

A CENTRALITY-BASED HISTORY PREDICTION ROUTING PROTOCOL FOR OPPORTUNISTIC NETWORKS

by

Amarpreet Bamrah

B.E in Electroncis and Communication, Punjab Technical University, India, 2012

A Thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2016

©Amarpreet Bamrah 2016

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

A Centrality-Based History Prediction Routing Protocol for Opportunistic Networks

©Amarpreet Bamrah, 2016

Master of Science
in Computer Science
Ryerson University

Abstract

Opportunistic networks are a subclass of delay tolerant networks based on a novel communication paradigm that aims at transmitting messages by exploiting direct contacts among nodes, without the need of a predefined infrastructure. Typical characteristics of OppNet include high mobility, short radio range, intermittent links, unstable topology, sparse connectivity, to name a few. As such, routing in such networks is a challenging task since it relies on node cooperation. This thesis focuses on using the concept of centrality to alleviate this task. Unlike other nodes in the network, central nodes are more likely to act as communication hubs to facilitate the message forwarding. In this thesis, a recently proposed History-Based Prediction Routing protocol is redesigned using this concept, yielding the so-called centrality-based HBPR protocol. The proposed CHBPR is evaluated by simulations using the ONE simulator, showing superior performance compared to HBPR without centrality and the Epidemic protocol with centrality.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Isaac Woungang, for having guided me throughout my graduate studies. His continuous support, patience and guidance during the course of my thesis has been invaluable. I would also like to thank Dr. Sanjay K. Dhurandher for his support and help throughout my research work. It was a great privilege working with them.

Thanks should go to all my committee members for their fruitful discussions and comments on this work.

My love and thanks goes to my family for their continuous support, prayers and encouragement throughout my study. A special note of thanks to Ssowjanya Harishankar, Khuram Khalid, and Navati Jain for being constant source of support through my masters program.

Contents

List of Figures	vii
List of Tables	viii
List of Abbreviations	ix
1 Introduction	1
1.1 Motivation and Research Problem	1
1.2 Approach	3
1.3 Thesis Contributions	3
1.4 Thesis Outline	4
2 Background and Related Works	5
2.1 Background	5
2.2 Related Work	9
3 Methodology	14
3.1 Network Models	14
3.2 HBPR Protocol	16
3.3 Proposed CHBPR Protocol	21
3.4 C-Epidemic Protocol	23
4 Performance Evaluation	25

4.1	The ONE Simulator	25
4.2	Simulation Settings	27
4.3	Performance Metrics	28
4.4	Simulation Results	29
5	Conclusion	34
	Appendices	36
A	Source Code	36
B	Configuration File	40
	Bibliography	43

List of Figures

3.1	Example of (a) Contact graph, and (b) the associated social graph connecting 8 nodes	15
3.2	An example scenario of the HBPR protocol	20
3.3	Flowchart of the HBPR	21
3.4	Pseudo-code of the CHBPR protocol.	24
4.1	Screenshot of the ONE simulator [7]	27
4.2	Number of messages delivered under varying thresholds	29
4.3	Comparison of HBPR and CHBPR for varying number of nodes and TTL.	30
4.4	Comparison of CHBPR and C-Epidemic while varying nodes and TTL.	31
4.5	Comparison of RandomWaypoint vs Custom Mobility Model	33

List of Tables

3.1	Example of the History Table of a node.	17
3.2	Example of the Home Location Table of a node.	18
3.3	Centrality Table of a node.	22
4.1	Simulation Parameters	29

List of Abbreviations

CHBPR	Centralized History Based Routing Protocol
CMM	Custom Mobility Model
DTN	Delay Tolerant Networks
HBPR	History Based Prediction Routing Protocol
RWP	Random Waypoint Model
TTL	Time-to-Live
ONE	Opportunistic Network Environment
OppNet	Opportunistic Networks

Chapter 1

Introduction

1.1 Motivation and Research Problem

In recent years, the advent of wireless technologies and ubiquitous computing have raised the expectation of a seamless and cheaper communication between wireless- based computing devices anytime and anywhere. In this setting, wireless mobile ad hoc networks play a central role since they are (by design) specialized in the deployment of mobile self-configurable devices, with no infrastructure assistance. Due to this feature, these networks are appropriate to be used in several applications such as military operations, disaster relief and emergency preparedness and responses, health care monitoring, just to name a few. Opportunistic networks (OppNets) are considered as specialized ad hoc networks [1] characterized by frequent intermittent connections, in which messages are routed in a store-carry and forward manner based on locally inferred knowledge about the behavior of the nodes. Consequently, routing and message forwarding in oppNets constitute a challenging problem since the assumption of an established end-to-end path between the source and destination nodes does no longer prevail [2] as it is the case in conventional routing protocols for ad hoc networks.

Most routing protocols for OppNets rely on partial knowledge and the prediction of future contacts based on historical contact information [2]. They also rely on social-metrics,

which themselves are dependent on the past information of nodes. The participation of nodes in the message forwarding process is not guaranteed without incentivizing them since the communication and information exchange occur opportunistically between mobile nodes that happen to be in the same communication range and most nodes tend to be reluctant in sharing their private resources (such as buffer space) for public uses. One way, among others, to promote the node participation in message forwarding is to utilize the concept of centrality of nodes [3].

In an OppNet, due to the dynamics of node mobility, one can consider the influence that a node may have over the spread of information in relation to how many other nodes (encounters) this node may have been in contact with [3]. The centrality of a node is determined by its capability to contact other nodes and the cumulative contact duration between that node and its encounters within a given period of time, as well as its ability to ferry the messages between these encounters under a limited network transmission capability. Typically, a centrality metric is defined that takes into account the contact time between the encounters. Based on this definition, each node in the network is assigned a centrality value, and nodes with highest centrality values (called central nodes) at various given times are more likely to act as communication hubs that facilitate the message forwarding.

In this thesis, a recently proposed History-Based Prediction Routing protocol (HBPR) [4] is re-engineered using the concept of centrality as described in [5], yielding an enhanced routing scheme called centrality-based HBPR (CHBPR). The performance of CHBPR is evaluated by simulations and compared against that of the HBPR [4] and Epidemic routing [6] protocols, showing superior performance in terms of number of messages delivered and routing overhead ratio, under varying number of nodes and Time-to-Live. Details on these results are given in Chapter 4.

1.2 Approach

The CHBPR protocol is designed by re-engineering the HBPR protocol [4] as follows:

(1) The same assumptions, message generation, and data structures that are used in HBPR [4] are kept.

(2) In HBPR [4], a node selects its next hop to forward a message based on an utility metric, which takes into account the stability of node movement, the prediction of the direction of the future movement, and the perpendicular distance of the neighboring nodes from source to destination. However, when doing so, there is no guarantee that the next encountered node will be the possible best forwarder of the message with the highest probability to meet the destination node. To cope with this issue, other network knowledge such as the social characteristics of a node are considered in the design of CHBPR, leading to the introduction of a social metric of a node (so-called node centrality [5]) as an additional parameter involved in the decision making by a node when selecting its next hop.

More precisely, the centrality value is assigned to each node according to the PeopleRank concept [5]. In CHBPR, when two neighbor nodes i and j meet, they exchange their current centrality values as well as the information on the number of neighbor nodes that each of them have, then they update their centrality values for the next encountered node. The message is delivered from node i to node j if the ranking of node j is discovered to be greater than or equal to that of node i or if node j happens to be the destination node. If no neighbor node has centrality greater than the current node i , then the next hop to node i is determined according to the HBPR rule [4].

1.3 Thesis Contributions

The contributions of this thesis are as follows:

- Design of an enhanced HBPR (called CHBPR) for OppNets.

- Redesign of the Epidemic routing protocol based on centrality concept.
- Performance evaluation of the proposed CHBPR routing protocol using the ONE simulator [7], showing that it outperforms HBPR without centrality and the Epidemic routing protocol [6] with centrality, in terms of predefined performance metrics.

1.4 Thesis Outline

This thesis is organized as follows:

- **Chapter 1** introduces the subject, motivation, and contributions of our research.
- **Chapter 2** presents some background information and related works.
- **Chapter 3** describes the proposed CHBPR protocol in-depth.
- **Chapter 4** describes the performance evaluation of the proposed CHBPR protocol.
- **Chapter 5** concludes the thesis and highlights some future work.

Chapter 2

Background and Related Works

2.1 Background

This section discusses the main characteristics of OppNets, as well as routing in such networks. The concept of centrality in social networks is also introduced as well as how it can be used to boost the message forwarding and routing in oppNets.

2.1.1 Main Characteristics and Requirements of OppNets

2.1.1.1 With Respect to Topology

As stated earlier, An OppNet is a kind of challenged mobile multi-hop ad hoc network characterized by unpredictable topology, prolonged disconnections and partitions. A most common illustrative architecture of OppNet is that of a network made of a small set of people equipped with smart devices (such as PDA, cell phones, etc) forming a seed network, which has the capability to grow opportunistically during its operations, becoming a network of large size depending on the targeted application. The benefit of this growth is the opportunistic leveraging of the diverse resources originated from these new nodes. As such, the main requirements for OppNets include: minimal starting configuration, high interoperability in terms of communication of diverse devices, embedded highly heteroge-

neous hardware and software, frequent intermittent connectivity, and harvesting of diverse resources whenever needed.

2.1.1.2 With Respect to Applications and Services

In [8], Lilien et al. have categorized OppNets into three categories, namely opportunistic communications Networks (OCNs) that focuses solely on the communication capabilities of nodes in OppNets, opportunistic data dissemination networks (ODNs) that deal with dynamic interconnection of devices and networks for data propagation purpose, and opportunistic capability utilization networks (OCUNs) that include the first two categories to utilize the combined capabilities that they both offer. Most representative OCUNs applications include: (1) pocket switched networks (PSNs) - where occasional transmission opportunities and user mobility are utilized to carry the message to its intended destination, (2) autonomic networks or self-managing networks that have been designed to overcome the complexity of Internet, by allowing different types of networks (so-called regions) on the Internet to be interconnected in an automatic fashion, and (3) socio-aware community networks - with focus on human-to-human communications supporting the intermittent connectivity, for instance, by adopting the DTN paradigm.

