

Effective communication for message prioritization in DTN for disaster scenarios

Erika Rosas (✉ erika.rosas@usm.cl)

Universidad Técnica Federico Santa María

Orlando Andrade

Universidad Técnica Federico Santa María

Nicolás Hidalgo

Universidad Diego Portales

Research Article

Keywords: Delay Tolerant Networks, Congestion, Disaster Scenarios, Resilience

Posted Date: May 23rd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1670527/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Effective communication for message prioritization in DTN for disaster scenarios

Erika Rosas^{1*}, Orlando Andrade¹ and Nicolás Hidalgo²

^{1*}Departamento de Ingeniería Informática, Universidad Técnica Federico Santa María, Santiago, Chile.

²Escuela de Informática y Telecomunicaciones, Facultad de Ingeniería, Universidad Diego Portales, Santiago, Chile.

*Corresponding author(s). E-mail(s): erika.rosas@usm.cl;
Contributing authors: orlando.andrade.14@sansano.usm.c;
nicolas.hidalgo@mail.udp.cl;

Abstract

When disaster strikes, effective communication is crucial for emergency responses and accessing a victim's location. Using smartphone-based Delay Tolerant Networks (DTNs) is the prevalent proposed approach to work around network disruptions. One of the challenges in these networks is high-traffic congestion, since buffer limitation produces message drop before they are delivered. In this article, a congestion control strategy using message prioritization is presented for DTNs to increase network resilience facing high message congestion. We measure congestion considering the free space in the *buffers* of the nodes involved in the routing or by counting the messages deleted from the *buffer* in a fixed period to improve the availability and immediacy of information for disaster scenarios. We evaluated the proposed strategies using The ONE simulator to test different mobility models and communication protocols. Results show that the strategy improved delivery rate, *buffer* usage, eliminated messages, *overhead*, and latency for the highest priority messages. The best results were obtained in terms of latency, which fits our disaster scenario since timely information is vital. The *trade-offs* are a slightly lower average delivery rate and a decrease in the lowest priority message delivery rate.

Keywords: Delay Tolerant Networks, Congestion, Disaster Scenarios, Resilience

1 Introduction

There has been continuous attention to the use of DTNs in the context of disaster scenarios, particularly after events that can severely damage communication infrastructure such as cell towers, removing the possibility of using cellular networks. Examples of these events are earthquakes, tsunamis, hurricanes, tornadoes, and floods. The networking research community predominantly promotes DTNs to support emergency response and recovery efforts, for example, in the process of victim's location, situation awareness, and emergency notifications [1–8].

In DTNs, each node in the affected area cooperates by forwarding messages to reach one or more destinations. They can work around network disruptions since they are powered up with the mobility of people carrying the devices. People's mobile devices are especially helpful because they have fair processing power, storage capacity, and battery life, enabling people to send relevant information through text messages, videos, photos, or audio.

DTNs use a bundle protocol architecture over the transport layer in the Internet Stack. Then, DTNs are independent of the communication technology used. The link layer could use WiFi, Bluetooth, some Low Power WAN technology, like are LoRa, or even satellite communication. The important issue that is resolved by DTNs is that it enables communication in challenged networks, without end-to-end path between source and destination of messages, tolerating high delays, avoiding issues when links have very different data rates, and allowing temporal disconnections.

The typical routing algorithms for DTNs are PROPHET [9], MaxProp [10], Epidemic [11], and Spray and Wait (SW) [12]. In Epidemic, each node sends to any other node it encounters the messages it does not have from its buffer. In SW, the protocol makes a pre-defined number of copies of the messages, saving one, and then waits until finding the destination. PROPHET and MaxProp take message routing decisions considering information about the node's mobility making assumptions and drawing conclusions from previous connections, learning from history and predicting optimal future encounters.

In disaster scenarios, people eager for information may spam the network without intention, jeopardizing availability, and decreasing performance. In particular, the authors in [13] probe that the communication drops after the event, but shortly afterward rises significantly during the phase of people asking for help, rescue efforts, and communication with friends and family. DTNs are not adapted to deal with high-traffic congestion, producing message loss and increasing latency of messages. In this kind of network, buffer limitation produces nodes drop old messages, affecting delivery rates and latency of messages.

In [14], the authors observed that the ad-hoc networks could not fully replace the infrastructure communication, so data prioritization in the network is essential. Message prioritization is a strategy to achieve communication in disaster scenarios where the number of messages that can arrive at their

destination may be affected by nodes' buffer space. Even though they are disseminated following opportunistic protocols, some messages may not be able to stay long in the buffers since the new messages have priority by default. However, the age of a message is not an indicator of its importance. Consequently, the default prioritization affects performance in the context of a disaster scenario.

To improve delivery rates and latency of messages under congested scenarios, we use prioritization of messages based on external classification, for example, the classification used in [15]. However, this work is compatible with other types of prioritization of messages, such as role-based or destination-based.

The priority of messages in our work changes the routing process when we detect congestion. In our protocol, the nodes exchange data to estimate the system's congestion locally, considering the messages drop rate and the buffer occupation. In particular, when we found congestion, we perform the following actions:

- We adapt the TTL value in the low priority messages in order to cause less overhead.
- We change the routing protocol to SW for low priority messages while using Epidemic or PRoPHET for high priority messages.

A challenge in this approach is to adjust the TTL value so that it is not too low to avoid the messages to reach its destination and also not too high that will congest the network. Also, properly detecting congestion and not confuse the symptoms (high message drop or high buffer utilization) with other cause. Our proposal provides a combination of routing algorithms that have different benefits to the network reducing the load of messages in the system, allowing to maintain the relevant information and discard the non-relevant one faster. This also allows a better data flow in all the nodes, so the transmitted information can arrive to its destination in a disaster scenario.

We evaluate our system with extensive simulation experiments using The One [16], considering different mobility models, levels of congestion, prioritization levels, among others.

The remainder of this article is organized as follows. In Section 2, we present the related work found in the literature. Section 3 describes the protocol designed for improving resilience in DTN under congestion. Experiments and discussion are presented in Section 4. Concluding remarks are in Section 5.

2 Literature review

Some works that tackle the communication problem for DTN in disaster scenarios focus on the routing protocols' performance. One approach is comparing opportunistic routing protocols considering mobility models specially tailored for disaster scenarios [17][18][19]. A second approach proposes new protocols with characteristics designed to improve performance in disaster scenarios [20].

A third approach proposes a set of opportunistic protocols in different situations [21] [22]. Given the characteristics of the disaster scenario, it is a common feature that the solutions minimize the energy spent in communication since it increases network survivability.

All the previously mentioned proposals treat messages equally; however, the delivery rate drops and latency increases under congestion. It has been reported that bandwidth availability of network channels is usually insufficient during a disaster period [13] [14] because of the large amount of information generated during a post-disaster and recovery phase. Lieser et al. [14] have highlighted the importance of message prioritization since they observed that a smartphone-based ad-hoc network could not fully replace the infrastructure-based communication but only provide limited communication.

2.1 Prioritizing messages in disaster scenarios

There are two questions to answer when defining prioritization in disaster scenarios: (1) what makes the difference between the messages? (2) how is the priority of a message going to influence the routing process of a message?

