Bulletin of the *Transilvania* University of Braşov
Series III: Mathematics and Computer Science, Vol. 2(64), No. 2 - 2022, 179-192
https://doi.org/10.31926/but.mif.2022.2.64.2.14

# MESSAGE PRIORITY MAXDELIVERY ALGORITHM APPLIED IN EARTHQUAKE COMMUNICATION SITUATION

**Corina-Ştefania NĂNĂU**[*,1]

**Abstract**

The study of Delay Tolerant Networks (DTNs) has considerably grown in recent years as communication contexts have emerged with needs that go beyond what the Internet could offer. For example, in case of a natural disaster that damages the classic communication network, a delay tolerant network can be implemented ad hoc, to face the challenges imposed by this context. The delay tolerant network is special because there are no permanent end-to-end path between nodes and links characteristics are time-varying. This paper aims to test the performance of the DTN MaxDelivery algorithm in establishing an efficient communication in the context of a post-earthquake situation, that affected the classical communication network. The goal of the algorithm is to maximize the number of high priority messages that manage to reach their destination. A series of simulations are presented to verify the optimal parameters that the network must comply with in order to maximize the message transfer rate.

2000 *Mathematics Subject Classification:* 93C35, 90B10, 68R10.
*Key words:* delay tolerant network, simulation, multiple priority, delivery rate, MaxDelivery algorithm.

## 1 Introduction

According to a United Nations (UN) report published in 2018, natural disasters, such as floods, earthquakes, hurricanes, cause losses of up to 300 billion dollars annually, and the number of people affected by them in the period 1998-2017, was detailed in the paper [1].

Natural disasters have been classified, from the point of view of the phenomenon that triggers them into hydrological, meteorological, climatological and geographical disasters. A disaster occurs when a natural phenomenon affects vulnerable communities and people.

In recent years, some of the most violent disasters have occurred in Indonesia, followed by the tsunami (2004), the earthquake in Haiti (2010) and Cyclone Nargis in Myanmar

[1*] *Corresponding author*, Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: corina.nanau@unitbv.ro

(2008). To this list we can add other events that took place in the period 2018-2020, such as vegetation fires in Greece (2018) and Australia (early 2020), against the background of the heat wave of in recent years, as well as the coronavirus pandemic of the last two and a half years that has affected the world and killed over 6 million people.

From the data provided in [1] it can be seen that the natural phenomena that cause the greatest loss of life and human knowledge are earthquakes, followed by storms and extreme temperatures. In most cases, in such situations, the existing telecommunications infrastructure is seriously affected.

In Tab. 1 we can observe the statistical data collected by the UN following the study of the impact of these phenomena on humanity, between 1998 and 2017.

Table 1: The impact of natural disasters on the population

| **Phenomenon** | **Affected population** | **Deaths** |
|---|---|---|
| Floods | 2 billions (45%) | ≈142.000 (11%) |
| Drought | 1.5 billions (33%) | ≈21.500 (2%) |
| Storms\Hurricanes | 726 millions (16%) | ≈32.000 (17%) |
| Earthquake | 125 millions (3%) | ≈750.000 (56%) |
| Extreme temperatures | 97 millions (2%) | ≈166.000 (13%) |
| Landslides | 4.8 millions (≈0.1%) | ≈18.000 (1%) |
| Fires and volcanic eruptions | 6.2 millions (≈0.1%) | ≈2.400 (0.2%) |

In an emergency situation that arises after an earthquake, coordination and communication are needed in order to be able to help the victims. In order to replace the traditional communication network affected by the earthquake, it is necessary to implement a DTN network. This type of network provides a powerful enough mechanism for storing data packets over a long period of time and redirecting them after re-establishing the connection. Their architecture is proposed by S. Burleigh & Co's articles in [2, 3].

According to [4], a DTN is a network that can connect devices and area on the earth that cannot be served by a traditional network. This is due to the fact that there can be no continuous communication between the end points of communication (the source and the destination of the message). However, in order to make communication possible, the intermediate nodes must take over the custody of the transferred data and pass it on as soon as an opportunity arises. Both nodes and links between them are unsafe in a DTN, and disconnections can be long. A critical challenge for DTN networks is the determination of the transfer route of the message without ever having a previously established connection from the source node to the destination node. Speaking of nodes and arcs, we can conclude that such a network can be modeled as a graph.

