Social Relationship based Routing for Delay Tolerant Bluetooth-enabled PSN communications

By

Sardar Kashif Ashraf Khan

SUBMITTED FOR THE DEGREE OF MASTER OF PHILOSOPHY

Supervised by Dr. Laurissa N. Tokarchuk

School of Electronic Engineering & Computer Science

Queen Mary University of London

March 2012

DEDICATED TO

MY LOVING FAMILY

AUTHOR'S DECLARATION

This is to certify that

1. The work presented in this thesis is the author's own.
2. Due acknowledgements has been made in the text to all other material used.
3. The thesis is less than 60,000 words in length (including footnotes), exclusive of tables, maps, bibliographies and appendices.

SARDAR KASHIF ASHRAF KHAN

School of Electronic Engineering & Computer Science

Queen Mary University of London

2012

ABSTRACT

Opportunistic networking is a concept derived from the mobile ad hoc networking in which devices have no prior knowledge of routes to the intended destinations. Content dissemination in opportunistic networks thus is carried out in a store and forward fashion. Opportunistic routing poses distinct challenges compared to the traditional networks such as Internet and mobile ad hoc networks where nodes have prior knowledge of the routes to the intended destinations. Information dissemination in opportunistic networks requires dealing with intermittent connectivity, variable delays, short connection durations and dynamic topology. Addressing these challenges becomes a significant motivation for developing novel applications and protocols for information dissemination in opportunistic networks.

This research looks at opportunistic networking, specifically at networks composed of mobile devices or, pocket switched networks. Mobile devices are now accepted as an integral part of society and are often equipped with Bluetooth capabilities that allow for opportunistic information sharing between devices. The ad hoc nature of opportunistic networks means nodes have no advance routing knowledge and this is key challenge. Human social relationships are based on certain patterns that can be exploited to make opportunistic routing decisions. Targeting nodes that evidence high popularity or high influence can enable more efficient content dissemination. Based on this observation, a novel impact based neighbourhood algorithm called Lobby Influence is presented. The algorithm is tested against two previously proposed algorithms and proves better in terms of message delivery and delay. Moreover, unlike other social based algorithms, which have a tendency to concentrate traffic through their identified routing nodes, the new algorithm provides a fairer load distribution, thus alleviating the tendency to saturate individual nodes.

TABLE OF CONTENTS

LISTS OF FIGURES

LISTS OF TABLES

ACKNOWLEDGEMENTS

First of all I am very thankful to ALLAH Almighty, who gave me strength, health and determination to complete this thesis. There are some important people in both my academic and personal life, who really helped and encouraged me throughout the course of my MPhil research.

My “Deepest Appreciation” goes to my principle supervisor Dr. Laurissa N. Tokarchuk for her kind guidance and invaluable academic/technical support. I am also thankful to Dr. Raul J. Mondragon for his understanding and encouragement. I would also like to thank academic and management stuff in EECS at Queen Mary University of London for providing such a nice academic and friendly environment. Also I would like to express my gratitude to all my friends, colleagues and faculty members who helped me one way or other in terms of sharing ideas and kept me motivated.

I would also like to thank Higher Education Commission of Pakistan for providing opportunities for the higher studies and praise their efforts for the betterment of the education of Pakistan.

My deepest prayers goes to my parents for their never ending prayers and blessings, kept faith in me, motivated me and praised me in every walk of life. My deepest appreciation goes to my siblings for their never ending love for me and their encouragements really keep meanings in my life.

Last but not the least, my special thanks goes to my dearest and loving wife, though she came in my life not so long ago but her understanding attitude, unconditional love and encouragement really boosted my moral especially in difficult situations.

SARDAR KASHIF ASHRAF KHAN

CHAPTER 1

INTRODUCTION

This chapter provides a summary of the proposed work that includes motivation and objectives behind this research. It also highlights the novelty and expected contributions of this research. The chapter ends with brief summary of upcoming chapters of this thesis.

1.1 MOTIVATION

In opportunistic networks, it is common that there is no priori information how to deliver messages between individuals. The easiest way is to broadcast messages throughout the network, but this kind of communication has severe consequences in terms of resource usage. A node in a broadcast network can receive messages over and over again and this may fill the buffer of the node and force the device to shut down. An improvement to this method is to use an Epidemic algorithm [64], where nodes that have received the broadcasted information are considered "infected" and they do not receive the same message again. This algorithm stops duplicate packets in the network. However, it still broadcasts at least one copy of packet to every node present in the network, thus increasing the overall communication cost, which eventually consumes too many resources, needlessly.

Epidemic routing has significantly reduced the number of duplicate packets, but it still relies on broadcasting and still sends packets everywhere other than the intended destination. To address this problem, Hui et al presented a social based forwarding algorithm known as Bubble Rap in [5]. Bubble Rap (BR) forwarding is based on the theory of community and node centrality: "*forward messages to those nodes that are more popular than the current* *node" [5].* Taken from the analogy of person popularity in his/her social circle such that if a person is considered more popular in a given community, is more likely that information can be transmitted to other members through that person. This message forwarding method tends to direct the traffic towards its intended destinations and also reduces redundant traffic in the network. However, BR has its own concerns; it increases the load of popular nodes in the network and the buffer on these nodes can overflow very rapidly. In other words, the resources of the popular individuals can be depleted very quickly.

The BR proposed that targeting key nodes in the opportunistic networks improves the routing efficiency. However, there are other algorithms that define key nodes in different prospective. For instance, Korn et al [75] presented metric known as *Lobby Index*. A node has high lobby index if its neighbours have at least equal or more neighbours than the node itself. Our LI algorithm is based on the lobby index which can also be formulated in terms of *the diplomat's dilemma [74];* i.e. a person has a strong influence in a society if he has relations with people. A diplomat has a high influence in a society because his connections are primarily with influential members of society. As a result, he has high reach in society with minimum effort of making personal relations (more power less connections/low cost). In real world analogy, a company boss does not need to know every person in his large company. If he has some query for a specific employee, he can pass the message to the manager of employee's department who will eventually deliver the message to the intended employee.

Social based algorithms can be used to reduce the unnecessary resource consumptions in opportunistic networks. These methods are based on the assumption that there are individuals in the network that can play a central role in the delivery of information. For example, popular individuals encounter many other individuals; therefore there is a good chance that they will meet the destination individual or someone that knows the destination individual. In order to identify these popular nodes that can be used as reliable message forwarders in the network, centrality measurements are used. Over the years three measures have become the standard for centrality measure in networks: betweenness [71], degree [72] and closeness [73]. Based on these centrality measures, many researchers [5, 75, 74] have proposed new interesting techniques to deliver information based on social interaction.

1.2 Research Challenges

Following are the key challenges while dealing message forwarding/routing in opportunistic networks:

1. In opportunistic networks nodes have no prior knowledge of the routes to its intended destinations. Normally nodes have to rely on its local information or information received from encountering nodes. Locating intended destination in such networks is the key challenge.
2. Simple broadcasting of messages in opportunistic networks is the easiest solution but this kind of message forwarding increases communication cost immensely. Message transfer without generating excessive traffic poses a challenge in these networks.
3. In opportunistic networks, messages are normally forwarded in store and forward fashion, thus causes variable delays in opportunistic forwarding. Designing of speedy message forwarding techniques is another challenge.
4. Social based forwarding algorithms are based on the assumption that there are individuals in the network that can play a central role based on popularity, centrality or betweeness and therefore can be exploited for efficient message delivery. However, nodes in networks can selfishly saturate these individual nodes and may deplete their

resources very quickly. Therefore, fairer distribution of traffic on these individuals is a distinct challenge.

1.3 Research Objectives

The main objective of this research is to design a routing algorithm based on human social relationships analogies that can improve overall message deliveries while keeping the cost minimum by targeting more influential nodes in large communities in general. In terms of technological aspect it should be capable enough to efficiently transmit messages in Bluetooth enabled pocket switched networks (PSN) opportunistic environments.

1.4 Novelty & Contributions

This research is novel and contributed following aspects:

1. A new generic opportunistic social forwarding technique is devised by combining characteristics of Bubble Rap [5] and Lobby index [75] known as Lobby Influence.

2. Detailed comparison of Lobby Influence algorithm with Bubble Rap and Lobby Index, which shows:

   - The new algorithm exploits two important analogies of human social relationships 1) popular nodes: how much a node interacts with other nodes in the network, defines its popularity 2) influential nodes: how well a node is connected with more influential nodes, defines its high reach/access in the network.

   - In Lobby Influence messages are not always forwarded to most popular node. Irrespective of node popularity if a node has strong neighbourhood connections in current network, the data may be transferred to that node which is less popular but high neighbourhood relationships.

- New strategy has eased the pressure on the most popular nodes in terms of resource usage by allowing less popular nodes but more influential nodes to take the responsibilities of message forwarding.

- The probability of locating destination or member of destination community is increased as a result good-put of message delivery increase in PSN network.
- The new algorithm makes message forwarding faster, a decrease in overall delay is observed, while transferring messages to intended destinations.

3. Author's Publication: Khan S., Mondragon R. & Tokarchuk L. (2012)," Lobby Influence: Opportunistic forwarding algorithm based on Human social relationship patterns" has been accepted for publication to IEEE PerCom (PerMoby 2012), in press.

1.5 THESIS STRUCTURE

This thesis is structured as follows:

Chapter 2 discusses about the difference between MANETs and opportunistic networks with its applications. This chapter also gives an overview of delay tolerant routing techniques proposed by different researchers. The key attributes of these algorithms are also highlighted in this chapter. This review discusses the environmental characteristics and operation conditions in which the algorithm proposed by this thesis must function.

Chapter 3 gives an overview of techniques proposed by other researchers for the detection of overlapping communities and hierarchical structures in social networks as well as social forwarding and centrality measure techniques. These techniques are thoroughly analysed and their weaknesses are discussed in this chapter. The algorithm proposed by this thesis employs social algorithm techniques in an opportunistic environment for efficient message forwarding. Therefore, the material in this chapter provides the theoretical basis for social forwarding algorithms presented in the next chapter.

Chapter 4 presents the proposed methodology and approach used in this research work. This chapter proposes a new generic social forwarding algorithm known as Lobby Influence for intermittent communication networks. The chapter contains background that motivates for the development of new social algorithm and it details.

Chapter 5 presents the experimental results for Lobby Influence algorithm. In this chapter, different sets of graph and discussion are shown for three different movement models namely, Random way point, Helsinki City Scenario and Working day movement models, respectively. Epidemic, Bubble Rap and Lobby Influence algorithms are tested against each of these scenarios and result comparison is shown.

Chapter 6 presents a concluding discussion of the Lobby Influence algorithm and suggests future directions for this work.

CHAPTER 2

REVIEW OF INTERMITTENT AND OPPORTUNISTIC NETWORKS AND ROUTING TECHNIQUES

MANETs (Mobile Ad-hoc Networks) is an active area of research these days, in MANETS any node can join or leave the network, however in such type of network topology, nodes must have the prior knowledge of routes to destinations. When a new node joins the network, it first needs to determine routes before transmitting data by using some ad-hoc routing algorithm such as AODV or DSDV. Opportunistic networks are slightly different kind of ad-hoc networks, although driven from MANETs. In opportunistic networks, there is no need of prior knowledge for the routes to destinations as required in MANETS, any node can be used opportunistically that come across the way with the probability of making destinations one step closer. Such kind of message forwarding generates excessive traffic in the network and thus poses the major challenge for researchers to deal with.

The rest of the chapter is organised as follows; section 2.1 introduces the concepts of delay tolerant networks and section 2.2 investigates opportunistic networks, it surveys some practical uses of this type of network. Section 2.3 examines the state of the art routing & forwarding techniques used in opportunistic networking. Finally, this chapter concludes with the summary.

2.1 DELAY TOLERANT NETWORKS (DTNS)

Internet Research Task Force (IRTF) has conducted extensive research in the field of Delay Tolerant Networks (DTNs) and defines a specification for DTN architecture [47].

There are several definitions that describe DTNs in particular scenarios. A DTN is a network that can efficiently work in situations such as space communications or on an interplanetary level, where it is expected to have delays due to the transmission disruption cause by the motion of planets or satellites. Data needs to be stored when no transmission medium is available and can be resumed when communication links are available. Same kind of situation can be observed in internet or telecommunication networks, some applications are designed to tolerate a certain amount of delay such as emails and SMS.

