

# DTN Routing Algorithm for Networks with Nodes Social Behavior

A.M. Dziekonski, R.O. Schoeneich

**Andrzej Marek Dziekonski, Radoslaw Olgierd Schoeneich\***

Institute of Telecommunications, Warsaw University of Technology  
ul. Nowowiejska 15/19, Warsaw, 00-665, Poland  
a.m.dziekonski@stud.elka.pw.edu.pl

\*Corresponding author: rschoeneich@tele.pw.edu.pl

**Abstract:** This article presents routing algorithm in Delay and Disruptive Tolerant Networks (DTN). The main idea of this work is routing method that is based on information about nodes social behavior and their social relations in sparse structure of network. The algorithm takes advantage of friendship relationships between nodes and uses historic information to create groups of friends for each node, which is used in buffer management and forwarding phase of routing. Beside the routing method, mechanisms of collecting and exchanging of maintenance information between nodes is described. The algorithm was tested using The ONE simulation tool especially designed for DTN scenario and compared with miscellaneous popular solutions.

**Keywords:** DTN, routing algorithm, social behavior.

## 1 Introduction

Communication has always been important part of different communities lives, both short and long distance. One of dynamically developing wireless networks are ad-hoc networks, that create connections directly between networks nodes - Mobile Ad-hoc Networks (MANET) [1]. This is a type of network, in which its nodes are only users and also they are the only elements required for networks existence and operating. One kind from MANET networks family are Delay Tolerant Networks (DTN).

DTN [2] is a kind of wireless networks that enables communication in sparse and disrupted mobile ad-hoc networks. There are not any central or privileged elements in the means of network operations. Depending on networks purpose, there might be nodes collecting data from others, or controlling them, but this is ensured simply by distribution of proper messages, not by other networks mechanisms. DTN network is generally very dynamic - its elements can freely move, so constant connections between nodes exist very seldom. Also periods between subsequent nodes contacts is not deterministic and usually is very long.

One of the essential parts of telecommunication, and also very important in DTN networks, is routing. Routing is the decision making process of finding the best paths that messages should follow to reach destination in networks. Without this mechanism nodes would not know which messages should be passed to which nodes to provide good network operation. In DTN networks routing is especially important, because contacts between nodes are rare and short lasting, so every opportunity should be perfectly used. Unfortunately, for the same reason, it is more complex than in traditional networks, where all connections are known for long periods and it is easier to find proper paths for messages.

Nodes in DTN networks are devices that move during network operation. Some nodes might be means of public transport like communication equipment connected to buses or trams, smart-phones, small devices connected to animals etc. - there is no limitation of DTN application. Still the most interesting and with the biggest possible usage are networks based on smartphones or other equipment held by people. That is why there is a big need for development of solutions

directed for networks with human mobility patterns, since people movement and behavior is not random, but more-or-less predictable.

## 2 State of the art

Routing protocols for DTN networks are specific solutions due to the necessity of dealing with complex requirements set by the network conditions, mostly in the means of disruption and long message delivery latencies. That is the reason why traditional algorithms from common MANET networks, like [3]: Ad-hoc On-Demand Distance Vector (AODV) or Dynamic Source Routing (DSR) cannot be used.

The easiest solution for delivering messages in DTN networks is Epidemic routing algorithm [4]. It floods the network with data without making decisions based on any criteria. Node is transmitting all of the messages from his buffer to any met node. The order of choosing messages is random. According to the fact, that it creates many copies of the same message in order to distribute them further, this algorithm belongs to the routing solutions group based on message replication. It is a very good reference point while comparing other solutions efficiency, because it chooses randomly proxy nodes and uses all available network resources. In perfect conditions, that are unlimited buffer capacities and infinite connection bandwidth, Epidemic routing acquire biggest possible number of successfully delivered messages to the destination node.

The next solution from replication-based types of routing algorithms is Spray and Wait [5]. It works much alike Epidemic routing, but it does not flood the network with unlimited amount of data. Each message can be replicated only in specified number of copies, which are transferred to other nodes. After creating and transmission of the last copy of selected message, all of the copies are kept in the buffers and transferred only while meeting the message destination node. Spray and Wait can work in two modes - normal and binary. In normal mode only the node that created the message is distributing copies to other nodes, which later can transfer them only directly to the destination node. In binary mode the node transmitting the message passes half of its lasted copies to the other node, and all the nodes that keep in their buffers more than one copy can pass them to any node until they have only one copy left.

Prophet algorithm [6] in contrast to Spray and Wait and Epidemic solutions is based on predictions of future nodes contacts. This is done using historic information collected during existence of the network. The main part of this algorithm is calculation of probability of meeting each of the other nodes from the network. In the first phase it calculates the new probability for the node that currently the connection is held. Second phase is actualization of the transitive probabilities, that is probability with the use of a proxy node (non-direct probability). Prophet solution implements also a mechanism that is prioritizing newest contacts before oldest ones. Nodes that are more active in the network, that establish more connections, have higher probabilities and higher priority in other nodes routing data collections. This algorithm creates many copies of the same message, theoretically infinite - there are no strict limits, but the network is not flooded with messages to the full extent - during contact only those messages are exchanged, that have higher probability of reaching destination while held by the node on the other end of the connection.

The algorithm that is best fitted for DTN networks based on complex, natural mobility models is MaxProp [7], despite the fact that it was developed for vehicle-based networks. In this solution each node keeps information about the network as a graph, which edges weights are calculated probabilities of the contact two chosen nodes. Then algorithm looks for the paths and makes decisions regarding order of the messages in the buffer. The creators did an observation based on many simulations, that transmission of newer messages, so the ones which passed by fewer nodes, in the first place can increase efficiency of routing. That is why in Maxprop algorithm

they developed a mechanism, that messages having hop count less than some calculated border value are transmitted in the first place, and those which passed by more nodes are ordered based on the probabilities calculated with the use of Dijkstra algorithm. One of the important elements of the solution is dealing with successfully delivered messages, that enables clearing nodes buffers from those messages and prevents the unneeded its further exchange.