Post-disaster challenges that a delay tolerant network have to face in the event of an earthquake:

- the identification of the affected areas and their delimitation

- the evacuation of the victims and granting of first aid
- limited access possibilities due to roadblocks
- communication difficulties due to the damaged networks (the telecommunications, the electricity, etc.)
- coordinating the transport operations of the victims to the hospital or to the shelter centers

The remainder of this paper is organized as follows. In Section 2, the context of simulation scenario is presented. Section 3 presents the MaxDelivery algorithm, used to simulate the context presented in the previous section. Following it, Section 4 provides the results offered after the simulation. In Section 5 there are the conclusions of this study.

## 2 The simulation context

The ONE (Opportunistic Network Environment) simulator [5, 6] is used to perform the simulations in this paper. This is one of the most used simulators for DTN networks due to the facilities offered and the simplicity of configuring the settings necessary to simulate a real context.

In the DTN network used in these conditions, there are several types of actors: rescuers, hospitals, ambulances, shelter centers (refuges) and a command center. These types of actors will represent the nodes of the DTN network that simulate the proposed context. The hospitals, the refuges and the command center are fixed nodes, and the rescuers and the ambulances are mobile nodes.

In a real-life use, the devices that make up the network are distributed in ambulances, hospitals and to the rescuers looking for victims and they use an updated version of the MaxDelivery routing protocol, proposed in the paper [7].

Thus, rescuers travel at an average speed of between 1 km/h and 5 km/h, and the stationary time in case of finding a victim is up to 15 minutes. Their mobility model involves identifying the shortest route between the current location and the location to travel.

The ambulances have an average speed between 30 km/h and 65 km/h, with a parking time of 2 minutes to 5 minutes for the ascent/descent of the victims. At the beginning of the simulation, they have a starting point near a hospital, and then move between a hospital and a position on the map, corresponding to a victim.

The speed and the stationary time were considered based on the context of a difficult movement due to roadblocks caused by the earthquake and the necessary time given to a victim needs.

The hospitals and the shelter centers coordinates are established using a wkt map, also included in the simulator. The operations are carried out on the default map provided by the ONE application, namely the map of Helsinki.

All these node mobility models are implemented in Java, using classes included in the mobility package. The list of the coordinates on the map of the hospitals is used for the ambulances mobility. The movement of an ambulance node reaches two states: that of moving to a victim and that of moving back to the hospital / refuge. The path performed by them is calculated using Dijkstra's algorithm for determining the shortest path in a graph, an algorithm pre-implemented in the simulator.

In the situation described above, messages with different priorities are identified. Similarly, in the papers [8, 9] was approached the problem of implementing a routing protocol in a DTN network, which would address massages with different priority degrees. Having different priorities, it is necessary to sort these messages in an earthquake situation. Thus we will classify the messages as:

1. **high priority messages** - those messages that come from rescuers, requesting the intervention of an ambulance, messages that transmit the location coordinates of a victim, messages that transmit the number of available seats in a certain hospital or in a certain safety shelter, etc.

2. **low priority messages** - messages that send the coordinates of a deceased person (it is assumed that the importance of locating their area is less than that of locating the area of a person who is still alive), messages with requested statistical information by the command center: number of deceased persons, number of wounded, etc.

The frequency with which the messages are generated differs depending on the category of actors involved in the system. From the group of victim rescuers, one message per minute will be generated on average, ambulances will generate one message per 1-2 minutes on average and the fixed nodes will generate one message per 30-60 minutes. This is the standard frequency of message generation, but this is one of the variable parameters of the simulation. We will study the evolution of the message delivery rate in the context of generating more messages versus generating a smaller number of messages.

For the verification of the message transfer rate, simulations were performed, varying several parameters.

