SNbR: Social Networks-based Forwarding in Delay Tolerant Networks

by

Nguyet Minh Nguyen

M.Sc., Vietnam National University, 2006

B.Sc., Vietnam National University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2012

Abstract

Recently, Web 2.0 has become very popular with the appearance of many social networks web sites such as Facebook, Google+, YouTube, Flickr, etc., which allow users to create and share contents. Besides Web 2.0, mobile networks have also become more popular than ever with the increasing popularity and diversity of mobile devices. Users of mobile devices also have the desire to connect and share their own contents everywhere and at any time. Supporting communications in mobile networks becomes an interesting research area which has attracted much attention of the research community in recent years. In the direction of investigating the application of social networks in supporting the communications in delay tolerant networks which are formed among mobile devices, in this thesis we propose a novel message forwarding method, called Social Networks-based Forwarding (SNbR), in DTNs which utilizes social networks among mobile users and the context of node relationships to facilitate the communications between mobile devices. Simulations in the scenario of a city with the working day movement model have shown that our proposed method achieves a better performance in terms of delivery ratio against delivery cost compared to many previous methods.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

SNbR	Social Networks-based Routing/Forwarding
DTNs	Delay Tolerant Networks
MANETs	Mobile Ad hoc Networks
PROPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
SimBetTS	Similarity, Betweenness and Tie Strength-based method
ONE	Opportunistic Network Environment
PROPICMAN	Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks
HiBOp	History Based Routing Protocol for Opportunistic Networks

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Son Vuong, for his guidance, support and encouragement. I would also like to thank to my co-supervisor Dr. Mike Feeley for the discussions and feedback for my MSc thesis research.

Thanks to the members in the NIC Lab for the feedback on my research work.

Finally, I would like to thank my family and friends for all their supports during my graduate studies at UBC.

Nguyet Nguyen

The University of British Columbia

April, 2012

Chapter 1

Introduction

With the advancement of wireless network technologies, nowadays mobile users are able to communicate anytime and anywhere. One major trend is the increasing popularity of mobile devices in a variety of forms. There has been active research in making networking between mobile devices possible which enables mobile users communicate and share information with each other in a peer-to-peer fashion, a trend that has been observed with the arrival of Web 2.0 and the appearance of numerous social networking sites such as Facebook, Google+, YouTube, Flickr, etc. It is the time that users can create and generate contents and share these contents with other users, making the world that we live in rich in communication and information sharing. Being able to create and share contents, therefore, is unavoidably what mobile users would also hope to be supported in mobile networks.

The ideas of Delay Tolerant Networks (DTNs) were introduced by K. Fall in a seminar paper in 2003 [13]. This is the concept about disconnected mobile ad hoc networks [1] where instantaneous connected paths are not available and communications are enabled by mobility of nodes and network storage. The ideas of DTNs have been well received by the research community in the area of mobile networks and this opens a future in which mobile devices could be able to communicate even when networks are not connected. DTNs are also supported with the Web 2.0 trend where mobile users can connect and share information even in the extreme conditions of the networks. Message routing and forwarding in DTNs is an interesting research area with the goals that are to increase delivery ratio, reduce network delay and to be cost effective because resources are limited in mobile networks compared to fixed networks.

1.1 Motivation

Unlike traditional networks, networks in DTNs are not connected and paths from source to destination have to be found through disconnected hops over time. When no information of the network is available, flooding-based methods such as epidemic algorithm [34], spray-and-wait [32], constrained flooding [28], and two-hop flooding [15] use multiple copies of messages to send in the network with the hope that some of them will reach the destination. Although flooding-based methods often provide high delivery ratio, they have high network overhead with a lot of redundancy. The overhead of flooding-based methods might be destructive to network operations because mobile devices often have limited resources.

To reduce the overhead of flooding-based methods, probabilistic methods such as PROPHET [23], MV [7], Mobyspace [22] use some information from the network to selectively decide relay nodes for messages. Unlike the early methods which assume that node mobility is random such as random waypoint, random walk, probabilistic methods assume that mobile nodes in DTNs often have repeated mobility patterns such as visiting some nodes or places frequently or periodically. The methods are proposed to exploit these repeated patterns to predict the future node encounters and help relay messages between nodes.

Working in the same manner as the probabilistic methods, the context-based methods such as HiBOp [4], PROPICMAN [27], SANE [26] perform the statistics on the contexts of encountering nodes to select carriers for messages. They are based on the assumption that if people often meet with other people having some certain context properties, then they will be a good carrier for a destination with context properties similar to the nodes that they have met.

Studies in DTNs such as [8], [6], [37] also show that because people carry mobile devices, mobile nodes often form communities driven by users' social behaviours. The studies have performed the analysis on mobile network traces and shown the detection of communities in DTNs from the encountering patterns of mobile nodes. The studies have also shown that node encounters in mobile networks are sufficient to build a connected graph, which is a small-world graph [35]. Because people having social relationships

often meet each other, they could be message relays for each other. In this direction, the social network analysis-based methods such as BubbleRap [16], SimBetTS [9] exploit social networks in DTNs for message forwarding. These methods utilize only the social network structure of nodes in DTNs. The idea is that nodes in the same community will be able to meet each other, and, therefore, they would be good message carriers for other members of the community. And nodes with high degrees of social centrality will have chances to meet more nodes and therefore could help to propagate messages in the network.

In this thesis, we design a novel social networks-based message forwarding method in DTNs called SNbR combining the social network structure of nodes and the searchability based on context in social networks. The social network structure exhibits the small-world links that come from the observation that individuals are often linked by a short chain of acquaintances. It has also been known that social networks are searchable based on the information such as geographical locations, occupation, etc. [36]. By combining these two metrics, messages are propagated to nodes which have high chances of encountering the destinations, and which are nodes that close to the destination than the current node in terms of social relationships.

1.2 Thesis contributions

The thesis proposes a method to combine the advantages of the social network structure and the searchability in social networks based on node context such as location, occupation, etc. Nodes having higher chances of encountering the destination are nodes that have high centrality degree in the social networks and have high context similarity with the destination. Based on the social network structure, the method exploits nodes which have better chances of communications with other nodes. Applying the results of the studies from social network studies, node centrality is calculated from the social networks discovered by a community detection algorithm. To estimate the context similarity of a node to destination, each node is attached with a context profile to indicate its context such as location, occupation, etc. Nodes make the forwarding decisions based on the popularity of a node and its matching context with the destination node.

The following is the details of the contributions of our study:

- Estimating the social centrality of nodes in social networks based on the centrality measures in social network studies [14] [24] and the social strength of nodes to each community.
- Providing a novel message forwarding mechanism called SNbR based on node context profiles and node centrality in social networks.
- Implementing the SNbR method in the ONE simulator [19].
- Evaluating the performances of SNbR in the ONE simulator compared to the other methods BubbleRap, PROPHET and Epidemic.

1.3 Thesis organization

The remainder of the thesis is organized as follows. In Chapter 2, we present the background and related work to the current study. Details of the design of the SNbR method in DTNs are presented in Chapter 3. In Chapter 4, we present the implementation of the SNbR method in the ONE simulator. Chapter 5 describes the performance evaluation of the SNbR method compared to the other methods BubbleRap, PROPHET and Epidemic using the simulations in the ONE simulator. Finally, Chapter 6 presents the conclusions and future work of the thesis.

Chapter 2

Background and Related Work

In this chapter, we present the background of DTNs and related work in message routing and forwarding in DTNs.

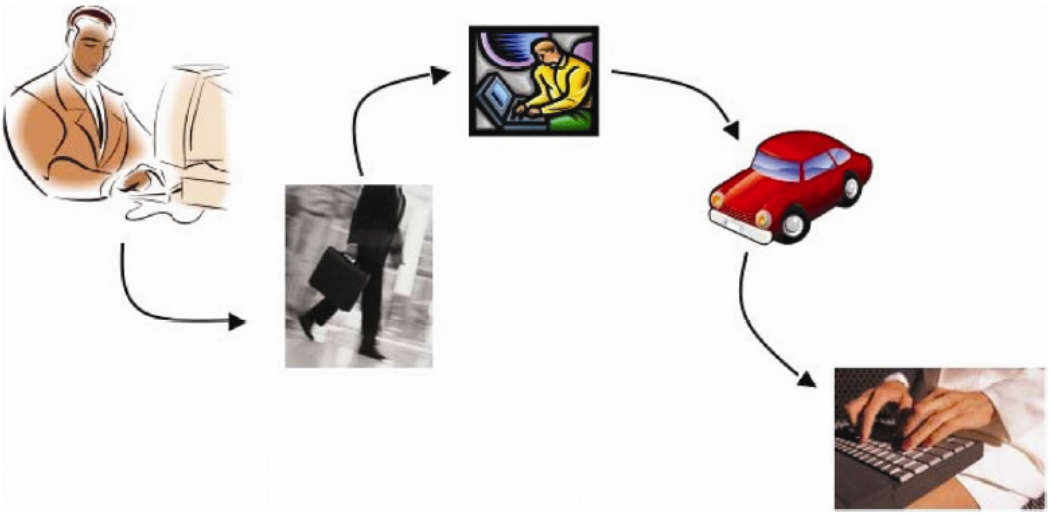
2.1 Delay tolerant networks

The penetrating of personal mobile devices has attracted the research in mobile ad hoc networks (MANETs) [1] which are originally developed for military applications. MANETs provide infrastructure-less multi-hop wireless networks between mobile devices which are constantly moving. In that dynamic environment of the networks, supporting communications and data transfer between mobile nodes via multi-hop paths is challenging. Routing in MANETs has been studied extensively. Many protocols have been proposed to deal with the dynamic network topology in MANETs, such as OLSR [2], AODV [31], DSR [18], etc. However, routing algorithms in MANETs implicitly assume that the networks are connected and there exist end-to-end paths between all pairs of source and destination. In reality, that may not be true when MANETs are sparse or node mobility is too high causing frequent disrupted connections, and nodes may conserve power causing the links shut down periodically. All of these problems can cause the networks disconnected and there may not be end-to-end paths between source and destination and routing protocols in MANETs cannot work properly in these scenarios. This leads to the research interests in DTNs.

DTNs become a new term to address disconnected mobile ad hoc networks. Because paths between pairs of source and destination are not present intermediately, messages have to be buffered at intermediate nodes, and the mobility of these nodes are exploited to bring messages closer to their destination by exchanging messages with the nodes that they opportunistically meet. The concept of DTNs can be demonstrated as in Figure 2.1 [5]. A desktop user wanting to send a message to a friend located far away

opportunistically transfers via a wi-fi ad hoc link to a user walking by in the street with the hope that this user is moving closer to the destination. This user may come to a train station and pass the message to a traveller heading to the same city as the destination's home. At the train station of the destination city, the driver of a car passing by is going to the same neighbourhood of the destination's home. The driver encounters the destination user on his way and the message is delivered to the destination user.

Figure 2.1 The delay tolerant network concept [5]



The method of routing in DTNs which buffers messages when next hops are not intermediately available for the current node to forward data and the current node carries the messages until it gets an opportunity to forward the messages is called store-carry-forward routing. It performs routing by moving the messages closer to the destinations one hop at a time. The challenging with routing in DTNs is to determine for each message the best nodes to forward the message, which are nodes that have the highest chances of successful delivery, i.e., the highest delivery probability. In the next section, we present the current routing strategies in DTNs.

2.2 Routing strategies in Delay Tolerant Networks

In this section, we start with the discussion of random methods or flooding, replication-based methods. If nodes know nothing about network states, then all nodes can do is to randomly forward messages to their neighbors. The routing algorithms belonging to this

category are epidemic and its various forms. If nodes can determine the forwarding probability of their neighbors, they can make better forwarding decisions. These routing algorithms are called estimation-based forwarding. There are many different approaches in estimation-based forwarding based on the information used to determine nodes' delivery probability. The first type of estimation-based forwarding methods that we will discuss is probabilistic routing which use the raw statistics of past node encounters, contacts, visiting places etc. The second type of this category is context-aware routing which performs the statistics of past contacts not on the frequency of encounters but on the contexts of the nodes that the current node encounters. Finally, we discuss social network analysis-based routing which makes forwarding decisions based on social network structure.