The importance of DTNs was first realized in respect to military applications in hostile situations, where communication has a vital importance. Since soldiers move in a particular formation not far from each other, use of DTNs can be ideal for such environments. Figure 2.1 depicts a battlefield, where soldiers exchange information by forming ad-hoc communication network. The information gathered from soldiers can be transferred to command centre far from battlefield using a rely node i.e. helicopter.

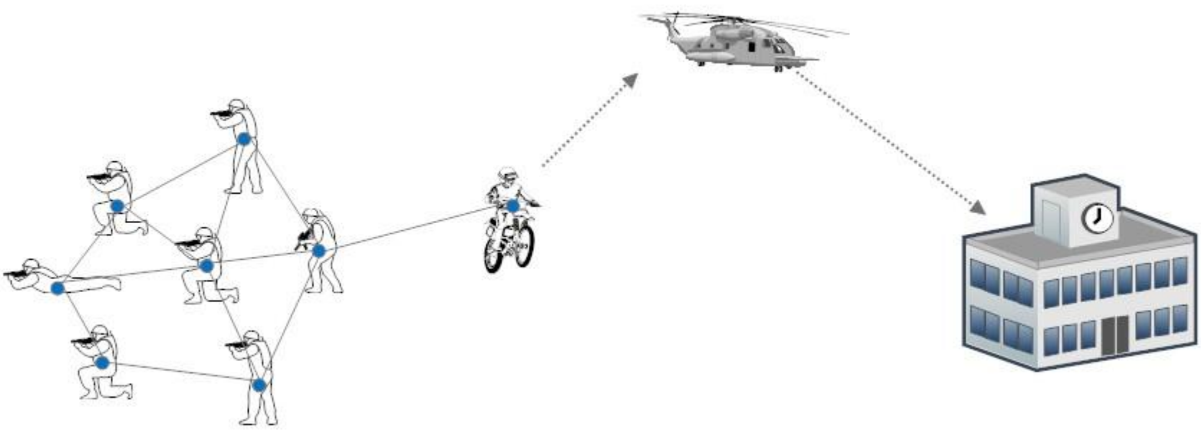

Figure 2.1 Ad-hoc communications in MANETs

Section 2.1.1 highlights the key characteristics of DTNs which makes them advantageous compared to its traditional counterpart such as internet. Section 2.1.2 describes the typical network architecture of DTNs followed by the key concept of message forwarding in DTNs in section 2.1.3. Section 2.1.4 and 2.1.5 introduce the typical protocol architecture of DTNS and basic role of node in DTNs, respectively.

2.1.1 WHY DTNS?

Many developing and prospective networks do not fit the traditional definition of internet; same is the case with DTNs. Following characteristics makes DTNs unique and advantageous compare to internet:

Discontinuous Connectivity: When there is no reliable path between source and destination means possibility of broken links then traditional internet protocol cannot apply in this scenario. One has to rely on such protocols that can operate in intermittent connectivity environments. DTNs can address this type of problem by introducing the concept of store and forward messaging. When links are not available store the message and resume it when links are available.

Unpredictable Delays: Traditional internet protocol such as TCP/IP can be unsuccessful in the situation where communication delays are longer than certain time period. In such an environment, a protocol is required that can cope with long and unpredictable delays. DTNs protocols can allow messages to be kept for longer period of time. In case, if one path is not available, it can select another to reach the intended destinations.

Variation in Data Rates: DTNs can address this issue by dividing large packets into smaller ones or by combining small packets into larger ones to meet the channel conditions.

Packet Loss: Traditional internet protocols such as TCP/IP, requires packet retransmit in case the packet gets drop along the transmission medium. Increase in packet loss will cause higher data error rates. In DTNs, in the case of packet failure, packets are required to retransmit on a node by node basis rather than end to end retransmission as typically done in TCP/IP.

2.1.2 Typical Architecture

The architecture of DTNs [47] is a combination of networks, where each network acts as an independent region having internet like connectivity within its scope. These independent regions are called DTN regions, where the communication between DTN regions is not permanent. Each region has its own set of protocols, physical infrastructure, communication technologies and independence. The DTN gateways are responsible for the communication among DTN regions; they isolate disconnection problems by not letting to affect the particular region. DTN protocols are designed in such a way that they sit on top of the traditional network layer protocols in order to make communication possible among DTN regions.

The concept is depicted in the figure 2.2, where DTN gateways interconnect regions running potentially dissimilar protocol stack. For instance, region A contains a mesh network, where each node is not only responsible for transferring its own data but also use as a relay for other nodes. Gateway of Region A is connected to Region C via satellite, if there is some disruption in communicating with satellite; DTN gateway is responsible for isolating problems without affecting the region. Similarly, Region B represents a wildlife monitoring project where movement of different animals are being studied. Animals are tagged with different antennas to track their movements. Data is being gathered using ad hoc network and stored in remote research office serving as DTN gateway, where researchers take it to research centre connected to region C for further processing.

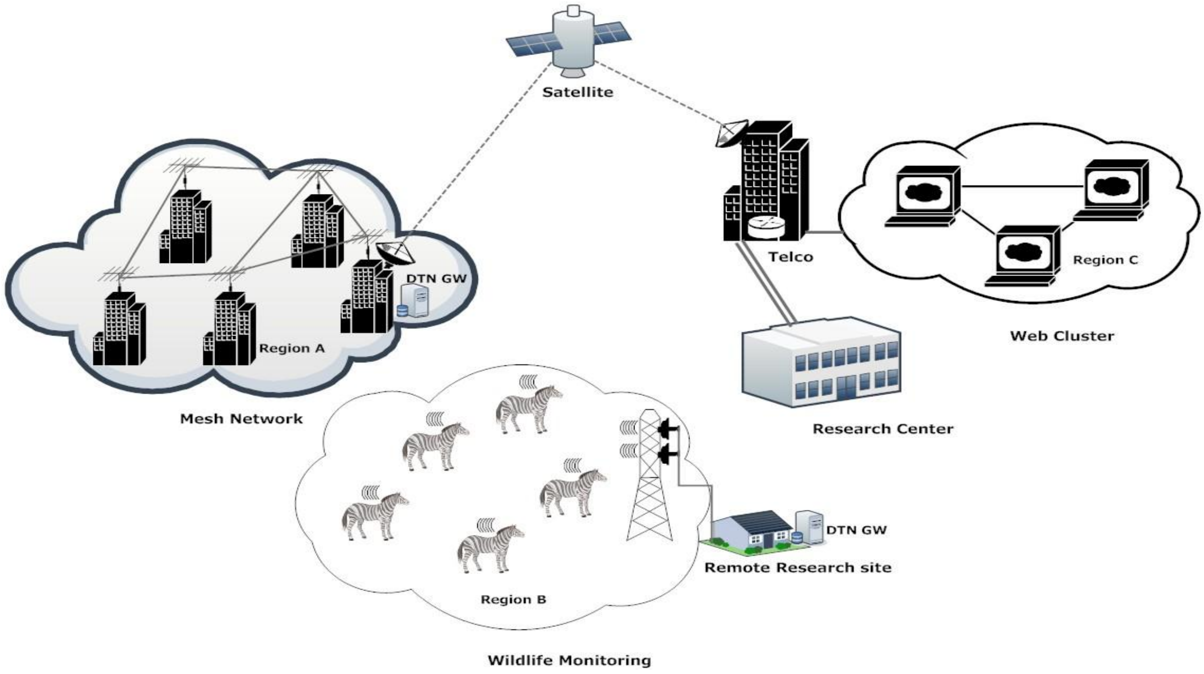

Figure 2.2 Typical architecture of nodes in DTN

2.1.3 KEY CONCEPT OF DTNS

A concept known as *store-and-forward message switching [48]* is used to deal with the situation, where networks have broken links, unpredictable delays, packet loss and variable data rates. A typical daily life example is the post office, where postal delivery system uses the same concept of store and forward letters to different post office outlets until reaches to the intended destinations. Email, SMS and voicemail also use the same store-and-forward messaging concept. In DTNs, when transmitting from source to destination, data is required to store and forward on node by node basis until it reaches the destination. The concept is shown in the figure 2.3.

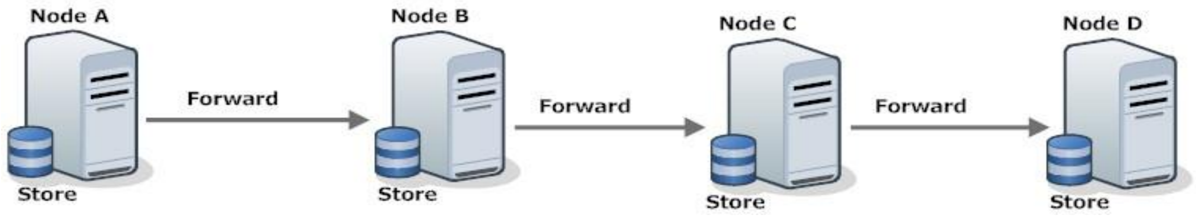

Figure 2.3 Key concepts in message transfer

The node that acts as a router requires continual data storage space for queues due to following factors:

- A communication link is broken for a long time then node has to keep data in its storage space until it receives a green signal for data transmission.
- If an error occurs during the transmission between node A and node B, node A has to keep the data in its storage space until transmission is successful by retransmitting it to node B.
- The transmission bandwidth can vary on different nodes, so overall transmission time is not predictable. Nodes require storage space in case of low transmission bandwidth.

2.1.4 THE DTN PROTOCOL LAYER

DTN protocols use a new higher layer protocol known as *bundle layer* that sits on the top of traditional internet layer protocols, to achieve the *store-and-forward* capabilities. The figure 2.4 shows the introduction of DTN protocol layer (Bundle layer) on top of traditional TCP/IP protocol architecture. The bundle layer is responsible for communicating between regions that are participating in DTN communication by storing and forwarding messages (bundles) at the bundle layer. The lower layers such as transport and below are region specific and work independently, as discussed in section 2.1.2.

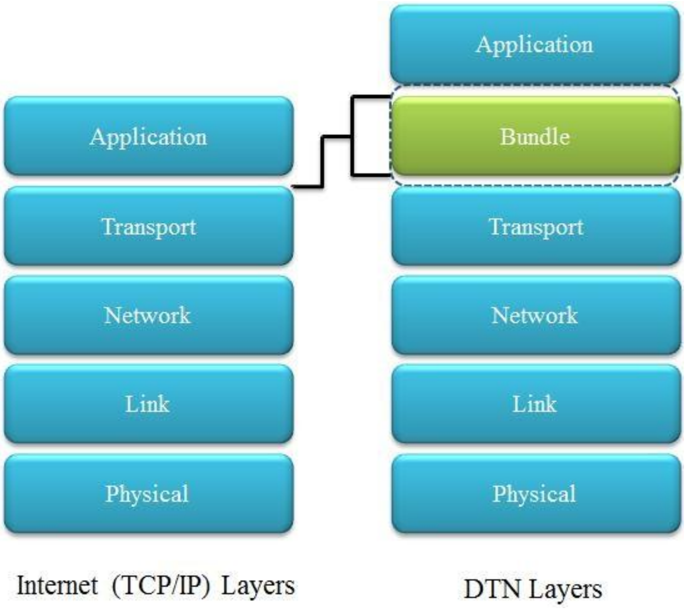

Figure 2.4 DTN protocol layer

2.1.5 Basic Role of a node in DTNs

In DTN networks, at the bundle layer, a node can act as source, destination or relay (forwarder). In DTN terminology, a node can called as *Host, Router or Gateway* depends upon the node's current role in the network.

Host: The host can be either source or destination to transfer bundles (messages), but do not use as relay. The host required high storage capacity in case there is delay due to unavailability of link. As soon as link is available the message can be resume to transfer to (or from) router.

Router: In single region, a router is used as a relay to transfer messages from source to destination. A node can act as router, normal node or both; but requires storage capacity to store messages in case of unavailability of links. The router can possibly nominate as retransmission point (custody transfer capability).