In the first phase, simulations were performed with:

1. message size granularity variation: 50k-500k, 125k-1M, 50k-2M

2. message lifetime (TTL) variation: 1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h, 10h

After that, the simulations were performed with:

1. the number of mobile nodes variation: 5 ambulances and 40 rescuers, 10 ambulances and 50 rescuers, 15 ambulances and 60 rescuers, 20 ambulances and 80 rescuers, 25 ambulances and 100 rescuers, 30 ambulances and 150 rescuers, 40 ambulances and 200 rescuers

2. the number of messages variation:

    - ambulances generate messages every 1-2 minutes, rescuers generate messages every minute and fixed nodes generate messages every 30-60 minutes

- ambulances generate messages every 30-60 seconds, rescuers generate messages every 30 seconds and fixed nodes generate messages every 15-30 minutes

Thus, several 24-hour simulations were performed in which all these parameters were varied, also performing combinations of them.

The algorithm used to check the message transfer rate is MaxDelivery, the one presented in the next section. Compared to the version described in [7], it has been adapted to deal with messages that have more priority degrees.

Considering that in the presented concrete life situation, not all the messages that are transmitted have the same importance and I considered it is essential that the really important messages have a delivery rate as high as possible, close to 90%-100%. In this context, the simulator has been modified to allow the generation of messages with several degrees of priority. For the simplicity of the calculation, two categories of messages were considered: those with high priority and those with low priority. But starting from the modification that was made, the application can be easily adapted so that the generated messages can support more priorities.

The version of the algorithm proposed in [7] treats all messages equally, and for this reason, the important messages, not being analyzed as such, had a high chance of being abandoned due to the transport nodes buffer being overloaded with messages. The version of the algorithm proposed in this article, adapted to the concrete situation of a natural disaster, when not all messages have the same priority, manages to maximize the delivery of messages with high priority as much as possible. The simulations performed showed that the messages with high priority also have a high delivery rate using the version of the algorithm proposed in this article, regardless of the parameter that are varied.

## 3 MaxDelivery - the routing protocol used for the proposed context

The goal of this algorithm is to maximize the number of messages that manage to reach their destination, especially those with high priority, by optimizing the selection of the message with the highest chances of delivery, avoiding congested routes and nodes. To achieve the objectives, the proposed algorithm is based on the following actions:

- discovering neighbors - to retrieve information about the network
- forwarding messages that increase the chance of delivery to the destination
- elimination of messages that can no longer reach their destination, in order to give the chance to other messages to be delivered
- periodic cleaning of the buffer

Each node in the network will maintain a list of nodes it has come in contact with. At the time of connection, they will transmit the occupancy of the buffer at that time, will update the list of delivered messages in order to delete their copies from the buffer and only after that will start the transfer of messages. Next, we will present the new version of the MaxDelivery algorithm, the one that takes into account the priority coefficient of the generated messages, detailed by its component parts.

## 3.1 Forwarding the messages

In this network, each node has the buffer divided into three priority queues, ready for message transmission. Messages will only be sent to nodes that have at most 90% of the buffer occupied. According to [7], the three priority queues are divided as following:

1. The first queue has priority 0 and contains messages destined for the contact node.

2. The second queue, with priority 1, contains messages destined for the neighbors of the contact node.

In these two cases, the messages are sorted by size. Smaller messages are sent first, ensuring a higher number of transmissions.

3. The third queue, with priority 2, contains the rest of the messages in the buffer, sorted by a utility function denoted $f_{fwd}(x)$, which favors the forwarding of messages that have been transmitted to less than half of the neighbors of the current node. The utility function has the following definition:

$$f_{fwd}(x) = \begin{cases} HC + MF, & \text{if } HC + MF < TN/2 \\ msgSize, & \text{otherwise} \end{cases} \tag{1}$$

HC represents the number of hops, MF represents the number of local transmissions of the message, TN represents the total number of the node's neighbors and *msgSize* represents the size of the message. The $TN/2$ value is a threshold used to avoid flooding the network with redundant messages.

In the case of messages with different priority degrees, the utility function in [7] has been modified, and it has reached the following form:

$$f_{fwd}(x) = \begin{cases} (HC + MF) \cdot coef_{fwd}, & \text{if } HC + MF < TN/2 \\ msgSize \cdot coef_{fwd}, & \text{otherwise} \end{cases} \tag{2}$$

where $coef_{fwd}$ represents the coefficient of the message forwarding.