2.2.1 Replication-based routing

In replication-based routing algorithms, the messages received at intermediate nodes are forwarded to all or part of the nodes' neighbors without using any estimation of the path forwarding probability of each node. The strategies base on making multiple copies of messages and hope that one of these copies will reach the destination. This approach is a natural approach when there is no information can be determined about the movement patterns of nodes in the network.

The examples of routing algorithms in this category are epidemic algorithm [34], spray-and-wait [32], constrained flooding [28], and two-hop flooding [15]. In [34], Vahdat and Becker propose an epidemic routing protocol for intermittently connected networks. When a message is received by an intermediate node, the node will flood the message to all of its neighbors. Hence, messages are quickly propagated through networks when carriers of messages come into contact with other nodes from node mobility. When nodes are in the communication range of each other, they exchange summary vectors containing the list of messages stored at each node. Based on received summary vectors, each node requests messages that are not yet in its buffer. Messages are delivered to the destination when the destination encounters a node carrying messages destined for it. This method guarantees delivery if a route is available but is expensive in terms of resources since the network is flooded.

In [32], Spyropoulos et la present the Spray-and-Wait solution which tries to reduce the number of copies of messages by taking a more constrained flooding approach. The Spray-and-Wait method assigns a replication number to each message and distributes message copies to a number of carrying nodes and then waits until a carry node meets the destination. Beside, Ni et la in [28] use a simple approach to reduce the overhead of flooding by only forwarding a copy with some probability $p < 1$, which is essentially randomized flooding. Another extreme is to let the source hold the message and deliver to the destination only when they are in communication range of each other. This approach obviously has minimal overhead but the delay could be long. In [15], Grossglauser et la propose a 2-hop forwarding approach. With this approach, a source node will give a message destined to a destination to another randomly chosen node one hop away in the network which is called receiver. When the receiver is in the communication range of the destination node, the receiver sends the message to the destination node.

The replication-based methods have been shown to work well in dense environments such as in classrooms or conferences and as well provide fair performance in sparse settings such as city-wide networks in terms of delivery ratio and delay. However, in terms of delivery cost, the replication-based methods are far from satisfactory because they create a lot of redundant messages as a side-effect of the delivery scheme, and the overhead could rapidly become unacceptable in contentious, vulnerable and resource-scarce mobile networks.

2.2.2 Probabilistic routing

In the methods of probabilistic routing, instead of blindly forwarding messages to all or some random neighbors, intermediate nodes estimate the chance for each neighbour node of eventually reaching the destination. These methods assume a non-random mobility of nodes and rather nodes have repeated mobility patterns which could be exploited to improve the delivery rate of messages.

The examples of the probabilistic methods are PROPHET [23], MV [7], Mobyspace [22], etc. In [23], the algorithm called PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity) is proposed, which exploits the frequency of

contacts between nodes. PROPHET uses past encounters to predict the probability of meeting a node again. Nodes that are encountered frequently will have an increased probability while the probability with older contacts will be decreased over time. Moreover, the transitivity of encounters is also exploited and nodes exchange encounter probabilities and the probability of indirectly encountering the destination node is evaluated.

In [7] and [22], information about nodes' mobility patterns and places nodes often visiting is exploited to forward messages to destination. In [7], the meets and visits (MV) protocol estimates the likelihood of forwarding messages by learning the frequency of meetings between nodes and visits to certain regions. The past frequencies are used to rank messages according to the likelihood of delivering messages through a specified path. MV determines a probability that the current node can successfully deliver a message to a region or a specific node within a number of transfers.

In [22], Leguay et la propose a mobility pattern-based method called MobySpace which exploits the knowledge of node mobility. MobySpace uses a high-dimensional Euclidean space to present nodes' mobility patterns. It is based on the assumption that two people with similar mobility patterns (i.e., encountering similar nodes or visiting similar places) will be more likely to meet each other and therefore will be able to communicate. Routing is done by forwarding messages to nodes that have mobility patterns more similar to the mobility pattern of the destination.

2.2.3 Context-based routing

The mobility patterns of nodes could be complimented with context awareness which is applied in the context-based routing algorithms. Context could be seen as a collection of information that describes the reality about the node and the history of social relationships with other nodes. Similar to the probabilistic routing algorithms, the context-based routing algorithms also assume that people are not likely to move around randomly, but rather move in a predictable fashion. The node's context information, similar to mobility patterns of nodes, plays an important role in predicting the mobility of nodes which is driven by the users' social behaviours. To describe a context of a node,

the information about the users that could be used is the name of the users, the user's residence, work place, and hobbies such as address of sports facilities, etc.

Haggle project [33] studies extensively the communication in Pocket Switched Networks. Most of its proposed works focus on the interactions between individuals and the information of who a node communicates with, which is stated as context awareness. Based on the history of node interaction and context, message forwarding is guided through passing messages to nodes with similar interests and history [4], [27]. In [4], Boldrini et la propose a context-aware routing framework for routing in DTNs called HiBOp. HiBOp forwards messages not only based on the current context of a node which is the information about the user and the current neighbors of the node, but also remember information about other users that the node met in the past. HiBOp uses a history table to record attributes seen in the nodes encountered in the past. The information in the history table allows HiBOp to estimate the probability of encountering nodes with these attributes in the future which contains far more information than just identity of encountered nodes as in probabilistic routing protocols. This allows HiBOp to exploit similarity between encountered nodes and destination. In [27], Nguyen et la propose a method called PROPICMAN (Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks) which uses context to guide the forwarding of messages by estimating the context of two-hop neighbourhood of a node.

In [26], Mei et la propose a social-aware stateless forwarding method called SANE (Social Aware NEtworking) based on interest profiles. The interests as defined in the paper could be the degree of interest of a node in a certain topic such as cinema, literature, etc., or the physical communities that the node belongs to such as a Facebook interest group, neighbourhood of residence, etc. The method bases on the assumption that the users' movement is guided by their personal interests and people with similar interests often have high chances to meet each other. Messages are forwarded to nodes that have higher degree of interest similarity with the destination.

In the thesis, we use the node context profile similar to HiBOP and PROPICMAN. However, we combine the context similarity of a node to destination with the social

popularity of a node to predict the probability of a node to deliver a message to destination.

2.2.4 Social network analysis-based routing

In social network studies, it is shown that social networks exhibit the phenomenon of small-world networks [35] in which people are often connected to each other in short chains. Recently, in [11] Eagle et la. perform some social networks studies using mobile phones to collect data about the interactions between individuals. The goal is to compare the information about the social relationships based on self-report data and mobile data. The studies show that mobile data not only accurately infer 95% of social relationships but also show the relationship between these social relationships and location and proximity. In [8], [6], [37], the studies also confirm that there exists a small-world phenomenon in mobile social networks. And the encountering of mobile nodes is enough to relay messages to any nodes in the networks. With the existence of social networks in DTNs, there are routing algorithms proposed in DTNs that exploit the social network structure between nodes to facilitate message forwarding in DTNs such as BubbleRap [16] and SimBetTS [9].

In [16], Hui et la devise a message forwarding method called BubbleRap with an assumption that nodes are in the same community will have the chances to encounter each other. Therefore, messages could be forwarded to members of the community of the destination and these members in the community of the destination will have chances to encounter the destination and deliver messages to the destination. In BubbleRap, each node has two rankings which are global and local rankings. The former ranking shows the popularity of the node in the entire network, the latter ranking shows its popularity in its own community. Messages are forwarded to nodes with higher global rankings until the messages reach the community of the destination. After that, the messages are forwarded based on the local rankings of nodes in the community.

In [9], Daly et la. propose the SimBetTS method which bases on three social measures such as betweenness centrality of a node, similarity of a node to destination and the tie strength with the destination. The betweenness centrality of a node measuring the extent to which a node lies in the shortest paths linking other nodes is estimated based on the

node's ego network [24]. The similarity metrics is calculated as the number of common neighbors between the current node and destination node. And the tie strength with the destination is measured based on three indicators frequency (i.e., the frequency with which a node is encountered), intimacy (i.e., the amount of time the node has spent connected to a given node), and recency (i.e., how recently two nodes encounter). When two nodes are in contact, a utility function combining these three social measures is calculated for each destination and the node with higher utility value for a destination is given the message.

In this thesis, we combine the social popularity of a node (i.e., its centrality degree in the social graph [14]) with its context similarity to the destination as the indicator of the node's delivery probability for forwarding the message to the destination. Our method will be able to take advantages of both the social popularity and context similarity of nodes. Nodes with higher social popularity will meet more nodes and will have high chances of bringing messages closer to the destination. Nodes with high context similarity with the destination will have high chances of encountering the destination. Using context similarity is supported by the results in social network studies on searchability in social networks which state that "people are able to find short paths even though they know very little about the target individual or the network" [35]. People are capable of forwarding messages through their network of acquaintances to reach a specific but distant target person in only a few steps. In [36], Watts et la provide a model to explain why social networks are searchable. In [20], Kleinberg explains about navigation in a small world network, how short chains are found through social networks. The searchability of social networks can be explained in terms of recognizable personal identities such as geographical location, occupation, etc. [3, 25, 29].

Chapter 3

System design

In this chapter, we present the design of our message forwarding method called social networks-based forwarding (SNbR) in DTNs which exploit the social network structure and searchability of social networks in DTNs to facilitate message forwarding in DTNs. Messages are forwarded to nodes which have higher degree of social popularity and context similarity to the destination than the current node. First, we present the overview of the system and the message forwarding process. Next we detail the use and management of context followed by the estimation of the node's social popularity which is social centrality degree of the node. Finally, we present how the node's context and social popularity degree are combined to make the message forwarding decisions.

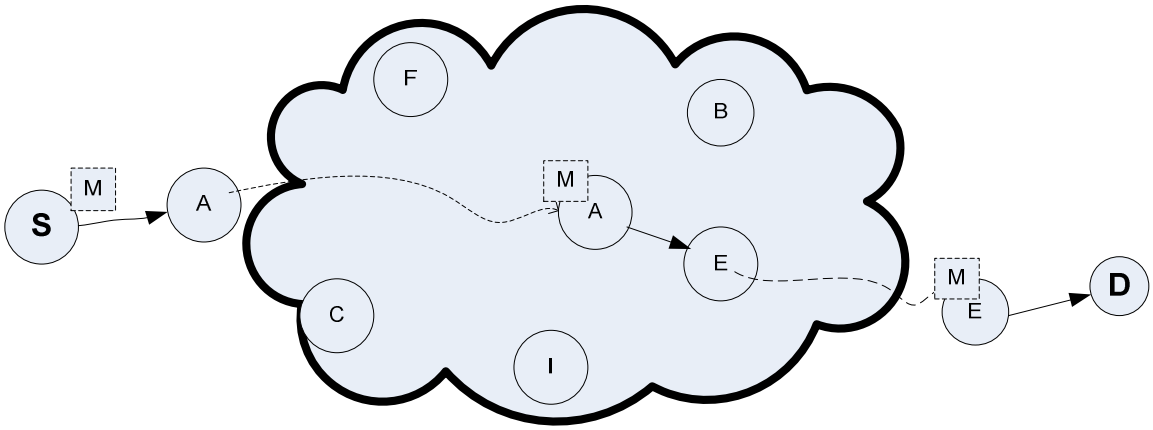
3.1 Overview of the system

Because networks in DTNs are not connected, message forwarding is based on the opportunities of contacts to reach a certain destination, which is decided by the probability of bringing messages closer to the destination. The main mechanism is that messages are forwarded only to nodes that heuristically have higher chances of getting closer to the destination. The goals of SNbR are to i) efficiently propagate messages through opportunistic contacts in DTNs using information of node popularity in social networks and context similarity to destination, ii) minimize the amount of resources spent in the message forwarding process and iii) maximize the number of messages that are delivered to the destinations successfully.