2.1.1.3 With Respect to Data Dissemination

OppNets are typically made of small devices (nodes) that can communicate with each other over a wireless link (using a short range communication) [1]. Due to node mobility, nodes can connect and disconnect since they move in and out of their communication range. The OppNet model constitutes a one-hop wireless communication scheme where data dissemination among nodes occurs in a spontaneous and ad-hoc manner using a predesigned data sharing protocol. Equipped with such protocol, a node must have the capability to send and receive the data to/from other node in a direct communication range whenever the opportunity to recognize other nodes in its physical proximity arises. Since no prior assumption is made

about the network topology, the delivery probability of a node (i.e. the probability that a node delivers a message to its intended destination) is often evaluated based on a forwarding strategy that takes into account the limited time in which nodes are in the radio ranges of each other as well as other criteria such as battery life of nodes, history of contacts of nodes, social behavior of nodes, to name a few [9].

2.1.1.4 Routing Protocols for OppNets

Traditional mobile ad hoc networks (MANETs) and Internet routing and forwarding techniques are not necessarily suitable for OppNets since they implicitly assume that the network, even if sparse, is connected and an end-to-end path always exists between the source and destination nodes. OppNets are quite different from legacy ad hoc networks in that the network is often disconnected as a rule rather than as an exception to deal with. This means that a message is forwarded and routed in a store-carry-forward fashion over an opportunistic link, then gets buffered at the next hop until the next link in the path becomes available, and so on and so forth, until it reaches its final destination. This operational mode imposes a new model for routing, which consists of independent, local forwarding decisions, based on the current connectivity information and the prediction of a future connectivity. If a message cannot be delivered immediately, the best carriers are those nodes with the highest chance of successful delivery.

In OppNets, the requirements for routing include: characterizing how long the contact durations (i.e. transmission opportunities) of nodes are, and determining the inter-contact times i.e. how long the time gaps between transmission opportunities are. Longer inter-contact times imply a longer storage period, which in turn requires a careful queue management. Based on these requirements, several categories of routing for OppNets prevail [10]. These include routing protocols for OppNets whose time-evolving topology is deterministic or stochastic, routing protocols designed for infrastructure-based and infrastructure-less OppNets, context-aware routing protocols - where the knowledge of the context in which

nodes operate is used to identify the best next hop of a given node [10], to name a few. Based on the utilized context information, context-based routing algorithms for OppNets are further classified into context oblivious routing schemes, mobility-based routing protocols, and social context-based routing schemes [10]. The routing protocol proposed in this thesis belongs to the class of social context-based routing protocols for infrastructure-less OppNets.

2.1.1.5 Concept of Centrality

In a frequently disconnected network such as OppNet, providing a comprehensive characterization of how the information propagates between mobile nodes is a challenge. In [11], it has been reported that few selected nodes qualified as bridge nodes can be identified, which possess the capability to broker the locally available information exchange among the nodes that are disconnected. This identification relies on the so-called node's centrality characteristics, which measures its network location importance.

Typically, for an OppNet, a contact graph is designed and used to analyze the social ties among the nodes. Based on this analysis, social metrics on nodes are derived and then used to quantify the node's centrality. It should be noted that the node's centrality can also be derived from other means, for instance, by designing an opportunistic forwarding mechanism that uses social metrics as criteria to make the forwarding decisions. Commonly used social metrics [12] include degree centrality - which measures the number of direct ties maintained by this node with other nodes, closeness centrality - which is derived from the shortest path between a node and other reachable nodes, betweenness centrality - which measures how often a node belongs to the route that links other nodes [12], [13], to name a few.

Depending upon the social metric used, a node with high centrality value is qualified as central node (or socially popular node), i.e. a node that has the capability to connect more often to other nodes in the network. In other word, a central node is likely to be part of the routing paths involving many other others nodes that it has encountered. In this thesis, we

have considered the mechanism proposed in [5] for calculating the centrality of a node.

2.2 Related Work

A number of routing protocols for OppNets based on the centrality principle have been proposed in the literature. Few of those that are more relevant to the scope of this thesis are [5], [11], [14]- [20] as described in the sequel.

In [5], Mtibaa et al. introduced PeopleRank, a forwarding algorithm that makes use of some social properties of nodes to lower the number of message retransmitted when performing the message routing in OppNets. In their scheme, the social interaction between nodes (so-called PeopleRank values) and how frequent a node meets with its encounters (so-called number of neighbors of the node in the social graph), are the relevant information used to design the routing metric that determines the next forwarder of a node. Trace-driven simulations using both real and virtual human mobility traces have also been conducted to assess the effectiveness of the PeopleRank protocol in terms of number of retransmissions, showing its superiority over a number of benchmark routing protocols for OppNets.

In [3], the relative importance of a node with respect to a group of nodes is studied, leading to the design of a partial-centrality-based opportunistic forwarding scheme (so-called OFPC scheme) using the concept of graph spectrum. In their scheme, the computing strength of the relationship between nodes is formulated as a Decayed Sum Problem. Its space complexity is analyzed, helping to track the aforementioned relationship between a node and any other node. The knowledge of the node's partial centrality and the overlapping community structure are used to make informed forwarding decisions. Through trace-driven simulations, their proposed OFPC scheme is shown to outperform few benchmark message forwarding schemes in terms of delivery ratio, delay and cost. However, their scheme has not been tested on other applications such as recommender systems.

In [11], Daly and Haahr proposed a centrality-based routing protocol (called SimBet)

that makes use of the so-called ego network analysis [10] to achieve efficient data forwarding in disconnected delay tolerant networks. In their scheme, any node wishing to send a data packet to a destination uses a combination of nodes centrality and social similarity (as routing metric) to identify a suitable forwarder until that data packet eventually reaches its destination. Through trace-driven simulations, the proposed protocol has been shown to be superior to encounter-based routing protocols such as Epidemic [11] and ProPHet [12]. Although this protocol showed such promises, it is yet to be tested on other types of distributed systems, for instance, on those network architectures where no knowledge of the global network topology information is available.

In OppNets, human mobility and device-to-device contacts are exploited to opportunistically create data paths over time. In this process, efficient routing relies on the identification of some influential nodes which could act as relays for data forwarding purpose. In [14], Gao et al. proposed a semi local centrality metric-based mechanism and its associated message forwarding algorithm to effectively identify the influential nodes for message forwarding in OppNets. The notions of nearest and next nearest neighbor's mobility patterns are exploited to significantly lower the computational complexity of their scheme compared to betweenness centrality-based schemes, degree centrality-based schemes, and closeness centrality-based schemes. The effectiveness of their proposed scheme is demonstrated by trace-based simulations the employ real world mobility traces and synthetic mobility traces. However, the proposed incentive mechanism, although a promising one, has not been tested on various other benchmark routing protocols for OppNets.

In [15], Xu et al. investigated the problem of choosing the centrality metrics that can effectively characterize the capability of a node to act as a relay for message forwarding in OppNets (so-called influential nodes) and proposed a destination unaware of forwarding algorithm that relies on contact durations between nodes and node's popularity in the underlying social graph network. The efficiency of their centrality-based message forwarding strategy compared to that of few message forwarding schemes is demonstrated through trace-driven

simulations involving human mobility and reality mining datasets, showing higher message delivery throughput and lower forwarding cost. However, in their simulation experiments, the transmission contentions and scheduling policies in the MAC layer are not taken into consideration.

In [16], Nikolopoulos et al. studied by experiments the efficiency of centrality-based routing schemes by investigating their impact based on three factors: (a) the contact graph on which the calculation of the centrality values of nodes depends, (b) the greediness nature of the forwarding decisions, and (c) the relevance of the global network centrality compared to their egocentric counterparts. Five basic routing primitive of centrality-based schemes are evaluated, showing the conditions under which centrality-based routing schemes are expected to be weak. However, in their scheme, a graph representation is introduced, where the contact trace is now transformed into a contact-vertex graph, leading to a new problem that should be addressed before taking full advantage of their proposal, i.e. the problem of investigating the impact of weighted contact-vertex graphs on centrality-based routing schemes.

In [17], Dhurandher et al. proposed a centrality-based acknowledgement forwarding mechanism to enhance the performance of the History Based Routing Protocol for Opportunistic Networks (HiBOp) [2]. Their scheme relies on the principle that the most central node on the path followed by the message being transmitted from source to destination is identified and appended to the message itself and an ACK mechanism is provided to confirm to the source that a message has been successful delivered to its expected destination. Their scheme is shown to improve the message forwarding in HiBOp in terms of message loss rate. One of its design features is that it can be implemented on top of any known routing protocols for infrastructureless OppNets. By simulations, HiBOp with centrality was shown to outperform HiBOp and the MaxProp protocol in terms of buffer occupancy, traffic overhead, and latency. However, no trace-driven simulations was performed.

In [18], Vastardis et al. proposed a multi-phase opportunistic mechanism called SAM-

PhO for routing in distributed mobile social networks. Their scheme involves the design of a centrality-based forwarding phase where global centrality metrics (such as betweenness) are utilized to route the messages to predefined foreign communities till they reach their destination communities, themselves identified by means of a network of central nodes. The effectiveness of the proposed multi-phase protocol in terms of scalability, memory utilization and delivery rates in highly social scenarios was demonstrated by simulations using OMNeT++ [19]. However, trace-driven simulations have not been performed.