Many research teams focused on analyzing human mobile patterns for the needs of telecommunication [8–12]. Therefore there are also few algorithms that were developed for networks which nodes are behaving like people. One of the algorithms of this type is Label Routing [13]. It is based on an assumption that people belong to some communities, so they meet with some of the nodes regularly. Authors developed a solution that create a label for each node that tells other nodes about his community. When message is supposed to be forwarded, it is passed directly to the destination node or to the node which is in the same community (has the same label) as the destination node. The problem is that messages are exchanged only if labels of potential next-hop and destination nodes match. In other cases messages are stored and waiting for a proper possibility to forward the message. This increases delivery delay and in some cases, for example with low and restricted node mobility, can have a big influence on routing efficiency.

Another interesting solution focused on social-based DTN networks is Bubble Rap Forwarding [14]. This algorithm takes advantage of two social characteristics - community and centrality. They assume that nodes belong to different communities and are active at different level. The algorithm allows each node to belong to more than one group - one group can be family, another colleagues at work, another friends from high school etc. In each of those groups nodes are prioritized by calculation of centrality of the node among others. This value is kind of popularity of the node and shows which ones have the highest probability to meet all others from that group. Since this value is calculated inside single group, each node has many values of centrality - one for each community to which it belongs. This parameter helps to forward message when it is already held by a node from the message destination node community. Another problem is to exchange message between different groups. For this purpose another centrality is calculated - global one, that takes into account whole network as one group. When new message is created, the first phase is forwarding the message to the destination node community using the nodes global centrality values and then the second phase is exchanging message using local centralities from current group.

Social Based Multicasting [15] is a multicast approach, that is creating many copies of the same message, to DTN routing that uses two social characteristics - centrality and community. In this solution, the centrality of the nodes and the cumulative probability of future contacts is calculated based on Poisson modeling of social networks. Then it uses unified knapsack problem to select relay nodes to assure proper delivery ratio. The main issue is the computational complexity.

On the other hand, the authors of SANE routing [16] focus on completely different social characteristics - interest and similarity. This solution comes from the assumption, that people with similar interests meet each other more often, than the ones that have nothing in common. The interests are represented as a vector, and messages are forwarded to nodes which interest profiles are close to the destination node one.

There are few other algorithms that focus on nodes social behavior and its application in order to develop an efficient routing [17–19]. Most of them focus on three social characteristics: community, centrality, friendship [20]. First of those assume that nodes can be divided into groups (communities) that has many regular contacts between each other and that will continue in further network operation. Thanks to that algorithms can divide its operation into two phases - between communities and inside destination group. Examples of solutions using community characteristic are described above Label and Bubble Rap, and also Friendship Based Routing

[21]. Second of frequently used social characteristics is node centrality, which is a measure of activity and importance of the selected node. This helps to differ nodes inside the network and find most important nodes in the means of potential better forwarding efficiency. The main algorithms taking advantage of centrality is described above Bubble Rap as well as SimBet [22] algorithms. The last social characteristic is friendship, which is measure of the relationship between two nodes. The number of contacts, the periods between consecutive connections, duration of contacts etc. - all of those parameters that can help calculate the grade of relation between two nodes can be taken into account and considered as friendship measure. The example of use of this characteristic is Friendship Based Routing algorithm.

### 3 The social based algorithm

This routing algorithm is based on information of nodes social behavior collected during network existence. Beside the main algorithm, also mechanisms of collecting and exchanging control information are important for making proper routing decisions.

The main assumptions while developing the new solution were that the algorithm should be aimed for networks with human-mobility patterns, but still work well in other types of networks. It should predict future nodes behavior based on historic data collected during network operation. Since in that type of networks resources should not be a big problem - smartphones and similar devices have large amounts of memory and communication interfaces with fast transmission speed - and the main goal is maximizing the number of delivered messages, while minimizing the delivery latency, the proposed algorithm is a full-replication-based type, so there is no limits of created messages copies. Secondary goal is also to minimize the length of the paths that messages follow, since it is proven that it helps the network to operate more efficiently [23].

#### 3.1 Information collected and exchanged for routing purposes

The key element of the solution are historic information regarding the network needed for making routing decisions. This data can be collected by the nodes themselves, or received from other nodes. Each node holds two information regarding whole network. (a.) First of them is maximum popularity value from all network nodes. It is used in mechanism that orders messages to be transferred to other nodes. (b.) Second stored data is the time of last reset of nodes popularity values. This helps to determine time, when table aging mechanism, that is the decrease of number of nodes contacts and groups popularities, should be launched.

For proper network operation it is needed to handle successfully delivered messages. For this purposes each node stores list of already delivered messages and updates and sends it during each new contact. After such an update node deletes from its buffer all messages that identification numbers are included in the received list. Essential for making routing decisions are two tables. First of them stores number of contacts with each node from the network. Based on it, the node decides which nodes belong to its friends group. The table is updated after each contact by increasing the number of connections with currently met node. Then the border value for belonging to nodes friends group is calculated. Identification numbers of the nodes that exceeds this parameter, so the ones that are chosen to friends group, and the popularity of the group, which is defined as calculated border value, are passed to each met node. Second essential information is table containing definitions of other nodes friends groups. For each node, with which at least once connection was established, the list of its friends identification numbers and the group popularity value is stored. This data is used in buffer management mechanism and during making decisions related to transferring messages to other nodes.

Except mentioned above, nodes store also information about the time of the last contact and the average period between those contacts for each node from the network. We use that data as one criteria in memory buffer management.

Nodes gets maintenance information in two ways. Some are collected by the nodes themselves during network operation - for example connection times with other nodes, number of meetings etc. Other information are collected from met nodes.