The SNbR forwarding process can be illustrated as in Figure 3.1. The source first sends the message into the network, and maybe replicates the message for the reliability reason with the number of message replication specified as an algorithm parameter. By exploiting nodes' mobility and contacts, the message proceeds in the network towards the destination. In our method, each node forwards the message only to nodes that have more

advantages than the current node in terms of social popularity and context similarity with the destination. The delivery probability of a node to a destination can be seen as a combination of these two factors. When a node carrying the message finds the destination the process stops.

Figure 3.1 Message forwarding process (adapted from [4]**)**



In the following sections, we will present the methods to estimate the social popularity of nodes and methods to compare the node contexts.

3.2 Node context profiles

Like HiBOp [4] and PROPICMAN [27], we use the context information to make message forwarding decisions. However, we combine this context information with the social popularity of a node to make the forwarding decisions which allow the message forwarding to take the advantages of the social roles of relay nodes and to look for the node with similar context as the destination. The social popularity of nodes will be discussed in the next section. In this section, we present the building of node context profiles.

The system provides a pre-defined set of context attributes and assigns a weight for each attribute which could be both drawn from the social network studies of homophily in social networks such as [25] (Table 3.1). Defining the set of context attributes and assigning weights to context attributes are in the discretion of the designer of the system

and may be based on specific scenarios that the system is applied. Different weights could be assigned to context attributes based on how important the information of the attribute is in assisting to find and reach the destination. Table 3.2 gives an example of a pre-defined set of five context attributes and the assigned weights to attributes which are determined heuristically. For example, the attribute "office location" in the example is assigned a higher weight than the attribute "name" because it indicates more important information to reach the destination. Two users may have the same name but they may never meet each other. However, if two users are in the same office, there are chances that they will meet each other.

Table 3.1 Pre-defined set of context attributes and assigned weights

Context attributes	**Weights**
Att_1	w_1
Att_2	w_2
…	…
Att_n	w_n

Table 3.2 Example of pre-defined set of context attributes and assigned weights

Context attributes	**Weights**
Name	1
Home location	4
Office location	5
Profession	2
Specialization	5

From the pre-defined set of context attributes, each node creates its own node context profile (Table 3.2). The node context profiles include a set of context attribute-value pairs. Table 3.3 gives an example of node context profile.

Table 3.3 A node context profile

Context attributes	Att_1	Att_2	...	Att_n
Values	v_1	v_2	…	V_n

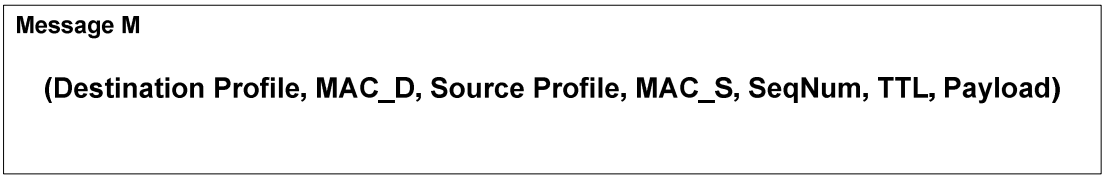
Table 3.4 Example of node context profile

Context attributes	Name	Home location	Office location	Profession	Specialization
Values	James Smith	Marine Drive	CS/UBC	Grad student	Computer Science

Node context profiles are important to determine the good relay nodes for messages. Destination's context profile is used to direct messages over networks. The values of node context profiles can be entered manually, registered to a trusted organization or detected automatically. Context attributes could be seen as a signature of a node. The combination of attributes is unique for each node.

When a source node sends a message M to a destination, it includes the destination profile and source profile in the header of message M which will be used by the SNbR routers to decide the relay nodes. The format of message M is given in Figure 3.2.

Figure 3.2 Format of a message M



3.3 Social popularity and social strength towards a community

3.3.1 Node social popularity

In a social network, not every node has the same popularity. Some nodes move often and encounter more nodes than other nodes. To estimate the importance of a node in a social structure, centrality in graph theory and network analysis [14] is used to quantify the relative importance of a vertex within a social graph. A central node often has a stronger capability of connecting other network members.

Degree centrality is measured as the number of direct contacts that involve a given node. A node with high degree centrality has relationships with many other network nodes. The social popularity of a node could be determined based on the ratio between the value of degree centrality and the total number of nodes in the network except the node itself as follows. If N is the number of nodes in the network, the **social popularity** of a node p_i, where $a(p_i, p_k)=1$ if p_i and p_k is directly linked in the social graph is calculated as:

$$SP(p_i) = \frac{\sum_{k=1}^{N} a(p_i, p_k)}{N-1} \qquad (1)$$

Notice that the value of SP is in the range of 0 and 1, i.e., $0 \leq SP \leq 1$. SP =0 when a node has no contact with other nodes, and SP=1 when a node have contacts with all the other nodes in the network.

3.3.2 Social strength towards a community

In [17], Hui et la proposes distributed community detection algorithms in DTNs which are able to detect both static and temporal communities. Communities are parts of the network in which the nodes are more highly connected to each other than to the rest of

the network [30]. Communities are also called clusters, cohesive groups or modules in the network. In the experiments presented in Chapter 4, we use the k-clique community detection algorithm proposed in [17] which is the distributed version of the community detection algorithm proposed in [30].

Here, we present the way to calculate the social strength of a node towards a community which shows how close the user is to a community and how likely a node gets in touch with the community. The social strength of a node towards a community is measured by the frequency of a node visiting the community. Using the community detection algorithm [17], the visiting frequency of a node could be measured by the time the node spends in different communities (i.e., for each community, the duration from the start of the first encounter of a node in a community to the end of the last encounter of a node in the community) and by monitoring transitions between communities. To calculate the frequency, we use a time window with the width of T. If $\sum t_{p_i}^{C_k}$ is the total time that a node p_i stays in a community C_k in the time window T, the frequency of the node p_i visiting community C_k in the current time window T is as follows:

$$CurrFreq_{p_i}^{C_k} = \frac{\sum t_{p_i}^{C_k}}{T} \qquad (2)$$

$CurrFreq_{p_i}^{C_k}$ is updated after each node visit to the community.

The average frequency of the node p_i visiting community C_k over time, which presents the social strength of the node p_i to community C_k, can be estimated as follows:

$$Freq_{p_i}^{C_k} = \beta \cdot Freq_{p_i}^{C_k} + (1-\beta) \cdot CurrFreq_{p_i}^{C_k} \qquad (3)$$

This is an exponentially weighted moving average that places an emphasis proportional to β on the past visit frequency of the node. To place more weight on the frequency of the current time window, β could be chosen with the value of 0.25.

3.4 Calculating context similarity

The context of a node contains the information about itself and also nodes belonging to the same communities as the node. Because a node can belong to many communities and

have different social strength towards each community, the node's context similarity with the destination has to take into account these factors.

For each node *p*, the similarity of its own context profile compared to a destination *d* is estimated by comparing the values of the attributes in p's context profile and destination d's context profile. The estimation is performed by the ratio of the sum of all the weights of the matching attributes between node p_i and destination *d* and the total weights of all the attributes specified in context profile. The **context similarity** of node p_i to destination *d* can be formulated as follows:

$$CS(p_i, d) = \frac{\sum_{i \in \{\text{matching attributes}\}} w_i}{\sum_{i \in \{\text{all attributes}\}} w_i} \qquad (4)$$

The context similarity of all the nodes belonging to a community C_k which is a set of nodes to destination is the maximum over all the context similarity of each node in the community.

$$CCS(C_k, d) = \max_{p_j \in C_k} \{CS(p_j, d)\} \qquad (5)$$

A node belongs to many communities and its visit frequency to each community estimated in formula (3) reflects the social strength of the node to the community. The social context similarity of the node to a destination *d* is computed as follows:

$$SCS(p_i, d) = \sum_k Freq_{p_i}^{C_k} CS(C_k, d) \qquad (6)$$

3.5 Node routing metrics

To estimate the chances of the node to successfully deliver the message to a destination, the routing metrics RM of the node to a destination node *d* is computed by combining the **social popularity** of the node with its **context similarity** compared to destination as follows:

$$RM(p_i, d) = \alpha \cdot SP(p_i) + (1 - \alpha) \cdot SCS(p_i, d) \qquad (7)$$

We want to take into account that node context similarity is more important than its social popularity, so α is chosen with the value of 0.25.

When two nodes meet, each node examines the messages it is carrying and computes the routing metrics for each destination. Messages are then exchanged where the message is forwarded to the node holding the highest routing metrics for the message destination node.

Chapter 4

Simulation Experiments

In this section we present the implementation of the SNbR router in the Opportunistic Networking Environment (ONE) simulator, a Java-based simulator developed by the Networking Laboratory at the Helsinki University of Technology [19]. ONE is specially designed for evaluating protocols developed for DTNs and opportunistic networks. It allows creating scenarios based upon movement models and offers a framework for implementing new routing and application protocols. By implementing the SNbR router in ONE, we could perform simulations with the SNbR routers and compare the performances of SNbR to other algorithms such as BubbleRap, PROPHET and Epidemic.

4.1 ONE simulation environment

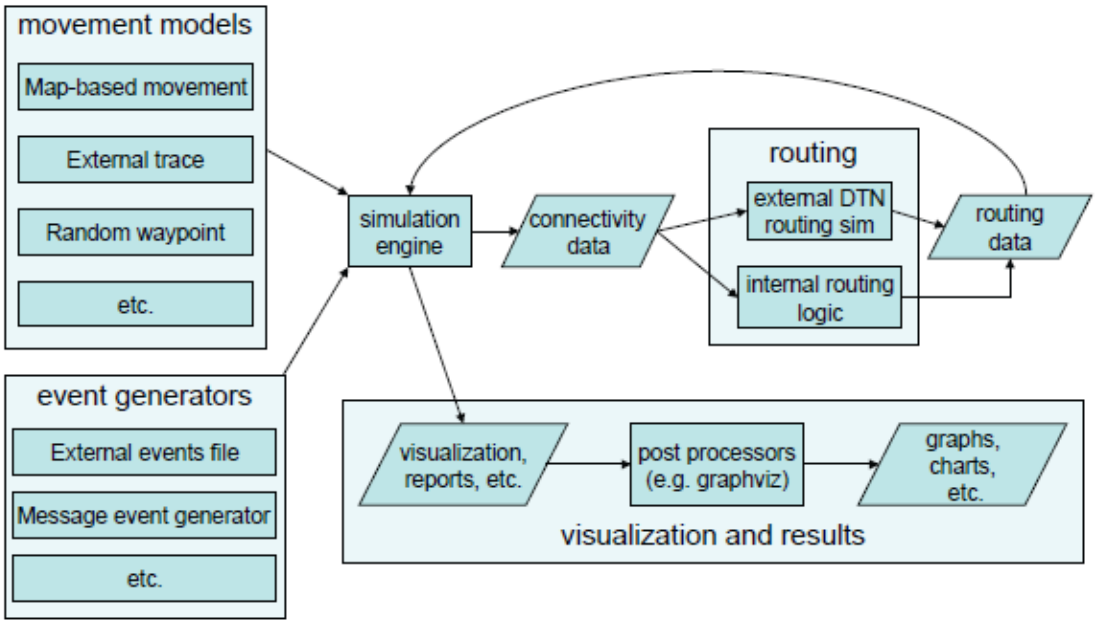
The ONE simulator is an agent-based discrete event simulation engine [19]. The main functions of the ONE simulator are to model node movement, inter-node contacts, routing and message handling. Result collection and analysis of simulations are performed through visualization, reports and post-processing tools. The elements and the interactions between them in the ONE simulator are shown in Figure 4.1.