In the context of the first question, some approaches classify messages in a pre-defined number of levels. Bhattacharjee et al. [15] proposed classifying messages in post-disaster scenarios using natural language processing, particularly the Naive Bayesian Classifier. The authors define five categories and associate a message priority to each one of them. Besides natural language processing, the context of the message also classifies them. Luqman proposed prioritizing based on the sender context information in [23], for example, the vital signs of a person, the battery level, or the location. The authors in [13] prioritize using importance, urgency, and uniqueness. They define three types of messages: critical, disaster, and non-disaster, all classified using a lexicon of a real disaster scenario and keywords. Other characteristics of the message are used to differentiate them [24], where the authors prioritize message multicast over unicast, assuming that they may be more critical in disaster scenarios.

In the context of the second question, the authors modified the traditional DTN protocols to adapt them to message prioritization. Joe and Kim [25] modified SW to deal with a 3-level prioritization. The authors proposed to change the Wait phase: if a message is classified high priority, this message is forwarded only if the speed of the other node is higher than the local speed; if a message is classified *normal priority*, the protocol maintains its normal behavior; if the message is considered low priority it is removed from the buffer.

Bhattacharjee et al. [15] uses a modified PRoPHET routing protocol to route messages. This approach uses two priority weights to balance the delivery predictability used by the regular PRoPHET and a new value that depends on the message's prioritization. A similar approach is taken by McAtee et al. [26] proposing the extension of Epidemic and PRoPHET to implement message prioritization. Both protocols use different probabilities of dissemination for each type of message.

Liu et al. [27] study message prioritization in the Epidemic routing protocol. The authors compare three approaches for selecting messages from the buffer. The simpler one is random prioritization, which randomly picks a defined number of messages from the buffer. Another approach is called *tiered* that favors new and short messages. The messages are divided into a 3-level priority, giving a different number of advertisements to each one. Finally, the one that obtained better results was the approach called *oblivious*, which favors the latest messages in the system. An adaptive buffer division is proposed in [24] to prioritize a set of messages without producing starvation of the low-ranked messages.

Moreover, in post-disaster scenarios, Lieser et al. [14] compare static prioritization with adaptive prioritization. Static prioritization refers to buffer reordering based on the message category. In the adaptive prioritization, priority changes are based on the most prominent message type. We argue that frequency is not an indicator of importance. We put dynamism in the level of prioritization made in the system: How much preference is given to one message over another will depend on the measures of congestion computed locally.

Our proposal approaches the second question about prioritization of messages, proposing new alternatives in routing for already classified messages. Unlike the state of the art, we propose a dynamic approach that changes with congestion observed.

2.2 DTN under congestion problem

TRIAGE [23] uses a queue occupancy-based method that defines a threshold in the buffer's length to determine when there is congestion. Moreover, the nodes transmit their congestion in the messages forwarded by marking a congestion bit in the outgoing header. TRIAGE is a centralized system, but given the network characteristics, congestion control decisions should be made autonomous by the nodes participating in the forwarding process.

MACRE (Message Admission Control based on Rate Estimation) [28] decides whether to admit a message according to the relationship between input rate and output rate of a node. Different from our work, this decision is only computed on the receiving side. Farooq and Bibi [29] propose a hybrid approach inspired in [28] that uses message admission control together with buffer space advertisement to control congestion in DTNs. In this approach, nodes advertise a message in the network to make every node aware of the buffer's state. In our approach, we piggyback information about congestion without introducing new messages in the network.

In the solution proposed by Thompson et al. [30], message replication is dynamically limited based on local information. The nodes define the congestion level exchanging information about message drops and message replications. Differently, Lakkakorpi et al. [31] consider the reception capacity of the next hop, proposing the use of buffer space advertisement to avoid congestion in DTNs. Nevertheless, our analysis shows that the protocols fill their buffers after a certain level of congestion, and the new messages are stored in

the buffers by dropping others. This approach has a better delivery rate and latency than replicating messages only when there is space in the next hop.

Token-based approaches [32] [33] have been used to bound traffic in DTNs to avoid congestion. For example, Wang et al. [33] divide the map of an urban map into a grid and establish fixed sink stations in specific grids, delimitating that a node is only able to transmit after obtaining one of the tokens associated with their cluster. However, in a disaster scenario, these restrictions may hinder and urgent message delivery rate and latency.

A survey in this context is presented by Silva et al. [34], which proposes a taxonomy of the congestion solutions for DTN. The taxonomy considers: how the congestion is detected, how the contacts are among nodes, which is the evaluation platform, among others. The authors declare that 44% of the surveyed proposals are evaluated using The One simulator [16]. Another classification proposed in this survey is *open versus closed loop control*. Closed-loop mechanisms use feedback from the network and reduce their traffic level. There is no feedback in an open-loop mechanism, but the nodes try to agree on the sending rate. The congestion avoidance approach can also be categorized as *proactive versus reactive*: proactive schemes try to prevent congestion from happening while reactive schemes wait for congestion to manifest, for example, through packet loss or buffer saturation. In a disaster scenario, congestion is the rule, not the exception, so we use an always-on approach that follows a closed-loop control.

3 Prioritizing messages in DTN under congestion

In a disaster scenario, where congestion is a significant problem, we define a system that modifies the standard routing protocol to provide Effective Communication Under Congestion (ECUC) for DTN in Disaster Scenarios (DS). Figure 1 shows the main parts of the system ECUC-DS: A module for message classification, a module to estimate congestion, and the routing module, where we implemented strategies to adapt the routing process under specific circumstances. In the following, we will detail the behavior of each module. When a new message arrives passes through the *Message Classification Module* to assign a priority to the message and a corresponding TTL. Then, the message is stored in the **Buffer**. The *Congestion Estimation Module* adapts the TTL value if the statistics obtained from the **Buffer** indicate so. In parallel, the messages from the *Buffer* are sent to the *Routing Module* to be sent to the neighbors.

3.1 Message classification

Figure 1 shows the defined *message classification module* that takes a newly received message and categorizes it in three or five prioritization levels. After the message is classified, the node stores it in its buffer. The first parameter of our system is P that represents the number of prioritization levels. In the

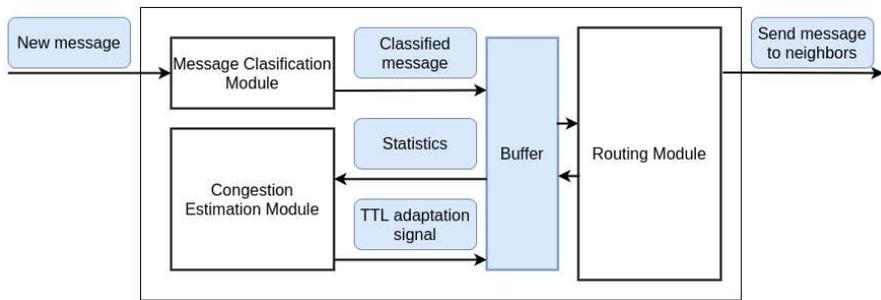


Fig. 1: Model of ECUC-DS

literature, three and five are the most common selection for this value [15][13][23].

In this work, we assumed that attributes for prioritization are pre-assigned. Moreover, a method as proposed by Bhatteercharjee et al. [15] is an excellent option to classify messages as they are received by a node. The authors use natural language processing with a Naive Bayesian Classifier, which has a good performance in terms of energy restrictions that is an essential characteristic of a disaster scenario.