Following a preliminary analysis, the simulation values for $coef_{fwd}$ are the following:

$$coef_{fwd}(msg) = \begin{cases} 5, & \text{if the message has high priority} \\ 1, & \text{if the message has low priority} \end{cases}$$

The algorithm used to prioritize messages for transmission to the link node is described in Alg. 1:

Alg. 1 calls four procedures:

- The $FILL\_DEST(DQ)$ procedure populates the DQ queue, which has the highest priority;

**Algorithm 1** The method for messages forwarding

```
procedure FWD
    DQ ← ∅                                   ▷ messages for the contact node
    NQ ← ∅                                   ▷ messages for a neighbor of the contact node
    OQ ← ∅                                   ▷ the other messages in the buffer
    FILL_DEST(DQ);
    FILL_NEIGH(NQ);
    FILL_OTHER(OQ);
    SET_TRANSMISSION_RANGE(DQ, NQ, OQ);
end procedure
```

- The $FILL\_NEIGH(NQ)$ procedure populates the NQ queue, which has medium priority;
- The $FILL\_OTHER(OQ, coef_{fwd})$ procedure populates the OQ queue, which has the lowest priority;
- The $SET\_TRANSMISSION\_RANGE(DQ, NQ, OQ)$ procedure sorts the three queues by priority.

The most interesting of the called procedures is the one that populates the OQ queue. Its algorithm is presented in Alg. 2. It is considered a *util* array that has the same number of elements such as the OQ queue, in which the values of the utility function are inserted. Thus, each message in the OQ will correspond to a value from the *util* array. The coefficient $coef_{fwd}$ is the one mentioned above, which indicates the range of prioritization of the message transmission.

The OQ queue is divided into two areas that have as a separator a threshold defined by half of the number of neighbors for the current node.

**Algorithm 2** Populate the OQ queue

```
procedure FILL_OTHER(OQ, coef_fwd)
    OQ ← allMessages − DQ − NQ;
    util ← ∅;
    for msg ∈ OQ do
        util(msg) ← coef_fwd · msg.Size;
    end for
    for msg ∈ OQ do
        if msg.HC + msg.MF < TN/2 then
            util(msg) ← (msg.HC + msg.MF) · coef_fwd;
        end if
    end for
    SORT_BY_UTILITY(OQ);
end procedure
```

### 3.2 Dropping the messages

According to [7], the formula for dropping the messages is:

$$f_{drop}(x) = RT \cdot (HC + MF) \tag{3}$$

HC and MF have the same meaning as for the utility function in forwarding case and RT represents the receiving time of the message by the current node.

Message dropping component of the MaxDelivery protocol is called when the buffer can't insert a new comming message because of its size. In this case will be removed the message with the highest value of the utility function $f_{drop}$.

In the context of the presence of different message priorities, this formula is changed by entering a priority coefficient for dropping the messages. The new utility function used for dropping messages becomes:

$$f_{drop}(x) = RT \cdot (HC + MF) \cdot coef_{drop} \tag{4}$$

The simulation values for $coef_{drop}$ are:

$$coef_{drop}(msg) = \begin{cases} 1, & \text{if the message has high priority} \\ 5, & \text{if the message has low priority} \end{cases}$$

### 3.3 Cleaning the buffer

The MaxDelivery protocol cleans the buffer of messages that have reached their destination. If the buffer reaches 90% of capacity, the messages created by the current node are also deleted, also using the priority coefficient of the message.

## 4 Simulations results

The first set of simulations was performed by varying the message time to live (TTL) between 1h and 10h. 20 ambulances, 80 rescuers, 5 hospitals and 10 refuges were considered. The ambulances and the rescuers (the mobile nodes of the DTN network) generate messages with size between 125Kb and 500Kb and the hospitals and the refuges (the fixed nodes of the DTN network) generate messages with size between 250Kb and 1Mb. The ambulances generate messages every 60 seconds, rescuers generate messages every 30-60 seconds and the fix nodes generate messages every 30-60 minutes. The considered buffer size of the nodes is of 5Mb.

The second set of simulations considers messages with a higher size variation. In this case, messages with random dimensions between 50Kb and 1Mb were generated by the mobile nodes and messages with random dimensions between 500Kb and 1Mb by the fixed nodes of the network. The values of the other parameters were kept unchanged.