In [20], Hui et al. proposed a social-based forwarding mechanism for DTNs called BUBBLE Rap, in which metrics from social network analysis (SNA) such as node centrality, which is typically computed over contact graphs, are introduced to formally characterize the social information of a node. Based on this feature, whenever a node wishes to route a message within its community, its community is identified, and centrality values are assigned to the encountered nodes within that community as well as to other nodes globally across the network. Simulations have shown that the proposed scheme is comparable to PROPHET in terms of delivery ratio. However, the proposed protocol has not been tested on available real mobility traces.

In [21], Yoneki et al. proposed a socio-aware centrality-based overlay protocol that can act as a backbone for publish/subscribe communication. In their scheme, some characteristics of nodes interaction are identified from real human users connectivity traces. Based on these characteristics, communities are detected and their centralities are evaluated, leading to the design of a multi-point event dissemination strategy, where message exchange between the nodes operate by means of clustering algorithms. The efficiency of their scheme in terms of overlay construction for message dissemination is proven using a trace-driven discrete event emulator. However, their study is yet to incorporate various other centrality characteristics for communication.

In [4], Dhurandher et al. proposed a routing protocol for infrastructure less OppNets in which the nodes behavioral information is utilized as a means to find its best next hop for

routing purpose assuming that nodes behaves as in a human mobility model, and cooperate with non-malicious intent. Typically, each node maintains the following tables: (1) a Home Locations table based on which the locations most frequently visited and rarely visited are identified, helping in predicting its future location using a Markov predictor, and (2) a History Table that records the node's locations at different timestamps, helping in determining the stability of the node's movements. Based on this, a utility metric is employed to determine the most appropriate next hop for message forwarding.

Unlike previous routing protocols for OppNets, in HBPR, it is possible to determine the proximity of a node based on its history of movement. In this thesis, the HBPR protocol is re-engineered by introducing the concept of centrality of a node to boost the message forwarding and thereby the routing, yielding an enhanced HBPR protocol (so-called CHBPR).

Chapter 3

Methodology

3.1 Network Models

This section introduces the considered network model and social relationship model, upon which the centrality of any node is computed.

In an OppNet, for the message to be delivered from one mobile node (called source node) to another node (called destination node), several opportunistic communication must occur among a set K of intermediate nodes (so-called contacts), providing that such communication between two contacts is possible only if they are within the radio range of each other. This prompts to the question of modelling the evolution of contacts in OppNets. In our approach, this is realized by a time varying contact graph $G1(t)=(N,E(t))$, where N is the set of vertices with $K = |N|$, and $E(t)$ is the set of edges, and it is assumed that the network starts to operate at time t_0 and ends its operation at time T .

Also, in our approach, the social relationships that exist among the nodes is modelled by means of a non-time varying and finite undirected graph $G2=(N_s,E_s)$ where N is a subset of N_s . An edge (X,Y) in E_s means that there is a social interaction between nodes X and Y , for instance, expressed in some form of correlation such as maintaining over time a similar number of neighbors, or more direct contacts than nodes that do not exhibit a correlation.

Whenever a source node generates a message at a certain time t_0 for a destination node, any node in $G_1(t)$ can be a forwarder of that message in a store-carry-and-forward fashion.

Using the above settings, the routing paths between the source and destination nodes are computed as a sequence of contacts that are followed as they appear in the $G_1(t)$ graph. Such path is not unique and the trick is to be able to compute only those paths that are delay-optimal. In [5], an algorithm (called PeopleRank algorithm) is proposed to compute such paths, along with a method for dynamically calculating the centrality value of a node whenever it has an encounter. It should be recalled that in the graph $G_1(t)$, each node is assumed to represent a device owned by a human user, and an edge $e(t)$ is assumed to represent a contact during the time interval t . With respect to graph G_2 , it can be assumed that two nodes have a social relationship if they share some common interest or are declared friends. In this thesis, the computation of the centrality value of each node is inspired from the PeopleRank algorithm [5].

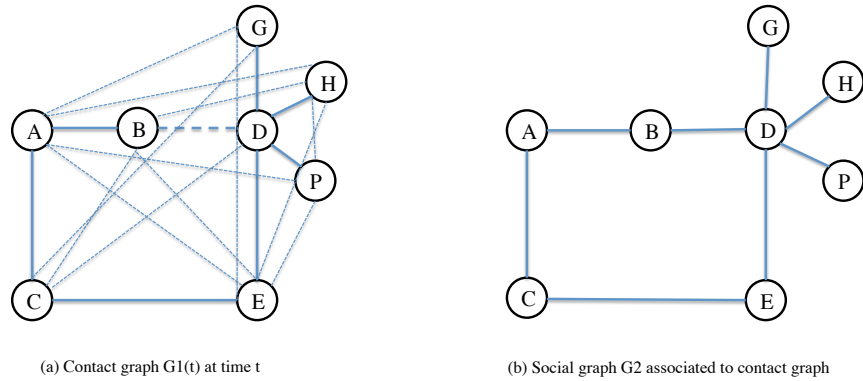


Figure 3.1: Example of (a) Contact graph, and (b) the associated social graph connecting 8 nodes

An example of contact graph and its associated social graph is depicted in Fig. 3.1. In

this contact graph, a solid line is used when nodes meet once every time unit, a dotted line is used when nodes meet once every 20 time units, and a dashed line is used when nodes meet once every 10 time units. Assume that node B want to send a message M to node P. By calculating the centrality of each node according to Equation (3.3) (as later described in Subsection 3.2.3 entitled Proposed CHBPR Protocol), we get $C_D > C_B > C_A$. This means that node B will have to wait until it encounters node D before forwarding the message M since node D has a higher centrality value compared to node B.

3.2 HBPR Protocol

3.2.1 Data Structures

For the design of the HBPR protocol [4], it was assumed that: (1) nodes in the network are not selfish and operate in a cooperatively manner; (2) the mobility of nodes is governed by a custom human mobility pattern [4]; (3) The simulation area is divided into cells, each of which is assigned a unique number representing a node location. If a node visits more frequently a cell, this is considered to be its home location. Based on these assumptions, the history of a node - i.e. its mobility pattern over a period of time, the direction of the node's movement, and the elapse of time taken by the node to meet an encounter, are the main parameters that are used to guide the message forwarding decision, using the following data structures.

3.2.1.1 History Table:

Each node in the network has a timer, which is set to 0 when the simulation starts (at this time, the node is at its home location). After a certain time period, say T, both the location (i.e. the node coordinate) and the node timer may have changed, leading to an entry of this table, in the form $\langle \textit{Timestamp}, \textit{Location} \rangle$. Without loss of generality, it is considered that a node can maintain a record of up to 100 entries and any eventual 101th entry

will be systematically deleted by the protocol in order to reflect the change in the node's movement behavior and preserve the memory storage. An example of a History Table where the simulation area includes cells numbered $\{23, 45, 45, 40, 45\}$ is depicted in Table 3.1.

Time	Location
0.45	23
1.2	45
2.6	45
2.9	40
3.3	45

Table 3.1: Example of the History Table of a node.

3.2.1.2 Home Location Table:

Whenever a node comes in contact with another one at a given time t , it keeps track of that node's home location ((i.e. $hostID$)) in the Home Location Table, in addition to keeping track of its own History of Table, leading to an entry of the form $\langle HostID, HomeLocation \rangle$ in the Home Location Table. At every encounter, the nodes that are involved share these tables; and in case the same entries are found, their timestamp fields are checked to determine if the entry should be updated or kept unchanged. This helps retaining only the most recent Home Location entry. An example of a Home Location Table where the simulation area includes cells numbered $\{68, 43, 45, 12, 6\}$ is depicted in Table 3.2.

3.2.2 How the HBPR Protocol Works

Using the above-mentioned data structures, the HBPR protocol operates as follows:

- Initially, all nodes flood their home locations so that each node is aware of the location of all other nodes in the network. In order for the History Table of the nodes to be

HostID	Home Location
A1	68
B1	43
H7	45
F3	12
A5	6

Table 3.2: Example of the Home Location Table of a node.

initialized, a head start is given to the network before the spread of the messages. This step is referred to as Home Location Initialization.

- At specific nodes, new messages are generated, each including the ID of the destination node. In their way to the destination, nodes are expected to frequently visit some locations and rarely visit some other locations. This information on the node movement is used to predict its future locations since if a change in the location of a node occurs, the said information will be flooded instantaneously to other nodes in the network so that they can update their Home Location Tables. This helps adjusting the changing relation and behavior of nodes. This step is referred to as Message Generation and Home Location Update.
- In order to determine the next hop of a node, a utility metric is designed, which involved three parameters: stability of node's movements, prediction of the direction of the future movement of the node, and perpendicular distance of the neighboring node from the line of sight of source and destination. Any node whose utility value is greater than a prescribed threshold (T) is assigned the message copy, with the hope that it will eventually reach the destination.

3.2.2.1 Predicting the Stability of Node Movements

Whenever a node moves, its location and time that recorded in its History Table are used to calculate its average speed over any two different positions, yielding a list of recorded average speeds for that node's journey. Based on this list, it can be determined if the change in average speeds is significant or nominal. In the former, the node movement is said to be unstable whereas in the latter, it is said to be stable. Without loss of generality, in our simulation settings, it is assumed that all nodes are initially assigned a stability value S ; and if the change in two consecutive average speeds is less than 10 units per second, the stability value does not change. Otherwise, the stability S is decreased according to the following equation [4].

$$S = S(1 - S) * S_i \quad (3.1)$$

3.2.2.2 Direction of Future Movement of Node

Based on the location information contained in the History Table of the node, a Markov predictor [4] is utilized to predict the direction of the future movement of that node, thereby its next location. In this case, the order of the predictor is considered to be the length of the past locations. Typically, for a given pattern of visits, a table recording the frequencies of visit for every location is maintained and the predictor uses it to determine the next location of the node. As an example, considering that the past history is

AGHBGTYGHIGHYKLOPWNGHGBKJDNHG

RJBFIJGHYKJFNHGYLKNKSGHWOKSADGH

the next location will be Y due to the fact it occurs the most as illustrated below

AGHBGTYG**H**IGHYKLOPWNG**H**GBKJDN**G**H

RJBFIJ**G**HYKJFN**G**HYLKNKSG**H**WOKSAD**G**H

3.2.2.3 Utility Metric

The perpendicular distance of the neighboring nodes from the line of sight of the source-destination (SD line) is a metric [4] used for selecting those nodes that are at closer distance to the SD line.