Due to limited node communication capabilities it is very important to assure effective and compact way of exchanging maintenance data. After establishing a new connection first and significant for solution effectiveness element of the algorithm is the exchange of successfully delivered messages identification numbers. This data is exchanged in both ways. According to them, nodes update their lists and clears their buffers from appropriate messages. Next the information about friends groups are exchanged. After update and rebuilding of nodes friends groups, each of the sides of the connection passes its friends identification numbers and popularity of his group to other node. This helps reduce amount of sent data in this phase of the algorithm. The node that receive this data refresh in its memory information about friends of the node on the other end of the contact. The last collected data from other nodes is information about most active router in the network - hub node. The activeness is measured as the number of contacts. Identification number of the node is not exchanged, only the value of the popularity of this biggest hub node.

### 3.2 Memory buffer management

After establishing a connection between two nodes, exchanging and processing maintenance information, the solution proceeds to the key phase of the algorithm - making routing decisions. It is divided into two stages. The first one concerns the management of buffer, the actions that need to be done when the memory is overloading. The second is directly connected with routing - it makes decisions in what order messages should be transmitted to connected nodes. This order has essential significance because of limited contact times and bandwidth speed.

The first stage is memory management. Nodes have limited space in buffers to store messages. It is highly probable situation, when a new message is created or received from other node, but there is no free space in the buffer. In this case the algorithm is launching decision-making procedure to free some space in the memory by choosing older message to be deleted. In the solution there is separate algorithm to compare messages for buffer management purposes. As a result of its operation all messages are sorted and placed in order to be removed, to free enough space for new message. It is done by directly comparing two messages and deciding which of them is less important and should be deleted before the second one. This algorithm is presented on the diagram below. In the buffer management decision-making process there are few criteria implemented. First of them, while choosing messages to delete, is the number of nodes that message already went through - number of hops. In case of its different values, the one which went through more nodes is chosen to be deleted before the other one. When compared messages have the same hop count, the algorithm checks predicted time of next contact with destination node. Prediction is based on historic information about the time of last contact and average period between them. This predicted time is not very precise, but taking into account non-deterministic nodes movement and the need to minimize resources, that are memory and computation time, it is sufficient to decide which messages should have higher priority to stay in memory. Additionally, to compensate inaccuracy, that difference of two predicted contact times has to exceed some border value. In other case the algorithm passes to the last criteria, which is comparison of paths through friends groups. In the case of memory management the distance is calculated and the message which distance is longer is chosen to be deleted. There is no need

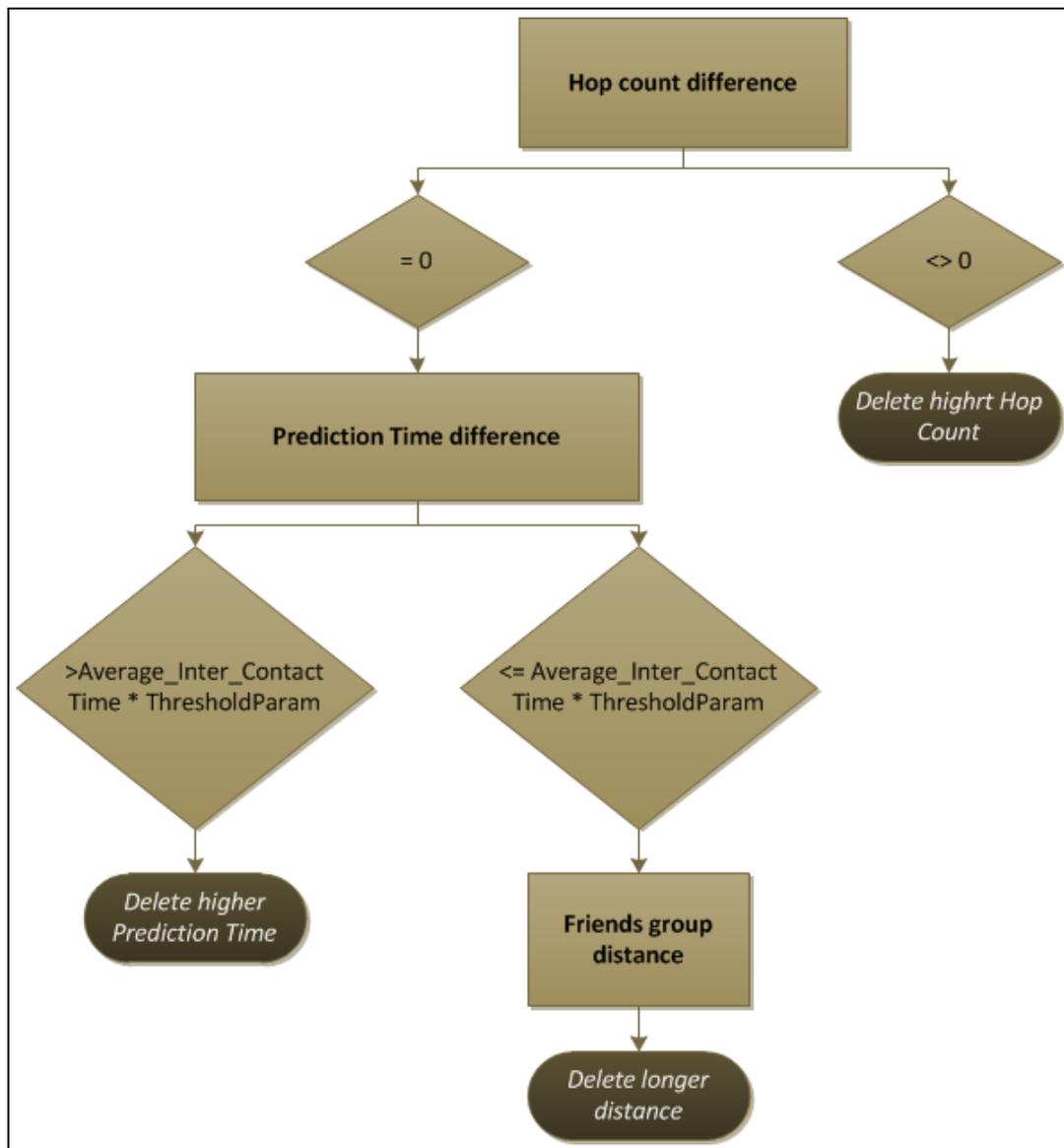


Figure 1: Algorithm to sort messages for buffer-management purposes.

to add any extra criteria - if all of described above steps do not differentiate priority of staying in buffer for two messages, it can be assumed that they are equally important.