Gateway: The functionality of a gateway is similar to the router except it is use to transfer messages (bundles) between two different regions.

2.2 Opportunistic Networks

Often opportunistic networks and delay tolerant networks are considered the same but there are important differences. As discussed in [49], while calculating the route from source to destination, DTNs depends on internet like topologies where links between gateways are permanent and nodes have to communicate after fixed, interval of time to the same gateway. In DTNs, nodes must have the prior knowledge of network topology in order to communicate between source and destination.

In the case of opportunistic networks, it is not necessary for a node to have prior knowledge of network topology. Every node can act as a gateway and is capable of keeping data until it finds a potential opportunistic node that can deliver the message to the destination. For route calculation, nodes have to rely on local information (local routing table) or information given by its neighbours (directly encountered nodes). Free choice of any node as a potential relay in opportunistic networks makes them more flexible in comparison to DTNs. An opportunistic network is shown in the figure 2.5, where source (left) and destination (right) communicates opportunistically with each other through number of participating nodes such as bus, car kid, bicycle etc., any node can be used as potential candidate for message transfer.

Pocket Switch Networks (PSNs) are specialised form of opportunistic networks, where devices such as mobile phones or PDAs are used for communication purposes. A mobile phone can be used as a device that is capable of transmitting messages to other mobile phones via using Bluetooth technology. The mobile phone can keep messages until it finds a potential mobile phone that is either the destination or can take data to the destination acting as potential forwarding node.

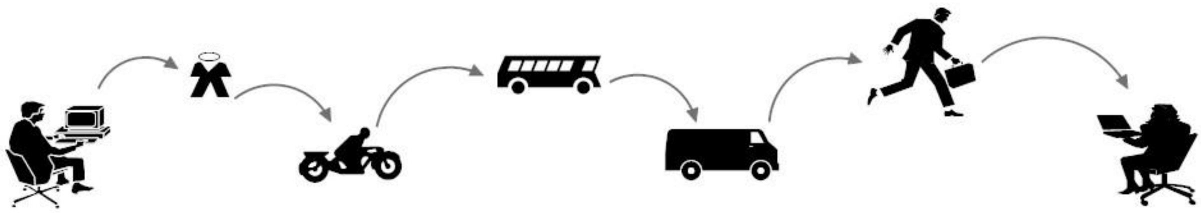

Figure 2.5 Opportunistic networks

This section will examine three popular application domains of opportunistic networks. Section 2.2.1 discusses about number of projects conducted to gather real life mobility traces for learning purposes. Section 2.2.2 highlights some of the project using opportunistic networks for wildlife studies. Similarly, section 2.2.3 describes some real implementation of opportunistic networks in rural area.

2.2.1 HUMAN MOBILITY CASE STUDIES

Opportunistic networks are different from traditional internet networks and require a different approach for designing new protocols. In opportunistic networks, nodes have no prior knowledge of the routes to intended destinations, this means that nodes have to rely on local information or the information gets from encountered node. Researchers have developed different real world experiments for opportunistic networks, where they learned about nodes behaviour such as contact frequency, contact time, contact duration etc. The information gained from these experiments are compiled and made publically available to other researchers for learning purposes, this collection of information is referred as real world mobility traces. These traces can help significantly for designing of new efficient protocols. Some examples of such experiments conducted for the purpose of gathering real world mobility traces are:

The Reality Mining Project: The Reality mining project run by MIT media lab [51] conducted several experiments to gather real time data sensed by machines related to human social behaviours. This new collection of data has opened new doors for the modelling of *conversation context*, *proximity sensing* and *temporospatial locations* of individuals in society.

In reality mining experiments mobile phones were used to gather information based on human social behaviours, either as an individual or in a group. They used 100 participants and captured their proximity, communications, daily activities and location information over the course of 2004-2005 academic years. According to reality mining project, they gather around 350,000 hours of continues data on human behaviour that is approximately equivalent to 40 man years. The data set produced by MIT media lab is widely used in several studies of human behaviours [51][46].

The Haggle Project: The Haggle project [52] funded by European commission has conducted many experiments specifically for opportunistic networks. The Haggle project works under the frame work defined in [53], where they are working on Pocket Switched Network (PSN). In these experiments, mobile phones and PDAs are used as communication nodes that can be used as opportunistic candidates for message transfer. The main focus of this project is to exploit *pair-wise [49]* contacts between device-to-device and user-to-device. The two parameters that define pair-wise contacts are *contact durations* and *inter-contact time*.

- When two mobile nodes come closer and have an opportunity to communicate with each other, the time they spend with each other scope is known as *contact durations*.
- When two mobiles nodes previously made contact, now once again make contact to each other, the time they spend with each other is known as *inter-contact time*.

For experiment purposes different mobile phones and PDAs are used in the field to collect data from the real world environment. This real world mobility traces gathered from this experiment provided a major breakthrough, now researchers and students are using these traces for the study of opportunistic networks.

2.2.2 WILDLIFE MONITORING

Opportunistic networks are also successfully deployed in the area of wildlife monitoring. These networks help biologists to study the changes and effect of ecosystem on animal's behaviour, their relation with other animals and information about their life cycle. In order to monitor wildlife activities, special tags with storage capability are attached with the animals that collect information and send this collected data to base station, which further takes that data to the processing centre. In such kind of network gathering information is very challenging because it is not possible to deploy base station for each animal, so one has to rely on opportunistic networks. Different animals have tags attached to them to form an opportunistic network based on store-and-forward mechanism, where information can transfer from one animal to other until that reaches to base station. Some of the well-known projects are discussed here:

ZebraNet: A project going on at Princeton University known as ZebraNet [54] [9] monitors the activities of zebras. The actual deployment of this project is in Savanna area of central Kenya. The zebras are equipped with special collars that are monitored by the researchers from a vehicle based base station. The vehicle moves around the Savanna Wild Park and gathers information from the zebras, encountered. Two algorithms: *flooding* and *history based* are used in ZebraNet. In *flooding*, data is transferred in a broadcast manner such that each collar (node) will transfer data to any collar it encounters until the destination is reached. Simulation results showed that the flooding protocol performed exceptionally well as compare to the direct protocol, in which each collar had to deliver data directly to the base station. In *history based* protocol, collar will select only one of its neighbouring collars with the assumption that chosen collar will eventually take data to the base station. The chosen collar is considered as potential neighbour if it has highest number of meeting-counters with base station. Every time when collar encounters a base station, there will be an increase in meeting-counters of collar and meeting-counters will decrease if collar has not seen base station for specific interval of times. Simulation results showed that the history based protocol is better than the flooding protocol, not only in terms of efficient information transfer but it also saves bandwidth and energy.

Shared Wireless Infostation Model (SWIM): In SWIM project [56], opportunistic network is deployed to monitor the behaviour of whales. Each whale has a tag attached with it; information is transferred from tag to tag (whale to whale) until it reaches to the SWIM station for further processing in research centre. The concept used in SWIM project is similar to the ZebraNet [55].

2.2.3 INTERNET FOR RURAL AREA

DakNet Project: DakNet project [57] is deployed in India, the aim of this project is to provide intermittent network based internet to the rural area where standard deployment of internet like technologies are not possible due to lack of telecommunication infrastructure. Kiosks are deployed in the villages with the capability of storing electronic data and equipped short range wireless connectivity to exchange data with different vehicles mounted with Mobile Access Points (MAPs). MAPs upload data from village kiosks and later uploads it to the internet when reach at nearby town. The same process takes place for downloading from the internet; MAPs download from the internet in nearby town and later download that data to village kiosks. Such kind of network is useful for emails exchange, audio/video messaging, mobile e-commerce, voting, health records and environmental sensor information.

Saami Network: The Saami Network Connectivity (SNC) project [58] used similar concept to provide intermittent internet connectivity to Saami population who live in the north of Sweden.

2.3 ROUTING/FORWARDING TECHNIQUES

Based on the nature of opportunistic networks, one can classify it into two broad categories *infrastructure based routing* and *infrastructure-less routing*. The aim is to look at ways where new routing/forwarding techniques can be used to guarantee message delivery and improve the delay time and communication cost caused during message transfer from source to destination. Before discussing different routing techniques in opportunistic networks, the next section provides a brief introduction of parameters used to gauge the performance of opportunistic forwarding algorithms.

2.3.1 PARAMETERS

Following parameters are considered to gauge the efficiency of any routing/forwarding technique.

Delivery ratio: Equation 2.1 gives the general expression of delivery ratio that use to check the overall average efficiency of message delivery from source to the destination. More details relating to the work done in this thesis will be discussed in section 5.1.

$$Delivery\, ratio = \frac{T_{re}}{T_{tr}} \qquad (2.1)$$

Where; T_{re} is total average amount of data received at all destinations and T_{tr} is total average amount of data send by all sources.

Delay: Another parameter to check the overall delay experienced by messages from sources to destinations. An expression for delay check can be derived from equation 2.1:

$$Time\ Delay = T_{re} - T_{tr} \qquad (2.2)$$

Where; T_{re} time at which data is received at destination and T_{tr} time when message is sent by source.

Communication Cost: Total number of messages forwarded to reach the destinations defines the overall cost of the network, which ultimately effects the system resource (bandwidth and energy) utilisation.

Delivery ratio, time delay and communication cost are the main three challenges which researchers had to deal when designing and proposing new routing/forwarding techniques for opportunistic networks.

2.3.2 INFRASTRUCTURE BASED ROUTING

Infrastructure Based routing can be deployed in a network in a variety of ways. This section discusses the core methods:

Node-to-Base station Communication only: In infrastructure based routing, static nodes (base station) are deployed sparsely throughout the network. The purpose of these static nodes is to gather information from the source node and then transfer it to the destination node. In such a network, the source node has to keep the message in its database until it comes in range of the static node, at which point the source node delivers the message directly to the static node. Similarly, the static node keeps the source node's message until the destination node comes within its range and then the message is delivered to the destination node. In this kind of deployment only source node- to-static node and static node-to-destination node communication is allowed. An example of such kind of routing is Infostations [59], where infostations are used as base station with high data rate wireless communication equipment. The purpose of infostations is to provide web services to the users that come within its range.

Node-to-Node and Node-to-Base station Communication: In this kind of network deployment, a node can transfer the message directly to the base station. If base station is not within reach, a node can use another node or number of nodes as a relay to transfer the message to the base station. Such a network allows node-to-node and node-to-base station transfer of messages. However, this deployment is energy inefficient but minimizes delays. SWIM [56] & ZebraNet [54] are typical examples of such kind of network, where node-to-node and node-to-base station communication is allowed as discussed earlier.

Mobile infrastructure based routing (Carrier-based Routing): In carrier-based routing, special mobile nodes (carriers) are injected in the network; the purpose of these nodes is to collect information from other nodes. These mobile nodes may be referred as *carriers*, *supports*, *forwarders*, *MULEs*, or *ferries* and follow pre-determined or arbitrary routes in the network to collect (or deliver) data from (or to) the nodes they encounter. In the situation where only node-to carrier message transfer is allowed, these mobile nodes (carriers) are responsible to deliver message directly to the destination. This type of communication is very energy saving but with disadvantage of additional delay. In situations, where nodes are sparsely located in the network, node-to carrier only communication is not sufficient and node-to-node communication is needed as well. This type of communication makes the message delivery mechanism very efficient but with cost of energy consumption.

In *Data-MULE system* [60], different wireless sensors are deployed at different locations in the area, which gather information from their surroundings. The main focus in this project is to make the network energy efficient. For that purpose a MULE (carrier) moves around the network and when it passes near a sensor, sensor-to-MULE communication takes place and information is uploaded to the MULE. When MULE is on the move and encounters an Access point (AP), the collected data from sensors is then transfers to the AP. APs are connected to the central database where data received from sensor stores and then eventually processed.

In the *Message Ferrying Approach* [61], extra mobile nodes known *as Message Ferries* are injected in the network; purpose of these nodes is to gather information from the source node. Two types of communications can be initiated in this approach:

- *Node Initiated Message Ferrying*: Each node in the network knows about the location of Ferrying node, because each Ferrying node moves in a pre-defined and fixed path. Source node will initiate the communication and transfer message to the Ferrying node by moving near to it.
- *Ferry Initiated Message Ferrying:* In this approach, *Ferrying* nodes continue to move on default path and every node knows about it current location. When a node wants to communicate with a Ferry node, it first transmit a request message to the Ferry node by the help of other nodes present in the area, if that node is very far from the current location of Ferry node. In response, Ferry node will come closer to the source node and then message transfer occurs.