It is important to notice that a message may be classified when created or each time it is forwarded. The first option saves processing power and fits a scenario with no malicious nodes since they may want to define all their messages as high priority messages.

3.2 Congestion estimation in DTN

Each node in ECUC-DS locally computes the level of congestion. We took two approaches; one uses the buffer occupation and the second approach uses the number of messages dropped from the buffer before the time to live (TTL) has expired.

In the first approach, each node maintains a percentage that indicates its buffer occupation and exchanges this value with the nodes it encounters in the DTN protocol. Every time a node encounters another, besides sending the messages to forward, it sends its buffer occupation, a value that is stored in the receiving node. We compute a small neighborhood's average buffer occupation, considering the last E nodes' values. We call this value *buff_level*, which is a moving average, that forgets old encounters.

Malfunctioning nodes malfunctioning that send corrupted information may affect the average congestion estimation and to prevent this issue, the values received by each node are evaluated to see if they are outliers compared with its neighbors' information. It is possible to assign thresholds and allow a certain error window to ignore corrupted information.

The second approach taken is complete local, each node counts the number of messages dropped in a pre-defined time window, TW , We call this value

dropped_messages. We took the approach of considering a local measure for congestion since DTN routing depends on the nodes' mobility, and one node may be receiving a large number of messages while others do not. The argument supporting this statement is that some routing protocols consider the peers' previous encounters before forwarding the messages.

We generalize both approaches in Figure 1. In the figure, we called *statistics* to the data from the buffer that feeds the congestion module. We defined two thresholds to identify the congestion states based on these three congestion levels: no congestion, medium congestion, and high congestion.

To indicate whether there is congestion or not, we check if the context parameter falls within a threshold to express different levels of congestion (No congestion - Medium congestion - High congestion):

No congestion (NC)	$\text{context_param} < \text{low_threshold}$
Medium congestion (MC)	$\text{context_param} < \text{medium_threshold}$
High congestion (HC)	$\text{context_param} > \text{high_threshold}$

3.3 Strategies to prioritize messages

To provide effective communication of messages in disaster scenarios under congestion, ECUC-DS takes the classification module's input and the congestion module: each message priority and the current congestion level. The routing module with this context modifies the routing process. We have evaluated different strategies and analyzed the impact on the performance of different well-known DTN protocols.

3.3.1 Hybrid routing protocols

We created hybrid protocols that use two different DTN protocols for different categories of message prioritization. Using the results obtained in [19], we consider that there are DTN protocols that prioritize delivery rates and latency, over energy reduction through less overhead (1) and other DTN protocols that do the exact opposite. With this in mind, we selected the protocol SW to route the messages with low priority classification since its results show very low overhead of messages. On the other hand, we selected the routing protocols called Epidemic and PRoPHET to route messages with high priority classification since these protocols show high delivery rates and low latency, characteristics we want to maintain for high priority messages.

$$\text{overhead} = \frac{(m_r - m_d)}{m_d} \quad (1)$$

To assign a protocol to a prioritization level, we defined a threshold called *hybrid_boundary*. This threshold defines which messages are routed with the low overhead DTN protocol (SW) and which messages are routed using the

low latency DTN protocol (Epidemic or PRoPHET). In the case $P = 3$ (three priority levels) the *hybrid_boundary* can take the values 1 or 2. In the first case, SW is used to route only the messages classified with the lowest message priority, and Epidemic or PRoPHET are used to route the other two levels. In the case $P = 5$, the *hybrid_boundary* can take the values 1, 2, 3 or 4, drawing the line between the use of each protocol.

SW routes the lowest priority messages and PRoPHET or Epidemic routes the other higher priority messages. We tested both combinations (SW and PRoPHET, and SW and Epidemic) to experimentally test which performed better.

3.3.2 Modification of Time To Live (TTL) value

The TTL is the time a message is allowed to travel in the network. It can be defined in terms of time or number of hops. Generally, every message is assigned the same TTL value. In ECUC-DS, we dynamically modify the TTL value of a message, depending on its priority level and the system's congestion.

Initially, we assign a priority level to every message in the buffer (*msg_prio*) which has a corresponding TTL value. Later, this TTL value is decreased considering the congestion level (*cng_lvl*). We consider that the lowest value of *msg_prio* the lowest priority, and the highest value is the highest priority.

To simplify assignments we define three values of TTL (*TTL_low*, *TTL_medium*, and *TTL_high*), the following algorithm shows how we assign these TTL values to the messages according to the congestion level:

Algorithm 1 TTL modification algorithm

Require: Message m with msg_{prio} and msg_{TTL}
Require: $cng_lvl \in \{NC, MC, HC\}$
if $msg_{prio} \leq threshold_low_TTL_{cng_lvl}$ **then**
 msg_{TTL} is adjusted to TTL_{low}
else if $msg_{prio} \leq threshold_medium_TTL_{cng_lvl}$ **then**
 msg_{TTL} is adjusted to TTL_{medium}
end if

The value of *cng_lvl* is in the set NC, MC, HC that indicates the congestion level. Therefore we define six thresholds, two for each level of congestion, that can take a value in the set $0, 1, \dots, P$. For example, if $P = 5$, we can define a $threshold_low_TTL_{MC}$ equal to 3, indicating that in case of medium congestion, the messages with priority equal to or lower than three will adjust their TTL values to TTL_{low} . Likewise, if $P = 3$, we can define a $threshold_medium_TTL_{HC}$ equals 3 that indicates all messages with priority equal to or lower than three will adjust their values to TTL_{medium} in case of high congestion. Notice the thresholds can take the value 0, that produces no message adjustment. Adjusting to a new TTL means that if the current

TTL of the message is greater than the new TTL, it will be decreased to that value. We only decrease the TTL and not increase its value.

The TTL is added to each message by each node at the moment of the message creation (taking into consideration the congestion level). Older buffer messages also adjust their TTL. This implies that the same message has different TTL's in different nodes if the congestion level of those nodes are different.

4 Simulation

The simulations in The One [16] were executed in a Xeon E5-2650 2.6 GHz, with 32 cores and 128GB Ram, with operating system GNU/Linux Ubuntu 14.04.5 LTS. We consider that mobile devices communicate using Bluetooth 4.0 technology, reaching a communication range of 100m and 1Mbps bandwidth.

Table 1 shows the default configuration used in our simulations. We configured the mobility scenario as the state of the art configurations. Messages generated between 2s and 5s produce congestion in the network. We will detail the parameter values defined for our system later in this section.

Table 1: Default parameter configuration

Configuration	Value
Simulation Time	80.000s
Communication range	100m
Bandwidth	1Mbps
Number of nodes	240
Pedestrians	160
Message generation interval	Between 2s and 5s
Buffer size	32MB
P	5
E	10
TW	1.000s
hybrid_boundary	2
Buffer low_threshold / high_threshold	50 / 75 %
Drop Number low_threshold / high_threshold	200 / 500
SW number of copies	10
SW binary mode	False
MaxProp Alpha	1.0
<i>threshold_low_TTL_{NC}/threshold_medium_TTL_{NC}</i>	0/0
<i>threshold_low_TTL_{MC}/threshold_medium_TTL_{MC}</i>	1/2
<i>threshold_low_TTL_{HC}/threshold_medium_TTL_{HC}</i>	2/3

We used three different disaster mobility models to test our simulations: the Post-Disaster Mobility Model (PDM) [35], the Valparaíso Mobility Model (VMM) [36], and the Cluster Mobility Model (CMM) [37].