### 3.3 Data exchange

The main part of developed routing solution is the way of passing messages to other nodes - methods of making the decisions about the order of messages to be exchanged with the nodes with whom connection is currently established. The primary assumption in passing the messages is the routing algorithm type, that is replication-based what means that many copies of the same message are created. While spreading many copies it is essential to give messages priorities, that tell which ones should be transmitted in the first place. For this purposes the sorting algorithm is developed.

Characteristic distinction between sorting method in this stage of routing algorithm compared to the one in buffer management are the items being sorted. In this case not only messages are ordered, but pairs of message-established connection. It helps better estimate probability of delivering data to destination, depending on which node will be next hop on for the message. Similarly to buffer management stage, in this phase also few criteria are implemented in order

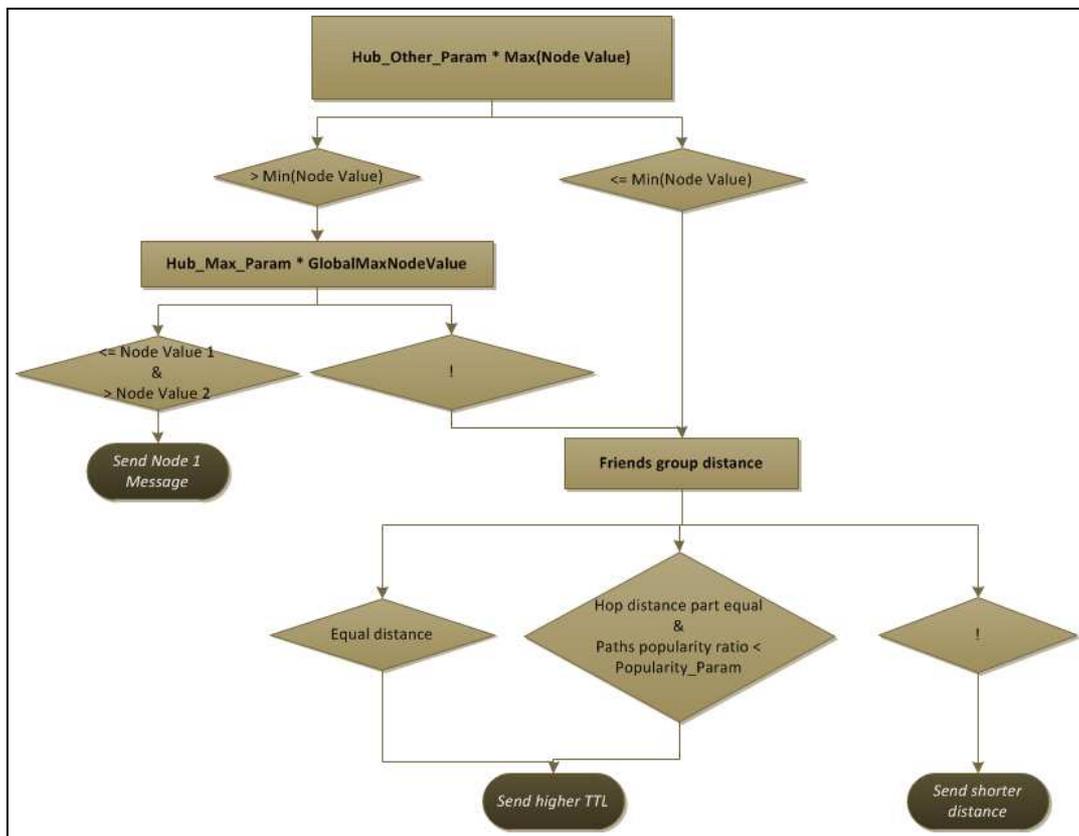


Figure 2: Algorithm to sort pairs messages-connection for data exchange purposes.

to sort elements. First of the criteria is detection, if the node being compared is a hub or not. Before checking that, to eliminate situations when both compared nodes are hubs or just the difference between them is very insignificant, the ratio of popularities of both nodes is calculated (bigger value to smaller) and it has to be bigger than border value set as a solution parameter.

In case, when this ratio is bigger than the border value, both of the compared nodes are checked if they can be classified as hubs. If only one is successfully classified, his pair, which is

the node and compared with its message, is receiving higher priority in sorting process. In other case, when both or none of them are classified, the algorithm advances to next criteria.

The next criteria in routing part of the algorithm is the most important one and makes decisions in most cases - distance based on friends groups. The main goal of the solution is working in networks, which nodes are personal equipment held by people, so the mobility is not random. People belong to different communities, have their own friends with whom they meet most often etc. Those characteristics are mostly constant or at least long lasting. The algorithm draws conclusions based on observations made during network existence - nodes spending most of the time in some location are establishing connections with chosen group of nodes. Additionally for each node those groups are different - family members spend a lot of time together, but they work in different places, take different routes using public transport or car, have different friends. That is why the creation of common groups is very complex and inefficient. In the solution each node creates his own independent friends group and passes its description, containing friends identification numbers and group popularity, to other network elements. Group popularity is calculated based on the number of connections required by nodes to belong to the friends group.