The utility metric $U(i)$ of node i is obtained as [4]:

$$U(i) = \sum_{j=1}^3 W(j) * V_i(j) \quad (3.2)$$

where $V_i(1)$ is the stability metric, $V_i(2)$ is prediction metric and $V_i(3)$ is the perpendicular distance metric for node i , and $W(j)$ is a weight parameter, $j=1,2,3$.

With this calculation, any node k whose utility value $U(k) \geq T$, where T is a prescribed threshold, can be a message forwarder.

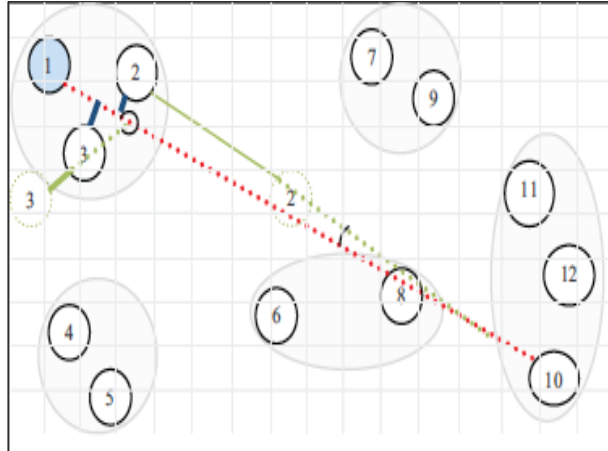


Figure 3.2: An example scenario of the HBPR protocol

An example of a network scenario [4] used to simulate the working of the HBPR protocol is shown in Fig. 3.2, where the SD line, perpendicular distance from the SD line, and future predicted locations of the nodes, are represented by the red line, blue line, and the dotted green circles respectively. Here, node 1 represents the source node (S) and node 10 represents the destination node (D), connected by an imaginary SD line. Also, nodes 2 and 3 are in the

radio range of the source node and their future predicted locations are shown. The future and present locations are connected by a straight line and the angle between the SD line and this straight line is shown, which can be calculated. The perpendicular distance of nodes 2 (resp. node 3) with the SD line is also shown using the dark blue line.

The flowchart of the HBPR is depicted in Fig.3.3

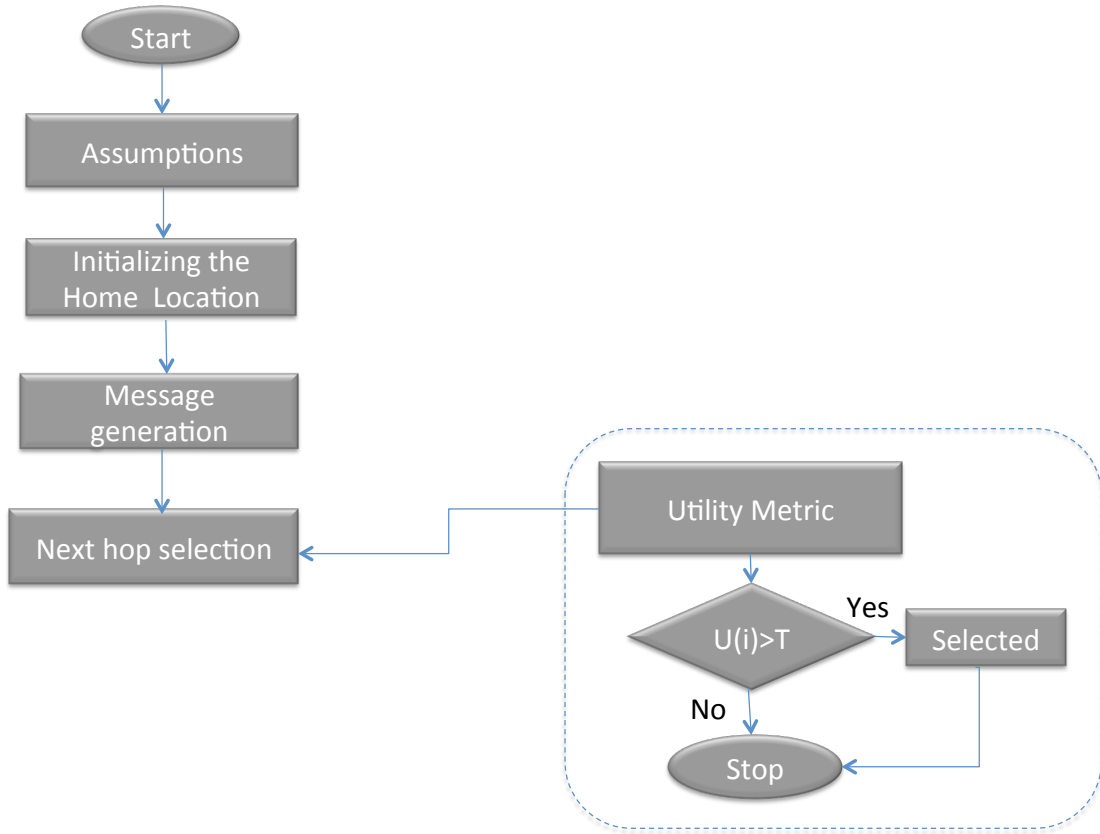


Figure 3.3: Flowchart of the HBPR

3.3 Proposed CHBPR Protocol

The CHBPR protocol inherits the same data structures and initialization settings that were utilized for the HBPR design [4]. In addition, each node maintains a new data struc-

ture called Centrality Table, which records its home location (i.e. hostID), the number of neighbor nodes it has (#neighbors), and the centrality values of all nodes it has encountered so far (denoted CentralityVal). Therefore, a record in such table is in the form $\langle HostID, centralityVal, \#neighbors \rangle$. Table 3.3 shows an example of a centrality table of a node. The CHBPR protocol is meant to reinforce the message forwarding part of the

HOST ID	Centrality Value	Number of neighbors
A1	3.3	3
B1	0.26	1
H7	7.9	5
F3	11	7
A5	4	2

Table 3.3: Centrality Table of a node.

HBPR protocol [4]. It works as follows: When two neighbour nodes (i,j) in the network's social graph come in contact, they exchange their centrality tables, from which two pieces of information are obtained: their current centrality values and the number of neighbors they have. Then, they update their centrality values according to the following equation inherited from [5]:

$$C(i) = (1 - d) + d \sum_{j \in F(i)} \frac{C(j)}{|F(j)|} \quad (3.3)$$

where $F(i)$ is the set of neighbors of node i , j is neighbor node in social graph and d is a damping factor defined as a probability that determines how much the social relationship between nodes can help improving their centrality values. For simulation purpose, d has been set to 0.87 as prescribed in [5]. A message is delivered from node i to node j if the centrality value j is greater than or equal to that of i or j is the destination node. If no neighbor node is found with centrality value greater than that of the current node, then the message forwarding is decided by the HBPR protocol [4]. The corresponding pseudo-code is shown in Fig. 3.4

3.4 C-Epidemic Protocol

Epidemic [22] was the first routing protocol designed for message forwarding in OppNets. In this protocol, the messages are flooded and infected among different nodes in the range of the source/intermediate node. Each node in the network maintains a so-called summary vector that contains the ID of each message. This summary vector is stored in its buffer. Whenever a node encounters another one, they exchanged their summary vectors and the messages that are not common to each other. In doing so, each node in the network can get the copies of messages from other nodes. That way, all the messages reach their destinations comparatively quicker with a high delivery probability. As reported in [22], the Epidemic scheme utilizes more bandwidth, more buffer space, and initiates a network congestion due to the generation of multiple copies of the same message. However, it has been shown to maximize the message delivery rate while minimizing the resources consumed when delivering the message. In this thesis, we have considered the Epidemic routing as benchmark scheme and have introduced the centrality concept in its core design, yielding the C-Epidemic routing protocol. The details of this code are provided in *Appendix A*.

Notations:

- $F(i)$: number of neighbors of current node i
- $|Y|$: cardinality of the set Y
- C_i : centrality value of node i
- $\text{buffer}(i)$: the buffer at node i
- M : message currently being sent
- DN : Destination Node
- $HBPR$: HBPR routing protocol

Require: $|F(i)| \geq 0$

```

while  $i$  is in contact with another node  $j$  do
  if  $j \in F(i)$  then
    if  $j$  is busy then
      select next node  $j=j+1$ 
    if  $j \in F(i)$  then
      send ( $C_i, |F(i)|$ )
      receive( $C_j, |F(j)|$ )
      update ( $C_i$ )using Equation 3.3
    end if
  end if
  while  $M \in \text{buffer}(i)$  do
    if  $j$  already has  $M$  then select next  $M=M+1$ 
    if  $C_j \geq C_i$  or  $j$  is equal to  $DN$  then
      Forward  $M$  to  $j$ 
    end if
  end if
  end while
end while
if (no node current centrality > neighbor centrality)
  follow HBPR for message forwarding
end if

```

Figure 3.4: Pseudo-code of the CHBPR protocol.

Chapter 4

Performance Evaluation

In this Chapter, the proposed protocol CHBPR is evaluated by simulations and compared against the HBPR and C-Epidemic protocols, where C-Epidemic is the Epidemic protocol with the same centrality concept used for HBPR embedded in it. We also studied the performance of CHBPR when the Random Waypoint model is used compared to the Custom Mobility Model. The simulator used is the Opportunistic Networking Environment (ONE) [7] version 1.5.1 RC2.