PDM is a mobility model specially designed for disaster scenarios that include a transportation network, population, and relief vehicle movement

pattern. 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 through 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 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 mobility patterns published by the Chilean National Office for Emergency (ONEMI). As an extension of PDM, VMM includes its same mobility patterns.

Both models use the same movements, but they differ in complexity. In particular, from the analysis performed in [36], the models are similar in the clustering coefficient, considering that VMM forms a larger number of clusters with fewer 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 than PDM since the latter has many nodes in some evacuation centers. In general, VMM is more challenging, given the higher degree of the network's partitioning, which creates a complex scenario for mobile ad-hoc communication protocols.

CMM is a cluster-based model composed of two types of nodes: the nodes responsible for carrying data from one cluster to another (or carrier nodes) and the nodes that belong to a cluster (or internal nodes). Carrier nodes move among clusters collecting and transporting data from one cluster to another. On the other hand, internal nodes move following a random walk pattern over an area limited by a predefined cluster radius. The number of clusters is also a parameter of the system. CMM is a simple mobility model; however, it has been used to test DTN protocols due to its capacity to fit several real post-disaster situations [37]. In this work, we apply this model to the city of San Francisco. Places such as fire stations, hospitals, police stations, shelters and collection centers can be mapped as cluster centers, and the people located at each place match the nodes of the cluster. In order to recreate a disaster scenario we propose an earthquake scenario where some points such as hospitals, police stations, schools, and universities are represented by the cluster centers. By default, we used CMM with 4 clusters, and 240 nodes in the scenario, 160 representing pedestrians. Also, by default, we assume that any node can select any other participant node to send a message.

Following the state of the art methodology [20], the number of vehicles versus vehicles was chosen considering that most of the population will remain in the clusters. Each simulation was executed 5 times with different seeds. The results shown are the average of the 5 executions, with its corresponding error.

4.1 Analyzing congestion scenarios in DTNs

The goal of our first analysis is to analyze congestion using well-known DTN protocols. Previous work presented in [22] showed that using PDM and VMM, the delivery rates and overhead are strongly affected by congestion. To complement these results, we executed the traditional DTN protocols over CMM. In particular, we considered the routing protocols Epidemic (EP) [11], MaxProp (MP) [10], PRoPHET (PR) [9], and Spray and Wait (SW) [12]. These protocols were considered since it gives us a high variability in the delivery logic on the messages and as a base to project the behaviour to similar protocols since our solution considers not the distribution logic of the messages itself, but ways to reduce the congestion in the message generation process itself.

We analyzed a scenario with messages being generated every 120,60, 30, and 1 seconds (four different congestion levels). We measure delivery rates, latency, overhead, buffer occupation, and drop rates of the messages stored in the node's buffers.

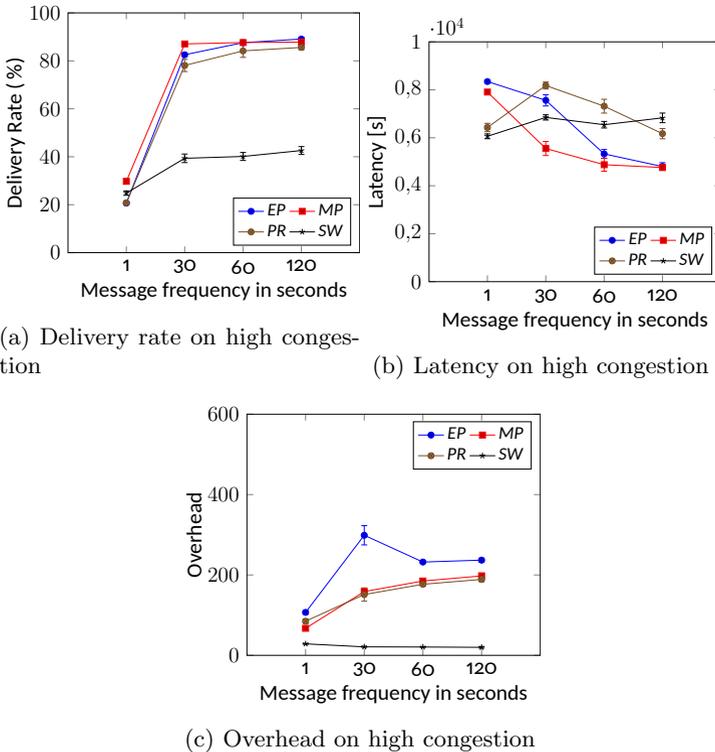


Fig. 2: Delivery rate, latency and overhead on high and medium congestion

Figure 2 compares delivery rate, latency, and overhead obtained in the DTN protocols without any modification of the proposal. We will call them *originals* in the following experiments. For DTN, delivery rates for Epidemic and PRoPHET are affected by congestion, decreasing the delivery rates from 90% to 20%. In these simulations that are considered higher connected networks like PDM and VMM, congestion has a higher impact dropping. SW tend to maintain their delivery rates despite the congestion in CMM, PDM, and VMM, since it maintains a fixed number of copies for each message. However, the delivery rates are constantly low. Maxprop has a good and stable delivery rate in CMM despite the congestion; however, in PDM and VMM, its delivery rates drop from 90% to 20% [22].

The overhead metric for all protocols under congestion tends to zero since congestion does not allow making copies of the messages. When the congestion decreases, the overhead increases and decreases again when the number of messages is small and does not cause too much overhead (Figure 2c).

Latency is a hard to analyze metric in terms of message frequency. The values are computed only with the delivered messages, a value that changes with the delivery rate. The higher the delivery rate, the average latency may increase since messages that take longer to deliver are also included in the metric. On the other hand, latency may tend to decrease if the messages are not dropped from buffers. Executing the protocols using CMM, the average latency values go from 3,000 seconds to 7,000 seconds (Figure 2b).

Figure 3 shows the analysis of what is occurring inside the buffers during a simulation with high congestion. Particularly, in Figure 3a, we observe that the buffers in all DTN protocols are almost full in the second half of the simulation, going closer to 100%. We complement this information with Figure 3b, which shows how some protocols drop many messages when the buffers are full. This is the case for Epidemic and, to a lesser extent, to PRoPHET. MaxProp reduces the number of exchanges of messages by considering the nodes' mobility information, but it is also affected by the congestion. SW is the exception since this protocol does not make more copies of the messages after

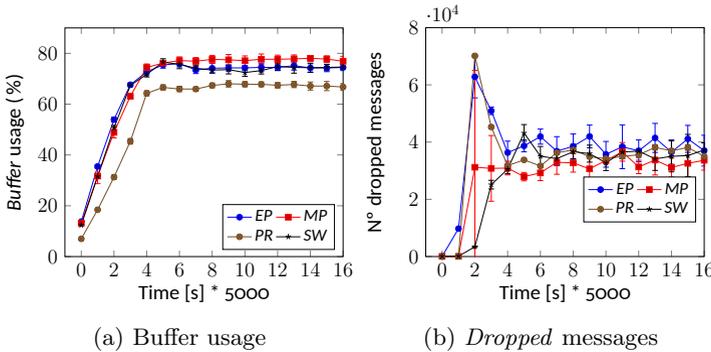
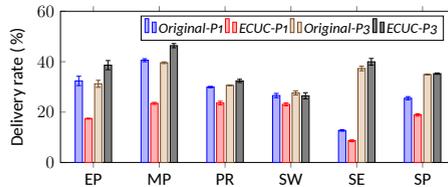


Fig. 3: Buffer analysis on high congestion and buffer size of 32M

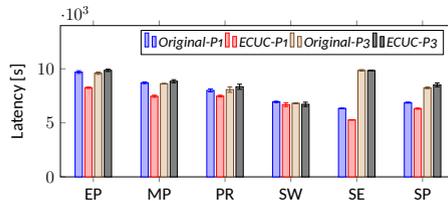
a certain threshold. Also it can be appreciated that PROPHET increases its message drop ratio drastically in $10 * 5000s$, since in that period of time, the buffer begins to be completely full, so it is more difficult for the buffer to make space for new messages, increasing the drop rate.