Calculation of the distance based on collection of friends groups is done by looking for a path through those groups to the destination node. In data exchange stage of the solution, the node that currently has an established connection is considered next hop for the message and based on its friends group next nodes on the message path are searched. After finding a path, the number of proxy groups is considered the calculated distance value. Furthermore friends groups have different importance - active nodes have higher border value that recognizes network elements as its friends, so they are more active and this suggests that this group should have higher priority. Popularities of the groups on the predicted path are not added, but averaged geometrically to prevent situations that would prefer the paths with bottleneck group inside, so one group with very small popularity that could decrease efficiency of the path. Still the length of the path calculated just as number of proxy groups is more important. That is why the popularity of the path is considered only when the length of compared paths is equal.

There is an additional limitation of maximum path length, that the search process is finished after reaching that value and if path to destination is not found, the distance is set to infinity. This limitation was introduced for two reasons - (a) to decrease computational complexity, usage of node resources and time needed to find paths, and (b) to introduce next criteria to compare pair message-neighbor node that can have better efficiency than comparison of very long potential message paths. This maximum path length is set as a variable parameter to the algorithm.

If any of mentioned above criteria do not resolve which compared pair should have priority, the algorithm passes to final criteria - comparison of TTL (Time To Live) parameter of the message. Justification is that messages that will not expire in short future have better probability of reaching the destination. It is worth noticing, that in this criteria only messages are compared - the connections in corresponding pairs are not influencing the result.

## 4 Simulations

Simulations were performed to check the efficiency of the solution using The ONE Simulator [24] - a tool developed on Aalto University in Helsinki. It is used for examination of DTN networks operations, mainly for checking network efficiency with the use of different routing algorithms and different nodes mobility patterns. This tool has many advantages that simplify simulations - good programming interfaces to easily add new algorithms, GUI that helps finding problems during network operation and implementation of few mobility patterns and most popular existing routing algorithms, i.e. MaxProp, PRoPHET, Spray&Wait, Epidemic.

Table 1: Values for parameters set for the simulations

<i>Parameter name</i>	<i>Value</i>	<i>Parameter name</i>	<i>Value</i>
Popularity	1,25	Hub Other	0,85
Hop Limit	4	Reset Seconds	21600
Border Value	1,4	Reset Divisor	2,0
Hub Max	0,95	Predict. Time Threshold	1,25

The efficiency of routing algorithms was determined by analyzing few parameters. The main ones are:

- *Delivered messages* - defined as ratio of number of successfully delivered messages to all generated messages.
- *Message delivery latency* - average time between generation of the message and time of its delivery to destination node. Only successfully delivered messages are taken into calculation. Dropped, expired or non-delivered are omitted.
- *Message hop count* - the number of nodes that message went through before reaching destination node. Rejected, expired and non-delivered messages are not included in calculation.
- *Message buffer time* - average time of holding each copy of the message in nodes buffers.
- *Overhead ratio* - represents the level of flooding network with messages. It is calculated as a ratio of number of all copies of all messages exchanged by nodes, reduced by number of delivered messages, to number of delivered messages. In case of Direct Delivery algorithm (that passes messages only when nodes meet destination node) overhead ratio reaches its minimum value - 0.

#### 4.1 Comparison with chosen existing algorithms

In order to compare efficiency of developed solution few simulations were performed, that differs mostly in used mobility patterns and simulation duration. They were executed using five different routing algorithms - proposed new solution, MaxProp, PРоPHET, Spray and Wait and Epidemic. To compare general efficiency of the new algorithm, values of its parameters that have influence on algorithm operation were set top-down, without looking for their optimal values, the same for all simulation scenarios. This way comparison with other solutions is fair and objective. They are presented in Table 1.

The values of the parameters were chosen to be universal and work well in different network types. We consider a node to be a hub if its popularity is at least 95% of the largest encountered popularity in the whole network (Hub Max parameter). This is important only if the difference between two nodes being compared in the message exchange process is larger than 15% (Hub Other parameter). Also each node consider other nodes to be their friends if they meet them at least 40% more often than the average number of all its encounters (Border Value parameter). The aging of the number of encounters mechanism is set to be executed each 6 hours (Reset Seconds parameter) and is done by dividing the number by 2 (Reset Divisor parameter). Maximum length of the path through friends groups is set to be 4 (Hop Limit parameter) and calculated difference in paths popularities must be at least 25% (Popularity parameter). The difference of predicted time of the next encounters for two pairs of nodes must be larger than 25% (Pred. Time Threshold parameter).

The two node mobility patterns that were used are Shortest-Path Map-Based Movement (SPMBM) and Working Day Movement (WDM). The second one is a pattern that simulates in the closest way the behavior of people during normal days of their life, so the one that algorithm was aimed for.

**Delivery ratio**

Delivery ratio, that is number of messages that were successfully delivered to destination, is the most important criteria while comparing routing algorithms. On the figure 3 the results of

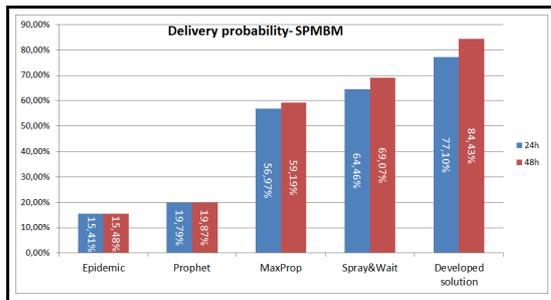


Figure 3: Delivery ratio for Shortest Path Map Based Movement scenario.

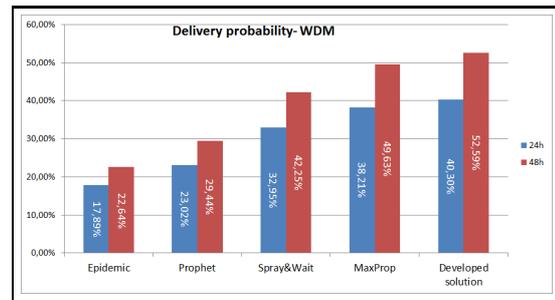


Figure 4: Delivery ratio for Working Day Movement scenario.

simulations for SPMBM mobility pattern scenario are presented. The developed solution reaches the best results from all tested algorithms, regardless of the simulation period. The important observation is that the longer simulation lasts, the greater increase in delivery probability this algorithm reaches from all others. Also it is easy to observe, that the difference between the best and the worst, which is Epidemic routing, is substantial.