The third set of simulations considers smaller message sizes. In this case, messages with random dimensions between 50Kb and 250Kb were generated by the mobile nodes and the messages generated by the fixed nodes have random dimensions between 250Kb and 500Kb. The values of the other parameters were kept unchanged too.

Table 2: The delivery rate values for high and low priority messages

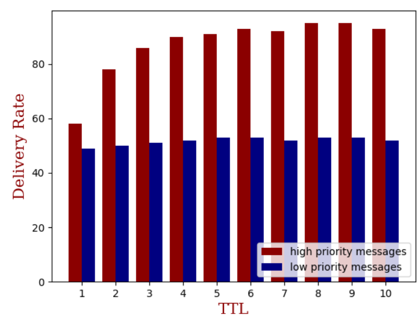
| TTL \ Msg size | **125Kb - 500Kb** | **50Kb - 1Mb** | **50Kb - 250Kb** |
|---|---|---|---|
| **1h** | h: 58 / l: 49 | h: 60 / l: 48 | h: 78 / l: 76 |
| **2h** | h: 78 / l: 50 | h: 78 / l: 53 | h: 95 / l: 93 |
| **3h** | h: 86 / l: 51 | h: 85 / l: 52 | h: 97 / l: 97 |
| **4h** | h: 90 / l: 52 | h: 90 / l: 52 | h: 98 / l: 96 |
| **5h** | h: 91 / l: 53 | h: 92 / l: 55 | h: 97 / l: 97 |
| **6h** | h: 93 / l: 53 | h: 92 / l: 53 | h: 97 / l: 97 |
| **7h** | h: 92 / l: 52 | h: 93 / l: 51 | h: 97 / l: 97 |
| **8h** | h: 95 / l: 53 | h: 94 / l: 48 | h: 97 / l: 96 |
| **9h** | h: 95 / l: 53 | h: 94 / l: 48 | h: 97 / l: 96 |
| **10h** | h: 93 / l: 52 | h: 93 / l: 49 | h: 97 / l: 97 |

The results for these three simulation sets can be seen in table Tab. 2 above and in the three images in Fig. 1. The delivery rate for high priority messages was denoted by "h" and the delivery rate for low priority messages was denoted by "l" into the table.

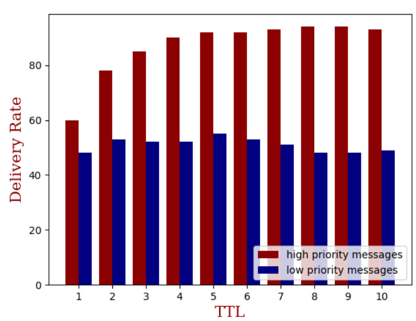
After the first three sets of simulations we can conclude that the delivery rate of high priority messages is very good, exceeding 90% for a TTL with values over 4 hours. A fairly good delivery rate can also be seen for low priority messages. Their delivery rate is approximately 50%. A major change occurs when the messages are smaller, because they are no longer required to be dropped frequently. In this case, the message delivery rate is over 95% for both high priority and low priority messages, at TTL values over 2h.

We will continue with three more sets of simulations, where the environmental conditions were changed. The lifetime of the message remains fixed, with the value of 5h, this being the optimal value, deduced from the previous simulations. This time, the parameter that varies is the number of nodes in the network. The variation will be made only for the mobile nodes, the fixed ones keeping the values from the previous simulations, namely: 5 hospitals and 10 shelters. In this sense, the number of ambulances will be 5, 10, 15, 20, 25, 30, 40, and the number of rescuers will be 40, 50, 60, 80, 100, 150, 200. The dimensions of the messages and the frequency of their generation will be those considered in the first set of simulations, namely: the ambulances generate messages of dimensions between 125Kb and 500Kb and the fixed nodes generate messages with size between 250Kb and 1Mb.

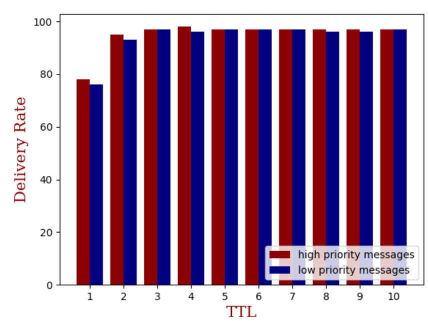
The table Tab. 3 and the figure Fig. 2 below show these results.