2.3.3 INFRASTRUCTURE-LESS ROUTING

Routing techniques that are used in Infrastructure-less opportunistic networks are based on the broadcasting. The idea behind such kind of routing is that no node has any prior knowledge of routes; the source node has to broadcast messages everywhere until it reaches the destination. This kind of routing is successful where nodes are densely populated in a network. Although the delivery process of a message from source node to destination node is very fast, the cost of high network traffic will eventually eat network resources such as bandwidth and energy consumption. The congestion in a network can be avoided by limiting the broadcast message to certain number of hops or by limiting the number of parallel copy-messages in a network. Some of very well-known opportunistic algorithms are discussed below:

Epidemic routing protocol: In [64], the message is spread across a network by means of *pair-wise contacts* between nodes like a virus or disease. A node is considered as *infected* node when it is acting as a source node or rely node. An infected node stores the message to its local database until it forward it to a rely node or destination node. Any node that has not yet received any message is considered as *susceptible* node and becomes an *infected* node when it receives the message. A node is considered *immune* when it passes the message to another node or destination node. In order to avoid congestion and loops, the message is allowed to transfer to certain number of *hop counts.* When hop count limit is equal to one this means that the message can only deliver directly to destination node.

MV routing protocol: The meeting & visits protocol [65] is more refined than Epidemic routing. In this kind of routing, message exchange is similar to Epidemic routing by exploiting pair-wise contacts but with extra refinement in terms of choosing a node. The node that is chosen as a potential candidate is the node that has higher ability to take the message to the destination. This can be achieved by looking at the node's recent-past history activities, if the *meeting* and *visits* of that node to the particular destination is more often will be considered as potential candidate and used as a relay for message delivery. A similar concept was used in the PROPHET algorithm [66].

PROPHET algorithm: [66] is a probabilistic routing protocol. PROPHET evaluates delivery predictability by employing probabilistic metric. This metric defines the probability of a node for delivering a message to the destination. When two nodes encounter each other, they exchange delivery predictability vectors. When same two nodes regularly meet each other, they are considered good message forwarders, thus have high delivery predictability for each other. Similarly, if two nodes are not meeting each other or it's been a while since they encountered, naturally they are not good forwarders and hence the delivery predictability metric starting to decrease as time passes. This algorithm showed a significant decrease in communication overhead compare to Epidemic [64].

N**etwork-coding-based algorithm:** [67][68] reduces significant amount of network traffic by using a coding mechanism. Suppose two nodes want to exchange data with each other and both nodes use the same middle node as a rely to reach each other. The middle node will gather the information of the two nodes and apply an exclusive OR (XOR) operation on data received from both nodes and broadcast the new XORed data to both nodes. The receiving node can reverse the process and extract its respective information. In this way 50 % of traffic on network can be minimized.

CORE: [82] is another coding-aware routing protocol, the priority of forwarding nodes in routing table varies dynamically depending on the coding opportunities. A node has more coding opportunities when it holds more coded packets. To avoid communication cost node must also be the neighbour of source node and at the same time more closer to the destination node than source node.

Context-Aware routing (CAR) protocol: In [69], routing is based on some prior knowledge of the node's context. Each node in the network has its own routing table, where it keeps record of potential neighbours as forwarders that can transfer the message to the destination node. The potential neighbours are elected based on context attribute such as, node's battery life, connection & disconnection rate, the possibility to reach to the destination, node's mobility pattern. CAR provides a framework based on *multi-attribute utility theory* to choose potential nodes as carriers in opportunistic networks.

MobySpace Routing: [70] is based on Context-based routing, where mobility pattern of nodes are used as context information for routing. A high dimensional Euclidean space named as MobySpace is constructed, where axis represents connection between two nodes and distance along axis represent probability of connection.

Contention-based geographic forwarding: [79] comes under the category of geographical information based routing protocols. In geographical based protocols the assumption is that each node has some knowledge of its location and neighbouring nodes with in the network. Each node transmits a one hop beacon messages to gather the co-ordinates of surrounding nodes and maintains location based table. In [79], source node transmits a control packet before sending actual data message to its surrounding nodes. Those nodes which received control packet from source node calculate their suitability as message forwarder by comparing their distance to the intended destination. There may be possibility that more than two nodes have close distance to the intended destination, in such situation these nodes will compete each other for the selection of message forwarding node. Only one node is allowed to become message forwarder to avoid communication cost. Once the message forwarder is decided, source node will transmit message directly to the selected node for further transmission. This process may increase the computation cost of surrounding nodes due to election and selection of message forwarder.

Location-Aided Opportunistic Routing: [80] selects potential forwarding nodes based on higher packet advancement metric. The packet advancement is the distance between the source node and destination node subtracting potential forwarding node and the destination node. This metric calculates the progress in distance when forwarding node transmits packets to the destination node. Mathematically[80],

$$D_{ir} = Dist(N_i, N_d) - Dist(N_r, N_d) \qquad (2.3)$$

Where; N_i is the source node, N_r is the forwarding node and N_d is the destination node.

To add the potential message forwarder into the forwarding list, the packet advancement metric of potential forwarder is compared to the highest packet advancement metric of all the neighbouring nodes of source node and must be less than predefined threshold value. This process allows nodes to include or exclude new potential forwarders.

Robust Geographic Routing: [81] uses the similar concept as proposed in [80] and exploits those potential forwarders that are geographically closer to the destination. The nodes supposed to know its current position and the position of its direct neighbours. Before transmission of data the source gathers location of destination and appends this information to the packet header along with its own location. The information gathering can be achieved by using one-hop beacon or piggyback in the data header’s packet.

2.4 Chapter Summary

This chapter has provided a review of two interlinked but yet different communication networks: DTNs and opportunistic networks. Fundamentally, both DTNs and opportunistic networks rely on the concept of store and forward messaging. However, calculation of routes from source to destination makes these networks different in nature. DTNs depend on internet like topologies where it is mandatory to have advance knowledge of routes, while in opportunistic networks it is not necessary to have prior knowledge of routes. In DTNs, links between gateways are permanent and mobile nodes transfer data in regular interval of time to the same gateway. But, in case of opportunistic networks, every node can act as a gateway and has capability of storing the data. Free choice of any node as a potential relay in opportunistic networks makes them more flexible in comparison to DTNs.

As the result of this study, it is clear that the efficiency of an opportunistic routing/forwarding algorithm can be mainly gauged on these attributes i.e. message delivery, delays and communication cost. The research presented in this thesis will implement different opportunistic forwarding techniques in similar simulation environments to better understand the efficiency of these algorithms. Based on the results obtained from these experiments, this thesis will propose new and efficient algorithm for opportunistic communication to improve the overall message delivery, less delays and low communication cost.

CHAPTER 3

COMMUNITY STRUCTURES IN SOCIAL NETWORKS

The previous chapter discussed delay tolerant networks (DTN). These types of networks can be used in situation where delays are expected due to transmission disruption such as satellite communication, data can be stored during absence of links and resume back when links are available. These kinds of networks are very feasible in situations where communication environments are intermittent. Furthermore, previous chapter investigated different opportunistic routing & forwarding techniques in DTNs and provided some of its practical uses. Based on these investigations, it is learnt that the major challenge in any opportunistic network is message routing/forwarding. Nodes do not have prior knowledge of routes to destinations; therefore they have to rely on local information or information gathered from encountered nodes. This means that node selection for message forwarding in opportunistic network is based on its probability of delivering messages to intended destinations

Formation of groups in any given human society is a natural phenomenon, where human forms tightly connected relationships (such as family, friends and employees etc.) to achieve common goals, such relationships are called as modules or communities in terms of Sociology. This type of network formation can also be seen in other areas such as biological [6], technological [5] and informational networks [1]. All these areas come under the same umbrella known as complex networks, for instance, formation of communities is not only valid for humans but also for other species such as dolphins and elephants, live in communities. To study these complex networks scientifically, researchers [21, 22, 32] employ different techniques of graph theory to understand underlying structure in complex networks. These techniques help to identify hierarchal and community (or sub-communities) structure present within the complex networks. Identification of such structures can greatly help in finding important nodes in scale free networks such as opportunistic networks. In opportunistic networks, sources have no prior knowledge of routes to the intended destinations; nodes have to rely on local information or information given by encountered nodes. Many researchers [44, 46, 74, 75] have proposed that the content forwarding can be achieved by exploiting key nodes which have influence or important position in these networks. For the identification of these key nodes, researchers rely mainly on three centrality measures betweenness [71], degree [72] and closeness [73], these centrality measures are driven from the concepts of graph theory. Therefore, this chapter will discuss key concepts of graph theory and related algorithms, also investigates different graph theory driven routing/forwarding techniques in opportunistic networks.

Section 3.1 discusses complex network analysis, including definitions of graph-theory terminology, a discussion of the importance of hierarchal & overlapping community structures and concludes with a survey of community detection algorithms. Section 3.2 introduces algorithms that utilise graph theory concepts in opportunistic routing/forwarding. Finally, section 3.3 provides a summary and highlights the importance of this study in the thesis.

3.1 COMPLEX NETWORK ANALYSIS

Social network analysis has gained huge attention since 1930 and a lot of research is going on in the field of sociology since then [9, 10]. The major area in complex networks is the study of sociology and in particular detection of *community structure* [11, 12]. In any social society (network), there are groups of people (nodes) who are much closer to some then the rest of the society. These groups form modules or communities [11, 13, 14]. In terms of complex networks, these modules or communities are formed due to the relationship based on common interest or goals. The same concept of community structure is also applicable to different areas such as the collection of web pages pointing to same subject matter [15], global structure of the worldwide air transportation network [16], taxonomic categories in food webs [17, 18], community formation in bottlenose dolphins [19], ecosystem management [20]. Thus detection of communities is very important in order to study any network but still, it is the most difficult task to deal with [7].

There are at least two facts that need to be dealt with while studying community structures. First, there are communities with in communities, small communities can form large one and vice versa for example a large company represent a community, within that company there are different departments at different levels such as sales, marketing, and productions etc., represent hierarchical structure as shown in figure 3.1(a). Such hierarchical structure present in any system i.e. company makes it very efficient, especially if modules i.e. departments formed are performing a dedicated task [21]. Therefore, concept of hierarchies in community structure makes it very rich, but this requires a method that can detect hierarchies in a community at different levels. The second area is overlapping communities [22, 23, 24] as shown in figure 3.1(b), where nodes belong to more than one community or module such as social communities formed by families, friends, professions, hobbies, etc. [25]. Standard methods of community detection cannot be applied in the situation where nodes belong to more than one module because overlapping nodes hide the important information and cause confusion, thus lowering the quality of the detected communities. Overlapping of nodes in any community structure is natural phenomenon; efficient methods are required to unearth the nodes that belong to multiple communities [7].

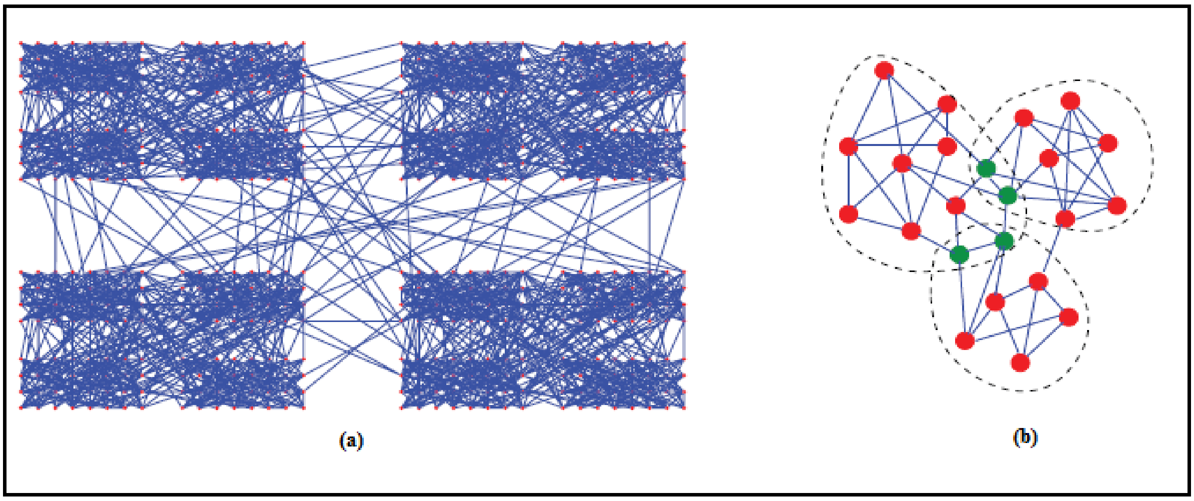

Figure 3.1 a) Hierarchical structures b) Overlapping communities [25]

3.1.1 Graph-Theory Terminology

In order to understand community structure of a network, one must be familiar with the terminology that is used in graph-theory. This section describes some of the most common and important terminologies use in graph-theory.

