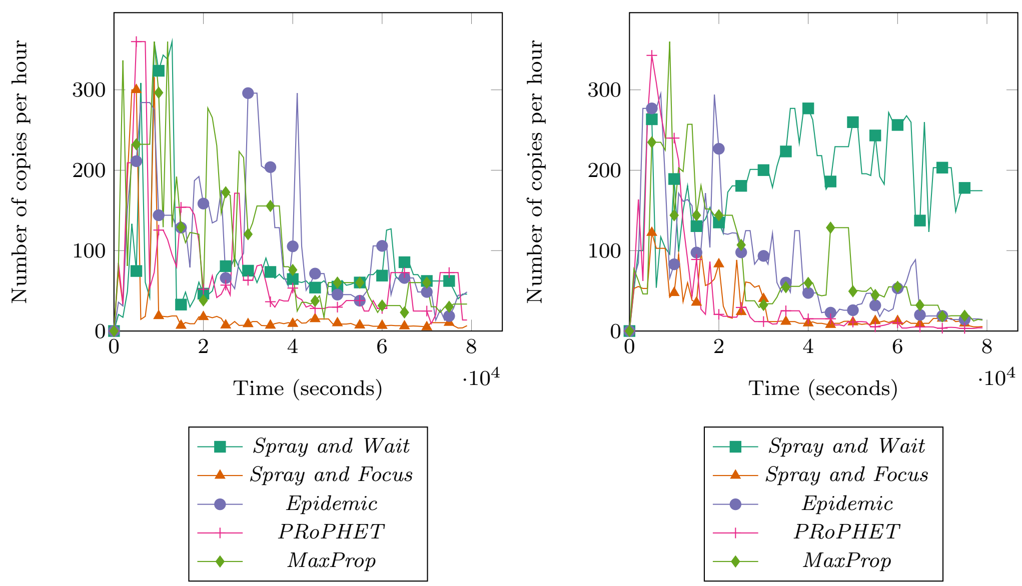
The proposed system called CSAR defines three parameters to set up before execution: the time to revise the protocol selection ($\Delta t$), the time a metric is valid, and the time that defines the current context information. Moreover, in the definition of the CSAR-DS, the system adapted to disaster scenarios, we have included a value that balances between the two metrics created for our scenario, called $\alpha$, and a value for normalizing the overhead metric ($overhead_{max}$). All these parameters may be set up using the simulation of different situation scenarios (which is our case), or in the absence of a simulated environment. These values can be defined locally, or globally by a command center and piggybacked to the nodes in the exchange of messages.

In the following section, we present the results of the simulations performed to experimentally select the values for the parameters that fit our case study and give guidelines to compute the values without the simulation phase. We have follow this methodology using simulation since each city is generally aware of the common threats they face, and how people move in such cases. We take as example, earthquake and tsunami prone locations. Other disaster scenarios that causes unexpected mobility in the network participant may compute their values on the fly.

### 5.3.1. *Maximum overhead*

In the following experiments, we analyze the behavior the overhead value for different protocols. We consider the overhead is the total number of messages generated in the communication network during a time frame. This metric is studied to establish the experimental boundaries to later normalize the parameter that depends on it: the overhead metric that is normalized. For the next experiments, we consider $\alpha = 0.5$ in order to equally balance the relevance of both metrics. For each parameter, two scenarios were simulated: one for each mobility model.

Figures 3a and 3b present the experimental results in terms of the number of copies generated by each protocol for PDM and VMM. We seek to establish the maximum number of copies generated to later normalize the metric related to the overhead value. In both figures, we can observe that the peak of copies generated is closer to 400 for the two mobility scenarios. In consequence, the parameter is set as 400. If a protocol generates 401 or more copies, the maximum value for the parameter remains at 400.



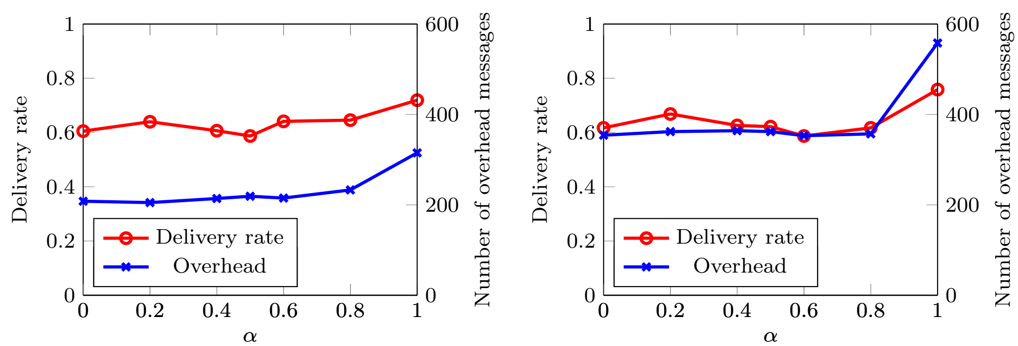

(a) PDM (b) VMM

Figure 3: Study of *maximum number of copies* observed for parameter's boundaries setup.

In a case without simulation, each node can store the maximum overhead value it has seen of each protocol to normalize its local metrics with that value. This value can be piggybacked in messages to converge with a common value in the community.

#### 5.3.2. $\alpha$ value

In this section we analyze the behavior of the hybrid protocol metrics in terms of the $\alpha$ value over the same two mobility scenarios. As we previously discussed, the $\alpha$ parameter balances the relevance of the hybrid protocol metrics, and we have included it in CSAR-DS in order to balance the relevance of the metrics related to *delivery rate* and *overhead*. Figure 4a presents the metrics behavior under the $\alpha$ parameter variation in the PDM scenario. Delivery rate and overhead are plotted on the same graph on the left and right $y$ axis respectively (as well as in the following results in the Parameter Selection section). We can observe a parallel trend between the $\alpha$ parameter with respect to the delivery rate and overhead. A higher delivery rate is associated with a greater number of message copies. In other words, more copies increase the probability to correctly deliver a message.