4.2 Evaluation of the system's communication

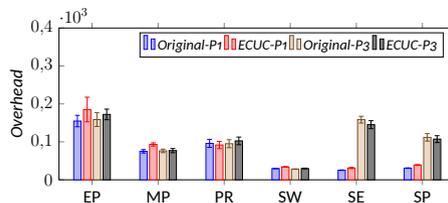
In these simulations, we consider message prioritization with three levels, from P1 to P3, given P3 as the highest priority. We compare the system with the original protocols when there is no use of the system ECUC that changes the TTL messages value when congestion is detected. We applied ECUC to the four protocols that we consider in the last section, adding two others: the hybrid proposals of SW with Epidemic (SW-EP), and SW with PROPHET (SW-PR). In Figures 4a, 4b, and 4c we show the metric results of messages with the lowest priority (P1) and highest priority (P3).



(a) Delivery rate comparison using dynamic TTL



(b) Latency comparison using dynamic TTL



(c) Overhead comparison using dynamic TTL

Fig. 4: Delivery rate, latency and overhead comparison using dynamic TTL

advance

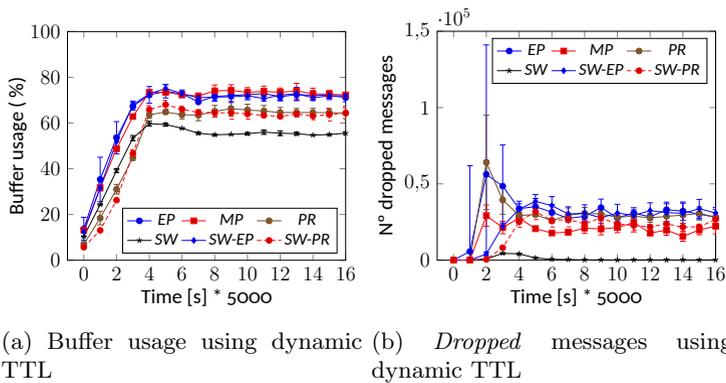


Fig. 5: Buffer usage and dropped messages using dynamic TTL

As expected, the values of the original protocols *Original-P1* and *Original-P3* are very close since these protocols do not display difference between the messages. What we would like to analyze is what happens with the different types of messages when we apply the TTL adaptation when congestion is detected.

For P3, our system increases delivery rate, varying the increment from protocol to protocol. The one with less impact is SW, since this protocol has its way to control congestion. Epidemic and MaxProp, on the other hand, benefit more from the ECUC modification. As they are mainly based in Epidemic and P_{Ro}PHET, the hybrid protocols behave similarly, and is SW-SE the one that performs the best. However, MaxProp is the one that has the best delivery rate. Latency is improved in all protocols for P3, which is a good result considering that we are increasing the delivery rate of the protocols. Latency is decreased in almost half in most of the protocols. Finally, overhead for P3 is worst for the hybrid protocols.

For the messages with the lowest priority, P1, the delivery rate decreases when applying ECUC, since we prioritize other messages. The impact of this result varies from protocol to protocol. In MaxProp, for example, the delivery rate for the lowest protocols falls almost to half its value. However, other protocols only slightly decrease the delivery rate, such as SW. This is the tradeoff of ECUC: the system decreases the delivery rate of the low priority messages, but increases the delivery rate of the high priority ones, avoiding messages that stay too long traveling through the DTN network.

Figures 5a and 5b shows how the buffer reacts to the system. Buffer usage drops below 80% for all protocols, with the lowest results SW, and with highest results Epidemic, which are the extremes in terms of the number of copies the protocol makes of a message. Comparing with Figure 3a we see a large improvement. The buffer usage is the average for all nodes in the system, and

some nodes are still dropping messages to face the high congestion, as shown in Figure 5b. However, the value is lower than in the original case for all DTN protocols.

4.3 Comparison with state of the art

We selected three *state of the art* techniques to compare our work. The first two are protocols presented in [26], where the message forwarding uses probabilities. The authors proposed three lists of messages, one for each message priority. Each list has different probabilities of exchanging messages. The two protocols associated with this work will be identified as *Epidemic-SOA* and *PRoPHET-SOA* since the first one uses Epidemic as the primary protocol and the second one uses PRoPHET. The third protocol used is based in [25] that uses SW as a primary protocol. In this protocol, each message's last copy is forwarded depending on the destination node's speed. This protocol is identified as *SW-SOA* in the results figures.

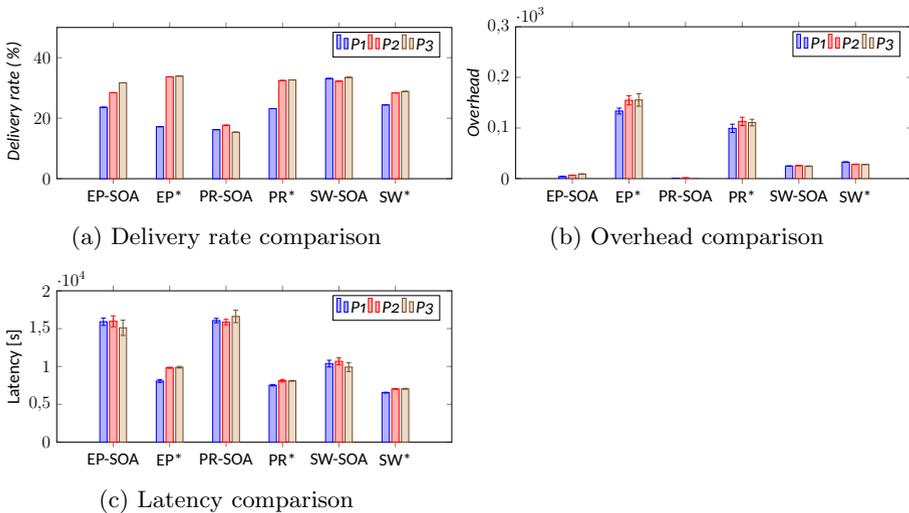


Fig. 6: State of the art comparison of delivery rate, latency and overhead

Figures 6a, 6b, and 6c shows the comparison between the selected state of the art protocols and the same primary protocols of the state of the art using the TTL adaptation. We named the protocols with an asterisk to represent that the protocols are not the original versions but the adapted versions. We can observe that the best protocols, in this case, were Epidemic modified (EP*) and PRoPHET modified (PR*). SW modified (SW*) did not perform as well as the state of the art protocol that used the same primary protocol.

However, the overhead shown in Figure 6b is more significant in the two protocols that achieved better results, obtaining many messages in the highest priority messages.