Graph

The complex network that is being studied under graph-theory is known as *Graph*. Typically a graph contains vertices and edges. There are two types of graphs:

- Undirected Graphs
- Directed Graphs

Some commonly used terminologies for undirected and directed graph are discussed below.

Vertices: Vertices are essentially points and are referred to as nodes in community structure study. Typically, they will be labelled on graphs.

Edges: In an undirected graph, edges are simply lines in between pairs of vertices. In undirected graphs direction does not matter, one line is used to connect two vertices. For example, suppose there are n vertices in a graph, the maximum number of edges will be:

$$_{n}C_{2} = n\,(n-1)/2 \qquad (3.1)$$

This is the number of edges in a graph. A graph is considered as a *complete/connected* graph where there exists an edge between all pairs of vertices.

Degree: A degree of a vertex in a graph can be defined as the number of edges that are incident on that vertex.

Walk: A walk can be defined as a series of edges that can be crossed one by one. This means that where one edge terminates at some point, next edge has to start from that point. A walk can have same starting and finishing point (like circle) or can terminate at different finishing points (like line).

Path: A walk which contains no repeated vertices is a path.

Cycle: A path is called a cycle if the first and last vertices visited are the same.

Trail: A trail is a walk which contains no repeated edges**.**

Circuit: A circuit is a trail that begins and ends at the same vertex.

Almost all definitions that are used in undirected graphs are also applicable in directed graphs, except some definitions have different meanings due to the nature of directed graphs that are listed below.

Edges (Directed Graph): The definition of edges in a directed graph has a different meaning as compared to edges in an undirected graph. In directed graphs direction does matter. For example there are two vertices say ‘a’ and ‘b’ present in a directed graph, now we have two edges one is directed from vertex ‘a’ to vertex ‘b’ and other is directed from vertex ‘b’ to vertex ‘a’. So, edges present in a graph not only connect pair of vertices but also defines direction as per line.

Degree (Directed Graphs): In directed graphs the degree of a vertex is consider in two ways: *in degree* and *out degree*. The sum of the edges that are coming into the vertex is known as *in degree* of vertex and the sum of the edges that are leaving a vertex is known as *out degree* of vertex.

Random Graph: A random graph is obtained by starting with a set of *n* vertices and adding edges between them at random such as Erdos and Renyi.

3.1.2 BASICS OF COMMUNITY STRUCTURE DETECTION

From sociological perspective, an individual forms a relation on the basis of common interest, thus forming communities, modules or units comprise of a set of individuals who have similar interests. It is also true that in a given community or unit some individuals are more tightly bonded to a sub-set of individuals than to the rest of the community. For example a university represents a community, students and teachers belonging to different departments are more tightly bonded to their respective departments compare to the other departments in a university. These differently weighted relationships form hierarchical structure within the community. Many researchers proposed different techniques based on graph theory to identify communities [6, 7, 31]. In terms of computing, isolating communities help locating the interested nodes in complex networks. Clauset et el [86] proposed a dendrogram technique to find the hierarchal structure of communities in complex networks. Newman and Girvan [6, 31] proposed different algorithms that can be used to detect subgroup of different communities present in densely populated networks. The goal is to identify the natural division of sub-communities present in a dense network. A very important concept was presented by Lancichinetti in [7]. This work proposed an algorithm that finds the hierarchal structure as well as overlapping communities. The algorithm is based on local optimization of fitness function, where peaks in the fitness histogram discover community structure.

As discussed in section 3.1, detection of communities is not as straightforward as it seems, a community may have both formations such as hierarchal and overlapping structures. Understanding of these structures is very important in complex networks to detect communities. This section discusses these two very important structures.

Hierarchies

Vertices in a graph can have different level of organization. For instance, consider the analogy of company that represents a community (or graph in terms of graph theory), as discussed earlier in section 3.1. In company different departments such as sales, marketing, productions represent clusters (or sub-graphs) form hierarchal structure within that company. Members (vertices) of each department (sub-graph) are tightly bonded and may occasionally move within company (graph).Such kind of hierarchal structure present in a company can make easier to find a particular member within that company. This shows that even within a large community there may be small sub-communities present and members of those sub-communities are more tightly bonded to each other than the rest of the community members. Therefore, identification of hierarchal structures (partitions or sub-communities) can indeed help to isolate interested nodes (vertices) and valid for many complex networks such as [5, 6, 1].

There is always possibility that a community structure may contain different small sub-communities (sub-partitions) or represent itself as a one large single community. Figure 3.1(a) illustrates a single graph which further splits into four sub-graphs, represents a hierarchal structure of a community. It is not easy to detect hierarchal structures within a community in terms of computer science. There are many question associated with the detection of hierarchal structures such as how to partition a community (logical division of sub-communities inside a community), how to compare those hierarchies and which partition is best among them with strongest community structure. One approach is to detect all partitions first and then compare them, however it is not practical to detect all partitions at the same time due to computational limitation as the number of partitions grows exponentially with graph size. Another issue arises when if there is no partition at all, as this must be recognized as well. Another aspect is to identify the clusters in communities; this is only possible if the graphs are sparse that mean it make no sense to identify clusters in communities if numbers of edges are much higher than vertices itself because of too many homogeneous edges.

Overlapping Communities

In any community structure, nodes can belong to more than one community. For instance, consider the analogy of a company where a manager may be responsible for two different departments or in a social network, a person may belong to different communities such as family or friends and thus have different relationships with other members based on particular community scenario. In such a case, algorithms need to deal with overlapping or non-overlapping communities, as shown in figure 3.1(b), where the green nodes in the middle belong to multiple communities. Locating overlapping nodes that represent the boundary of any community is perhaps one of the most difficult tasks for any community detection algorithm to deal with. In simple words, one can easily detect an overlapping node by estimating its degree of participation in different communities. However it is not an easy task for community detection algorithms, because one has to make assumptions by putting a node forcefully into a single community in order to run a particular algorithm. This assumption is not always true and not fit for all real community networks [22]. Standard algorithms may identify overlapping nodes by checking the stability of partitions by applying different variations in the community structure [22], but most community detection algorithms do not address this problem. G. Palla et al. [32] presented a very interesting algorithm to detect overlapping communities, which is discussed in section 3.1.3.

3.1.3 COMMUNITY DETECTION ALGORITHMS

Community detection algorithms can be classified into two broad categories based on the nature of their purpose: Hierarchy detection algorithms and Overlapping detection algorithms. This section discusses some of important algorithms relevant to the community structure detection.

Newman & Girvan's Modularity Function

When defining any community structure, there is always probability of large number of partitions in any given community. It is always important to identify those partitions that possess real community structure. In order to identify the quality of a partition in a community structure, a quality function is needed that uses a quantitative criterion to determine the goodness of a partition. The quality function proposed by Newman & Girvan [30] has become the basic criteria for many researchers in determining the quality of partitions. The equation can be written as [27]

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta\left(C_i, C_j\right) \qquad (3.2)$$

Where; $\sum ij$ compute for all pairs of vertices in graph. A is the adjacency matrix. The value of A_{ij} of adjacency matrix is 1 if vertices i and j are connected and 0 if not connected. K_i is the degree of vertex i and K_j is the degree of vertex j. m represents total number of connection (edges) of the graph. δ is equal to 1 if vertices i and j are belong to the same community otherwise value is 0.

In equation 3.2, total sum includes only those values, when vertices belong to the same cluster i.e. δ is equal to 1. There is no point including those values when vertices are not in the same cluster, because in that case δ is equal to 0, so it has no contribution in overall equation. Based on above information the equation 3.2 can also be written as [27]:

$$Q = \sum_{s=1}^{n_m} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m}\right)^2\right] \tag{3.3}$$

Where; n_m is the number of modules in a graph, l_s is the number of connection (edges) joining vertices present in module s, d_s is the sum of the degrees of the vertices of s, and m is the total number of connection (edges) present in a whole graph.

Equation 3.3 gives two important pieces of information, the first term deals with the real graph, where it calculates the fraction of edges present within module. The second term deals with the random graph and estimates the expected number of edges, with the assumption that each vertex has similar degree ratio. In this situation, a vertex can form a pair with any other vertex in the graph and the probability of their connection is proportional to the product of their degrees.

Based on the information provided by eq. 3.3 the definition of community is when numbers of internal connection (edges) among vertices (nodes) are higher in sub-graph (of a graph) as compared to the expected number of connections in the modularity's null model, then that sub-graph is a module because vertices are much tightly bonded inside than outside. Summation process in eq. 3.3 shows that if the result of the summation is positive this means that sub-graph is a module, where difference between real and expected edges is much higher. If the result is negative, it means that the sub-graph has vertices with negative value.

This shows that the graph has no community structure or one might say that if the negative value is very large this could means that sub-groups have very a few connections inside than outside. Therefore, large number of positive value of Q shows that there are good partitions present in a graph.

The modularity function has become basis for many other algorithms to identify partitions in the graph and has opened many doors for researchers to explore new areas. Coming section will discuss some algorithms which are based on this original modularity function of Newman and Girvan.

Edge Centrality Algorithm

The edge centrality algorithm proposed by Girvan and Newman in [31] has become a benchmark for community detection research. The algorithm is based on the concept of *edge centrality*, where a measure known as betweenness is used to calculate the number of shortest paths between pairs of nodes that come across it. In case, multiple shortest paths are present between a pair of nodes then each path will be assigned an equal weight such that the net weight of all paths is equal to 1. If a network community structure comprises of different small sub-communities and these sub-communities are connected to each other with very few edges, then it is certain that the shortest path between these sub-communities must go through some or one of these edges. Therefore, the edges that are participating among sub-communities have high edge betweeness and by removing these edges the sub-communities can be isolated so that underlying community structure of the network can be discovered.

The steps of algorithm are:

1. Calculate centrality of all edges present in the graph.
2. Eliminate the edge, which has highest centrality.
3. Recalculate centrality of remaining edges.

4. Keep doing this process from step 2, until partitions get isolated.

The authors also discussed three different definitions of betweenness: edge, current-flow and random walk. Among these definitions, the edge betweenness proved to be more fast and scalable in time as $O_{(mn)}$. The techniques used to calculate edge betweenness are generally based on breadth-first-search (BFS) [32] [33]. In graph theory, BFS is a searching algorithm, its starts searching from root node and then discovers all of the neighbouring nodes then explore undiscovered neighbours of the neighbouring node and keep on doing it until finds the target.

Random Edge Betweenness Algorithm

In [34] Tyler proposed a modified form of the Girvan-Newman algorithm. The purpose of this algorithm is to improve the calculation time of the algorithm. In order to gain speed, rather than calculating the betweenness values of all pairs, the algorithm chooses vertices in pairs randomly and calculates the edge betweenness between those pairs. Due to random selection, there are statistical errors present in this algorithm; as a result the vertices present at the boundary between partitions may be assigned to different partitions at different times. However, the algorithm proves to be very fast in comparison to the original algorithm of Girvan-Newman, but it introduces potential partition error because a node that belongs to one partition may show member of another partition on next simulation process.