Node mobility in simulations is generated by movement models. Connectivity between nodes is based on their location, communication range and the bit rate. The routing function is implemented by routing modules that decide which messages to forward over existing contacts. And messages of applications in DTNs are generated through event generators.

The ONE simulator includes report modules that report simulation results during the simulation run. Report modules receive the events such as message events or connectivity events from the simulation engine and generate results based on them. The results generated may contain logs of events that could then be further processed by external post processing tools or they may be aggregate statistics calculated in the simulator such

as delivery ratio, overhead ratio, average latency, etc. The simulator also includes a graphical user interface (GUI) that displays a visualization of the simulation state showing the locations, active contacts and messages carried by the nodes.

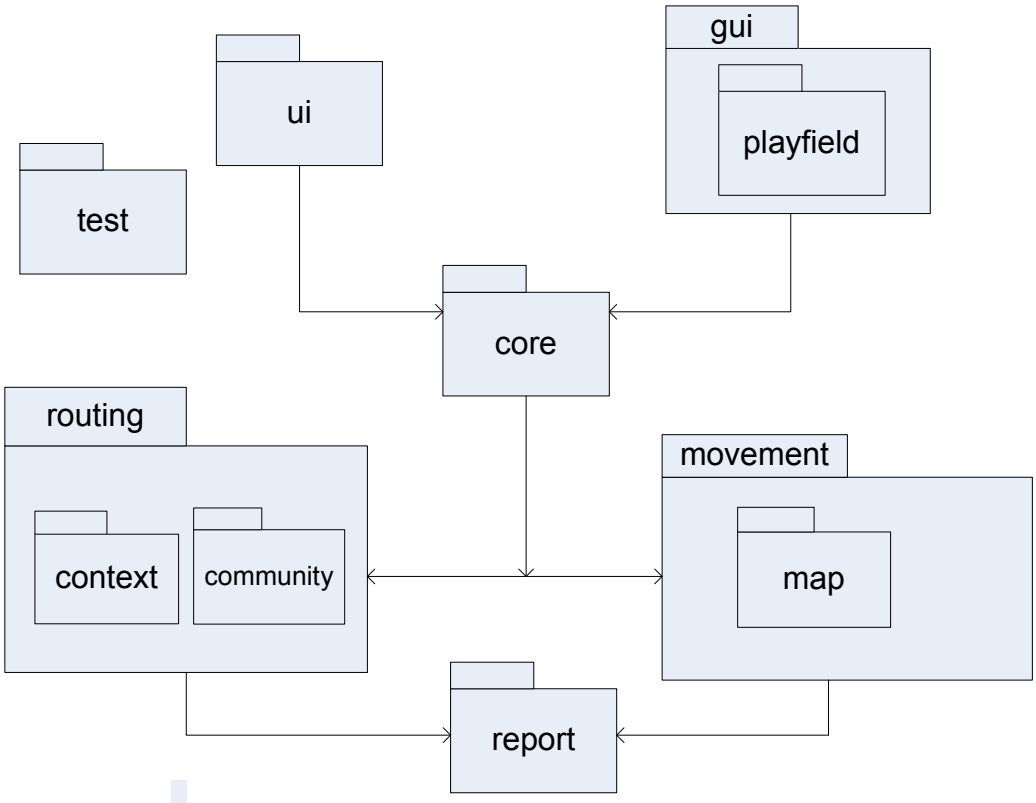
Figure 4.1 Overview of the ONE simulation environment [19]



4.1.1 Software architecture

Figure 4.2 depicts the packages in the ONE simulator and the new packages added into the ONE simulator to implement the functions of the SNbR router. The figure shows the dependencies between the packages. Core components of the simulator such as classes presenting a DTN host or a connection are in the core package. GUI-related classes are in the gui package which also has the playfield sub-package containing classes presenting graphical objects in the play field view. Generic user interface class and the text-based console class are stored in the ui package. The G(UI) classes instantiate SimScenario and World classes from the core package which in turn create instances of routing modules from the routing package and movement models from the movement package. During the simulation, routing and movement modules provide output for report modules from the report package. The test package is not directly related to the simulator, but it contains a set of unit and system tests that can be run to check if the system performs as expected.

Figure 4.2 The ONE software packages



We extend the ONE simulator to include the context profile component in DTN hosts. The format of messages is also adjusted to contain the profiles of destination and source. We particularly enhance the core package and the routing package of the simulator. In the core package, we extend the implementation of the DTNHost class and Message class. We add a new package routing.context to implement the ContextProfile class which implements context profile vectors and related classes. We extend the classes in routing.community to calculate the node centrality and the community visit frequency. The SNbasedRouter class is added implementing the computation of the SNbR algorithm.

4.2 SNbR implementation in the ONE simulator

In this section, we detail the implementation of the SNbR router in the ONE simulator. Specifically, the implementation of the SNbR router extends the classes in the core package and routing package. In the following sub-sections, we present the core class dependency, routing class dependency, and the sequence diagram of the routing decider module.

4.2.1 Core class dependency

Figure 4.3 details the core class dependency. To use the context information for making forwarding decision, the ContextProfile class is created to define the format of context profile vectors. The ContextProfile class includes the attributes to store the context information of a node such as home location, office location, host ID. The ContextProfile class also specifies the weights for attributes.

An attribute of the type ContextProfile is added to a new class ContextEmbeddedDTNHost which extends the class DTNHost. The new class ContextEmbeddedDTNHost inherits all other attributes and operations from the class DTNHost and adds new operations to act on the new attribute of ContextProfile.

The Message class contains the attributes of the source DTNHost (from) and the destination DTNHost (to). Managing messages in DTNHost is done by the MessageRouter class. Each DTNHost has a component of MessageRouter. The MessageRouter manages the collections of incoming messages, messages that a DTNHost is currently carrying, and delivered messages which are the messages that the current node is the final destination.

4.2.2 Routing class dependency

The dependency of the routing classes is depicted in Figure 4.4. In the ONE simulator, all routing modules must extend the MessageRouter class. The MessageRouter class provides the basic interface and functionality for routing modules such as simple buffer management and callbacks for various message-related events. These callbacks are invoked by the simulator engine for all kinds of events, e.g., when a new message is created or a message is sent to the node. A specific router module needs to handle these events and also define actions to be carried out at every time step and the behaviour when a new node comes into or leaves the node's radio range. The following is the details of some of the operations performed by the MessageRouter class.

When a node wishes to try to transfer a message to another node, it asks the related connection object to start the transfer, which in turn forwards the request to the other node. The other node calls its router module's receiveMessage method. The router

module can check whether it wants to receive the message. The router module may rejects the message, e.g., if it already has it in its message buffer.

In addition to the message receiving method, the MessageRouter class also has methods that are called when a message was transferred successfully, aborted, or when a new message should be created or should be deleted from the buffer. The MessageRouter class also has two important methods that it does node provide implementations for: the changedConnection and update methods. The former is called every time a new connection comes up or an old connection goes down. And the latter is called on every update round. The subclasses of the ActiveRouter class which extends the MessageRouter class have to provide implementation for both of these methods since with them they are able to react to connection opportunities and also handle the ongoing transactions.

The two classes extending the MessageRouters are ActiveRouter and PassiveRouter. The PassiveRouter class is used to interact with the real-time testbed. The ActiveRouter class extending the MessageRouter class provides many utility functions that all active routing modules use. These functions include sending messages that can be delivered to a directly connected host or trying to send a set of messages in a certain order using a given set of connections. It also provides a simple implementation for the update method. The update implementation finishes ready transfers, abort transfers if the connections was torn down before the message transfer finished and also drops messages whose time-to-live (TTL) has expired. All the active routing modules in the ONE simulator need to extend the ActiveRouter class.

The DecisionEngineRouter class extends the ActiveRouter class to provide a routing framework which when an event occurs, an object implementing the interface RoutingDecisionEngine will be consulted to make the actual decisions. The SNbasedRouter class implements the RoutingDecisionEngine interface. To make the routing decision the SNbasedRouter also needs to interact with the objects of the classes CommunityDetection and Centrality.

4.2.3 Routing sequence diagram

In this section, we describe the details of exchanging messages to make the routing decisions. The two main functions of the router modules are to manage and control the events with messages and connection changes. The events with messages include make message transferring requests, make decisions of receiving messages. The events also include when a message was transferred successfully, when a transfer is aborted, when a new message should be created or when a message should be deleted from the buffer. The second function of the router modules is to deal with the events of changedConnection and update. The update method is called on every update round of the simulation. In the following, we discuss in more details the events of changedConnection. The changedConnection event occurs when a new connection comes up or an old connection goes down.

Figure 4.5 depicts the routing decider sequence diagram. When the ConnectionInterface object detects a new connection comes up, it calls the connectionUp method of the DTNHost object. The DTNHost calls the changedConnection method of the DecisionEngineRouter. The DecisionEngineRouter notifies the SNbasedRouter by calling the method ConnectionUp and then calls the method shouldSendMessageToHost to make the decision to forward the message to the other host of the connection. The SNbasedRouter then calls the CommunityDetection object with the method getLocalCommunity to get the set of nodes in its local communities and calls the object Centrality to calculate the node centrality. The SNbasedRouter also calls its own method getContextSimilarity to compare the context of the node with the destination. The decision of sending the message to the other node will be based on the values of node centrality and context similarity. When a new connection is up, the DecisionEngineRouter also calls the method doExchangeForNewConnection of the SNbasedRouter which then calls the method newConnection of the object CommunityDetection to update the properties of the social connection between two nodes in the connection.

When the NetworkInterface object detects a connection is down, it calls the method connectionDown of the DTNHost. The DTNHost then notifies the DecisionEngineRouter

by calling the method changedConnection. The DecisionEngineRouter calls the method connectionDown of the SNbasedRouter. The SNbasedRouter notifies the CommunityDetection object by calling the method connectionLost which updates the connection history of the two nodes in the connections such as the contact time and inter-contact time.