On the figure 4 the results for WDM scenario are shown. In this case also the best delivery results are reached by described in the article new solution, but in this case the difference is not that significant. The important thing to observe is that in this scenario, so the one with human mobility patterns, with the longer simulation time the results are improving notably. It shows that nodes are making better routing decisions when they have more data collected, so they learn during network operation.

**Delivery latency**

The second most important criteria while checking efficiency of routing algorithms is delivery latency, because in many situations it is important to deliver messages quick, in other case they can be useless. In the SPMBM scenario (Fig. 5) the new solution gets the longest average time

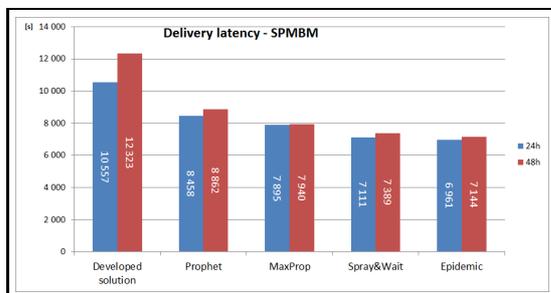


Figure 5: Mean delivery latency for Shortest Path Map Based Movement scenario.

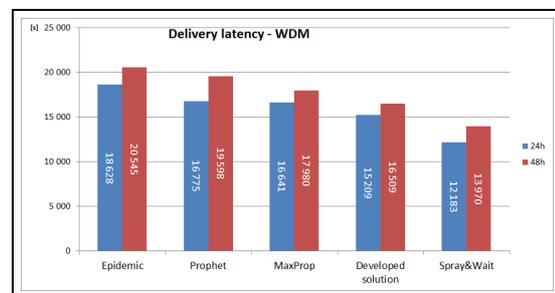


Figure 6: Mean delivery latency for Working Day Movement scenario.

for the messages to reach destination. It is not expected result, since one of the goals was to minimize latency. But it is important to look at this result together with the acquired delivery

ratios. In other case the conclusion would be that the Epidemic routing is the most efficient, since the delivery latency is the smallest. The developed solution has big latency values, because it successfully delivered messages, that other algorithms dropped. This messages often take a lot of time to reach destination, so it overstates the measured average delivery latency.

In WDM scenario the results look even better, than in SPMBM. Despite the fact, that new developed solution has the best delivery ratio, the average latency has quite similar values that all other algorithms. This means that not only it delivers more messages than other algorithms, but also average time of the same messages delivered by different algorithms is smaller for described in the article solution.

### Average message hop count

The figures 7 and 8 present the results of the average number of proxy nodes on the path (hops)

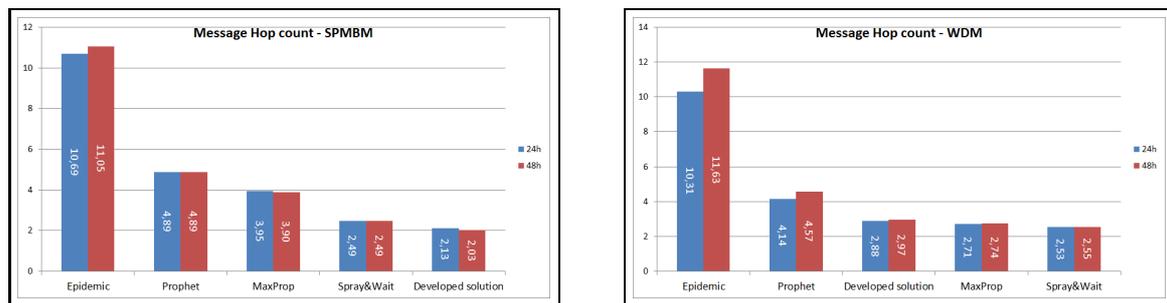


Figure 7: Mean message hop count for Shortest Path Map Based Movement scenario. Figure 8: Mean message hop count for Working Day Movement scenario.

that messages followed before reaching destination. In networks using Epidemic routing the messages passed many nodes before getting to the destination node. This is caused by the fact, that there is no decision-making process and messages are transferred in random order. The best results for WDM scenario, so the ones when messages go through a small number of nodes, are reached by new developed solution, MaxProp and Spray&Wait. They are very close to each other, so the conclusion is that is almost optimal value. In SPMBM scenario also described in this article algorithm and Spray&Wait allow messages to reach destination in shortest paths calculated in number of proxy nodes.

## 4.2 Different network conditions

Each algorithm works the best in certain environment. Different conditions in network can have significant influence on the obtained results. Even if all of the parameters of the network and its elements change the way that network operates, there are few characteristics that have the biggest influence. The main are mobility patterns, that were described in previous chapter, number of hosts, transmission speed and range and buffer sizes.