Information Centrality Algorithm

In [36], Fortunato proposed an algorithm that is based on information centrality, where edges with the highest information centrality are removed in an iterative manner to obtain hierarchical partitions in graph. The concept is based on efficiency [37], which determines how quickly information transfer from one vertex to another, using the shortest paths in a graph. This algorithm removes edges based on the decreasing value of information centrality.

This method produces good results for dense communities by better classification of vertices. However, this algorithm appears to be slower compare to the Girvan-Newman's algorithm, it scales as $O(n^4)$ on sparse graph.

Random Walk

In [38], Zhou uses the concept of random walks to identify communities. The idea is that if the community is highly populated this means that density of edges is high as well, therefore, a random walker spends lot of time inside a community may become part of a communication process as edge betweeness for different pairs of vertices. So, random walks calculate the distance between pairs of vertices. A distance between two vertices in a graph depends on the number of edges a random walker has traversed to reach from source vertex to destination vertex. The concept of local and global attracter is used, the local attracter should always put near to a vertex as a neighbour, thus forms a local community. The global attracter is the closest node to a vertex but not as neighbour, thus forms global community. A criterion is used in the graph to allow minimal number of sub-communities means no smaller sub-graphs are allowed in communities.

Zhou proposed successive algorithm that calculates the dissimilarity between vertices in the graph on the random walk concept. Basically this idea is similar to distance based on structural equivalence. Equation 3.4 [27] shows the dissimilarity measure based on the concept of structural equivalence, where a vertex is consider structurally equivalent to another vertex if both vertices have same number of neighbours, irrespective of distance between them. Whereas, if two vertices have different number of neighbours, they are considered very distant from each other even though they are very close in reality.

$$x_{ij} = \sqrt{\sum_{k \neq i,j} \left(A_{ik} - A_{jk}\right)^2} \qquad (3.4)$$

Clique Percolation Method

Almost all algorithms discussed so far deal with the detection of partitions or hierarchies in a community. Very few researchers have proposed techniques to solve the problem of overlapping communities. One method proposed by Palla et al. in [39] can trace the community structure in a graph as well as deals with overlapping nodes that may present in multiple communities. The concept is based on the cliques formed by the edges in the graph. The edges that are present inside a community are more likely to form cliques because of their strong relationships, while the edges that belong to different communities will not show such kind of bond. The number of k vertices to define k-clique as a complete graph. There are different assumptions used, which are:

- Two K-cliques will be considered as neighbour if they share k-1 vertices.
- Merging of neighbouring k-cliques will form a chain, called as k-clique chain.
- Two k-cliques can communicate to each other if they are connected to the k-clique chain.
- The largest community that can be detected from the graph is identified by combining all k-cliques which are connected to each other, thus forming a chain and known as k-clique community.

By nature, k-clique communities share their vertices, so there is always a possibility of overlapping in the graph. The vertices that do not belong to the neighbouring k-cliques but are part of the k-clique chain can be reached via different paths and may lead to a different partitions or clusters. The searching of k-clique communities in a graph comes with the expense of excessive time that grows exponentially with the graph size. However, the author claims that it is reasonably fast and takes very less time compare to the anticipated time for the graphs with up to 10^5 vertices. This algorithm has been used in many other cases such as analysis of community structure in social networks [40], weighted [41] and directed [42] graphs and provided satisfactory results.

Overlapping and Hierarchical Algorithm for complex networks

In [25], an algorithm that claims to be the first algorithm that can detect both overlapping and hierarchical community structure in complex networks is presented. The method proposed, searches the natural community of each node by exploring the network locally. For this purpose, nodes may be visited multiple times irrespective of whether they belong to some community or not. In this way overlapping communities can be naturally recovered. For detection of hierarchies, a resolution parameter is introduced, by varying the value of this parameter one can estimate the size of communities and eventually explore all hierarchical levels present in the network.

In this method the sub-graph that can be identified from graph is based on the fitness of it nodes. An expression used in this algorithm to calculate fitness of a node is given as [25]:

$$f_{\mathcal{G}} = \frac{k_{\text{in}}^{\mathcal{G}}}{(k_{\text{in}}^{\mathcal{G}} + k_{\text{out}}^{\mathcal{G}})^{\alpha}} \qquad (3.5)$$

Where; $f_{\mathcal{G}}$ is the fitness of sub-graph, $k_{in}^{\mathcal{G}}$is the total number of internal degrees of nodes in module $\mathcal{G}$, $k_{out}^{\mathcal{G}}$ is the total number of external degrees of nodes in module $\mathcal{G}$ and α is a resolution parameter used to control the size of communities and must be in positive real value.

This algorithm was tested on artificial networks where the hierarchical and overlapping structure of the graph was already known and algorithm seemed to work as expected. Then, they applied this algorithm to some real networks and algorithm produced satisfactory results. The authors also claimed that this algorithm can be extended to weighted networks where edges carry weight.

The Diplomat's Dilemma

In [74], authors emphasis on the position and influence of the node in the network. According to the authors it is goal of the professionals to maintain power or position of influence in the society; in other word it is known as "*The diplomat's dilemma"* i.e. a person has a strong influence in a society if he has relations with many people. This is exactly a diplomat wants that he can influence many people in society with minimum effort of making personal relations (more power less connections/low cost).

In this paper, an agent has the power to take strategic decision in the society, the agent keep optimizing their score functions by adopting different strategies based on six factors i.e. MAXD, MIND, MAXC, MINC RND and NO. These factors define how an agent makes it ties with others up to the range of 1^{st} degree network neighbourhood relationship. The increase in score function describes the influence of that agent in the society. The drawback of this work is that there is possibility that node starts to selfishly optimise their positions in the network where they require less number of connections without knowing the network than the second neighbourhood.

The Lobby Index

In [75] authors presented a solution for The Diplomat's Dilemma problem by giving the concept known as *lobby index.* In lobby index, a node has high lobby index if it has neighbours having at least equal or more neighbours then node itself in current communication environment. Mathematically [75]:

$$l(x) = \max\{\, k : \deg(y_k) \geq k\} \qquad (3.7)$$

Where, $l(x)$ = Lobby index of node x

k= Degree on node

y_k= Neighbors of node x; k=1, 2,...n

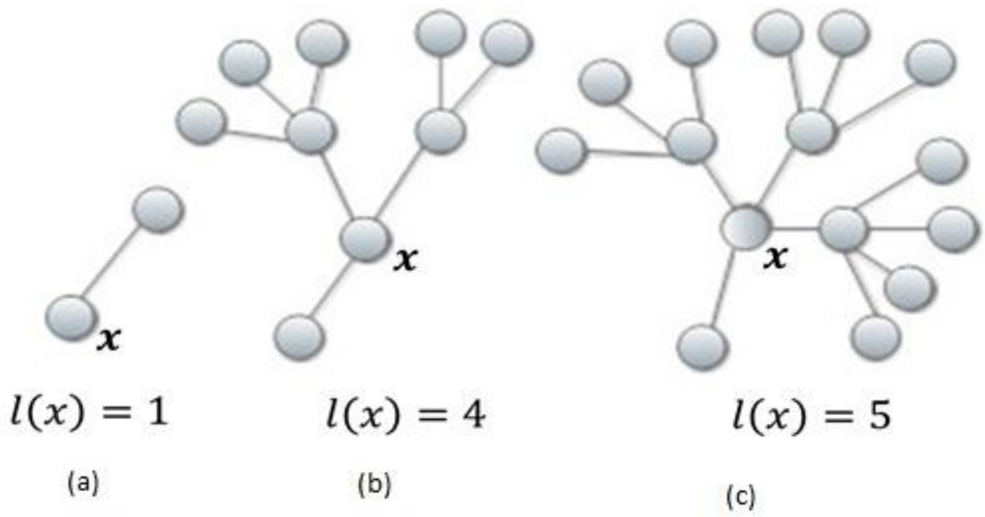

Figure 3.2 Lobby Index concepts

In figure 3.2 the concept of lobby index is depicted. Fig. 3.2(a) shows that when only two nodes are communicating with each other, the lobby index of node 'x' is 1. In fig. 3.2(b), one of the neighbours of node 'x' has 4 neighbours; therefore lobby index of node 'x' is 4. Similarly fig. 3.2(c) shows that lobby index of node 'x' is 5 because one of its neighbours has

5 neighbours. A node having high lobby index means that it has neighbours having high connections in the society, thus have high reach. This means that this node can refer messages to its neighbours for the dissemination in the network at the same time keeping its cost low.

3.2 GRAPH THEORY DRIVEN OPPORTUNISTIC FORWARDING ALGORITHMS

In scale free networks such as opportunistic networks, where there is no prior knowledge of number of nodes, a lot of study is done to find out different centrality measures based on graph theory in order to improve efficiency of communication in community networks. Efficient communication in any society or ecological system means that a node has more knowledge of nodes or high access/reach in network with minimum price to pay (in terms of resource usage). Over the years three measures have become the standard for centrality measure in scale free networks: betweenness [71], degree [72] and closeness [73]. Based on these centrality measures, many researchers [44, 46, 74, 75] have proposed new interesting ideas based on graph theory to target those nodes in the network which have influence or important positions and use them as message forwarder to reduce the communication cost of the network. This section provides some of very well-known opportunistic forwarding algorithms that employs concept of graph theory.

Bubble Rap Algorithm

In [44], Hui et al introduced a social forwarding algorithm based on human mobility traces in social community structures that can be implemented in Pocket Switched Networks (PSNs). The algorithm is known as Bubble Rap (BR) and deals with two aspects of a society: community and centrality. Human society is structured and people living in that society make communities. In those communities some people are more famous and have more relationship to others thus have high centrality. The idea behind the Bubble Rap forwarding algorithm is to exploit popularities of people in society, since people living in a society have different roles thus have varying popularities levels. The popularity of the nodes in BR is calculated on the average of previous course of encounters with other nodes using betweeness centrality. Betweeness centrality defines how many times a node participates as message forwarder between source and destination. In PSNs, allowing popular nodes to take responsibility as message forwarder help reduce the communication cost and at the same time increase the probability of message delivery to intended destinations. Forwarding strategy of message in Bubble Rap algorithm is based on:

1. Locate more popular nodes than the current node and forward messages to newly located nodes.
2. Identify the nodes that belong (or may take one step closer) to the target communities and use them as relays.

This algorithm based on the following two assumptions:

- Each node must be assigned to at least one community and nodes may only belong to one community.
- Each node has a global and local ranking. Global ranking means a node's popularity (centrality) across the whole graph. Local ranking means a node's popularity across the local community (sub-graph). If a node is a member of multiple communities then it has multiple local rankings.

Authors proposed three different algorithms as part of BR algorithm for the detection of communities, namely SIMPLE, K-CLIQUE and MODULARITY. Based on their experiments they recommend that K-CLIQUE has 85% accuracy. Authors also proposed that centrality measured in the past can be used as future predictor by taking degree average in unit times throughout experiments, referred as DEGREE of a node. This DEGREE represents the global and local centrality values of a node. However, during experiments they found out that total degree (unique nodes seen by a node throughout experiments) is not good approximation for node centrality measurement. For this purpose, two types of windows are recommended: Single (S)-Window and Cumulative (C)-Window, which calculates degree in per unit time. S-Window compares how many unique nodes found by comparing it with pervious unit time slot (i.e. 6 hours recommend by authors). C-Windows takes the average of all previous unit time windows (i.e. 6 hours). Among these three techniques, C-Windows technique proved to be the most efficient as recommended by authors.

Bubble Rap algorithm seems applicable for the practical implementation of PSN. This message forwarding method tends to direct the traffic towards its intended destinations and also reduces redundant traffic in the network. However, Bubble Rap has its own concerns; it increases the load of popular nodes in the network and the buffer on these nodes can overflow very quickly, this can make a node disappear from the network very quickly. Furthermore, while the authors anticipated that it would work for hierarchical structure, it has only been tested for flat community structure.