Moreover, the modified protocols with our techniques achieved better latency metrics, as shown in Figure 6c, where the protocols of the state of the art measure average latencies of almost two times higher than the proposed techniques. For emergency cases, the information should be available as early as possible to act quickly, so this is a valuable result in this scenario given that the delivery rate of the higher priority message was maintained or even increased.

4.4 Analyzing strategy behavior under different scenarios

We evaluated the effectiveness of the strategy with different mobility models. In particular, CMM and VMM, which we described in detail at the beginning of this section.

Figures 7a, 7b, and 7c show the results obtained using the VMM, Figures 7d, 7e and 7f show the results obtained using the PDM, and finally, Figures 7g, 7h, and 7i show the results obtained using the CMM.

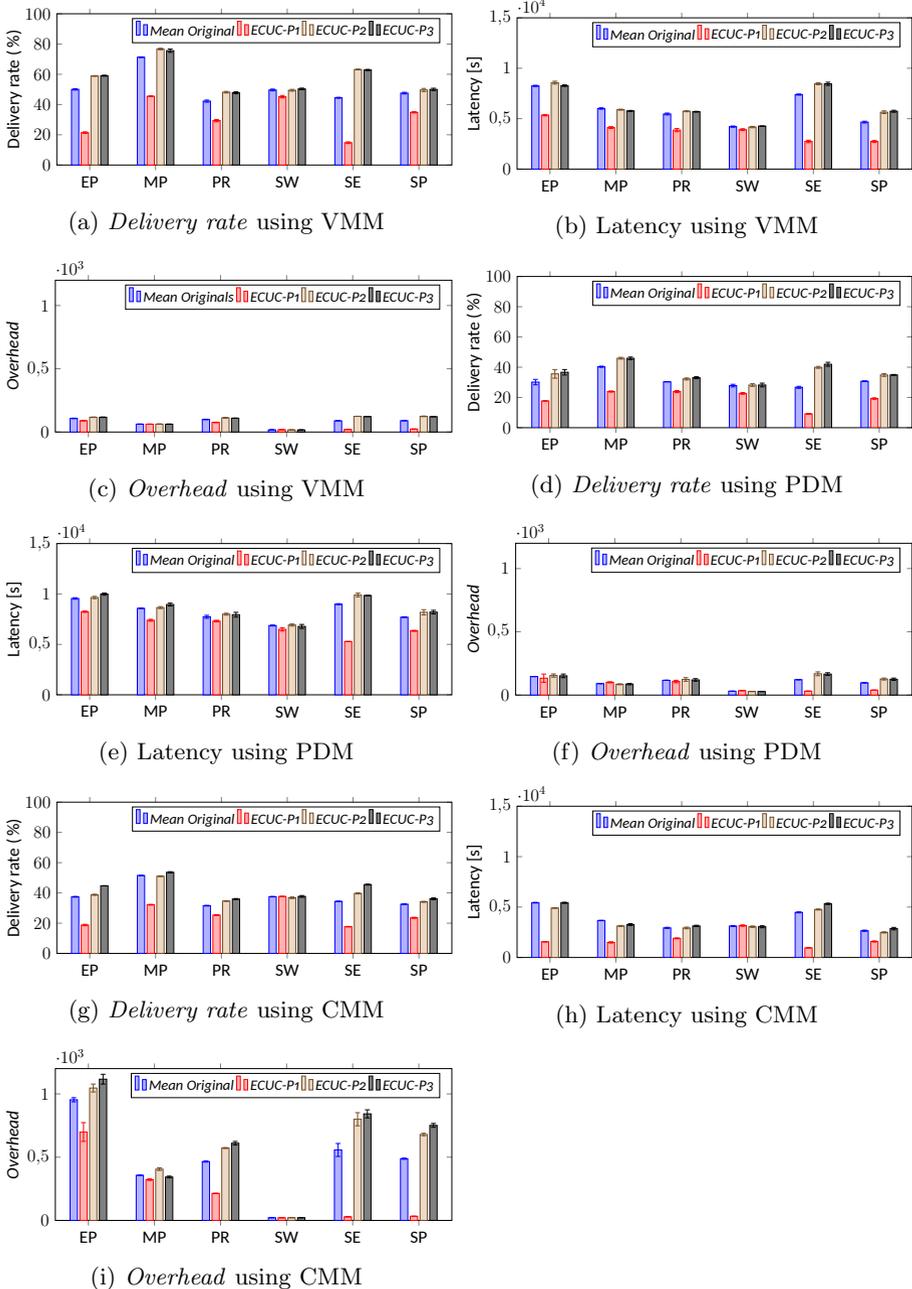
We observe from the figures that our system performs the same protocols in all mobility models, improving most DTN protocols delivery rate but decreasing it in one of them. SW, given its behavior, performs equally without being impacted by congestion and MaxProp, which considers the nodes' movement to forward a message.

Overhead, on the other hand, is much smaller in VMM and PDM for all protocols, given the network's different levels of partition. It has almost no impact on the application of the proposed techniques. Finally, the latency is improved in all scenarios and all protocols by a large amount since we decrease the TTL of the low priority messages, removing messages that stay long in the buffers, consuming storage resources, except for a few.

Also, for more insight on the prioritization results, it can be observed that for the medium level of prioritization it behaved as expected, with higher delivery rate than the lower priority message and also with less delivery rate than the higher priority message. For the latency we obtained a similar behaviour but in some cases the latency is similar to the highest priority message. This results is similar when analyzing the overhead. In the other hand, with both the hybrid protocols and the TTL adjustment the medium priority messages reached higher delivery rate with similar latency.

4.5 TTL modifications

Other simulations were performed in order to evaluate the parameter selections. For example, Figures 8a, 8b, and 8c compare differences in metric results for low and high TTL values in ECUC. In most cases, the delivery rate for low priority messages (P1) increases their value and high priority messages (P3) also increases the delivery rate but increasing latency instead. An exception is

**Fig. 7:** Metrics using different mobility models CMM, VMM, PDM

the SW protocol since it can manage congestion by fixing the number of copies of each message.

Latency increases with higher TTL messages since messages that perform more hops are delivered to the destination. Despite the messages that take longer to arrive are now considered, delivery rates are not improving with higher TTL. Finally, concerning the overhead, the higher the TTL, the higher the overhead for each type of priority of messages. An optimal selection of the TTL value for best performance is part of our future work.