(a) PDM (b) VMM

Figure 4: Study of delivery rate and overhead for different values of $\alpha$ over two mobility scenarios.

We have repeated our previous simulation with the VMM scenario, and the results are presented in Figure 4b. Here we observed a similar behavior as in our previous scenario in terms of the relation between the delivery rate and the overhead of messages. In this case, however, there is a larger number of message overhead while maintaining the delivery rate. These results follow the selected DTN protocols in $S$ even though there are protocols that can maintain a very low overhead of messages as Spray and Focus. This protocol is not selected since it also has low delivery rates compared to other DTN protocols in $S$. Moreover, the number of messages of overhead increased in the VMM scenario, which can be explained by the characteristics of the mobility model: as described in [20]. VMM is more challenging for DTN protocols since it has a less average number of neighboring nodes, which produces more exchanges of messages to achieve good delivery rates (a behavior that repeats in all the experiments). From both scenarios, we conclude that the best balance between both metrics is reached for an $\alpha$ value of 0.2, even though any value below 0.6 would perform similarly.

In the absence of a simulated environment to tune the function designed to select the protocol in $S$, it would be important to select a default configuration and globally analyze the results using a command center or the vehicles that are part of the scenario. They can gather more information and define a value that fits the overall requirements.

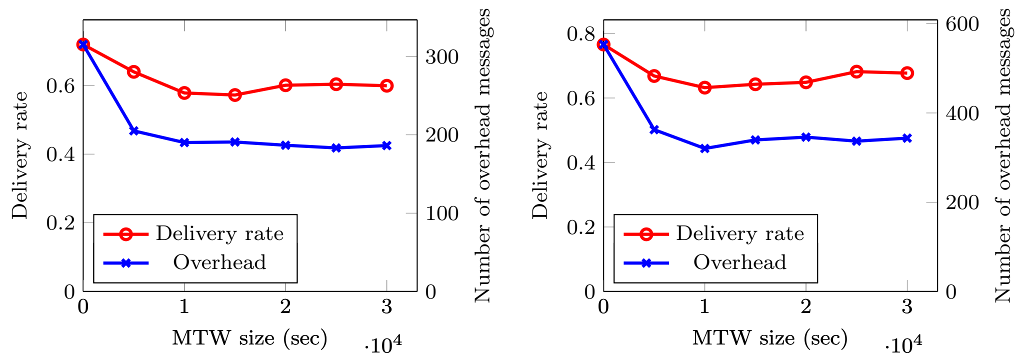
*5.3.3. Metric Time Window*

We also study the impact over the performance of the parameter *metric time window* (MTW). This parameter limits the amount of data used by the node to evaluate the metrics. A greater value means a more extensive range of data to evaluate the metric behavior. However, the amount of data considered may compromise the quality of the metric estimations by considering outdated data, then noise is introduced to the estimation. The results of this study are presented in Figure 5a for the PDM and in Figure 5b for the VMM scenario. In both scenarios, we study the influence of this time window over the metrics *delivery rate* and *overhead*.

For both scenarios, we observe that the delivery rate is maximized for smaller time windows at the price of a higher overhead. This occurs given that there is no past history to take into account the requirements of the selection function. For the scenarios in the case study, *overhead* and *delivery rate* are stabilized after a window size of 5000 seconds, meaning that larger time windows do not improve the decision made based on the information the nodes have gathered. We have selected this value as MTW size in the following experiments. On the other hand, for the VMM scenario, the same behavior is observed after a size of 15000 seconds. In this case, the parameter value required a trade-off analysis based on the objetive function values. Here we observed a greater trade-off when the MTW size is 25000 seconds.

The results from both scenarios differ in the message overheads, maintaining the same shape of the curves since they perform the same process of passing from a random map-based movement to a evacuation to safe zones that produce clusters of nodes [20].

In the absence of a simulation environment to decide on this value, we suggest to take into account the amount of information each node is able to gather about all protocols before erasing old information that may be useful for computing the metric values that build the objective function for each DTN protocol in $S$. A low amount of information will maintain the system doing random protocol selections without considering the requirements of the scenario, and the proposed system would not converge to a stable performance.

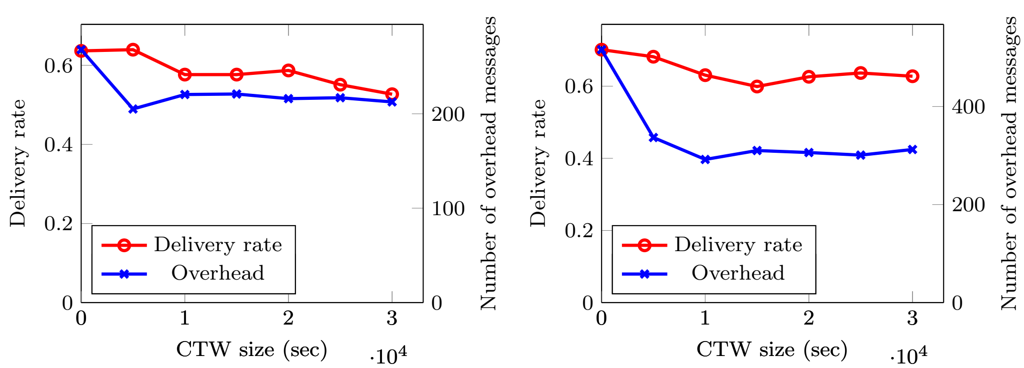



(a) PDM (b) VMM

Figure 5: Study of *delivery rate* and *overhead* for different values of *metric time window*.

### 5.3.4. Context Time Window

The *context time window* (CTW) limits the amount of data used to estimate the node's context. In our case study, this parameter influences the estimation of the density of neighboring nodes and the average contact time. For this parameter, a good size may better reflect the actual context while a size too large or too small may introduce noise to its estimation in dynamic environments.




(a) PDM (b) VMM

Figure 6: Study of *delivery rate* and *overhead* for different values of *context time window*.