Human Mobility Predictability

In [46], Hui presented an algorithm that can improve forwarding efficiency in terms of delivery ratio and delivery cost by predicting human mobility patterns in day-to-day base. The experimental data set used for this algorithm test is taken from reality mining project. The algorithm basically uses vertex similarity to measure the predictability of human interactions for this purpose classic Jaccard measurement is considered for vertex similarity. The idea is simple, compare vertex similarity of the contact graphs over two days and extract the human interaction similarity. The result presented in this work was based on Jaccard measurement and mathematically the equation is given as [46]:

$$\sigma_{Jaccard} = \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i \cup \Gamma_j|} \qquad (3.6)$$

Where; Γ_i is the neighbourhood of vertex i, which is connected to set of vertices connected to vertex i via an edge. $|\Gamma_i|$ is the cardinality of set Γ_i, that is equal to the degree of the vertex i.

SimBet

In [83], the selection of forwarding nodes is based on two attributes: similarity utility and betweeness utility. Selection of best message forward carrier may cause decision problem due to presence of multiple attributes. This problem is addressed by using a pairwise comparison on the normalised relative weights of the attributes. Mathematically [83]:

$$SimBetUtil_n(d) = \alpha \frac{Sim_n(d)}{Sim_n(d) + Sim_m(d)} + \beta \frac{Bet_n}{Bet_n + Bet_m} \qquad (3.7)$$

Where; α and β are tuneable parameters and $\alpha+\beta=1$.

PEOPLERANK

In [84], the approach used is derived from the idea proposed in PageRank [85]. PageRank was originally used by google for the ranking of web pages; this algorithm measures the relative importance of the page within the web. Inspired by the concept of PageRank, PeopleRank has proposed similar technique in a social graph to rank the nodes. The idea is that nodes with higher PeopleRank are considered more socially active and thus connected to other important nodes of the network. As a result communication cost is significantly reduced by targeting higher PeopleRank nodes that plays central role in opportunistic network.

3.3 IMPLICATIONS FOR RESEARCH

As a result of this study, this thesis realises that the very same concepts of community detection are also applicable in opportunistic networks, where important nodes in the network can be identified by using graph theory concepts. These important nodes can be effectively used for the purpose of content forwarding in opportunistic networks. For instance, Bubble Rap algorithm uses the concept of node's popularity; this idea is taken from the analogy of human social relationships in a society. A person who is more popular in a society has naturally more information about other people, Bubble Rap takes this idea and allow nodes to forward messages to those nodes which are more popular. Lobby index is another concept that also utilises human social relationship analogy. This concept is based on the analogy of lobbying such as lobbyist or politician, where aim is to keep the knowledge of those people who have high connections. By making such connections, a lobbyist makes sure that he has access to majority of a society without any effort of making personal connections to each individual, less effort more power.

This thesis aims to design an efficient content forwarding technique for opportunistic network. By studying Bubble Rap content forwarding algorithm, it is revealed that there are some problems associated with it. Bubble Rap content forwarding has decreased communication cost significantly. However, this kind of message forwarding has drawbacks as well; it will exert more pressure on popular nodes and resources of these nodes can deplete very quickly as a result nodes can be forced to shut down. Furthermore, delays in Bubble Rap content forwarding will also increase because popular nodes only forward messages to more popular nodes and if not able to find one, the message could be dropped either due to buffer full or TTL expires. This thesis believes that the problem associated with Bubble Rap can be addressed by incorporating lobby index concepts. Lobby index allows nodes to make connection with more influential neighbouring nodes that have more knowledge of the network, thus gives more access in the network. The idea of this thesis is not merely depend on popular nodes but also unpopular nodes with popular neighbours can be used in the network for opportunistic content forwarding. Such kind of content forwarding can improve over all message delivery ratios and also decreases delays. In future, this thesis investigates and implements this idea in a suitable simulation environment and obtained results will be compared with previously proposed algorithms such as epidemic and Bubble Rap algorithms.

3.4 Chapter Summary

This chapter highlighted the concept of community structures in social networks such that a community may comprise of hierarchal or overlapping structures. Detection of these structures in social networks can be uncovered by using graph theory concepts. Based on graph theory different algorithms are proposed that can detect community structures and indeed help isolate the intended nodes in social communities. Furthermore, this chapter also gave a survey of algorithm that uses graph theory concepts in opportunistic communication networks.

CHAPTER 4

PROPOSED ALGORITHM & APPROACH

Two major factors have influenced the design of the novel algorithm presented in this chapter. Firstly, if the popularity of a node based on the average of previous encounters in a community is exclusively used, this may be exhaustive in terms of cost because every other node in a community used always seeks the most popular nodes. Furthermore popular nodes have undesirable increased loading as they may have to keep the message for longer period of time unless a suitable node is found for further transmission. Secondly, the most popular nodes do not always guarantee that they actually lead information to its recipient or always resolve a query, there are more chances that TTL (time to live) of message may expire before it can actually be delivered to the destination or buffer of most popular nodes may exceeds as a result dropping of important messages. Of course in such decentralised environment nothing guarantees for hundred per cent accuracy in terms of message delivery.

This chapter proposes the novel algorithm Lobby Influence. Lobby Influence uses the concept of community and centrality as represented in [5] but incorporates the use of lobby index as represented in [75], please see section 3.2 and 3.1.3, respectively. Lobby index indicates how well connected a node is with its neighbours provided that neighbours have got equal or more degree than the node itself. This concept ensures that a node with high lobby has neighbours which are further well connected with their neighbour, thus have high reach in the network. This new algorithm extends the capabilities of Bubble Rap by targeting the influence of neighbours of neighbours and simultaneously, it ensures that these neighbours are currently well connected with each other. Lobby influence increases the probability of message delivery (good put) while minimising the overall delay.

4.1 LOBBY INFLUENCE FORWARDING DECISIONS

Based on the arguments discussed previously, the forwarding decision of Lobby Influence not only depends on the node popularity (Bubble Rap) (as discussed in section 3.2) but also on the strength of the neighbourhood relationships (high lobby index) (as discussed in section 3.1.3). In order to illustrate how this new algorithm increases the probability of message delivery and while also keeping the cost of popular nodes low, consider the situation 'A' in figure 4.1. If only Bubble Rap communication is present, the most popular nodes will be used for message transfer. In this situation, node '1' is more popular than node '6' and thus node '1' will not transfer message to node '6' because node '6' has low popularity level. Node '1' will keep that message until it finds suitable node or time to live (TTL) of message expires. However, as depicted, node '6' has high neighbourhood relationship in current network and thus transfer to this node would increase the probability of message transmission to its destination. Lobby influence addresses this issue and allows more popular nodes to transmit message to less popular nodes provided that node has a strong neighbourhood relationship (high lobby index) in current network. Thus allowing less popular node to take responsibility as a result pressure on most popular nodes become less.

Now consider situation 'B', if only lobby index based network communication is present. Node '26'has high lobby index and it is well-connected with its neighbours, however node '26' is not able to forward messages because it could not able to locate a suitable node in its network that can take information to the destination, therefore waiting for suitable node to connect to the network or TTL of message expires. Lobby Influence addresses this issue by allowing high lobby index nodes to transfer messages to those nodes which are more popular, thus allowing messages to keep on forwarding until destination meets. Therefore, using Lobby Influence node '26' will transfer message to node '29', as although it has low lobby

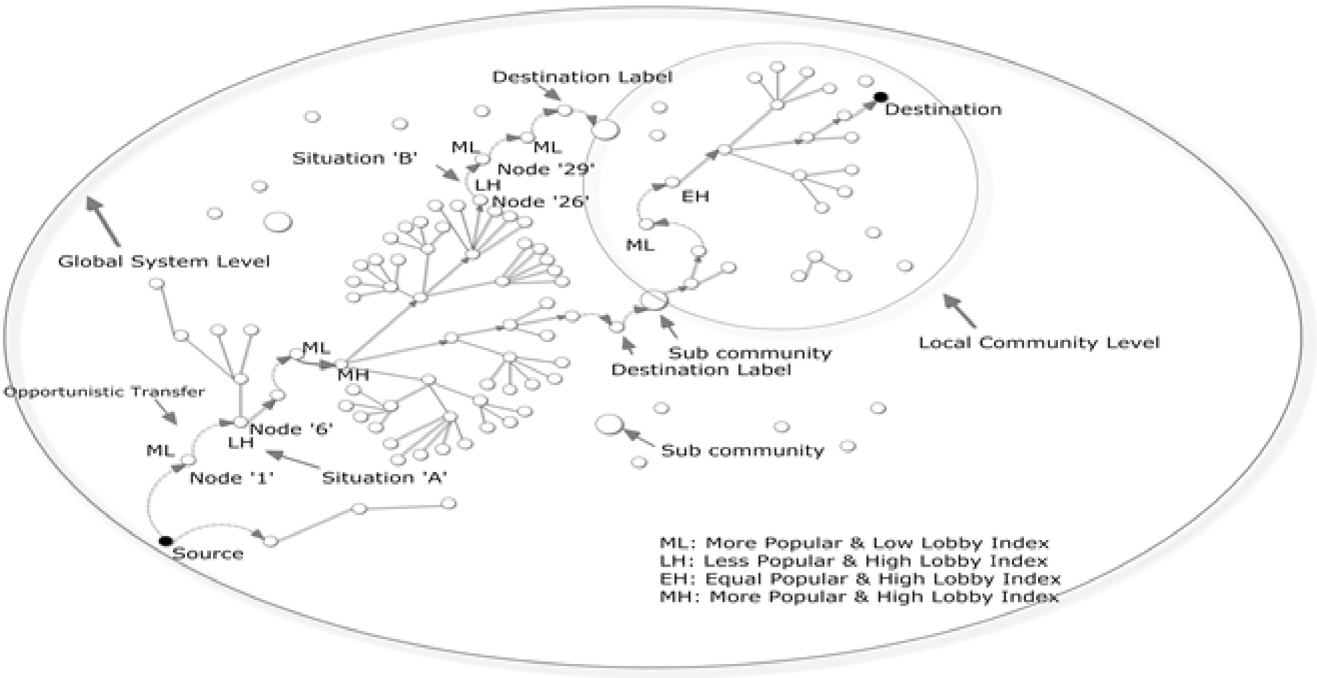

Figure 4.1 Sketch of Lobby Influence

index but it has a high popularity level and as a result improves the probability of message delivery significantly.

In summary, lobby influence addresses the shortcomings of both Bubble Rap and Lobby Index. In the case of Bubble Rap, by allowing a message to be transferred to node with high lobby index irrespective of their popularity means that now messages may be transfer to those nodes which are less popular but have high lobby index. In case of Lobby Index, when a well-connected node is unable to find destination in its network it can now rely on the concept of Bubble Rap by allowing transfer of message to popular nodes, to keep forwarding the message until destination meets. This new strategy has increased the probability of the message delivery at the same time decreased the overall delay.

4.2 THE ALGORITHM

Table 4.1 summarises different variation used in the forwarding decision for Lobby Influence as a combination of popularity and Lobby Index. Based on these variation this algorithm is little bit tilted towards Lobby Index, closely look at variation (d), (f), (g) and (i) as shown in table 4.1. The current node transmits a message to the encountered node irrespective of its popularity. This is because the LI algorithm supports the fact that Lobby Index has more significance because it represents connectivity with its neighbour of neighbours in the current network; whereas popularity of node depends on the previous encounters more optimistic compare to the Lobby Index. If we consider variation (c) this algorithm will exactly behaves like Bubble Rap. In the situation where a node encounters two different nodes at the same time, the forwarding decision will be made on the basis of lobby index priority. If lobby index of both nodes are equal in that case, the decision will be based on popularity level. However if the popularity is low and the lobby index of both nodes are equal than nodes with equal lobby index can forward messages provided that lobby index of both nodes are greater than 1.

The Lobby Influence is the extension of Bubble Rap algorithm, therefore the Label scheme [76] used in BR has been adopted. Each node is given a label which describes the node's associativity with a community (a node may have associativity with different communities thus can have multiple labels). Those nodes will be targeted for message delivery whose community association is similar to the destination node or keep on forwarding the message until member of targeted community is encountered. For this algorithm, three assumptions are considered:

- Each node must have label to show its association with at least one community.
- Each node has one global rank and one local rank to define its global centrality (popularity) across the whole system and local centrality within its local community, respectively. A node may have multiple local ranks due to its association with different local communities, thus may have multiple labels.
- Each node has its lobby index which indicates how well a node is connected to its neighbours in current network.