4.1 The ONE Simulator

The ONE simulator [7] is Java-based that has a broad range of delay tolerant network (DTN) protocols simulation capability. It supports mobility and event generation, DTN routing and application protocols, visualization and analysis interfaces, importing and exporting mobility traces, as well as basic notion of energy consumption at each node. Its main functions are the modeling of node movement, inter-node contacts, routing, and message handling.

A Node is a basic agent in the simulator. It is considered mobile such as a pedestrian, car or tram, with the required hardware. Each group of nodes is configured with different capabilities. Basic capabilities of a node are radio interface, persistent storage, movement, energy consumption and routing. On the other hand, some basic simulation parameters

associated with a node include its position, current movement path and current neighbors. These parameters allow one to implement context-aware specific algorithms such as the one proposed in this thesis.

In ONE [7], mobility models are meant to govern the way that nodes move in the simulation. They provide coordinates, speeds, and pause times for the nodes. A Few movement models have so far been included, namely, Random Waypoint movement, Map-constraint random movement, and Human behavior based movement, and Working Day Movement. In this thesis, we will use the Random Waypoint movement, as well as the Custom mobility model that have been introduced in [4].

With respect to routing, the routing modules are meant to define the route for the messages. The ONE simulator [7] comes with the implementations of some of the popular DTN protocols such as First Contact, Epidemic, Spray and Wait, Direct Delivery, PProPHET, and MaxProp. To add a new routing protocol in the ONE simulator, a new routing module should be designed, which inherits the basic functionalities of the MessageRouter module [7] (such as buffer management and callbacks for various related message events) and the ActiveRouter module - which deals with message transfers and aborts.

With respect to application support, there are two ways in the ONE simulator [7] to generate the application messages: using message generators or external event files. In the former, messages are created with random or fixed source, destination, size and interval whereas in the latter, messages are generated with specified message ID, fixed source-destination and at specific times.

With respect to reporting and visualization, a typical screenshot of the Graphical User Interface of the ONE simulator is shown in Fig. 4.1, where the main window depicts the node location, the current paths, and the nodes's radio range.

To start the simulation, there is pause/play button on the top left corner of the GUI. The right side of the GUI window displays the number of nodes in the simulation. In the event log section located at the bottom of the GUI window, the created connections and

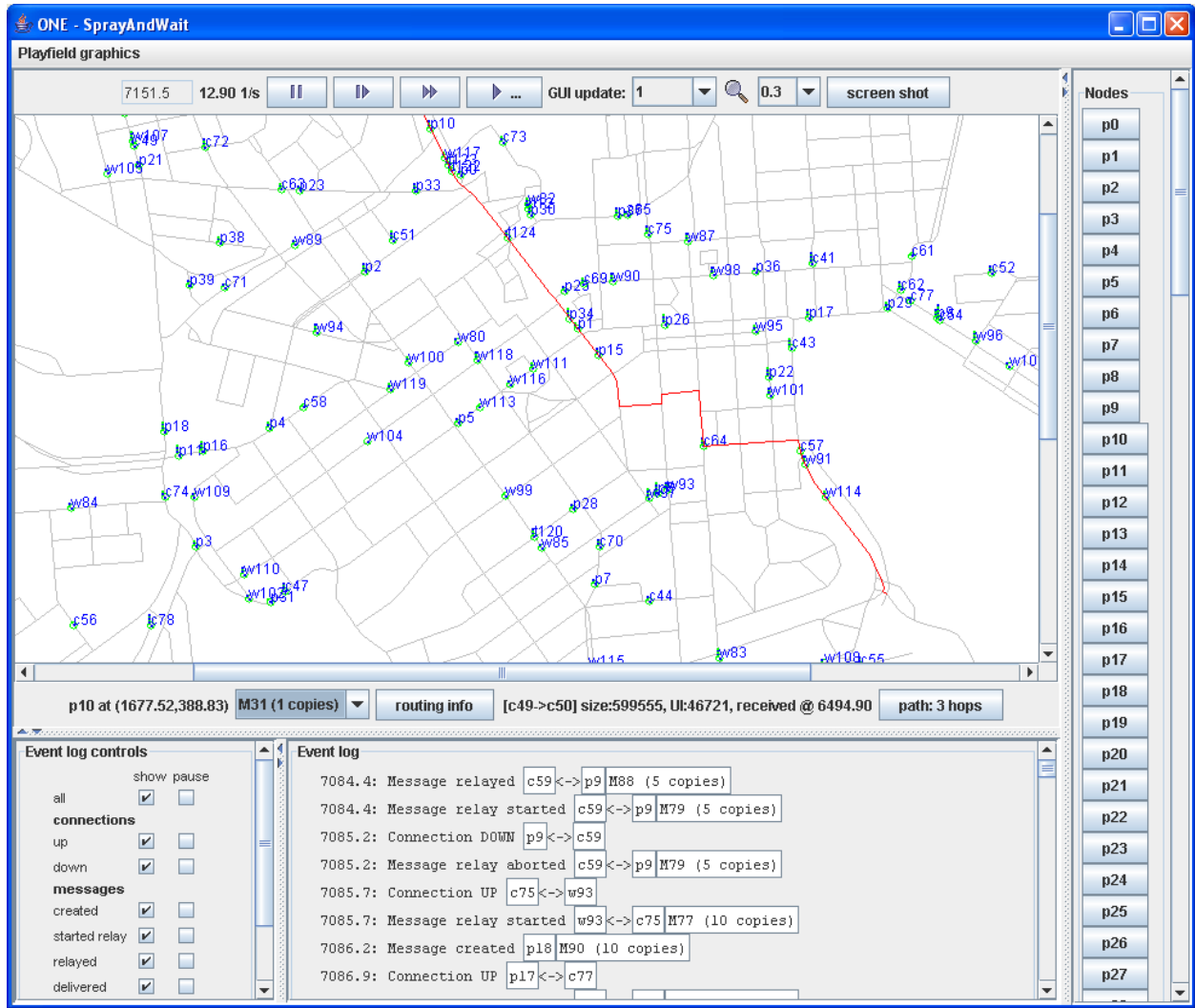


Figure 4.1: Screenshot of the ONE simulator [7]

message transmitted etc, are displayed. A statistic report module is also included in the ONE simulator [4] to help gathering the statistics on the performance such as Delivery ratio, Message Delivered, Overhead ratio, to name a few.

4.2 Simulation Settings

To simulate the proposed protocols for OppNets, the HBPR routing protocol [4] has been replicated and coded in Java. The custom mobility model movement model [4] and the

human mobility model are considered. The nodes are mobile and are grouped into six communities, each having 25 nodes. Each community also has a Human Location cell of 100x100m meters. The first and third group of nodes are groups of pedestrians, with a speed between 0.5-1.5 m/sec. The second group is a group of cyclists with a speed varying between 2.7-13.9 m/sec. The fourth, fifth and sixth group of nodes are cars with speeds varying between 7-10m/sec. The mobile nodes have a transmission range of 10 meters and a transmit speed of 2 Mbps. After every 25-35 seconds, a new message is generated, with the size varying from 500KB to 1MB. The weights given to the parameters in the Utility function of HBPR [4] are $W(1) = 0.4$ for the stability metric, $W(2) = 0.4$ for the prediction metric $W(2) = 0.4$, and $W(3) = 0.2$ for the perpendicular metric.

4.3 Performance Metrics

The performance metrics used to evaluate the developed protocols are: message delivered and overhead ratio.

- *Message Delivered*: This represents the total number of messages received by the destination. Mathematically, it is defined as:

$$\text{Message Delivered} = \frac{\text{Total number of packets received by destination}}{\text{Total number of packets send by the source}} \quad (4.1)$$

- *Overhead ratio*: This is a metric that helps measuring the additional number of packets needed by the routing protocol to actually deliver the data packets. It can be defined as:

$$\text{Overhead ratio} = \frac{\text{Number of packets relayed} - \text{Number of packets delivered}}{\text{Number of packets delivered}} \quad (4.2)$$

The various simulation parameters used are shown in Table 4.1

Simulation Parameter	Value
Mobility Model	Human Mobility Pattern (Variable)
Routing Protocol	HBPR, CHBPR, C-Epidemic
Number of Nodes	240 (Variable)
Simulation Time	43200 secs
Simulation Area	4500X3500m
Node Speed	0.5-13.9 m/sec
Threshold	0.6
TTL	300 min (Variable)

Table 4.1: Simulation Parameters

4.4 Simulation Results

This section describes the simulation results obtained when evaluating the studied routing protocols.

4.4.1 Varying Thresholds

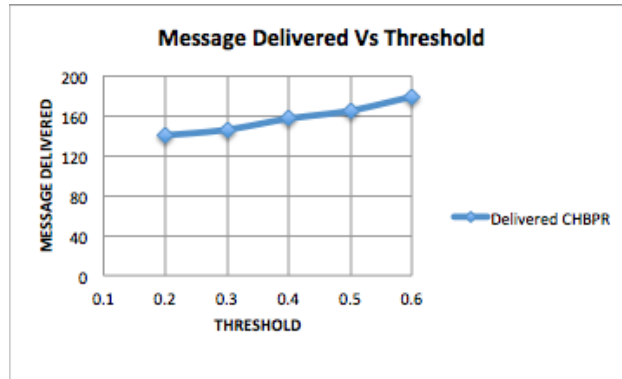


Figure 4.2: Number of messages delivered under varying thresholds

The threshold T is varied and the impact of this variation on the number of messages is investigated. The results are captured in Fig. 4.2. In Fig. 4.2, it can be observed that with the increase in the threshold, the number of messages delivered is gradually increased. This is attributed to the fact that CHBPR selects the node that has the highest utility

metric to relay the message. The increase in the number of messages delivered shows that lesser messages get dropped and flooding is controlled. For the following set of simulation experiments, we will use the threshold $T = 0.6$ as per the recommendation made in [4].