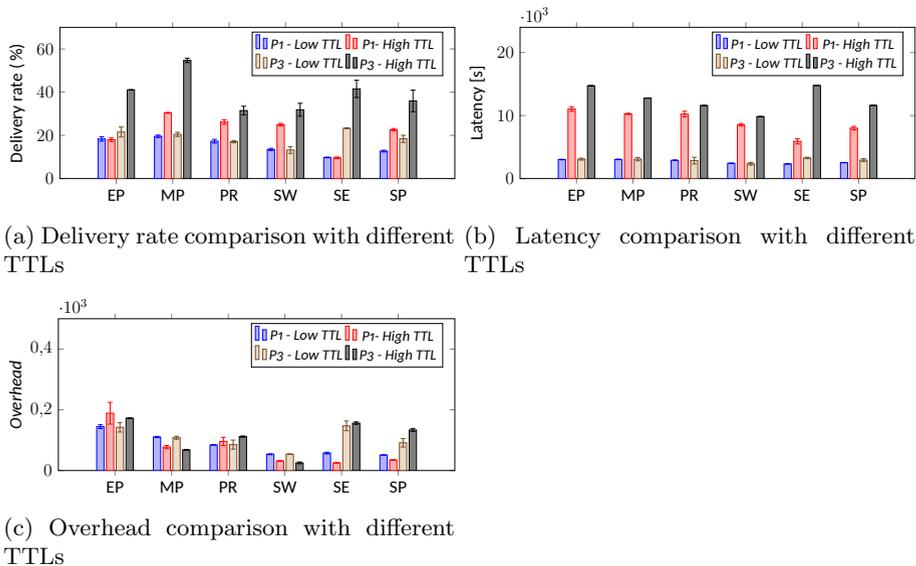


Fig. 8: Delivery rate, latency and overhead with different TTLs

5 Conclusion

In this work, we proposed ECUC-DS, a system built with information congestion control strategies for disaster scenarios that reduce network congestion. ECUC-DS prioritizes messages and adjusts TTL, considering the congestion level. We compared the system with state-of-the-art strategies and evaluated ECUC-DS using different mobility models.

From the simulation results, we conclude that the strategy fulfilled its role of alleviating network congestion, observing a noticeable improvement in the message latency, reducing times by practically 50% in some simulations, which is of significant importance in disaster scenarios since it is crucial to obtain information as quickly as possible. Regarding the delivery rate, in some protocols suffered a small decrease, while in others it increased. In all the simulations, the highest priority messages delivery rate increases, which is beneficial in disaster scenarios since those messages are considered vital information from which we want to be informed as soon as possible. Similarly,

ECUS-DS controlled the *overhead* of the messages using the hybrid protocol technique, allowing prioritizing the delivery of the highest priority messages.

Also, as seen in the results, prioritizing messages allows the system to distinguish different messages in order to select those that are more valuable in the emergency scenario and increase its delivery rate while lowering the latency to reach its destination. But on the other hand, sacrifices the metrics of less important messages to achieve this upgrades. On a real world application, this is of high importance since it is not very useful if the network is filled with messages that do not provide rich information to authorities or that congest the system so that the important messages can't be delivered, for example the ones that can saves lifes.

On the other hand, the hybrid protocols allow us to get the best of different routing algorithms such as high message replication and fast delivery for high priority messages (EP or PR) and low overhead and message replication for less important messages (SW). Implementing this system on a real world scenario could provide the previously explained results, making the most important messages more accessible to the authorities, yet, further work has to be done to check if these protocols are the best for the selected scenarios, since a lot of combinations could be performed to obtained an ever better hybrid protocol.

As future work, we would like to adjust the TTL setting according to each protocol's logic and particular scenario. Protocols such as *Spray & Wait* and *Spray & Focus* behave in a similar way in all scenarios. Similarly, by analyzing each protocol's behavior, hybrid protocols could be redesigned with different protocols and adjusting their optimal TTLs. Moreover, we would like to dynamically modify the hybrid protocols that forward the different message priorities depending on the congestion level since we have observed that no protocol always outperforms the others.

6 Declarations

6.1 Ethical Approval and Consent to participate

Not Applicable.

6.2 Human and Animal Ethics

Not Applicable.

6.3 Consent for publication

Not Applicable.

6.4 Availability of supporting data

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

6.5 Competing interests

The authors declare they have not competing interests as defined by Springer, or other interest that might be perceived to influence the results and/or discussion reported in this paper.

6.6 Funding

The work of the authors Erika Rosas and Orlando Andrade were funded by Agencia Nacional de Investigación y Desarrollo, project Fondecyt Iniciación N^o 11181028.

6.7 Authors' contributions

E.R. did the conceptualization of the project. E.R., O.A. and N.H. participated in the research process. E.R. and O.A. defined the methodology. O.A. prepared all the figures and evaluation results. E.R. and O.A. wrote the main manuscript text. All authors reviewed the manuscript.

6.8 Acknowledgments

Authors want to thank the project Fondecyt Iniciación N^o11181028, ANID, Chile. Nicolás Hidalgo would like to thank the project STIC-AMSUD code 20-STIC-01, ANID, Chile.

6.9 Authors' information

Erika Rosas is an assistant professor in the Department of Informatics at the Universidad Técnica Federico Santa María, Santiago, Chile. Ms. Rosas obtained her Ph.D. degree in computer science in 2011 from Pierre and Marie Curie University in Paris. She is a former postdoctoral researcher of Yahoo! Labs Santiago. Her research interests include scalable, dynamic and resilient distributed systems, real-time stream processing, mobile ad-hoc communications, in the context of Internet of Things and challenged networks.

Orlando Andrade is an Informatics Engineer and assistant researcher. He obtained his engineer degree at Universidad Técnica Federico Santa María, Santiago, Chile, in 2020. Currently, he works at the private sector in robotics and Artificial Intelligence field. His research interests include ad-hoc networks, Internet of Things and security.

Nicolás Hidalgo obtained his Ph.D. degree in computer science at the Pierre and Marie Curie University of Paris in 2011, Paris. Currently, Mr. Hidalgo is an assistant professor at the Universidad Diego Portales, Santiago, Chile. Mr. Hidalgo is a former assistant professor of Universidad de Santiago de Chile and former full time researcher from Yahoo! Labs, Santiago. His research areas cover highly scalable distributed systems, stream processing, and mobile computing.

References

- [1] Basu, S., Roy, S., Bandyopadhyay, S., Das Bit, S.: A Utility Driven Post Disaster Emergency Resource Allocation System Using DTN. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(7), 2338–2350 (2020). <https://doi.org/10.1109/TSMC.2018.2813008>
- [2] Uchida, N., Fukumoto, M., Ishida, T., Shibata, Y.: Static body detecting methods by locational and rotational sensors on smartphone for dtn based disaster information system. In: Barolli, L., Amato, F., Moscato, F., Enokido, T., Takizawa, M. (eds.) *Web, Artificial Intelligence and Network Applications*, pp. 605–613. Springer, Cham (2020)
- [3] Bhattacharjee, S., Roy, S., Das Bit, S.: Post-disaster map builder: Crowd-sensed digital pedestrian map construction of the disaster affected areas through smartphone based DTN. *Computer Communications* **134**, 96–113 (2019)
- [4] Basu, S., Roy, S., Bit, S.D.: *Reliable Post Disaster Services over Smartphone Based DTN*. Springer, 152 Beach Road, 21-01/04 Gateway East, Singapore 189721 (2019)
- [5] Roy, A., Mahanta, S., Tripathy, M., Ghosh, S., Bal, S.: Health condition identification of affected people in post disaster area using dtn. In: 2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), pp. 1–3 (2016). <https://doi.org/10.1109/UEMCON.2016.7777878>
- [6] MATSUZAKI, R., EBARA, H., MURANAKA, N.: Rescue support system with DTN for earthquake disasters. *IEICE Transactions on Communications* **E98.B**(9), 1832–1847 (2015). <https://doi.org/10.1587/transcom.E98.B.1832>
- [7] Baumgärtner, L., Lieser, P., Zobel, J., Bloessl, B., Steinmetz, R., Mezini, M.: Loragent: A dtn-based location-aware communication system using lora. In: 2020 IEEE Global Humanitarian Technology Conference (GHTC), pp. 1–8 (2020). <https://doi.org/10.1109/GHTC46280.2020.9342886>
- [8] Solpico, D., Tan, M.I., Manalansan, E.J., Zagala, F.A., Leceta, J.A., Lanuza, D.F., Bernal, J., Ramos, R.D., Villareal, R.J., Cruz, X.M., dela Cruz, J.A., Lagazo, D.J., Honrado, J.L., Abrajano, G., Libatique, N.J., Tangonan, G.: Application of the v-hub standard using lora beacons, mobile cloud, uavs, and dtn for disaster-resilient communications. In: 2019 IEEE Global Humanitarian Technology Conference (GHTC), pp. 1–8 (2019). <https://doi.org/10.1109/GHTC46095.2019.9033139>