Figure 4.3 Core class dependency

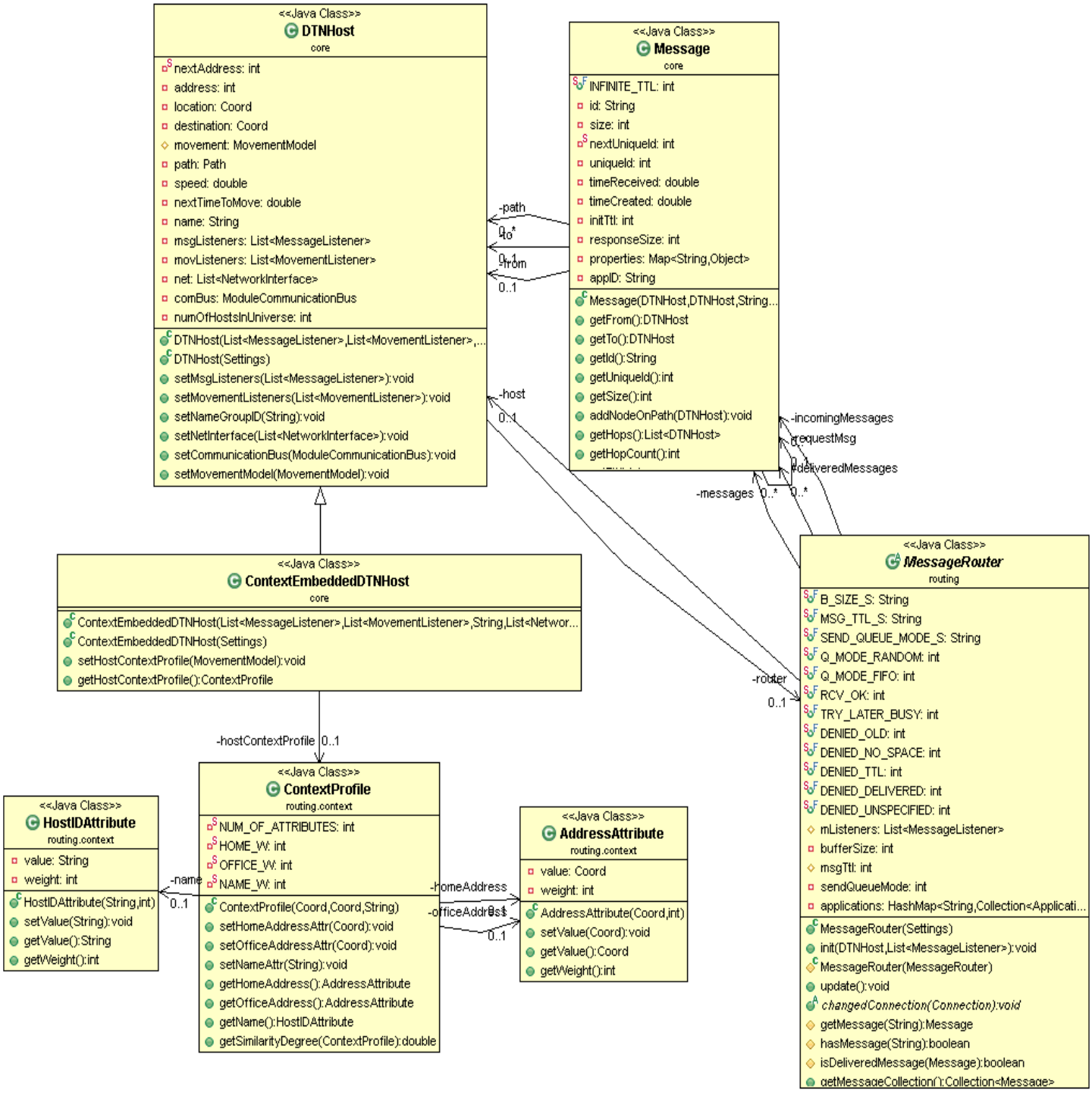


Figure 4.4 Routing class dependency

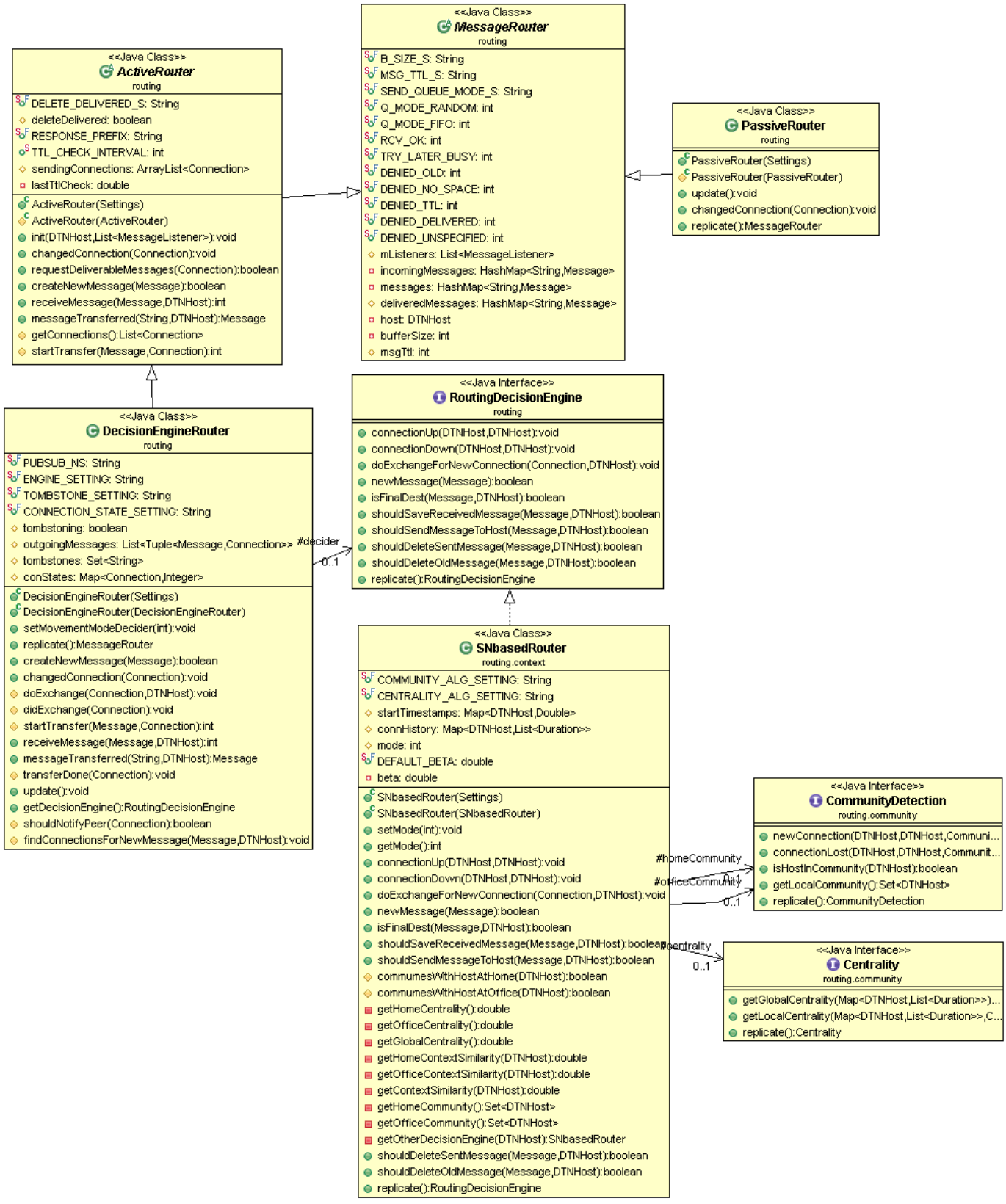
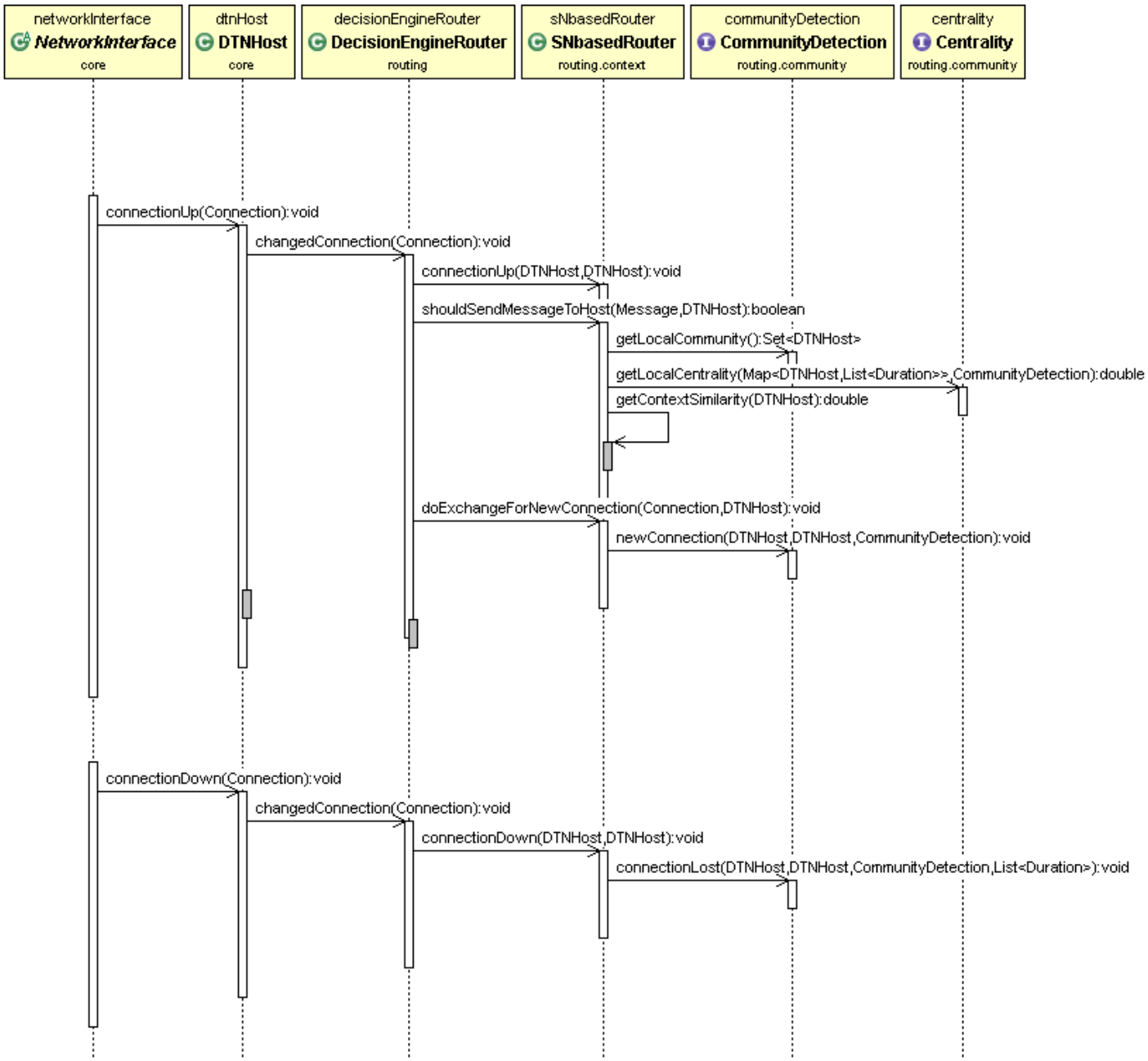


Figure 4.5 Routing decider sequence diagram



Chapter 5

Performance Evaluation

In this section, we present the evaluation of the SNbR method via the ONE simulator. We compare the performance of the SNbR method with other methods such as Epidemic [34], BubbleRap [16], and PROPHET [23]. The Epidemic algorithm is chosen because it is the basic method of message forwarding. It uses flooding and, therefore, is expected to have the highest delivery performance and short delays trading by a high overhead. The BubbleRap algorithm is chosen because it is a social network analysis-based method which uses only the information of the social network structure in the forwarding decisions. The PROPHET algorithm presents for probabilistic-based algorithms and context-aware algorithms because it has the same assumption with these algorithms using repeated mobility patterns observed with mobile nodes. To compare the performance of SNbR with other methods, we focus on three performance criteria: *delivery ratio, overhead* and *latency*.

5.1 Simulation settings

To evaluate the performance of SNbR and other protocols, we create the network simulations in the ONE simulator with the working day movement model [12] which simulates the everyday activities of people. We selected the working day movement model because the model has been validated against real user traces in terms of inter-contact times, contact durations and contacts per hour [12]. In the literature of DTNs, these measures are commonly used for characterizing connectivity in DTNs. The working day movement has also been used in some studies such as [38], [39]. In the following, we first provide a description of the working day movement model and how the simulation scenarios are created within the working day movement model. We then present the simulation parameters for the simulation scenarios.

5.1.1 Working day movement model

The working day movement model models the everyday life activities of people including going to work in the morning, spending the day at work, and commuting back home in the evenings. To model these activities, the working day movement model combines different movement model elements together. These elements are called submodels. On a more detailed level, the activities differ from each other. These submodels are repeated everyday, resulting in periodic repetitive movement.