In Figure 6a we present the metric behavior for different *context time window* values over the PDM scenario. Greater time window values may prevent the system from detecting sudden movement changes. For this reason we can notice that the delivery rate decreases while overhead remains stable. To set up the parameter value, we consider the window size that delivers a good trade-off between overhead and delivery rate. Following this criteria, we set the *context time window value* to 5000 seconds.

A similar approach is followed to set the parameter for the VMM scenario. Figure 6b presents the results for this scenario. In VMM the best trade-off is observed in second 10000. In consequence, the parameter *context time window value* is set to 10000 seconds.

In the absence of a simulated environment to tune this parameter, it is a good approach to analyze the dynamism of the network, observing the density and average contact value, as well as their variances in time. The larger the variances, the smaller the size of the CTW to consider.

### *5.3.5. $\Delta t$ parameter*

The $\Delta t$ parameter represents the minimum time a protocol must be used to evaluate its change. This window time is critical since it limits protocol fluctuations. Its value may significantly affect the proposal performance. A greater value may introduce more stability to the process; however, our proposal may be unable to detect changes in the scenario and performance should be closer to the traditional use of a single protocol. If we select a small time window, the protocol may change frequently, making it difficult to reach an improvement over the performance; however, if enough data is collected, the protocol may adapt better to highly dynamic scenarios.

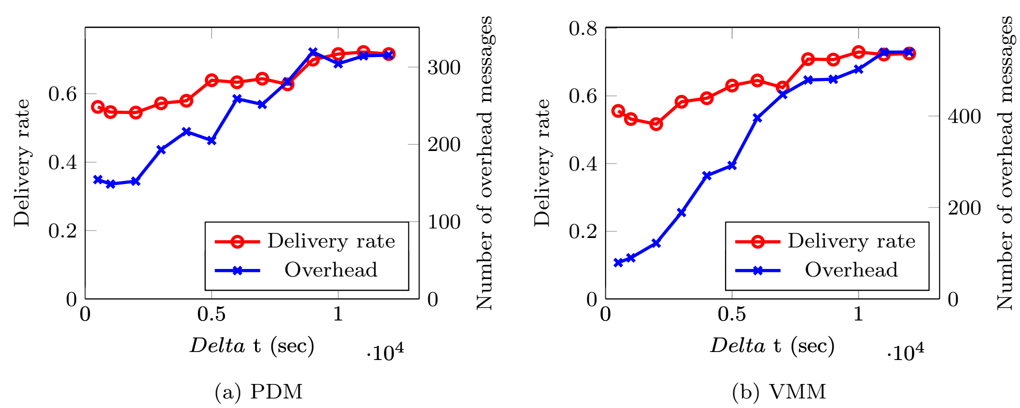



Figure 7: Study of *delivery rate* and *overhead* for different values of $\Delta t$.

In this section we study the impact of varying $\Delta t$ parameter in order to experimentally set parameter values. The behavior curve is used to analyze in detail the trade-off over the two metrics. Figures 7a and 7b present the results obtained in our testing scenarios. From the figures we can infer that a window of 5000 seconds presents a good trade-off between the two metrics. Our further analysis shows that we can reach a better trade-off when we set $\Delta t$ value to 7000. Here we increase the delivery rate by 17% at the price of 61% of overhead. After this threshold, overhead increases abruptly.

Results are similar for the VMM scenario; however, in this scenario, to obtain a gain in terms of delivery rate, we need to pay a higher price in terms of overhead. For this reason, we set the parameter $\Delta t$ value to 5000 in this scenario.

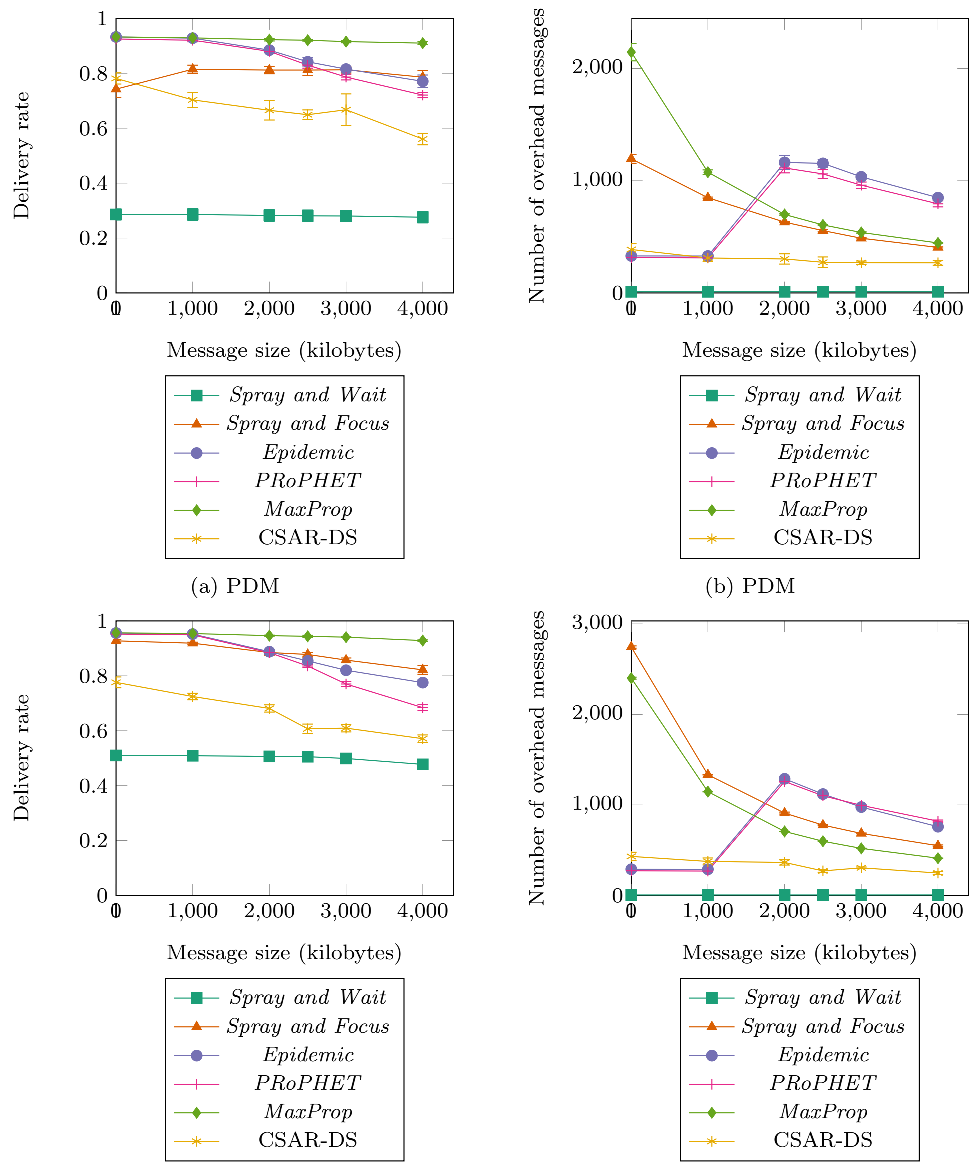
In the absence of a simulation environment to tune this parameter, we suggest to take a value lower than CTW and MTW in order to handle changes in dynamic environments.