(a) Message size variation between 125Kb and 500Kb




(b) Message size variation between 50Kb and 1Mb

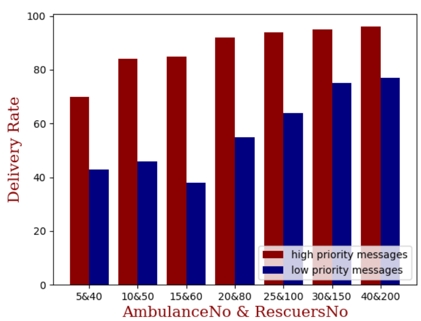



(c) Message size variation between 50Kb and 250Kb

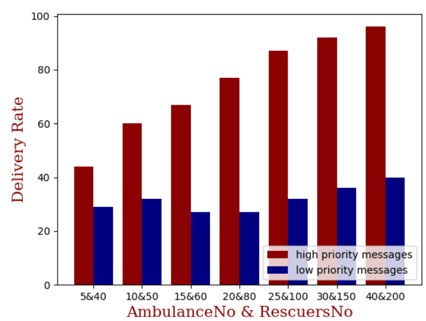
Figure 1: The delivery rates for high and low priority messages

Table 3: The delivery rate values for high and low priority messages

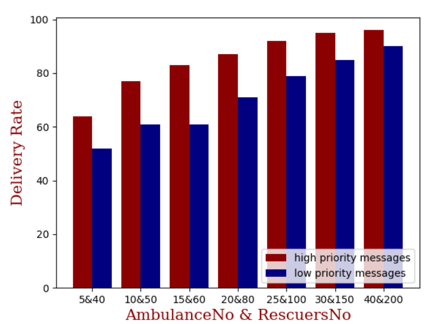
| **Node variation** \ **Sim. cases** | **Case 1** | **Case 2** | **Case 3** |
|---|---|---|---|
| **5 + 40** | h: 70 / l: 43 | h: 44 / l: 29 | h: 64 / l: 52 |
| **10 + 50** | h: 84 / l: 46 | h: 60 / l: 32 | h: 77 / l: 61 |
| **15 + 60** | h: 85 / l: 38 | h: 67 / l: 27 | h: 83 / l: 61 |
| **20 + 80** | h: 92 / l: 55 | h: 77 / l: 27 | h: 87 / l: 71 |
| **25 + 100** | h: 94 / l: 64 | h: 87 / l: 32 | h: 92 / l: 79 |
| **30 + 150** | h: 95 / l: 75 | h: 92 / l: 36 | h: 95 / l: 85 |
| **40 + 200** | h: 96 / l: 77 | h: 96 / l: 40 | h: 96 / l: 90 |




(a) Simulation Case 1




(b) Simulation Case 2 - increasing messages number




(c) Simulation Case 3 - increasing buffer size

Figure 2: The delivery rates for high and low priority messages

The cases addressed by the following three sets of simulations are:

**Case 1:** The ambulances generate messages every 60 seconds, rescuers generate messages every 30-60 seconds and the fix nodes generate messages every 30-60 minutes. The considered buffer size of the nodes is of 5Mb.

**Case 2:** The ambulances generate messages every 30-60 seconds, rescuers generate messages every 30 seconds and fixed nodes generate messages every 15-30 minutes. The considered buffer size of the nodes is of 5Mb, too.

**Case 3:** Due to the low delivery rates obtained in the previous simulation, the node buffer size was increased from 5Mb to 10Mb, and the other parameters are kept identical as in Case 2.

With the increase of the buffer size, the delivery rate of messages has increased, both for those with high priority and for those with low priority. From the presented data it can be seen that the results following the variation of the number of nodes are slightly lower than those obtained when the time to live of the messages varies. This difference is especially visible in the case of low priority messages.