The working day movement model allows the forming of communities and social relationships when a set of nodes are doing the same activity or in the same location. For example, nodes with the same home location are family members, and nodes with the same office location are work colleagues.

Mobile nodes (people) do the activities on a daily basis starting from home in the morning. Each node is assigned a wake-up time which determines when the node should start from home. The node uses the same wake-up time every morning during the whole simulation. At the wakeup time, nodes leave their homes and use different transport methods to travel to work. Nodes travel between activities either by car or by bus via different submodels. The working time is configurable. After the working hours, the nodes decide whether they go out for the evening activity or return home. Different submodels are used for transitions between the locations. Different user groups have different locations where the activities take place.

All nodes move on a map. The map defines the space and routes in which the nodes can moves. The map contains all the information of the locations of the houses, offices and meeting spots, as well as the bus routes and bus stops. The design of the map is an important part of the mobility model. Since all the movement of the nodes is determined by activities with specific locations, the placement of these locations define how nodes are moving on a larger scale, i.e., in which areas of the map, nodes will be doing different activities. The positions of these locations can be node group specific. The map can be used to limit node movement to small areas, which increases the locality. The houses and offices and meeting spots are spread randomly on the map, thereby, having very little locality and nodes meeting easily.

5.1.2 Simulation parameters

Table 5.1 Simulation parameters

Simulation area	10000 x 8000 m^2
Number of nodes	100
Number of buses	2
Number of offices	5
Number of meeting spots	2
Number of houses	20
Message event generator	Constant bit rate generator
Number of event sources	25
Message interval	3600s
Message size	[500KB,1MB]
Simulation time	430000s

The simulation scenarios use a city map of the city of Helsinki with the simulation area of 10000x8000 m^2, i.e. 80 km^2. There are 100 nodes and 2 buses in the simulation networks. The number of offices is 5 and the number of houses is 20. For the evening activities, the number of meeting spots is 2. All nodes have the Bluetooth network interface with the transmission speed of 250KBps and the transmission range of 10m. The nodes move based on the working day movement model described above.

To generate messages in the network, we use a constant bit rate message generator. The set of 25 senders is chosen uniformly at random at the beginning of the simulations. The

interval between the generations of two consecutive messages at the same sender was modeled as 3600 seconds.

The summary of the simulations parameters are as shown in Table 5.1. The details of the simulation settings scripts are presented in Appendix A including the default setting script which is Epidemic setting script, the SNbR routing script, the BubbleRap setting script and the PROPHET setting script.

5.2 Performance criteria

We compare the performances of SNbR with BubbleRap, PROPHET and Epidemic in the simulation networks using the following three performance criteria:

- *Message delivery ratio*: Ratio between the number of messages successfully delivered and the number of messages that were created in a given simulation experiment
- *Average latency*: the average amount of time taken for messages to be delivered successfully
- *Overhead* or *average number of duplicate messages*: the average number of messages that are still floating around the network at the time of the first delivery.

Based on the operations of the algorithms, we predict the performances of the algorithms as in Table 5.2. Epidemic is a flooding algorithm, and we expect it has a high delivery ratio with short delay, but these performances are traded with high overhead. PROPHET presents probabilistic-based routing and context-aware routing methods with the assumption of repeated mobility patterns of nodes. However, when such statistical information is not available, PROPHET goes back to the epidemic method. Therefore, we expect PROPHET has a high delivery ratio, short delay but with the overhead nearly equal to the overhead of Epidemic. BubbleRap is a social network analysis-based method. However, it works as a constraint-flooding method, which uses only the central nodes to make the replications of messages until the messages reach the destination's community. Consequently, we expect that BubbleRap has a high delivery ratio but does not have the latency as short as Epidemic and PROPHET, with a high overhead but still less than the overhead of the Epidemic and PROPHET methods. With SNbR, we expect a

high delivery ratio, a latency performance comparable to BubbleRap with a light overhead because SNbR is not exploited the flooding method at any part. We present the simulation results in the next section which confirm the performance predictions of the algorithms.

Table 5.2 Prediction of the performances of the algorithms

Routing algorithms	**Delivery ratio**	**Latency**	**Overhead**
Epidemic	High	Short	High
Prophet	High	Short	Nearly High
BubbleRap	High	Short-Medium	Medium
SNbR	**High**	**Short-Medium**	**Light**

5.3 Performance evaluation

We ran the simulation networks with the configurations in Section 5.1 against SNbR and the other algorithms BubbleRap, PROPHET and Epidemic. The simulations are repeated with different values of TTL: 1 hour, 6 hours, 12 hours, 1 day and 2 days. Figure 5.1 shows the comparison of the simulation results of delivery ratio between SNbR, BubbleRap, PROPHET and Epidemic. The results show that Epidemic has the highest delivery ratio among the algorithms. The delivery performance of PROPHET is less than that of Epidemic but higher than that of BubbleRap. SNbR has a comparable delivery performance with the other algorithms, especially SNbR does not use any form of flooding in operations. Figure 5.2 shows the average delivery ratio of the algorithms. Overall, SNbR delivers 79%, 76%, 75% as good as BubbleRap, PROPHET and Epidemic respectively. From the simulation results, it also shows that while the delivery ratios of BubbleRap, PROPHET, and Epidemic are not affected much with the different values of TTL. SNbR has better delivery ratios with longer TTLs, delivering nearly 80% of the

messages with TTL of 2 days compared to 90% of the messages delivered by Epidemic which is the algorithm with the best delivery ratio.

Figure 5.1 Delivery ratio

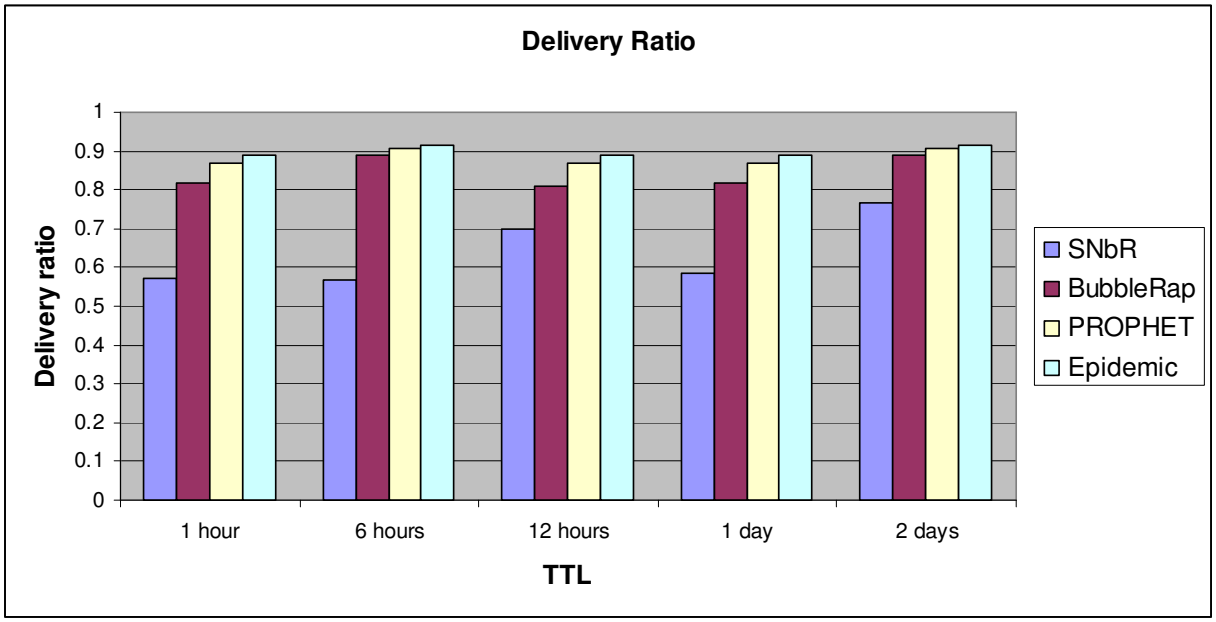
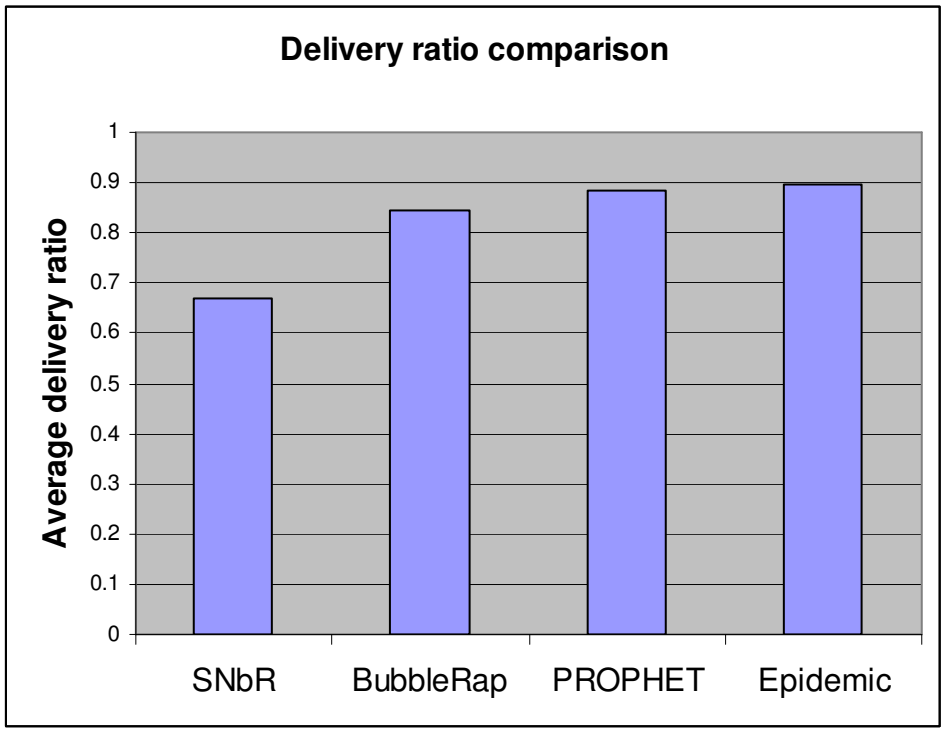


Figure 5.2 Comparison of delivery ratio



The simulation results of overhead of the algorithms are shown in Figure 5.3. SNbR produces much less overhead compared to BubbleRap, PROPHET and Epidemic. Overall, SNbR produces only 10.19%, 9.28% and 8.84% of the costs of BubbleRap, PROPHET and Epidemic respectively. Combining these results with the delivery performances of the algorithms (Figure 5.2), SNbR achieves a better performance in terms of delivery ratio against delivery cost compared to BubbleRap, PROPHET and Epidemic.