### *5.4. Protocols Comparison*

In order to compare our proposal (*CSAR-DS*), with the state of the art protocols, we evaluate the behavior of some of the most well-known state of the art DTN protocols. In our tests, we use the same performance metrics as our previous studies: delivery rate and overhead, and two mobility scenarios: PDM and VMM. We study the behavior of the protocols in terms of four parameters: the message size, node density, the workload, and the buffer size. For all our tests, our proposal follows the configuration presented in Table 1. We compare our proposal with: *Spray and Wait*, *Spray and Focus*, *Epidemic*, *PRoPHET*, and *MaxProp* protocols.

#### *5.4.1. Size of messages*

Our first test consists of evaluating the impact of the message size, which may be different considering the application using the protocol We vary the message size from 1KB to 4MB. This range simulates the transmission in the network of simple text messages to small images. For the PDM scenario, Figure 8a shows the total delivery rate in the simulation if we exchange messages from 1KB to 4KB for our protocol and the five state of the art DTN protocols we are comparing it with. Figure 8b presents the total overhead of messages for the same scenario and protocols. Figures 8c and 8d present the behavior of the tested protocols over the VMM scenario.




(a) PDM (b) PDM

(c) VMM (d) VMM

Figure 8: Study of *delivery rate* and *overhead* for different message sizes.

From the figures, we can observe that *Maxprop* outperforms the other protocols in both scenarios in terms of the deliver rate. Its performance in terms of the delivery rate remains almost stable under a message size variation. Other protocols with good delivery rate are *Spray and Focus*, *Epidemic*, and *PRoPHET*: all of them present a better delivery rate than our proposal; however, they generate a higher overhead at the same time. This is an expected result since all these protocols prioritize the delivery rate. If we observe the overhead figures, we can notice that CSAR-DS provides a better trade-off than their competitors since it balances both metrics better. In terms of message size, CSAR-DS overhead remains stable while its delivery rate is slightly compromised when using bigger messages. These results are repeated in PDM and VMM scenarios.

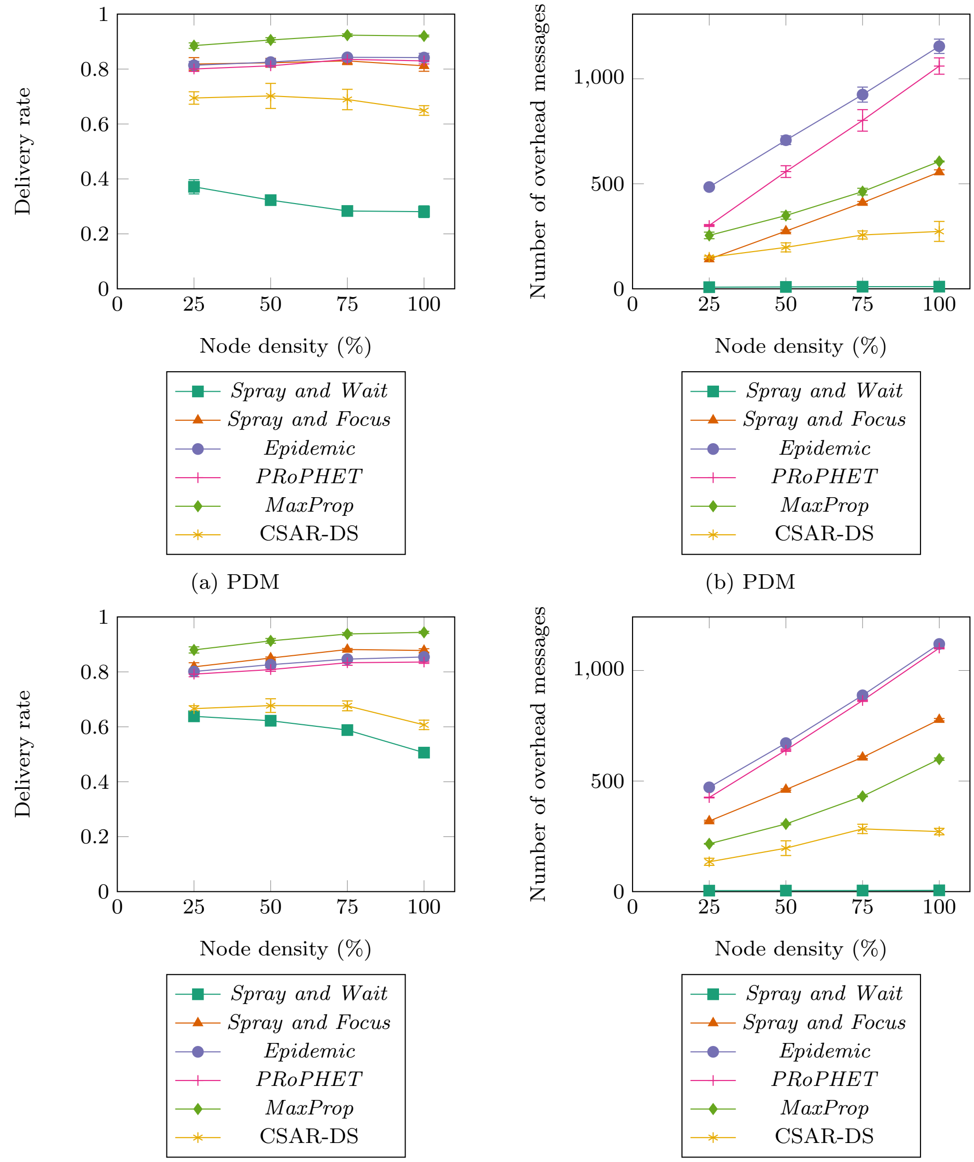
The results follow the design decision in CSAR-DS that is considering both metrics in the objective function: the system is not selecting the protocol based only in the delivery rate but also in the overhead of messages to save energy in a disaster scenario. This trade-off is balanced with the $\alpha$ value that was analyzed in previous experiments. The slight decrease in the delivery rate responds to the less amount of messages that can be exchanged when the messages are larger in some protocols, which affects our proposal since some nodes are selecting those protocols in some parts of the simulation. The same happens with the overhead of messages, which decreases when the size of the messages is larger since there is a smaller number of messages to exchange in buffer, and they can be transmitted in shorter contact times between neighbor nodes.