### Number of nodes in DTN network

We performed the simulation with the use of two algorithms - the new described in the article and Spray&Wait - to have a reference point to existing solutions. The results are presented on the figure 9. In the small network (100 nodes) both algorithms obtain the same delivery ratio, but Spray&Wait need less time for that. With the increase of the network size, the developed algorithm is gaining much better results. The delivery ratio is increasing more dynamically and saturates at higher level than the other algorithm. The delivery latency is also changing in the good way - Spray&Wait needs more and more time to deliver messages with the growth of the size of network, while new solution is obtaining smaller delivery latency in bigger networks. It

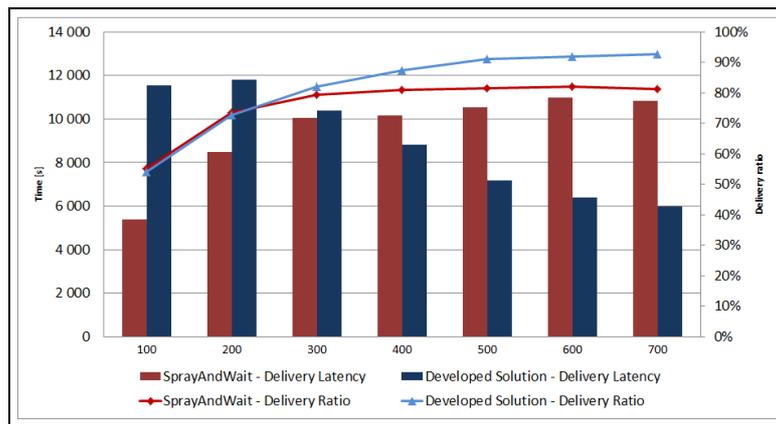


Figure 9: Impact of number of hosts in network.

shows that the algorithm described in the article works best in big networks, but still get good results in smaller ones.

**Transmission speed and range**

On the figure 10 and 11 the influence of changes in transmission are shown - change in transmis-

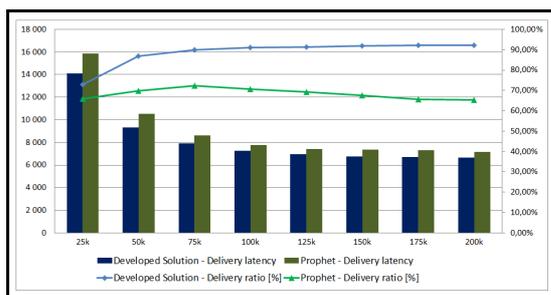


Figure 10: Impact of transmission speed.

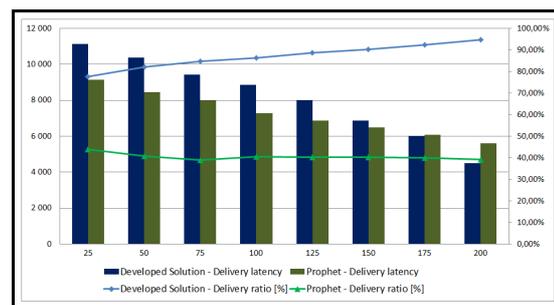


Figure 11: Impact of transmission range.

sion speed, that allows to exchange more data during contact, and change in transmission range, that causes more contacts between nodes. It is clear that with the growth of these parameters, the conditions are becoming better, so the algorithms should work more efficiently. The charts show that new developed solution works exactly as expected - with the growth of transmission speed or range, the delivery ratio is increasing and the latency is decreasing. The improvement in the routing efficiency is better in the new developed solution in comparison to Prophet algorithm.

**Buffer size**

The last examined network nodes characteristic that has big influence on the results is the buffer size. The increase of this parameters allows routers to hold more messages without the need to drop some of them. The figure 12 shows that the described in the article algorithm is more efficient when buffers are smaller and reaches its efficiency maximum for the much smaller buffer size than Prophet algorithm. This means that the buffer management mechanisms and correlated with it decision making process is very efficient.

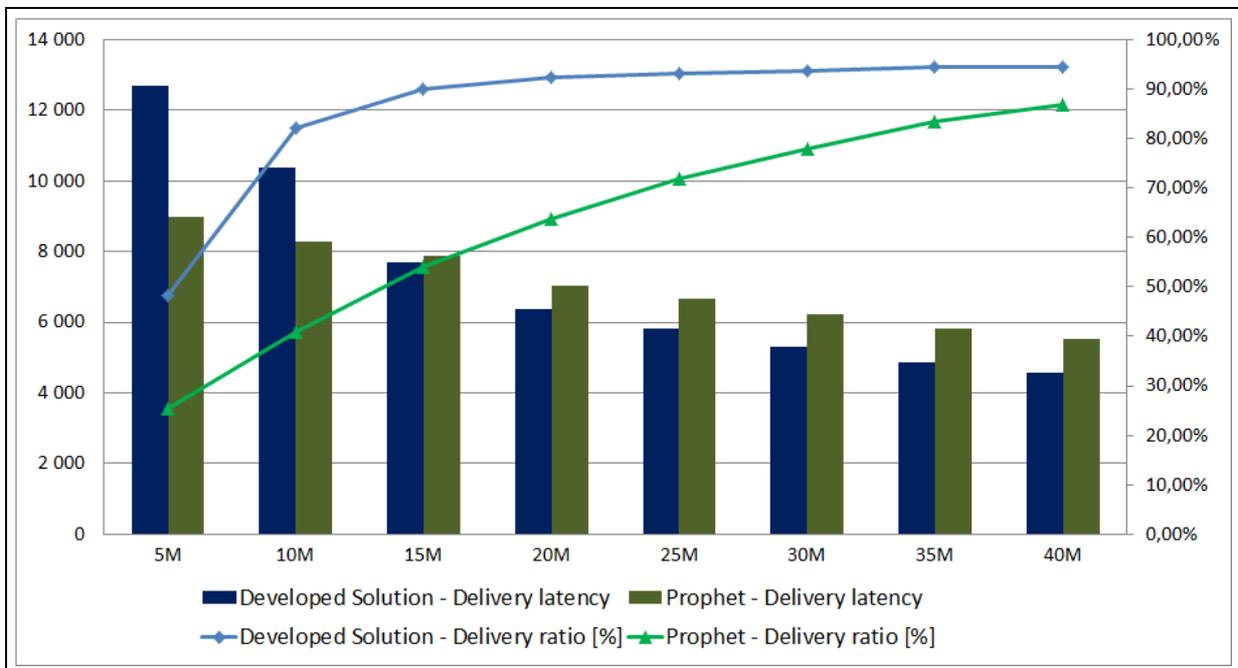


Figure 12: Impact of buffer size.