Figure 5.3 Overhead

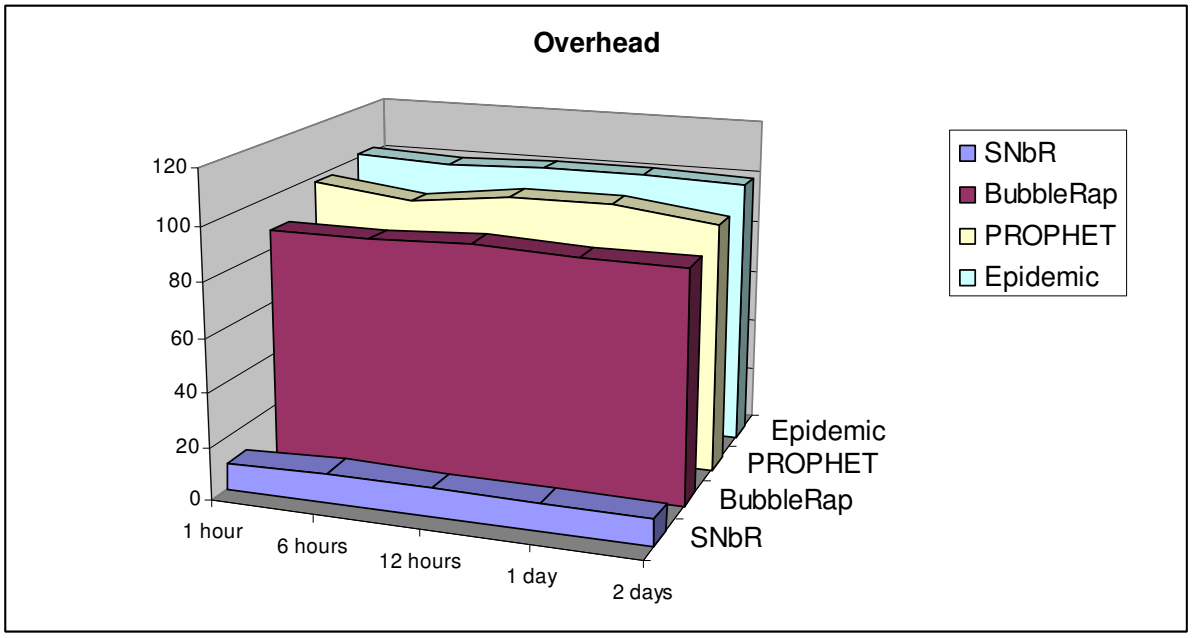
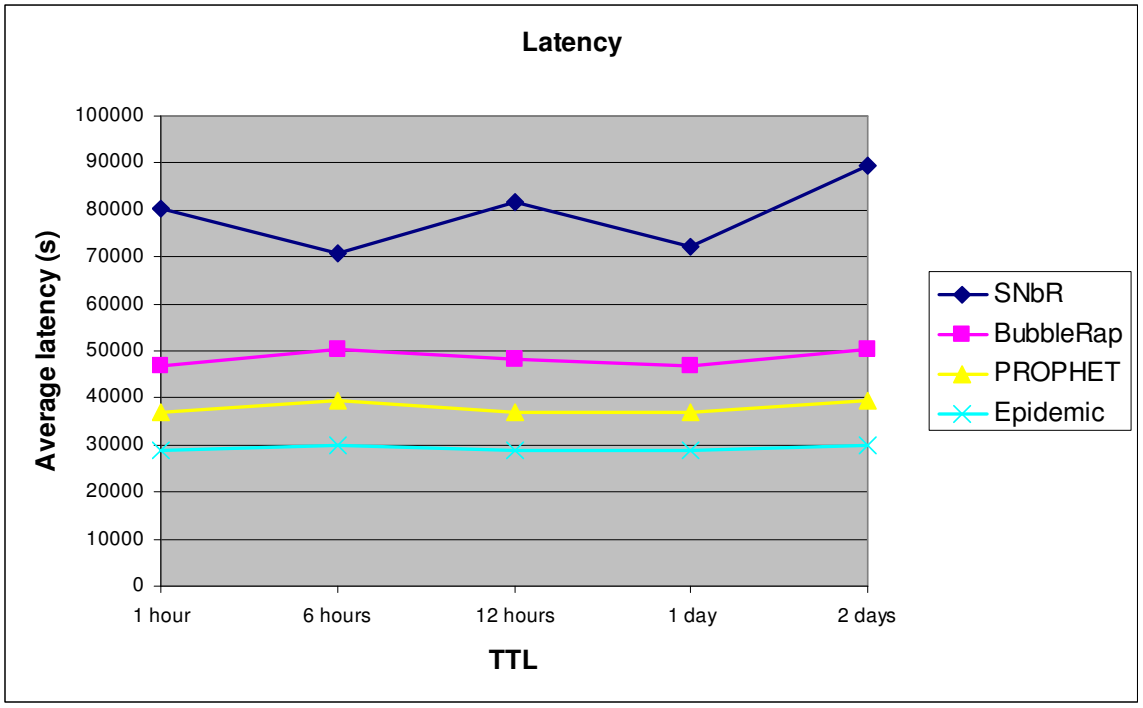


Figure 5.4 shows the average latency of SNbR, BubbleRap, PROPHET and Epidemic. As expected, Epidemic has the shortest latency. PROPHET has longer latency than Epidemic but shorter than BubbleRap. Because SNbR use fewer copies of messages in the network, it is expected that SNbR has higher latency compared to the other algorithms. However, by adding a replication policy in SNbR, SNbR can achieve a comparable latency with other algorithms.

Figure 5.4 Latency



Chapter 6

Conclusions

6.1 Summary

Supporting communications in DTNs has been attracted attention of researchers recently with the increasing popularity of personal mobile devices. Applications in DTNs demand a good performance from the message forwarding algorithms. Beside the high delivery ratio, one of the important factors is network overhead because resources are limited in DTNs. Current message forwarding algorithms in DTNs take different approaches to provide high delivery ratio while reducing the overhead of flooding-based algorithms. Probabilistic methods are based on the repeated mobility patterns of mobile nodes to predict the future encounters of nodes. However, to get to a node with a mobility pattern similar to the destination, probabilistic methods may need to apply flooding. Context-based methods provide a different way of estimating the repeated mobility patterns of nodes. However, they suffer the same problem of probabilistic methods which is the need to use flooding until messages reach nodes with similar context to the destination. With the present of social networks formed among mobile devices, utilizing the social network structure of nodes as in social network analysis-based methods provides a different way to attack the problem. However, because of the oblivious about the context of the destination, social network analysis-based methods only rely on the difference in the social popularity of nodes to propagate messages which is essentially a constrain flooding method.

In the thesis, we provide a message forwarding solution in DTNs called SNbR that combines the advantages of the social popularity of nodes and the context awareness in selecting relay nodes to forward messages. The goal of the SNbR forwarding algorithm is try to move the message closer to the destination. And the two important factors for choosing a node to achieve that goal are the ***social centrality*** of a node and the **context similarity** of the node to the destination. The simulation results show that SNbR achieves

a better performance in terms of delivery ratio against delivery cost compared to the other algorithms including BubbleRap representative for social network analysis-based methods, PROPHET preventative for context-aware methods and the flooding method Epidemic.

6.2 Limitations and Future work

In the future, SNbR can be improved in some of the following directions.

The detection and use of context in message forwarding in DTNs could be explored further in the future. Currently, using context attributes is simply based on the matching of values. More complicated methods could be used to compare the contexts. Besides, defining the specific format of context profiles for different scenarios and dynamic weighting of the context attributes will be also investigated in the future.

In the future, we also would like to perform the performance evaluation of SNbR with the real traces such as the mobility traces from the MIT reality mining project [40] and in more scenarios with different mobility models. The algorithm parameters would also be investigated further to see how the changes of the parameters would affect the algorithm performances.

To be applied in real systems, the security problems in SNbR including cooperation enforcement, encryption and robustness against the attacks needs to be address. First, cooperation is important in the operations of SNbR. To enforce the cooperation of nodes, some reward mechanisms can be proposed to acknowledge the contributions of nodes and motivate the cooperation of nodes.

Besides the contents of the messages, SNbR operations are based on the exchanges of encounter history and context of a node. Therefore, the privacy problem is an important issue and needs to be addressed. To ensure the privacy, confidentiality and integrity, some key management mechanism and encrypting schemes needs to be designed.

Protections against the attacks are also important. Methods need to be provided to prevent the security attacks to routing operations such as dropping packets, falsifying routing tables, and counterfeiting the context and centrality information.

Bibliography

[1] Ad Hoc Networking. 2001. Addison-Wesley Longman Publishing Co., Inc, Boston, Ma, Usa.

[2] Optimized Link State Routing Protocol (Olsr) Rfc. 2003.

[3] Blumstein, P. And Kollock, P. 1988. Personal Relationships. *Annual Review Of Sociology* 14, 467-490. .

[4] Boldrini, C., Conti, M., Jacopini, J. And Passarella, A. 2007. HiBOp: A History Based Routing Protocol For Opportunistic Networks. In *World Of Wireless, Mobile And Multimedia Networks, 2007. Wowmom 2007. Ieee International Symposium On A,* Anonymous , 1-12.

[5] Boldrini, C., Conti, M. And Passarella, A. 2008. Autonomic Behaviour Of Opportunistic Network Routing. *International Journal Of Autonomous Adaptive Communications Systems (Ijaacs)* 122-147. .

[6] Bulut, E., Zijian Wang And Szymanski, B.K. 2009. Impact Of Social Networks On Delay Tolerant Routing. In *Global Telecommunications Conference, 2009. Globecom 2009. Ieee,* Anonymous , 1-6.

[7] Burns, B., Brock, O. And Levine, B.N. 2005. MV Routing And Capacity Building In Disruption Tolerant Networks. In *Infocom 2005. 24th Annual Joint Conference Of The Ieee Computer And Communications Societies. Proceedings Ieee,* Anonymous , 398-408 Vol. 1.

[8] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R. And Scott, J. 2007. Impact Of Human Mobility On Opportunistic Forwarding Algorithms. *Ieee Transactions On Mobile Computing* 6, 606-620. .

[9] Daly, E.M. And Haahr, M. 2007. Social Network Analysis For Routing In Disconnected Delay-Tolerant Manets. In *Proceedings Of The 8th Acm International Symposium On Mobile Ad Hoc Networking And Computing,* Montreal, Quebec, Canada, Anonymous Acm, New York, Ny, Usa, 32-40.

[10] Dubois-Ferriere, H., Grossglauser, M. And Vetterli, M. 2003. Age Matters: Efficient Route Discovery In Mobile Ad Hoc Networks Using Encounter Ages. In *Proceedings Of The 4th Acm International Symposium On Mobile Ad Hoc Networking \& Computing,* Annapolis, Maryland, Usa, Anonymous Acm, New York, Ny, Usa, 257-266.

[11] Eagle, N., Pentland, A.S. And Lazer, D. 2009. Inferring Friendship Network Structure By Using Mobile Phone Data. *Proceedings Of The National Academy Of Sciences* 106, 15274-15278. .

[12] Ekman, F., Ker\Anen, A., Karvo, J. And Ott, J. 2008. Working Day Movement Model. In *Proceedings Of The 1st Acm Sigmobile Workshop On Mobility Models,* Hong Kong, Hong Kong, China, Anonymous Acm, New York, Ny, Usa, 33-40.

[13] Fall, K. 2003. A Delay-Tolerant Network Architecture For Challenged Internets. In *Sigcomm '03: Proceedings Of The 2003 Conference On Applications, Technologies, Architectures, And Protocols For Computer Communications,* Karlsruhe, Germany, Anonymous Acm, New York, Ny, Usa, 27-34.

[14] Freeman, L.C. 1979. Centrality In Social Networks Conceptual Clarification. *Social Networks* 1, 215-239. .

[15] Grossglauser, M. And Tse, D.N.C. 2002. Mobility Increases The Capacity Of Ad Hoc Wireless Networks. *Networking, Ieee/Acm Transactions On* 10, 477-486. .

[16] Hui, P., Crowcroft, J. And Yoneki, E. 2010. Bubble Rap: Social-Based Forwarding In Delay Tolerant Networks. *Mobile Computing, Ieee Transactions On* Pp, 1-1. .

[17] Hui, P., Yoneki, E., Chan, S.Y. And Crowcroft, J. 2007. Distributed Community Detection In Delay Tolerant Networks. In *Proceedings Of 2nd Acm/Ieee International Workshop On Mobility In The Evolving Internet Architecture,* Kyoto, Japan, Anonymous Acm, New York, Ny, Usa, 7:1-7:8.

[18] Johnson, D.B., Maltz, D.A. And Broch, J. 2001. Dsr: The Dynamic Source Routing Protocol For Multi-Hop Wireless Ad Hoc Networks. In *In Ad Hoc Networking, Edited By Charles E. Perkins, Chapter 5,* Anonymous Addison-Wesley, , 139-172.