*5.4.2. Density*

We also study the effects of node density over the protocols' performance. We test 4 density scenarios with 25%, 50%, 75%, and 100% of nodes of the original scenario participating in the simulation. In other words, in the 100% density there are 330 nodes in the PDM scenario and 270 nodes in the VMM scenario while in the 50% scenario there are 165 nodes in the PDM scenario and 135 nodes in the VMM scenario. Results show the total delivery rate and overhead of messages for the total simulation in Figures 9a and 9b for the PDM scenario, and in Figures 9c and 9d for the VMM scenario.

From figures the we can observe that CSAR-DS can provide an acceptable delivery rate and overhead compared to the other protocols. In terms of overhead, it can reduce overhead considerably when compared to protocols such as Epidemic or PRoPHET. This important overhead reduction enables the protocol to spend less energy than most of the state of the art protocols tested. The only exception is Spray and Wait, but this protocol provides lower delivery rates in all cases.

We conclude that density affects the protocols mostly in terms of overhead




(a) PDM (b) PDM

(c) VMM (d) VMM

Figure 9: Study of *delivery rate* and *overhead* for different nodes' densities.

of messages since more encounters provide the opportunity of a larger number of exchanges, which do not necessarily mean larger delivery rates. Our design is able to cope with this challenge by considering the overhead metric inside the objective function. As delivery rate is not affected for the analyzed mobility models, in this case, the protocol is able to get the sufficient feedback to make good decisions in terms of the trade-off discussed.

*5.4.3. Workload*

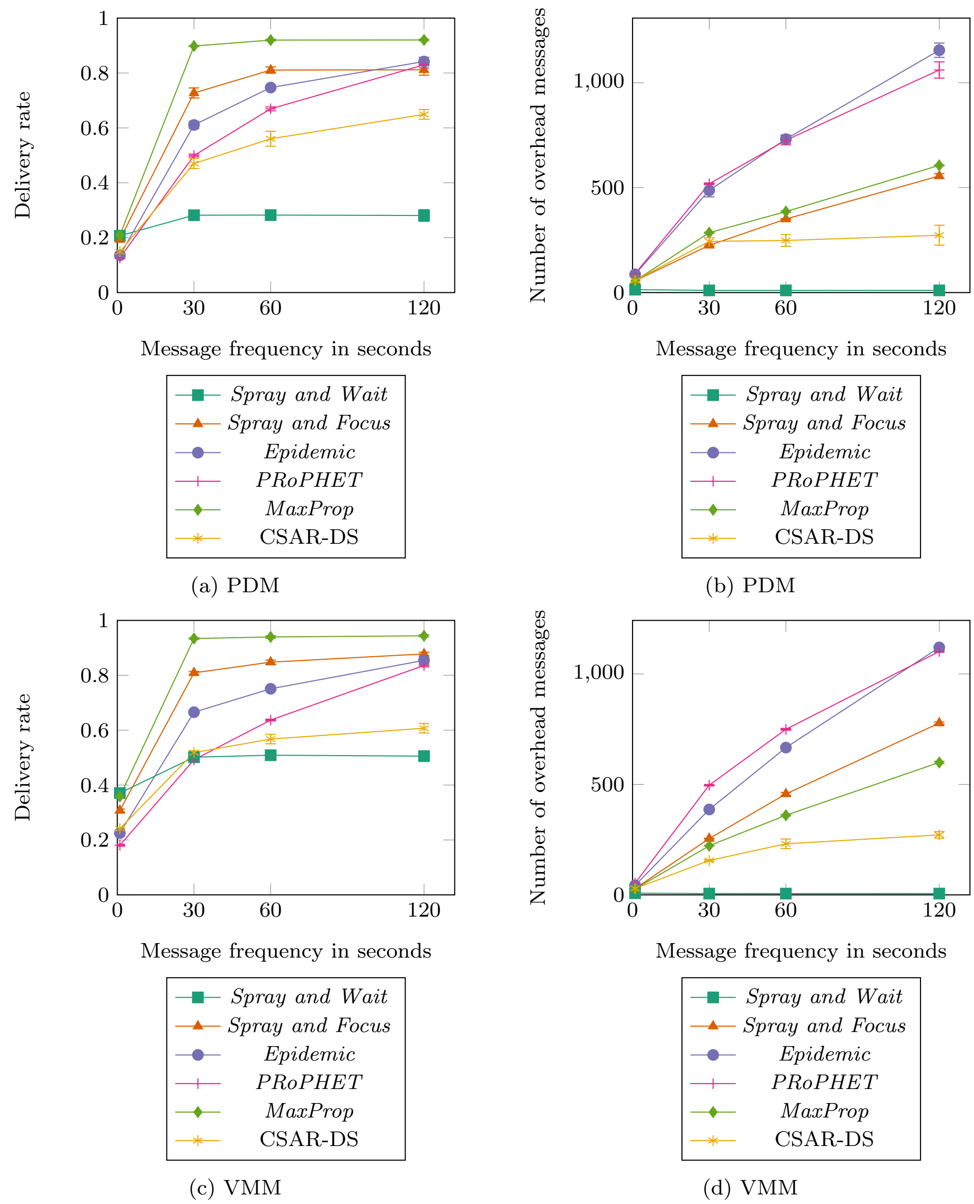
We study the workload impact over performance in our proposal and the state of the art DTN protocols. To this end, we vary the frequency of the message traffic injected in the simulation: one message per second, one message every 30 seconds, one message every 60 seconds, and one message every 120 seconds. Results for the delivery rate and overhead of messages for the entire simulation are presented in Figures 10a and 10b for the PDM scenario, and in Figures 10c and 10d for the VMM scenario.