4.4.2 Comparison of HBPR vs CHBPR

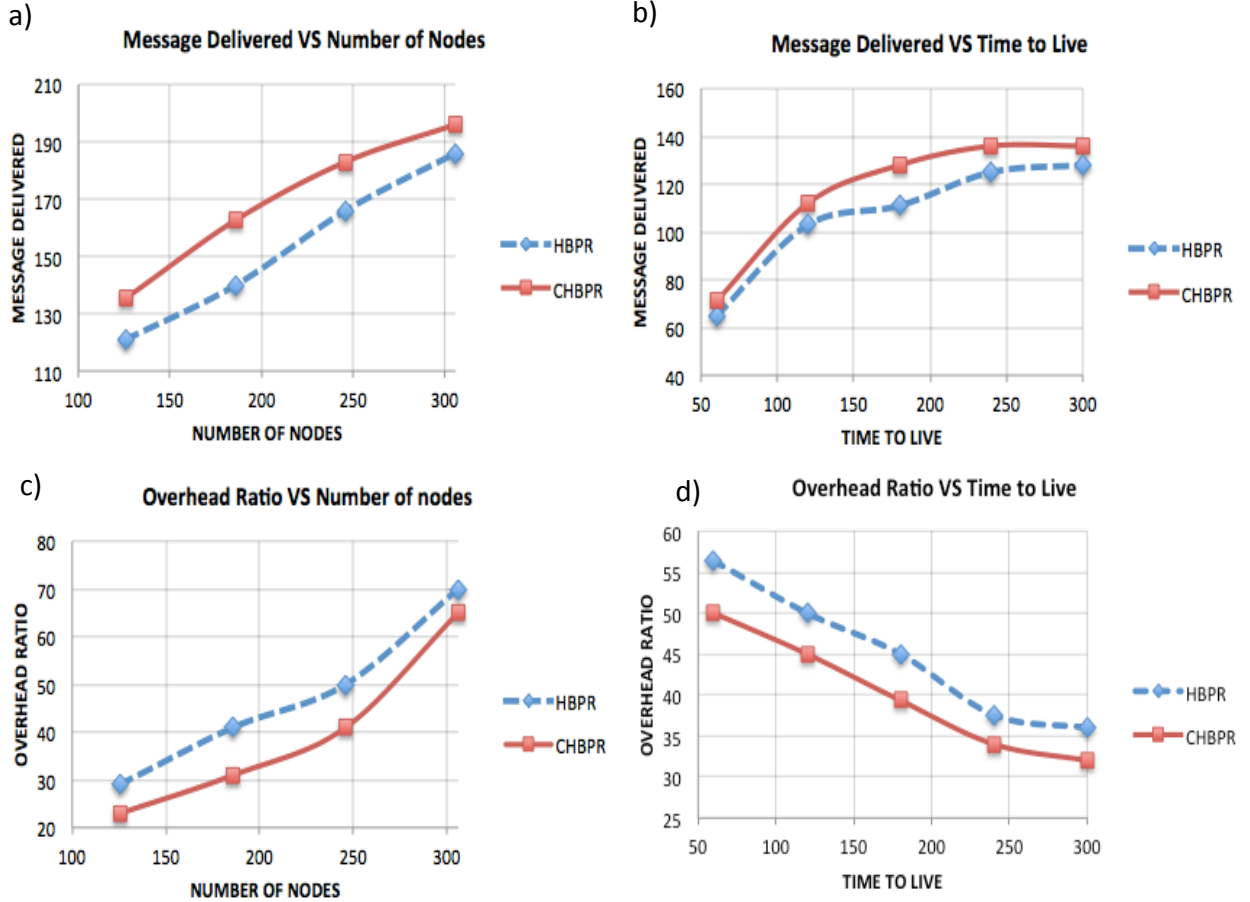


Figure 4.3: Comparison of HBPR and CHBPR for varying number of nodes and TTL.

The CHBPR algorithm is compared against the HBPR algorithm using the aforementioned performance metrics, by varying the number of nodes and varying Time-to-live (TTL). The results are depicted in Fig 4.3.

In Fig. 4.3a) and Fig. 4.3b), it can be observed that the number of messages delivered using CHBPR is higher than that obtained using HBPR. Also, in Fig. 4.3c) and Fig. 4.3d), it

is observed that the overhead ratio obtained for CHBPR is significantly lower than obtained using HBPR. These results are attributed to the fact that compared to HBPR, CHBPR is likely to yield a better forwarding path to deliver the message to the destination since such path is built with nodes that are likely characterized by high centrality values compared to nodes that are not part of the routing paths.

4.4.3 Comparison of CHBPR vs C-Epidemic

The CHBPR is compared against the C-Epidemic. The results are depicted in Fig 4.4. In Fig

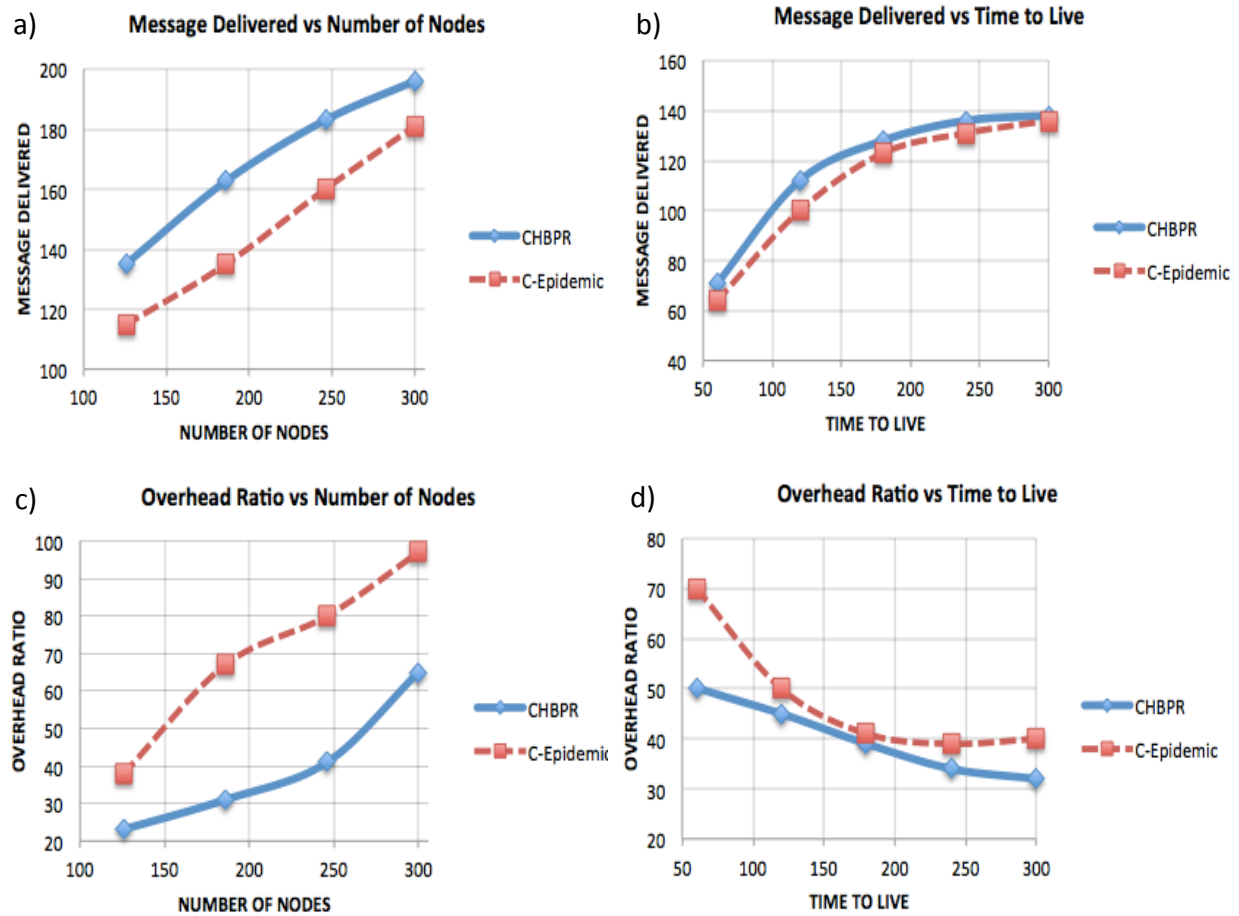


Figure 4.4: Comparison of CHBPR and C-Epidemic while varying nodes and TTL.

4.4a) and Fig 4.4b), it can be observed that CHBPR delivers more messages to destination compared to C-Epidemic. This result is also attributed to the fact that in CHBPR, the

routing path is constructed by nodes that likely have higher centrality values or better utility values. The results obtained with C-Epidemic can be justified by the fact that using only socially centralized nodes does not necessarily guarantee a good network performance. Fig. 4.4c) and Fig 4.4d), it is observed that C-Epidemic yields a higher overhead ratio under varying number of nodes and varying TTL. This is mainly attributed to the fact that C-Epidemic follows a blindfold forwarding mechanism, therefore a large amount of network resources are consumed when making the number of copies of the message to be sent.

4.4.4 Comparison of Random Waypoint vs. Custom Mobility Models

We have studied the performance of CHBPR when using the Random Waypoint model (RWP) vs. the Custom Mobility model (CMM). The results are depicted in Fig. 4.5

In Fig.4.5a), it can be observed that when the number of nodes increases, the message delivered also increases. But, this increase is more pronounced when the CMM is used compared to the RWP. This is due to the fact that in the CMM, the nodes visit their home locations more frequently and follow a specific pattern to reach the destination whereas in RWP, node moves randomly. In Fig. 4.5b), it can be observed that with the increase in TTL, the messages delivered also increases. This is attributed to the fact that when using CMM, the messages remains in the buffer for a longer period of time, leading to a higher chance to meet the destination. In Fig. 4.5c), it is observed that when the number of nodes increases, the overhead ratio also increases. In Fig. 4.5d), it is observed that when the TTL, the overhead ratio decreases. These observations are again due to the fact that the messages reside in the buffer for a longer period of time and are not dropped on their way to destination, leading to a limited number of retransmissions of a single message, therefore to a low overhead ratio.