- [9] Lindgren, A., Doria, A., Schelén, O.: Probabilistic routing in intermittently connected networks. *Mobile Computing and Communication Review* **7**(3), 19–20 (2003). <https://doi.org/10.1145/961268.961272>
- [10] Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: Maxprop: Routing for vehicle-based disruption-tolerant networks. In: *Proc. the 25th IEEE Int. Conf. Computer Communications*, pp. 1–11 (2006)
- [11] Vahdat, A., Becker, D.: Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report, Duke University (July 2000)
- [12] Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In: *Proc. the ACM SIGCOMM Workshop on Delay-tolerant Networking*, pp. 252–259. ACM, ??? (2005)
- [13] Klinsompus, P., Nupairoj, N.: Critical message scheduling for disaster response and recovery phases. In: *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 65–70 (2015). <https://doi.org/10.1109/ICTC.2015.7354496>
- [14] Lieser, P., Richerzhagen, N., Luser, S., Richerzhagen, B., Steinmetz, R.: Understanding the impact of message prioritization in post-disaster ad hoc networks. In: *2019 International Conference on Networked Systems (NetSys)*, pp. 1–8 (2019). <https://doi.org/10.1109/NetSys.2019.8854517>
- [15] Bhattacharjee, S., Basu, S., Roy, S., Bit, S.D.: Best-effort delivery of emergency messages in post-disaster scenario with content-based filtering and priority-enhanced prophet over dtn. In: *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–7 (2016). IEEE
- [16] Keränen, A., Ott, J., Kärkkäinen, T.: 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*, p. 55 (2009)
- [17] Bhattacharjee, S., Roy, S., Bandyopadhyay, S.: Exploring an energy-efficient dtn framework supporting disaster management services in post disaster relief operation. *Wireless Networks* **21**(3), 1033–1046 (2015)
- [18] Martín-Campillo, A., Crowcroft, J., Yoneki, E., Martí, R.: Evaluating opportunistic networks in disaster scenarios. *Journal of Network and Computer Applications* **36**(2), 870–880 (2013). <https://doi.org/10.1016/j.jnca.2012.11.001>
- [19] Rosas, E., Garay, F., Hidalgo, N., Marin, O.: 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, pp. 193–199. IEEE, ??? (2015). <https://doi.org/10.1109/ICT-DM.2015.7402038>
- [20] Uddin, M.Y.S., Ahmadi, H., Abdelzaher, T., Kravets, R.: 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, pp. 1–9 (2009)
- [21] Martín-Campillo, A., Martí, R.: Energy-efficient forwarding mechanism for wireless opportunistic networks in emergency scenarios. *Comput. Commun.* **35**(14), 1715–1724 (2012)
- [22] Rosas, E., Garay, F., Hidalgo, N.: Context-aware self-adaptive routing for delay tolerant network in disaster scenarios. *Ad Hoc Networks* **102**, 102095 (2020). <https://doi.org/10.1016/j.adhoc.2020.102095>
- [23] Luqman, F.: Triage: Applying context to improve timely delivery of critical data in mobile ad hoc networks for disaster response. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 407–408 (2011). <https://doi.org/10.1109/PERCOMW.2011.5766923>
- [24] Begerow, P., Krug, S., Schellenberg, S., Seitz, J.: Robust reliability-aware buffer management for dtn multicast in disaster scenarios. In: 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM), pp. 274–280 (2015). <https://doi.org/10.1109/RNDM.2015.7325240>
- [25] Joe, I., Kim, S.-B.: A message priority routing protocol for delay tolerant networks (dtn) in disaster areas. In: International Conference on Future Generation Information Technology, pp. 727–737 (2010). Springer
- [26] McAtee, F., Narayanan, S., Xie, G.G.: Performance analysis of message prioritization in delay tolerant networks. In: MILCOM 2012 - 2012 IEEE Military Communications Conference, pp. 1–6 (2012). <https://doi.org/10.1109/MILCOM.2012.6415765>
- [27] Liu, X., Chen, Y., Li, C., Taylor, W., Moore, J.H.: Message prioritization of epidemic forwarding in delay-tolerant networks. In: 2014 International Conference on Computing, Networking and Communications (ICNC), pp. 573–577 (2014). IEEE
- [28] An, Y., Luo, X.: Macre: A novel distributed congestion control algorithm in dtn. In: Advanced Engineering Forum, vol. 1, pp. 71–75 (2011). Trans

Tech Publ

- [29] Farooq, F., Bibi, N.: Message admission control along with buffer space advertisement to control congestion in delay tolerant networks (dtns). *EAI Endorsed Trans. Energy Web* **5**(17), 13 (2018)
- [30] Thompson, N., Nelson, S.C., Bakht, M., Abdelzaher, T., Kravets, R.: Retiring replicants: Congestion control for intermittently-connected networks. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9 (2010). <https://doi.org/10.1109/INFCOM.2010.5461934>
- [31] Lakkakorpi, J., Pitkänen, M., Ott, J.: Using buffer space advertisements to avoid congestion in mobile opportunistic DTNs. In: International Conference on Wired/Wireless Internet Communications, pp. 386–397 (2011). Springer
- [32] Coe, E., Raghavendra, C.: Token based congestion control for DTNs. In: 2010 IEEE Aerospace Conference, pp. 1–7 (2010). <https://doi.org/10.1109/AERO.2010.5446944>
- [33] Wang, H., Lv, H., Wang, H., Feng, G.: DCAR: DTN congestion avoidance routing algorithm based on tokens in an urban environment. *Journal of Sensors* **2017**, 9 (2017). <https://doi.org/10.1155/2017/6523076>
- [34] Silva, A.P., Burleigh, S., Hirata, C.M., Obraczka, K.: A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks* **25**, 480–494 (2015). <https://doi.org/10.1016/j.adhoc.2014.07.032>. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication
- [35] Uddin, M.Y.S., Nicol, D., Abdelzaher, T.F., Kravets, R.: 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, pp. 2785–2796 (2009). <https://doi.org/10.1109/WSC.2009.5429249>
- [36] Garay, F., Rosas, E., Hidalgo, N.: When a tsunami strikes: A mobility model for coastline cities. In: Hellingrath, B., Gojmerac, I. (eds.) 4th International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2017, Münster, Germany, December 11-13, 2017, pp. 1–7. IEEE, ??? (2017)
- [37] Saha, S., Sushovan, Sheldekar, A., Joseph C., R., Mukherjee, A., Nandi, S.: Post disaster management using delay tolerant network. In: Recent Trends in Wireless and Mobile Networks vol. 162, pp. 170–184. Springer, ??? (2011)