We can observe that results are similar for both scenarios given the similarities in the underlying contact graph (see Section 5.1). The higher the traffic workload, the higher the race for the resources (i.e bandwidth, buffers, etc.). The race affects the delivery rate by making it decrease while the overhead also decreases due to a higher number of messages in the network that are sharing the resources and not arriving to the destination. Our proposal again seeks to provide a good trade-off between the delivery rate and overhead when compared to other protocols. In high traffic scenarios, CSAR-DS can considerably reduce energy consumption hence improving the network's survivability. However, CSAR-DS is not able to make any difference when the protocols are not delivering any messages and therefore, the feedback to build information about the protocols is not sufficient, and the system only achieves an average of the behavior of the performance of the protocols in $S$ as it can be seen in the configuration of one message per second.

*5.4.4. Buffer size*

We study the buffer size impact over the performance of the protocols in order to compare this behavior under different buffer size configurations. We measured the same metrics and compared the same protocols tested in previous sections. DTNs protocols are *store-carry-and-forward*, so buffer size plays an important role in the capacity of nodes to carry messages and disseminate them. In our test, buffer size ranged from 32 MB to 512 MB. Figures 11a and 11b present the results for the PDM scenario, and Figures 11c and 11d the results of the VMM scenario.

As can be seen, we have similar results in both scenarios where *Maxprop* reaches the better delivery rate, followed by Spray and Focus. Hence, the majority of protocols improve their delivery rate and reduce the overhead

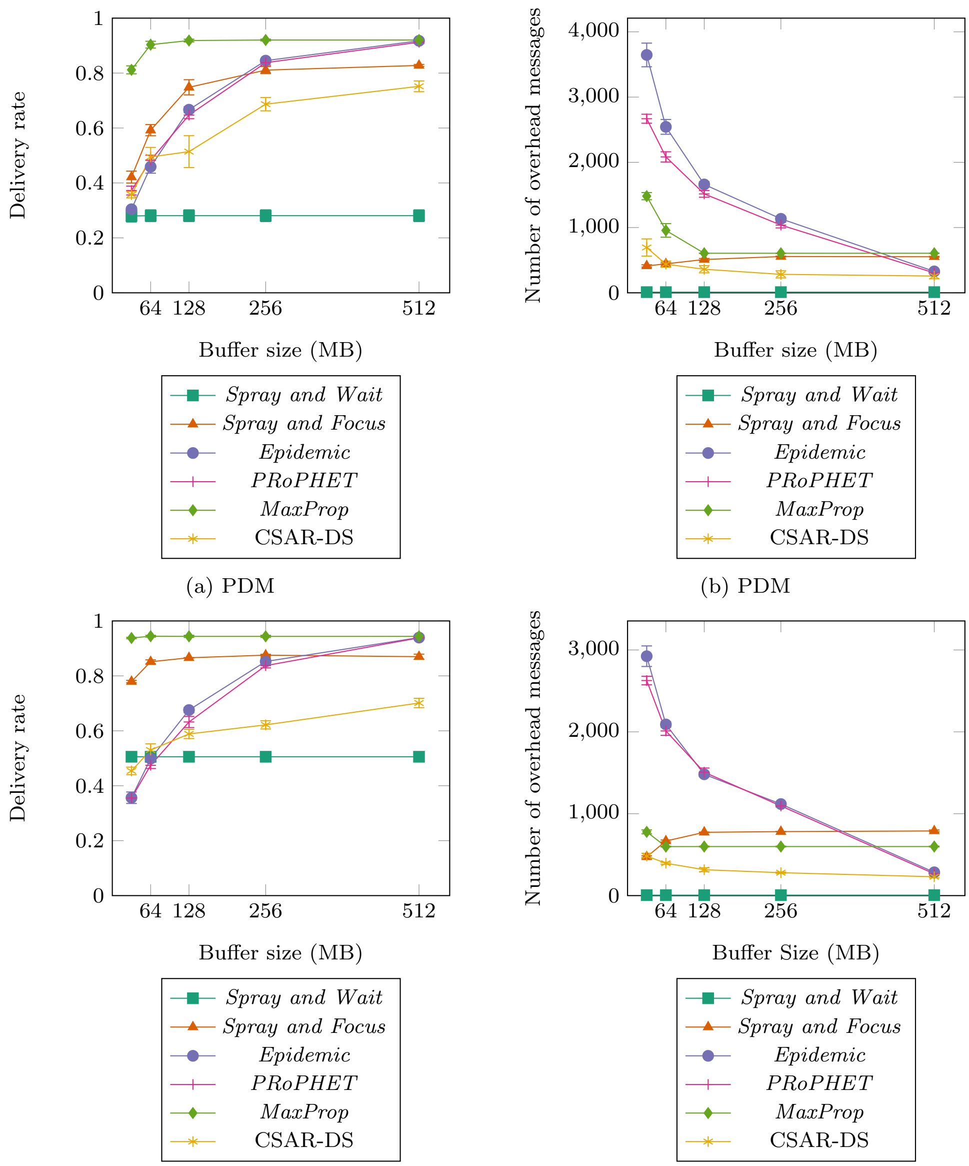



(a) PDM (b) PDM

(c) VMM (d) VMM

Figure 10: Study of *delivery rate* and *overhead* for different workloads.