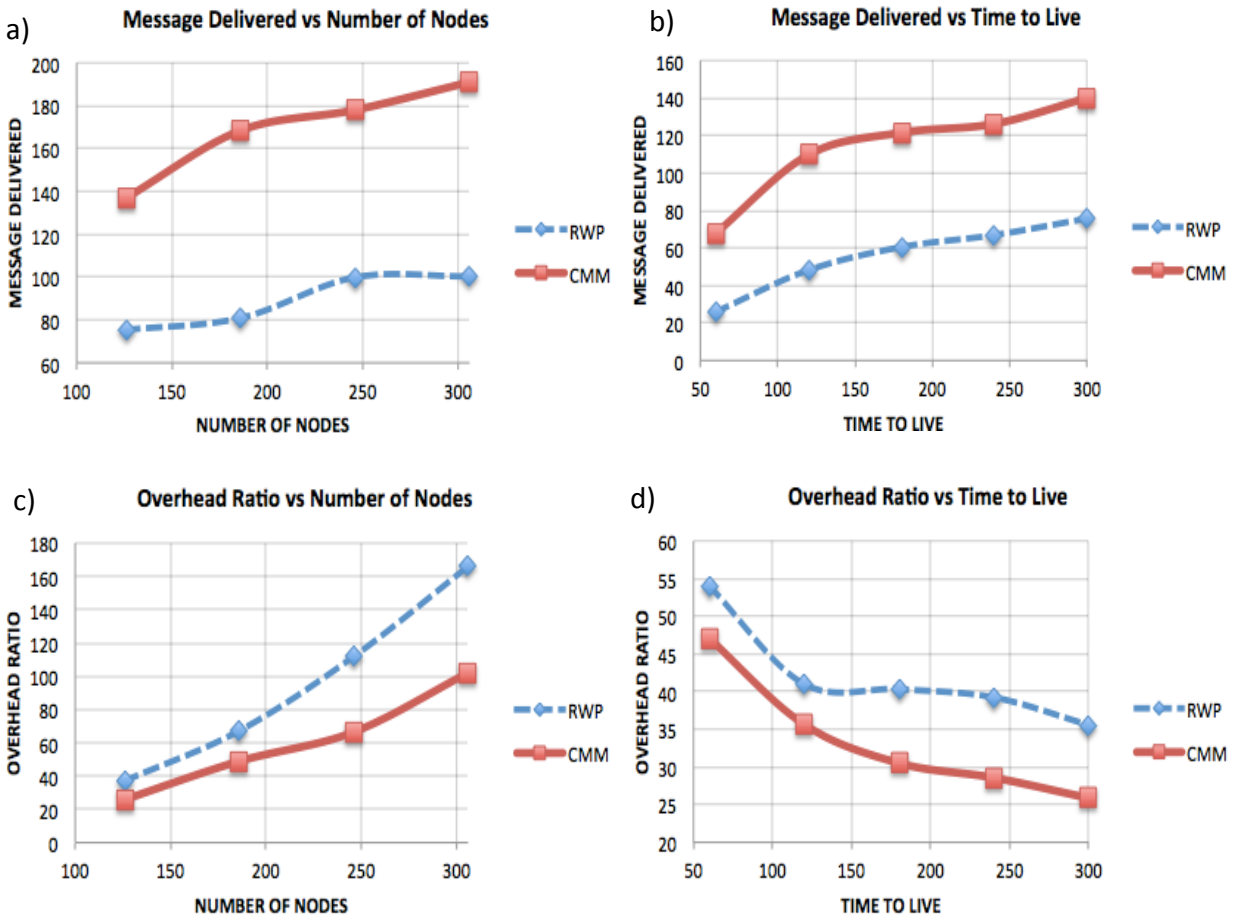


Figure 4.5: Comparison of RandomWaypoint vs Custom Mobility Model

Chapter 5

Conclusion

In this thesis, an enhanced version of the HBPR protocol for OppNets is proposed (so-called CHBPR) which uses the centrality concept to boost the message forwarding process in HBPR. Simulation results on the performance of CHBPR in comparison with C-Epidemic, chosen as benchmark protocol reveal that:

1. CHBPR significantly outperforms HBPR in terms of message delivered and overhead ratio under varying number of nodes and TTL. Due to this, it can be concluded that in the case of CHBPR, a node that is likely to participate in the message routing makes an intelligent decision in choosing the next best node to forward the message.
2. The C-Epidemic performance falls off in terms of message delivered and overhead ratio when compared against that of CHBPR. Due to this, it can be concluded that using only socially connected nodes is not enough to guarantee a good performance.
3. CHBPR under the CMM outperform CHBPR under RWP in terms of number of messages delivered and overhead ratio. Due to this, it can be concluded that CHBPR is likely to perform well when using communities such as human scenarios, and structures that have recurring pattern.

As future work, we plan to make the proposed CHBPR protocol energy efficient. Security is another aspect that deserve further attention. Finally, experimenting the simulation of CHBPR using realistic mobility traces is desirable.

Appendix A

Source Code

```
//=====Calculation of node'neighbors=====//
public int number_Of_Neighbours(CHBPR router, int loc) {
    Collection c = router.home.values();
    Iterator i = c.iterator();
    int size_of_N = 0;
    while(i.hasNext())
    {
        if((int)i.next() == loc)
        {size_of_N++;}
    }
    return size_of_N;
}
//=====Algorithm end=====//
//=====Calculation of PeopleRank=====//
public double update_centrality(Double J_Centrality, int j_neighbours) {
    double centralityi = 0.0d; //initialize centrality value
    double d = 0.87;
    centralityi = (1-d)+d* (J_Centrality/j_neighbours);
    return centralityi;
}
//=====Algorithm end=====//
//=====Send and Receive Centrality=====//
public void send_Centrality(CHBPR jn_router ,double ICentrality, int Neigh_i) {
    ArrayList<Object> cent_values = new ArrayList<Object>(2);
    cent_values.add(ICentrality);
    cent_values.add(Neigh_i);
    jn_router.cent_Table.put(this.getHost().toString(), cent_values);
}
public ArrayList<Object> recieve_cent(CHBPR jn_router,int loc) {
    double cent = jn_router.get_centrality();
    int Neigh_j = jn_router.number_Of_Neighbours(jn_router, loc);
    ArrayList<Object> cent_values = new ArrayList<Object>(2);
    cent_values.add(cent);
    cent_values.add(Neigh_j);
    this.cent_Table.put(jn_router.getHost().toString(), cent_values);
    return cent_values;
}
//=====Algorithm end=====//
```

```

List<Object> cent_values = new ArrayList<Object>();
public Map<String,ArrayList> cent_Table; // Centrality Table

//=====Algorithm for CHBPR =====//
int Neigh_I = this.number_Of_Neighbours(this, loc);
if(Neigh_I >= 0) //Step 1: Require:  $|F(i)| \geq 0$ 
{
    for (Connection con : getConnections()) { // Step 2: while i is in contact with
                                                another node j do
        DTNHost jn = con.getOtherNode(getHost()); //Neighbour Node j is one hop
                                                neighbour of current node
        CHBPR jn_router = (CHBPR)jn.getRouter();

        if(jn_router.isTransferring()) //if j is busy then go to step 2
        {
            continue;
        }
        if(home.containsKey(jn_router.getHost())) // if j is in F(i)
        {
            this.send_Centrality(jn_router,this.get_centrality(), Neigh_I ); //send  $(C_i, |F(i)| )$ 
            j_cent_values = this.receive_cent(jn_router, loc); //receive  $(C_j, |F(j)|)$ 
            this.centrality = this.centrality +
            this.update_centrality(((double)j_cent_values.get(0),(int)j_cent_values.get(1));
            //update  $(C_i)$  using Equation 3.3
        }
        double j_centrality = (double)j_cent_values.get(0);
        for(Message m: msgCollection) //Step 3: while M buffer(i) do
        {
            if (jn_router.hasMessage(m.getId())) //if j already has M then go to step 3
            {continue;}
            DTNHost dest = m.getTo();
            if(j_centrality >= this.centrality || jn_router.getHost().equals(dest))
            //if  $C_j \geq C_i$  or j is equal to DN then
            {
                messages.add(new Tuple<Message,Connection>(m,con));
                //Forward M to j
            }
        }
    }
}
// Step 4: if (no node current centrality > neighbor centrality)

// follow HBPR for message forwarding

//=====Algorithm End =====//

```

```

/=====Epidemic embedded with Centrality=====/
public class EpidemicRouter extends ActiveRouter {
    public List<Object> cent_values = new ArrayList<Object>();
    public Map<String,ArrayList> cent_Table; // Centrality Table
    private double centrality;
    private double th;
    public EpidemicRouter(Settings s) {
        super(s);
        //mycode
        this.cent_values = new ArrayList<Object>(2);
        this.cent_Table = new LinkedHashMap<String,ArrayList>(100)
        {
            protected boolean removeEldestEntry(Map.Entry eldest)
            {
                return size() > 100;
            }
        };
        this.centrality = 0.0;
        Settings s = new Settings();
        th=s.getDouble("Threshold");
    }
    protected EpidemicRouter(EpidemicRouter r) {
        super(r);
        //TODO: copy epidemic settings here (if any)
    }
    @Override
    public void update() {
        super.update();
        if (isTransferring() || !canStartTransfer()) {
            return; // transferring, don't try other connections yet
        }
        // Try first the messages that can be delivered to final recipient
        if (exchangeDeliverableMessages() != null) {
            return; // started a transfer, don't try others (yet)
        }
        // then try any/all message to any/all connection
        this.tryAllMessagesToAllConnections();
    }
    protected Connection tryAllMessagesToAllConnections(){
        List<Connection> connections = getConnections();
        List<Connection> sendCons = new ArrayList<Connection>();
        if (connections.size() == 0 || this.getNrofMessages() == 0) {
            return null;
        }
        int Neigh_I = this.number_Of_Neighbours(this, loc);
    }
}