## 5 Conclusions and future work

From the above analysis we can see that as the number of rescuers and ambulances increases, so does the percentage of message delivery. This is due to the high mobility and frequent contacts between nodes. Another important remark is the very high percentage (over 90%) of high priority message delivery, using the adapted MaxDelivery algorithm. The messages generated in this paper had two degrees of priority: low and high; but the algorithm can be easily adapted for messages with a wider range of priorities.

This approach, which involves the presence of messages with different degrees of priority, can also be applied in the works [10, 11]. There are presented networks composed of fixed and mobile nodes, the mobile ones being represented by drones. In those situations, messages with multiple degrees of priority can be entered, and the MaxDelivery algorithm used will be the one in the updated version.

Another interesting direction is to modify the MaxDelivery algorithm so that it uses a position-based strategy to determine better node to forward messages as well as to drop messages from buffer. Approaches in this sense can be found in the papers [12, 13, 14].

## References

[1] Wallemacq, P. and House, R., *Economic losses, poverty & disasters: 1998-2017*, United Nations Office for Disaster Risk Reduction, Technical Report, 2018, doi:10.13140/RG.2.2.35610.08643.

[2] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K. and Weiss, H., *Delay-tolerant networking architecture*, Network Working Group http://tools.ietf.org/html/rfc4838RFC:4838, Apr. 2007.

[3] Scott, K. and Burleigh, S., *Bundle protocol speciffication*, Network Working Group, http://tools.ietf.org/html/rfc5050RFC:5050, Nov. 2007.

[4] Jones, E., *Practical routing in delay-tolerant networks masters thesis*, Waterloo, Ontario, Canada, 2006.

[5] Keranen, A., Ott, J. and Karkkainen, T., *The ONE simulator for DTN protocol evaluation*, Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SimuTools 2009), Rome, Italy, article no. 55, 1-10, 2009, doi: 10.4108/ICST.SIMUTOOLS2009.5674.

[6] Keranen, A., *Opportunistic network environment simulator*, Special Assignment report, Helsinki University of Technology, Department of Communications and Networking, May 2008.

[7] Nănău, C.Ş., *MaxDelivery: a new approach to a DTN buffer management*, Proceeding of 21ST IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WOWMOM 2020), Cork, Ireland, doi:10.1109/WoWMoM49955.2020.00023.

[8] Inwhee, J., and Sang-Bo, K., *A message priority routing protocol for delay tolerant networks (DTN) in disaster areas*, Lecture Notes in Computer Science, vol 6485, Springer, Berlin, Heidelberg, 2010, doi: https://doi.org/10.1007/978-3-642-17569-5_72.

[9] Wang, E., Yang, Y.J., Wu, J. and Liu, W.B., *A buffer scheduling method based on message priority in delay tolerant networks*, Computer Networks and Distributed Computing, **31** (2016), no. 6, doi: 10.1007/s11390-016-1694-7.

[10] Udroiu, R., Deaconu, A. and Nănău, C.Ş., *Data delivery in a disaster or quarantined area divided into triangles using DTN-based algorithms for unmanned aerial vehicles*, Sensors, **21** (2021), no. 11, 1-18, doi: https://doi.org/10.3390/s21113572.

[11] Deaconu, A.M., Udroiu, R. and Nănău, C.Ş., *Algorithms for delivery of data by drones in an isolated area divided into squares*, Sensors, **21**(16) (2021), 1-19, doi: https://www.mdpi.com/1424-8220/21/16/5472.

[12] Sabeetha, K., Kumar, V.A., Wahidabanu, R.S.D. and Othman, W.A.M., emphEncounter based fuzzy logic routing in delay tolerant networks, Wireless Networks, **21** (2014), no. 1, 173-185, 2014, doi: 10.1007/s11276-014-0780-4.

[13] Neelam, M., Shailender, G. and Bharat, B., emphA fuzzy based routing protocol for delay tolerant network, International Journal of Grid Distribution Computing, **8** (2015), no. 1, 11-24, doi: 10.14257/ijgdc.2015.8.1.02.

[14] Rahimi, S. and Jamali, M.A.J., *A hybrid geographic-DTN routing protocol based on fuzzy logic in vehicular ad hoc networks*, Peer-to-Peer Networking and Applications volume, **12** (2019), 88-101, doi: 10.1007/s12083-018-0642-4.