when the *buffer size* is increased. This is a straightforward result since a bigger buffer increases the capacity of the nodes to carry more messages and allows their movement to deliver messages better. If we focus on the overhead of messages, we can notice that *Maxprop* presents a higher over-




(a) PDM

(b) PDM

(c) VMM

(d) VMM

Figure 11: Study of *delivery rate* and *overhead* for different workloads *buffer size*.

head compared to *CSAR-DS*, which means more energy is used. Once again *CSAR-DS* presents a good alternative in terms of the trade-off between good delivery rates and the overhead generated. Moreover, the results of CSAR-DS and Maxprop are very similar in terms of the shape of the curve, which indicates that CSAR-DS is choosing this protocol to perform the exchange of messages, while combining it with the one that spends less overhead of messages: Spray and Wait. The latter does not see any change in its performance when changing buffer size because its overhead is very low and its limited by the number of copies made for each message.

## 6. Conclusion

In this article we proposed the CSAR model which is a context-aware protocol that is able to self-adapt under changes of the nodes' characteristics and environment, based on the current and past performance of a set of DTN protocols. This model is able to create a family of hybrid protocols changing the metrics to be evaluated for each node when deciding the routing protocol to used. To evaluate and exemplify the CSAR model, we proposed CSAR-DS, whose goal is to fit the requirements of a disaster scenario, maximizing the delivery rate while minimizing energy costs.

The experimentation methodology used was based on a simulated environment, using two mobility models for disaster scenarios, studying first the parameters of the solution and then comparing the results with other DTNs, varying the size of the exchanged messages, the density of the nodes, the workload frequency, and the buffer size. The results show that CSAR-DS is able to balance the considered metrics, achieving a good tradeoff between the delivery rate and overhead of messages that impact the energy used by the nodes.

As future work, we would like to study how the idea of selecting different routing protocols can be extended so that the appropriate routing protocols are able to tune their configuration based on the evidence gathered by the system. Moreover, we would like to analyze how to deal with buffer congestion in these particular environments in order to prioritize relevant messages.

## 7. Acknowledgment

Authors want to thank the project FONDECYT Nº 11181028, CONICYT, Chile.

## References

[1] B. R., W. P., Annual disaster statistical review 2017, brussels: CRED (2018).

[2] Google Crisis Response, http://www.google.org/crisisresponse/.

[3] Red Cross Apps, http://www.redcross.org/prepare/mobile-apps.

[4] G. Association, The mobile economy 2018, tech report, https://www.gsma.com/mobileeconomy/ (2018).

[5] S. M. George, W. Zhou, H. Chenji, M. Won, Y.-O. Lee, A. Pazarloglou, R. Stoleru, P. Barooah, Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response, IEEE Communications Magazine 48 (3) (2010) 128–136.

[6] T. Hossmann, P. Carta, D. Schatzmann, F. Legendre, P. Gunningberg, C. Rohner, Twitter in disaster mode: Security architecture, in: Proc. the Special Workshop on Internet and Disasters, ACM, 2011, pp. 7:1–7:8.

[7] N. A. Mahiddin, N. I. Sarkar, B. Cusack, An internet access solution: Manet routing and a gateway selection approach for disaster scenarios, The Review of Socionetwork Strategies 11 (1) (2017) 47–64. doi:10.1007/s12626-017-0004-3.
URL https://doi.org/10.1007/s12626-017-0004-3

[8] F. Coriat, A. Fladenmuller, L. Arantes, O. Marin, Crowdsourcing-based architecture for post-disaster geolocation: A comparative performance evaluation, in: 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), 2016, pp. 1–9. doi:10.1109/NCA.2016.7778583.

[9] Z. Lu, G. Cao, T. L. Porta, Networking smartphones for disaster recovery, in: 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2016, pp. 1–9. doi:10.1109/PERCOM.2016.7456503.

[10] A. Martín-Campillo, J. Crowcroft, E. Yoneki, R. Martí, Evaluating opportunistic networks in disaster scenarios, Journal of Network and Computer Applications 36 (2) (2013) 870–880. doi:10.1016/j.jnca.2012.11.001.

[11] E. Rosas, N. Hidalgo, V. G. Costa, C. Bonacic, M. Marín, H. Senger, L. Arantes, C. A. C. Marcondes, O. Marin, Survey on simulation for

mobile ad-hoc communication for disaster scenarios, J. Comput. Sci. Technol. 31 (2) (2016) 326–349. doi:10.1007/s11390-016-1630-x.

[12] E. Rosas, F. Garay, N. Hidalgo, O. Marin, Mobility-aware DTN protocols for post-disaster scenarios, in: 2nd International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2015, Rennes, France, November 30 - December 2, 2015, IEEE, 2015, pp. 193–199. doi:10.1109/ICT-DM.2015.7402038.

[13] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad Hoc Networks, Tech. rep., Duke University (July 2000).

[14] J. Burgess, B. Gallagher, D. Jensen, B. Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks, in: Proc. the 25th IEEE Int. Conf. Computer Communications, 2006, pp. 1–11.

[15] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proc. the ACM SIGCOMM Workshop on Delay-tolerant Networking, ACM, 2005, pp. 252–259.

[16] T. Spyropoulos, K. Psounis, C. Raghavendra, Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Proc. the 5th Annu. IEEE Int. Conf. Pervasive Computing and Communications Workshops, 2007, pp. 79–85.