```

```

if(Neigh_I >= 0) {
for (Connection con : getConnections()) {
DTNHost jn = con.getOtherNode(getHost());
EpidemicRouter jn_router = (EpidemicRouter)jn.getRouter();
if(jn_router.isTransferring()) {
continue;
}
if(home.containsKey(jn_router.getHost())) {
this.send_Centrality(jn_router,this.get_centrality(), Neigh_I );
j_cent_values = this.recieve_cent(jn_router, loc);
this.centrality = this.centrality +
this.update_centrality((double)j_cent_values.get(0),(int)j_cent_values.get(1));
double j_centrality = (double)j_cent_values.get(0);
for(Message m: msgCollection) {
if (jn_router.hasMessage(m.getId()))
{continue;}
DTNHost dest = m.getTo();
if(j_centrality >= this.centrality || jn_router.getHost().equals(dest)) {
sendCons.add(con);
}
}
}
if(sendCons.size() == 0)
return null;
List<Message> messages = new ArrayList<Message>(this.getMessageCollection());
this.sortByQueueMode(messages);
return tryMessagesToConnections(messages, sendCons);
}
@Override
public EpidemicRouter replicate() {
return new EpidemicRouter(this);
}
=====End Algorithm=====

```


Appendix B

Configuration File

```
#Settings for the simulation
## Scenario settings
Scenario.name = CHBPR_Filename
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 43200s == 12h
Scenario.endTime = 43200
Threshold=0.6
# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10
# High speed, long range, interface for group 4
highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000
# Define 6 different node groups
Scenario.nrofHostGroups = 6
# Common settings for all groups
Group.movementModel = HumanWalk1
Group.router = CHBPR
Group.bufferSize = 5M
Group.waitTime = 0, 120
# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface
# Walking speeds
Group.speed = 0.5, 1.5
# Message TTL of 300 minutes (5 hours)
Group.msgTtl = 300
Group.nrofHosts = 40
```

```

# group1 (pedestrians) specific settings
Group1.groupID = p
# group2 specific settings
Group2.groupID = bc
# cycles can drive only on roads
Group2.okMaps = 1
# 10-50 km/h
Group2.speed = 2.7, 13.9
# another group 3 of pedestrians
Group3.groupID = w
# The Cars groups
Group4.groupID = c1
Group4.speed = 7,10
Group5.groupID = c2
Group5.speed = 7, 10
Group6.groupID = c3
Group6.speed = 7, 10
## 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 = 25,35
# Message sizes (500kB - 1MB)
Events1.size = 500k,1M
# range of message source/destination addresses
Events1.hosts = 0,126
# Message ID prefix
Events1.prefix = M
## Movement model settings
# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed = 1
# World's size for Movement Models without implicit size (width, height; meters)
MovementModel.worldSize = 4500, 3400
# How long time to move hosts in the world before real simulation
MovementModel.warmup = 1000
## Map based movement -movement model specific settings
MapBasedMovement.nrofMapFiles = 4
MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt
## Reports - all report names have to be valid report classes
# how many reports to load

```

```

Report.nrofReports = 1
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports (can be overridden per Report with output setting)
Report.reportDir = reports/thesis/
# Report classes to load
Report.report1 = MessageStatsReport
## Default settings for some routers settings
ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true
## Optimization settings -- these affect the speed of the simulation
## see World class for details.
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 = 100
# Regular Expression log filter (see Pattern-class from the Java API for RE-matching
details)
#GUI.EventLogPanel.REfilter = .*p[1-9]<->p[1-9]$

```

Bibliography

- [1] L. Lilien, Z.H. Kamal, V. Bhuse, and A. Gupta, “The Concept of Opportunistic Networks and Their Research Challenges in Privacy and Security”, Chapter 5 in: S.K. Makki, P. Reiher, K. Makki, N. Pissinou, and S. Makki (Eds.), *Mobile and Wireless Network Security and Privacy*, Springer Science+Business Media, Norwell, Massachusetts, 2007, pp. 85-117.
- [2] C. Boldrini, M. Conti, I. Iacopini and A. Passarella, “HiBOP: a History Based Routing Protocol for Opportunistic Networks”, *Proceedings of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, 2007 (WoWMoM 2007)*, Espoo, Finland, June 18-21, 2007, pp. 1-12.
- [3] P. Yuan, P. Liu, S. Tang, “Exploiting Partial Centrality of Nodes for Data Forwarding in Mobile Opportunistic Networks”, *Proceedings of IEEE 17th International Conference on Computational Science and Engineering (CSE)*, Chengdu, China, Dec. 19-21, 2014, pp. 1435-1442.
- [4] S. K. Dhurandher, Deepak K. Sharma, I. Woungang, S. Bhati, “HBPR: History Based Prediction for Routing in Infrastructure-less Opportunistic Networks”, *Proceedings of 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013)*, Barcelona, Spain, March 25-28, 2013, pp. 931-936.
- [5] A. Mtibaa, M. May, C. Diot, M. Ammar, “PeopleRank: Social Opportunistic Forwarding”, *Proceedings of IEEE INFOCOM 2010*, San Diego, CA, USA, March 14-19, 2010,

pp. 1-5.

- [6] A. Vahdat, D. Becker, “Epidemic routing for partially connected ad hoc networks”, Technical Report CS-200006, Duke University, Durham, NC, USA, 2000.
- [7] A. Keranen, J. Ott, T. Karkkainen, “The ONE simulator for DTN protocol evaluation”, Proceedings of the 2nd Intl. Conference on Simulation Tools and Techniques (SIMUTools 2009), Rome, Italy, Mar. 2-6, 2009.
- [8] L. Lilien, Z.H. Kamal, A. Gupta, I. Woungang, E. B. Tamez, “Quality of Service in an Opportunistic Capability Utilization Network”, Chapter 7 in: M. Denko et al. (Eds.), *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, Auerbach Publications, Taylor and Francis Group, Boca Raton, Florida, 2011, pp. 173-204.
- [9] C-M Huang, K-C Lan, C-Z Tsai, “A Survey of Opportunistic Networks”, Proceedings of 22nd International Conference on Advanced Information Networking and Applications Workshops (AINAW 2008), Okinawa, Japan, Mar. 25-28, 2008, pp. 1672-1677.
- [10] S. Palazzo, R. Scopigno, M. Spirito, I. Korpeoglu, “State of the art of research on opportunistic networks, and definition of a common framework for reference models and performance metrics”, EU Deliverable, 216715 NEWCOM++ Report, 2008, 54 pages.
- [11] E. M. Daly, M. Haahr, “Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs”, Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBiHoc’07), Montreal, QC, Canada, Sept. 9-14, 2007, pp. 32-40
- [12] L. C. Freeman, “Centrality in social networks conceptual clarification”, *Social networks*, Vol. 1, Issue 3, 1978, pp. 215-239.

- [13] L. C. Freeman, A set of measures of centrality based on betweenness, *Sociometry*, Vol. 40, No. 1., March 1977, pp. 35-41.
- [14] Z. Gao, Y. Shi, S. Chen “Identifying Influential Nodes for Efficient Routing in Opportunistic Networks”, *Journal of Communications*, Vol. 10, No. 1, Jan. 2015
- [15] K. Xu, V. Li, J. Chung, J, “Exploring Centrality for Message Forwarding in Opportunistic Networks”, *Proceedings of the 2010 IEEE Conference on Wireless Communications and Networking (WCNC)*, Sydney, NSW, Australia, April 18-21, 2010, pp. 1-6.
- [16] P. Nikolopoulos, T. Papadimitriou, P. Pantazopoulos, M. Karaliopoulos, I. Stavrakakis, “How much off-center are centrality metrics for routing in opportunistic networks”, *Proceedings of the 6th ACM workshop on Challenged networks (CHANTS’ 11)*, Las Vegas, Nevada, Sept. 23, 2011.
- [17] S.K. Dhurandher, I. Woungang, A. Rajendra, P. Ghaie, P. Chatzimisios, “A Centrality-based ACK Forwarding Mechanism for Efficient Routing in Opportunistic Networks”, *Proceedings of International Conference on CYber physiCaL systems, IoT and sensors Networks (Cyclone 2015)*, Rome, Italy, Oct. 26, 2015. In Press.
- [18] N. Vastardis, K. Yang, S. Leng, “Socially-Aware Multi-phase Opportunistic Routing for Distributed Mobile Social Networks”, *Wireless Personal Communications*, Vol. 79, Issue 2, Nov. 2014, pp. 1343-1368.
- [19] OMNET++ Simulator, <https://omnetpp.org> (Last visited Jan 4, 2016).
- [20] P. Hui, J. Crowcroft, E. Yoneki, “BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks”, *IEEE Transactions on Mobile Computing*, Vol. 10, Issue 11, Dec. 2010, pp. 1576-1589
- [21] E. Yoneki, P. Hui, S. Chan, J. Crowcroft, “A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks”, *Proceedings of the 10th ACM Symposium*

on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07), Oct. 22-26, 2007, Chania, Crete Island, Greece, pp. 225-234.

- [22] A.Vahdat, D.Becker, “Epidemic routing for partially connected ad hoc networks”, Technical Report CS-200006, Duke university, April 2000.