## 5 Conclusion

DTN networks are a young field of study and are not standardized. That leaves plenty of space for development new solutions for them. One of the very important part of network operation is routing and in case of DTN networks, in which nodes are connecting with others seldom and for short times, its efficiency is essential.

The algorithm that we developed aims for the networks, where nodes follow human mobility patterns, so the ones where networks are made of devices held by people. Our algorithm can be divided into three stages. The first one is the exchange and collection of historic control data needed for predicting future contacts and making decisions in further stages of the solution. The second phase is buffer management - this consists of processing control data and making decisions of releasing space in buffer when it is overflowed. The last stage is strictly making decision process about the messages to be transferred other nodes. Contacts are rare and short-lasting, so it is important to choose which messages to transfer and in what order, because in most cases there is no possibility of exchanging all the messages between nodes during contact. In case of developed solution, which is full-replication based, nodes try to exchange all the messages with other nodes - they do not choose which ones - but focus on the order in which messages are tried to be send.

The developed algorithm gets very good results, especially in the most important criteria, that is successful delivery probability. Compared to existing algorithms (Spray&Wait, Epidemic, MaxProp, Prophet), our new solution if very efficient, especially in the networks with human-mobility patterns.

## Bibliography

- [1] T.Omari; G.Franks; M.Woodside (2005); On the effect of traffic model to the performance evaluation of multicast protocols in MANET, *Electrical and Computer Engineering, Canadian Conference on*, IEEE: 404-407.
- [2] L. Pelusi; A. Passarella; M. Conti (2006); Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, *Communications Magazine*, IEEE 44(11):134-141.
- [3] S. Corson; J. Macker (1999); Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, *IETF RFC 2501*, 1-11.
- [4] A. Vahdat; d. Becker (2000); *Epidemic routing for partially connected ad hoc networks*, Technical Report CS-200006, Duke University.
- [5] T. Spyropoulos; K. Psounis; C.S. Raghavendra (2005); Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks, *Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*: 252-259
- [6] A. Lindgren; A. Doria; E. Davies; and S. Grasic (2012); Probabilistic routing protocol for intermittently connected networks, *IETF RFC 6693*, 1-8.
- [7] J.Burgess; et al. (2006); MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks, *INFOCOM*, 1-11, DOI: 10.1109/INFOCOM.2006.228.
- [8] D. Karamshuk; C. Boldrini; M. Conti; A. Passarella (2011); Human Mobility Models for Opportunistic Networks, *IEEE Communications Magazine*, 49(12):157-165.
- [9] D. Karamshuk; C. Boldrini; M. Conti; A. Passarella (2012); An Arrival-based Framework for Human Mobility Modeling, *Proceedings of the IEEE International Symposium WoWMoM*: 1-9.

- 
- [10] A. Passarella; M. Conti; C. Boldrini; R. I.M. Dunbar (2011); Modelling Inter-contact Times in Social Pervasive Networks, *Proceedings of the ACM MSWiM*: 333-340.
- [11] C. Boldrini; M. Conti; A. Passarella (2007); Users Mobility Models for Opportunistic Networks: the Role of Physical Locations. *Proceedings of the WRECOM07*: 1-6
- [12] C. Boldrini; M. Conti; A. Passarella (2009) The Socialble Traveller: Human Travelling Patterns in Social-Based Mobility, *Proceedings of the MobiWAC*: 34-41.
- [13] P. Hui; J. Crowcroft (2007); How small labels create big improvements, *Procedeengs ot the IEEE PerCom*: 65-70.
- [14] P. Hui; J. Crowcroft; E. Yoneki (2011); Bubble rap: Social-based forwarding in delay-tolerant networks, *Mobile Computing, IEEE Transactions*, 10(11): 1576-1589.
- [15] W. Gao; Q. Li; B. Zhao; G. Cao (2009); Multicasting in delay tolerant networks: a social network perspective networks, *Proceedings of the ACM MobiHoc*: 299-308.
- [16] A. Mei; G.Morabito; P. Santi; J.Stefa (2011); Social-aware stateless forwarding in pocket switched networks, *Proceedings of the IEEE INFOCOM*: 251-255.
- [17] Y. Zhang; J. Zhao (2009); Social network analysis on data diffusion in delay tolerant networks, *Proceedings of the ACM MobiHoc*: 345-346.
- [18] W. Gao; G. Cao (2011); User-centric data dissemination in disruption tolerant networks. *Proceedings of the IEEE INFOCOM*: 3119-3127.
- [19] F. Fabbri; R. Verdone (2011); A sociability-based routing scheme for delay-tolerant networks, *EURASIP Wireless Communications and Networking*: 1-13.
- [20] Y. Zhu; B. Xu; x. Shi; Y. Wang (2013); A survey of social-based routing in Delay Tolerant Networks: positive and negative social effects, *IEEE Communications Surveys and Tutorials* , 15(1):387-401.
- [21] E.Bulut; B. K. Szymanski (2010); Friendship based routing in delay tolerant mobile social networks, *Proceedings of the IEEE GLOBECOM*, 10.1109/TPDS.2012.83, 23(12): 2254-2265.
- [22] E. M. Daly; M. Haahr (2007); Social networks analysis for routing in disconnected delay-tolerant manets, *Proceedings of the MobiHoc*: 32-40.
- [23] C. Boldrini; M. Conti; A. Passarella (2012); Less is More: Long Paths do not Help the Convergence of Social-Oblivious Forwarding in Opportunistic Networks. *Proceedings of the ACM/SIGMOBILE MobiOpp*: 1-8.
- [24] A. Keranen; J. Ott; T.Karkkainen (2009); The ONE simulator for DTN protocol evaluation, *Proc. of the SimuTools*: DOI: 10.4108/ICST.SIMUTOOLS2009.5674.