[19] Ker\Anen, A., Ott, J. And K\Arkk\Ainen, T. 2009. The One Simulator For Dtn Protocol Evaluation. In *Simutools '09: Proceedings Of The 2nd International Conference On Simulation Tools And Techniques,* Rome, Italy, Anonymous Icst, New York, Ny, Usa.

[20] Kleinberg, J.M. 2000. Navigation In A Small World. *Nature* 406, 845-845. .

[21] Ko, Y. And Vaidya, N.H. 2000. Location-Aided Routing (Lar) In Mobile Ad Hoc Networks. *Wirel.Netw.* 6, 307-321. .

[22] Leguay, J., Friedman, T. And Conan, V. 2006. Evaluating Mobility Pattern Space Routing For Dtns. In *Infocom 2006. 25th Ieee International Conference On Computer Communications. Proceedings,* Anonymous , 1-10.

[23] Lindgren, A., Doria, A. And Schel\'En, O. 2003. Probabilistic Routing In Intermittently Connected Networks. *Sigmobile Mob.Comput.Commun.Rev.* 7, 19-20. .

[24] Marsden, P.V. 2002. Egocentric And Sociocentric Measures Of Network Centrality. *Social Networks* 24, 407-422. .

[25] Mcpherson, M., Smith-Lovin, L. And Cook, J.M. 2001. Birds Of A Feather: Homophily In Social Networks. *Annual Review Of Sociology* 27, 415-444. .

[26] Mei, A., Morabito, G., Santi, P. And Stefa, J. 2011. Social-Aware Stateless Forwarding In Pocket Switched Networks. In *Infocom, 2011 Proceedings Ieee,* Anonymous , 251-255.

[27] Nguyen, H.A., Giordano, S. And Puiatti, A. 2007. Probabilistic Routing Protocol For Intermittently Connected Mobile Ad Hoc Network (Propicman). In *World Of Wireless, Mobile And Multimedia Networks, 2007. Wowmom 2007. Ieee International Symposium On A,* Anonymous , 1-6.

[28] Ni, S., Tseng, Y., Chen, Y. And Sheu, J. 1999. The Broadcast Storm Problem In A Mobile Ad Hoc Network. In *Proceedings Of The 5th Annual Acm/Ieee International Conference On Mobile Computing And Networking,* Seattle, Washington, United States, Anonymous Acm, New York, Ny, Usa, 151-162.

[29] Okasha, S. December 2005. Altruism, Group Selection And Correlated Interaction. *The British Journal For The Philosophy Of Science* 56, 703-725. .

[30] Palla, G., Derenyi, I., Farkas, I. And Vicsek, T. 2005. Uncovering The Overlapping Community Structure Of Complex Networks In Nature And Society. *Nature* 435, 814-818. .

[31] Perkins, C.E. And Royer, E.M. 1999. Ad-Hoc On-Demand Distance Vector Routing. In *Proceedings Of The Second Ieee Workshop On Mobile Computer Systems And Applications,* Anonymous Ieee Computer Society, Washington, Dc, Usa, 90.

[32] Spyropoulos, T., Psounis, K. And Raghavendra, C.S. 2005. Spray And Wait: An Efficient Routing Scheme For Intermittently Connected Mobile Networks. .

[33] Su, J., Scott, J., Hui, P., Crowcroft, J., De Lara, E., Diot, C., Goel, A., Lim, M.H. And Upton, E. 2007. Haggle: Seamless Networking For Mobile Applications. In *Proceedings Of The 9th International Conference On Ubiquitous Computing,* Innsbruck, Austria, Anonymous Springer-Verlag, Berlin, Heidelberg, 391-408.

[34] Vahdat, A. And Becker, D. 2000. Epidemic Routing For Partially Connected Ad Hoc Networks. .

[35] Watts, D.J. And Strogatz, S.H. 1998. Collective Dynamics Of /`Small-World/' Networks. *Nature* 393, 440-442. .

[36] Watts, D.J., Dodds, P.S. And Newman, M.E.J. 2002. Identity And Search In Social Networks. *Science* 296, 1302-1305. .

[37] Wei-Jen Hsu And Helmy, A. 2006. On Nodal Encounter Patterns In Wireless Lan Traces. In *Modeling And Optimization In Mobile, Ad Hoc And Wireless Networks, 2006 4th International Symposium On,* April, Anonymous , 1-10.

[38] Doering, M., P\Ogel, T. And Wolf, L. 2010. Dtn Routing In Urban Public Transport Systems. In *Proceedings Of The 5th Acm Workshop On Challenged Networks,* Chicago, Illinois, Usa, Anonymous Acm, New York, Ny, Usa, 55-62.

[39] Karvo, J. And Ott, J. 2008. Time Scales And Delay-Tolerant Routing Protocols. In *Proceedings Of The Third Acm Workshop On Challenged Networks,* San Francisco, California, Usa, Anonymous Acm, New York, Ny, Usa, 33-40.

[40] Eagle, N. And (Sandy) Pentland, A. 2006. Reality Mining: Sensing Complex Social Systems. *Personal Ubiquitous Comput.* 10, 255-268. .

Appendices

Appendix A: Simulation Settings Scripts

A.1: Default settings script

```
Scenario.name = Epidemic-HostType-%%Group.hostType%%-TTL-
%%Group.msgTtl%%-SimulationTime-%%Scenario.endTime%%-NumOfOffice-
%%Group1.nrOfOffices%%-NumOfMeetingSports-%%Group1.nrOfMeetingSpots%%
Scenario.simulateConnections = true
Scenario.updateInterval = 1
Scenario.endTime = 430k
Scenario.nrofHostGroups = 2

Report.warmup = 11k
Group.msgTtl = [3600; 22k; 43k; 86k; 172k]
Group.hostType = DTNHost

## Interface-specific settings:
# type : which interface class the interface belongs to
# For different types, the sub-parameters are interface-specific
# For SimpleBroadcastInterface, the parameters are:
# transmitSpeed : transmit speed of the interface (bytes per second)
# transmitRange : range of the interface (meters)

# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10

########################################
## Common settings for all groups
Group.movementModel = MapBasedMovement
Group.router = EpidemicRouter
Group.bufferSize = 1000M

# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface

Group.transmitSpeed = 100k
Group.waitTime = 0, 0
```

```
# walking speeds
Group.speed = 0.5, 1.5

Group.workDayLength = 28800
Group.probGoShoppingAfterWork = 0.5

Group.officeWaitTimeParetoCoeff = 0.5
Group.officeMinWaitTime = 10
Group.officeMaxWaitTime = 100000
Group.officeSize = 100

Group.nrofHosts = 0

Group.timeDiffSTD = 7200
Group.minGroupSize = 1
Group.maxGroupSize = 50
Group.minAfterShoppingStopTime = 3600
Group.maxAfterShoppingStopTime = 7200

#####################################
Group1.groupID = n
Group1.waitTime = 0, 0
Group1.nrofHosts = 100
Group1.movementModel = WorkingDayMovement
Group1.busControlSystemNr = 8
Group1.speed = 0.8, 1.4
Group1.ownCarProb = 0.2
Group1.shoppingControlSystemNr = 8
Group1.nrOfOffices = 5
Group1.nrOfMeetingSpots = 2

Group2.groupID = BUSS
Group2.speed = 7, 10
Group2.waitTime = 10, 30
Group2.nrofHosts = 2
Group2.movementModel = BusMovement
Group2.routeFile = data/demo_bus.wkt
Group2.routeType = 2
Group2.busControlSystemNr = 8

## Message creation parameters
# How many event generators
Events.nrof = 1
# Class of the first event generator
Events1.class = MessageEventGenerator
# (following settings are specific for the MessageEventGenerator class)
```

```
# Creation interval in seconds (one new message every 25 to 35 seconds)
Events1.interval = 3600
# Message sizes (500kB - 1MB)
Events1.size = 500k,1M
# range of message source/destination addresses
Events1.hosts = 0,25
# Message ID prefix
Events1.prefix = M

# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed = 8372

# World's size for Movement Models without implicit size (width, height; meters)
MovementModel.worldSize = 10000, 8000
# How long time to move hosts in the world before real simulation
MovementModel.warmup = 43000

## Map based movement -movement model specific settings
MapBasedMovement.nrofMapFiles = 2

MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt

## Reports - all report names have to be valid report classes
# how many reports to load
Report.nrofReports = 1
Report.reportDir = reports
# Report classes to load
Report.report1 = MessageStatsReport

## Default settings for some routers settings
ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true

## Optimization settings
Optimization.connectionAlg = 2
Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true

## GUI settings

# GUI underlay image settings
GUI.UnderlayImage.fileName = data/helsinki_underlay.png
# Image offset in pixels (x, y)
GUI.UnderlayImage.offset = 64, 20
```

```
# Scaling factor for the image
GUI.UnderlayImage.scale = 4.75
# Image rotation (radians)
GUI.UnderlayImage.rotate = -0.015

# how many events to show in the log panel (default = 30)
GUI.EventLogPanel.nrofEvents = 200
# Regular Expression log filter (see Pattern-class from the Java API for RE-matching
details)
#GUI.EventLogPanel.REfilter = .*p[1-9]<->p[1-9]$
```

A2: SNbR settings script

```
Scenario.name = SNbasedRouting-HostType-%%Group.hostType%%-TTL-
%%Group.msgTtl%%-SimulationTime-%%Scenario.endTime%%-NumOfOffice-
%%Group1.nrOfOffices%%-NumOfMeetingSports-%%Group1.nrOfMeetingSpots%%

#Host settings as ContextEmbeddedDTNHosts
Group.hostType = ContextEmbeddedDTNHost

#Social networks-based routing

Group.router = DecisionEngineRouter

DecisionEngineRouter.decisionEngine = context.SNbasedRouter
DecisionEngineRouter.communityDetectAlg =
routing.community.KCliqueCommunityDetection
DecisionEngineRouter.K = 5
DecisionEngineRouter.familiarThreshold = 700

DecisionEngineRouter.centralityAlg = routing.community.CWindowCentrality
```

A3: Bubble Rap settings script

```
Scenario.name = BubbleRap-TTL-%%Group.msgTtl%%-SimulationTime-
%%Scenario.endTime%%-NumOfOffice-%%Group1.nrOfOffices%%-
NumOfMeetingSports-%%Group1.nrOfMeetingSpots%%

#BubbleRap routing

Group.hostType = DTNHost
Group.router = DecisionEngineRouter

DecisionEngineRouter.decisionEngine = community.DistributedBubbleRap
DecisionEngineRouter.communityDetectAlg =
routing.community.KCliqueCommunityDetection
```

DecisionEngineRouter.K = 5
DecisionEngineRouter.familiarThreshold = 700

DecisionEngineRouter.centralityAlg = routing.community.CWindowCentrality

A4: Prophet settings script

Test scenario using Prophet router and Points of Interest (POIs)

Scenario.name = PRoPHET-%%ProphetRouter.secondsInTimeUnit%%siu-TTL-
%%Group.msgTtl%%-SimulationTime-%%Scenario.endTime%%-NumOfOffice-
%%Group1.nrOfOffices%%-NumOfMeetingSports-%%Group1.nrOfMeetingSpots%%
Group.router = ProphetRouter

ProphetRouter.secondsInTimeUnit = 30

# Define POI data files
PointsOfInterest.poiFile1 = data/ParkPOIs.wkt
PointsOfInterest.poiFile2 = data/CentralPOIs.wkt
PointsOfInterest.poiFile3 = data/WestPOIs.wkt
PointsOfInterest.poiFile4 = data/shops.wkt

# Define probabilities for different groups selecting POIs from different POI files
Group1.pois = 1,0.3, 2,0.1, 3,0.1, 4, 0.1
Group2.pois = 2,0.3, 3,0.1