[17] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, Mobile Computing and Communication Review 7 (3) (2003) 19–20. doi:10.1145/961268.961272.
URL `http://doi.acm.org/10.1145/961268.961272`

[18] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools 2009, Rome, Italy, March 2-6, 2009, 2009, p. 55.

[19] M. Y. S. Uddin, D. Nicol, T. F. Abdelzaher, R. Kravets, A post-disaster mobility model for delay tolerant networking, in: Proceedings of the 2009 Winter Simulation Conference, WSC 2009, Hilton Austin Hotel, Austin, TX, USA, December 13-16, 2009, 2009, pp. 2785–2796. doi:10.1109/WSC.2009.5429249.

[20] F. Garay, E. Rosas, N. Hidalgo, When a tsunami strikes: A mobility model for coastline cities, in: B. Hellingrath, I. Gojmerac (Eds.), 4th International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2017, Münster, Germany, December 11-13, 2017, IEEE, 2017, pp. 1–7.

[21] Z. Zhao, T. Braun, D. Rosário, E. Cerqueira, Caor: Context-aware adaptive opportunistic routing in mobile ad-hoc networks, in: 2014 7th IFIP Wireless and Mobile Networking Conference (WMNC), 2014, pp. 1–8. doi:10.1109/WMNC.2014.6878878.

[22] M. Elias, A. Khattab, K. M. F. Elsayed, Corb: Context-aware opportunistic resource-based routing for stationary wireless sensor networks, in: 2015 International Conference on Computing, Networking and Communications (ICNC), 2015, pp. 166–170. doi:10.1109/ICCNC.2015.7069335.

[23] C. Boldrini, M. Conti, F. Delmastro, A. Passarella, Context- and social-aware middleware for opportunistic networks, J. Network and Computer Applications 33 (5) (2010) 525–541. doi:10.1016/j.jnca.2010.03.017.

[24] S. Yasmin, R. N. B. Rais, A. Qayyum, A multi-attribute routing protocol for opportunistic network environments, in: 2014 23rd International Conference on Computer Communication and Networks (ICCCN), 2014, pp. 1–6. doi:10.1109/ICCCN.2014.6911876.

[25] R. Cabacas, I. Ra, CMP: A context information-based routing scheme with energy-based message prioritization for delay tolerant networks, Int. J. Fuzzy Logic and Intelligent Systems 14 (4) (2014) 295–304. doi:10.5391/IJFIS.2014.14.4.295.

[26] M. Musolesi, C. Mascolo, CAR: context-aware adaptive routing for delay-tolerant mobile networks, IEEE Transactions on Mobile Computing 8 (2) (2009) 246–260. doi:10.1109/TMC.2008.107.

[27] E. C. R. de Oliveira, E. F. Silva, D. G. Passos, J. F. Naves, D. C. Muchaluat-Saade, I. M. Moraes, C. Albuquerque, Context-aware routing in delay and disruption tolerant networks, International Journal of Wireless Information Networks 23 (3) (2016) 231–245. doi:10.1007/s10776-016-0315-2.

[28] S. Gao, L. Zhang, H. Zhang, Energy-aware spray and wait routing in mobile opportunistic sensor networks, in: 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), 2010, pp. 1058–1063. doi:10.1109/ICBNMT.2010.5705251.

[29] F. D. Rango, S. Amelio, P. Fazio, Enhancements of epidemic routing in delay tolerant networks from an energy perspective, in: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013, pp. 731–735. doi:10.1109/IWCMC.2013.6583647.

[30] B. B. Bista, D. B. Rawat, Ea-prophet: An energy aware prophet-based routing protocol for delay tolerant networks, in: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), 2017, pp. 670–677. doi:10.1109/AINA.2017.75.

[31] M. Uddin, H. Ahmadi, T. Abdelzaher, R. Kravets, A low-energy, multi-copy inter-contact routing protocol for disaster response networks, in: IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009, pp. 1–9.

[32] Y. Zeng, J. Wu, N. Xiong, D. Li, Energy-efficient routing and rate allocation for delay tolerant networks, in: 2012 32nd International Conference on Distributed Computing Systems Workshops, 2012, pp. 260–266. doi:10.1109/ICDCSW.2012.69.

[33] S. L. F. Maia, E. R. Silva, P. R. Guardieiro, A bayesian game based optimization strategy proposal for routing in energy constrained dtns, in: 2015 XXXIII Brazilian Symposium on Computer Networks and Distributed Systems, 2015, pp. 1–9. doi:10.1109/SBRC.2015.12.

[34] A. Martín-Campillo, R. Martí, Energy-efficient forwarding mechanism for wireless opportunistic networks in emergency scenarios, Comput. Commun. 35 (14) (2012) 1715–1724.

[35] J. Lakkakorpi, M. Pitkänen, J. Ott, Adaptive routing in mobile opportunistic networks, in: Proceedings of the 13th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2010, Bodrum, Turkey, October 17-21, 2010, 2010, pp. 101–109. doi:10.1145/1868521.1868539.

[36] C. Raffelsberger, H. Hellwagner, Overview of hybrid manet-dtn networking and its potential for emergency response operations, Electronic Communications of the EASST 56.

[37] Y. Kawamoto, H. Nishiyama, N. Kato, Toward terminal-to-terminal communication networks: A hybrid manet and dtn approach, in: Proc. the Int. Workshop Computer Aided Modeling and Design of Communication Links and Networks, 2013, pp. 228–232.

[38] T. Miyakawa, A. Koyama, A hybrid type dtn routing method using delivery predictability and maximum number of replication, in: Int. Conf. on Advanced Information Networking and Applications Workshops, 2015, pp. 451–456.

[39] F. Garay, E. Rosas, N. Hidalgo, Reliable routing protocol for delay tolerant networks, in: 21st IEEE International Conference on Parallel and Distributed Systems, ICPADS 2015, Melbourne, Australia, December 14-17, 2015, IEEE Computer Society, 2015, pp. 320–327. doi:10.1109/ICPADS.2015.48.