Table 4.1 Lobby Influence variation table

	Bubble Rap: Popularity level (Global or Local)	**Lobby Index**	**Lobby Influence ***
a)	High (H)	High	HH → Transmit Message
b)	High	Low (L)	HL → Transmit Message
c)	High	Equal (E), if index>1	HE → Transmit Message
d)	Low	High	LH → Transmit Message
e)	Low	Low	LL → Keep Message
f)	Low	Equal, if index>1	LE → Transmit Message
g)	Equal	High	EH → Transmit Message
h)	Equal	Low	EL → Keep Message
i)	Equal	Equal, if index>1	EE → Transmit Message

*HH= First two letters refer to popularity and lobby index, respectively, H=high, L=low, E= equal (valid for all subsequent routing decisions)

The algorithm (see Fig. 4.2) works in such a way that a node may come across with two situations a) Node within a community b) Node within Global system.

a) Node within a local community: The first part of algorithm deals with the situation, where a node is present inside a local community. This means that the label of current node is equal to the label of destination node and local rank (popularity) with lobby index will be used to take decision for message forwarding. If the encountered node has high lobby index irrespective of its rank (either high or low) the message will be transferred. However, if the lobby index of encountered node is similar to the current node then the message will be transfer based on high rank. If none of these conditions meet then node will keep this message to itself until finds some suitable node or TTL of message expires.

```
Begin

Foreach (encounteredNode_i) do
        if (getLabel(CurrentNode) == getLabel(Destination))
        {
                if (getLabel(EncounteredNode_i) == getLabel(Destination)) &&
                getLobbyInfluenceLocalStatus(EncounteredNode_i) == true)
                {
                EncounteredNode_i.addMessageToBuffer(message);
                }
        }
        else
        {
                if (getLabel(EncounterNode_i)==getLabel(destination)) ||
                getLobbyInfluenceGlobalStatus(EncounteredNode_i)==true)
                {
                EncounteredNode_i.addMessageToBuffer(message);
                }
        }
End


getLobbyInfluenceLocalStatus(EncounteredNode_i)
        {
        if ((getPopularityLocal(EncounteredNode_i)==high) ||
        (getLobbyIndex(EncounteredNode_i)==high) ||
        (getLobbyIndex(EncounterNode_i)==Equal))
                return true;
        else    return false;
        }

getLobbyInfluenceGlobalStatus(EncounteredNode_i)
        {
        if ((getPopularityLocal(EncounteredNode_i)==high) ||
        (getLobbyIndex(EncounteredNode_i)==high) ||
        (getLobbyIndex(EncounterNode_i)==Equal))
                return true;
        else   return false;
        }
```

Figure 4.2 Lobby Influence algorithm

b) Node within Global system: The second part of algorithm deals with the situation, where a node is looking for a destination at global level. At this level source or encountered node keep forwarding the message until finds a suitable node that belongs to the same community as the destination node. Here an assumption is made that whenever a node finds a member of a destination community the current node will transfer message to the encountered node and remove the original message from it database to reduce the cost, with the hope that the member of a destination community will lead info to the destination community or destination itself. In order to forward message to a suitable node at global system level, when a node meets with another node, the message will transfer to the encountered node if the lobby index of the encountered node is high (irrespective of global rank). However, if the lobby index of the encountered node is similar to the current node the message transfer will occur on the basis of global rank (popularity). If none of the condition is met then source node will keep the message into its database until if finds suitable node or remove it if TTL of the message expires.

4.2 Chapter Summary

This chapter has provided the concept and motivation behind Lobby Influence algorithm. The new algorithm is based on two observations such that in opportunistic networks not only popular nodes but also un-popular nodes with popular neighbours can be considered as efficient messages forwarder. By targeting these nodes high access and reach in the network can be achieved, as a result high delivery rates and latency is expected.

The next chapter will discuss the simulation setup and mobility models used to simulate proposed algorithm and two other algorithms i.e. Bubble Rap and Epidemic. Furthermore, comparison graph of these algorithms and discussion will be presented.

CHAPTER 5

SIMULATION AND RESULTS

This chapter presents the results for new algorithm known as Lobby Influence (LI). In order to measure the performance of this algorithm, two other algorithms namely Bubble Rap (BR) [5] based on more famous node selection and Epidemic [64] based on flooding are considered for comparison.

5.1 SIMULATION SETUP

To evaluate these algorithms a simulator known as Opportunistic Networking Environment (ONE) [77] is used, which is specifically designed for delay tolerant networks. For this simulation, mobile phones or similar devices with Bluetooth are considered as communication means between mobile users. Most of the devices in this simulation are operating at 2 Mbit/s data rate with 10m range. Algorithms are evaluated by varying the queue size of the nodes. Queue size affects the overall performance in network communication; as queue size increases, the number of messages delivered at destination increases as well. This is natural; with large queue size more messages can be stored and less risk of dropping as a result message delivery increases. In order to fully justify the algorithms performance, it is very important that the mobility of nodes should be very close to the real world. Three synthetic mobility scenarios are taken into account: Random Way Point (RWP) [77], Helsinki City Scenario (HCS) [77] and Working Day Movement model (WDM) [78], these represent different type of mobility patterns. Each scenario is relatively different due to node movement patterns and thus have different simulation world sizes. Detail of each mobility scenario along with simulation parameters in tabular form will be discussed in later sections. Two algorithms Bubble Rap and Lobby Influence are logically very close to each other, because LI extends the characteristics of BR by implementing Lobby Index concept on top of BR algorithm. To ensure the maximal performance of BR, the recommendations made by authors in [5] (as discussed in section 3.2), for the best efficiency is used in this experiment. It is possible that there are superior settings for LI, but for the experiment conducted in this thesis, settings shown in Table 5.1 is valid and satisfactory. Therefore, same settings are used throughout the course of experiments. Only exception is “Familiar threshold parameter” which is adopted according to ONE simulator requirement by Dillon [87]. “*A node using K-Clique keeps a record of all the nodes it has met and the cumulative contact duration it has had with each. Once this total contact duration for one of these nodes exceeds a configurable parameter “Familiar threshold”, the node is added to the host’s familiar set and local community and the node's familiar set is added to an approximation of all the familiar sets of the host's local community”*[87]. The value of K labels the parameter used in K clique community detection algorithm [39] (as discussed in section 3.1.3) to trace the community structure in the network, which is part of both BR and LI algorithms. To obtain an optimal value of K as suggest in [5], BR and LI both are tested against different values of K i.e. 1, 2, 3, 5 in WDM scenario and K=3 proves to be the optimal value for this simulation, as shown in figure 5.1.

Table 5.1 Algorithm settings for BR and LI

Parameters	**Value**
Community detection algorithm	K Clique
K	3
Familiar threshold	700
Centrality algorithm	C window
Centrality time window	3600 (s)
Computation interval waiting time	300 (s)
Number of time intervals to average	3

Several experiments are performed for each scenario; RWP, HCS and WDM are each simulated 30, 15 and 10 runs with different seeds, respectively and present the mean values in resulting graphs. The reason for using different scenarios with different numbers of runs is because of computation cost. RWP scenario seems to be quickest to simulate, HCS comes 2nd and WDM takes lot of time to finish just one experiment. For these experiments, total 24 dual core linux based computers are used with 8 GB RAM each. The algorithms performances are measured against three metrics: 1) *Message delivery ratio:* In ONE simulator, delivery ratio is considered for the entire network by taking average across all nodes, it is the ratio of total no. of messages received at all destinations to the total no. of messages generated by all sources. 2) *Delays*: how long it takes for messages to reach at destinations and 3) *Forward messages*: defines the cost in terms of the exchange messages between nodes, which ultimately effects the system resources (bandwidth and energy) utilisation. In ONE simulator, nodes generate messages throughout simulation time period (from start to the end of simulation) between message creations intervals defined in simulation settings.

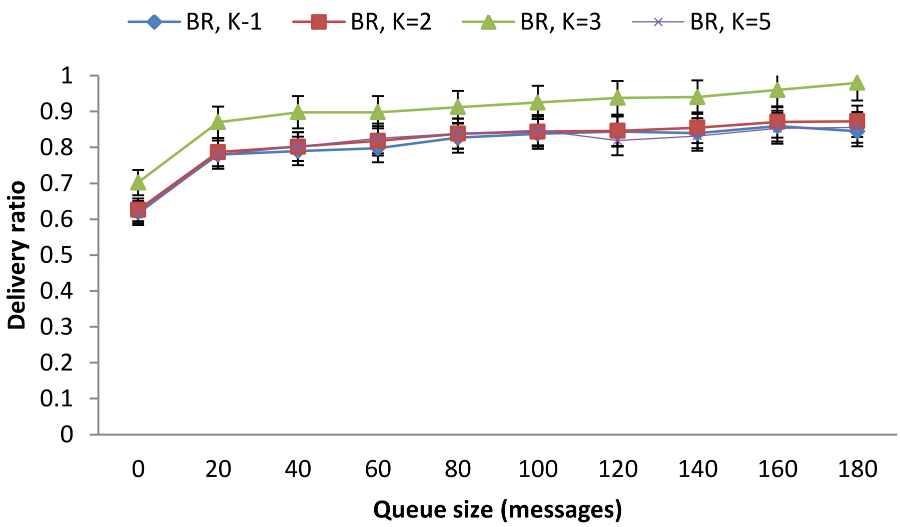

Figure 5.1 Optimum K value establishment for simulation

Each of the graphs given here contains 3 curves represents behaviour of BR, epidemic and LI algorithms, respectively. The x-axis in each graph represents different queue sizes, against which the performance of each algorithm is measured. The next sections: 5.2, 5.3 and 5.4 discuss different mobility scenarios with experiment parameters and results.

5.2 Random Way Point (RWP) Mobility Scenario

As a starting point, the random waypoint movement model is considered for nodes movement, as described in [77]. In this model random coordinates are given to each node in simulation area. Nodes follow the zigzag paths until they reach their coordinates. All nodes are considered as pedestrians moving in random speed of 0.5-1.5 m/s with pause time of 1-1500 s. Pause time means that after reaching their coordinates, nodes wait for a certain period of time and then moves to the next random coordinate. A screen shot of RWP movement model is shown in Fig. 5.2.

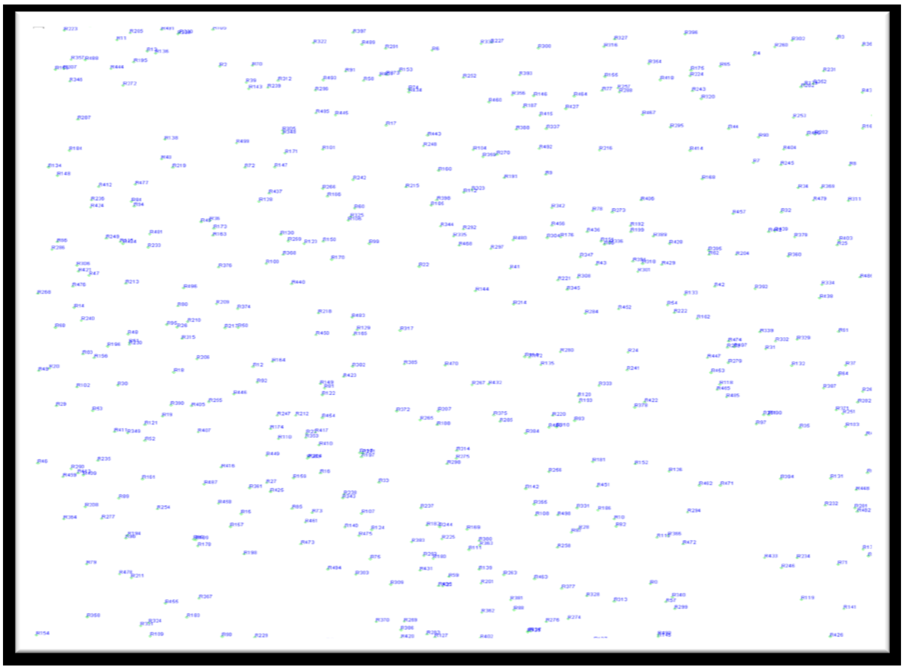

Figure 5.2 Screen shot of RWP model in ONE simulator

Table 5.2 summarize the parameters used in this experiment. The world size of movement models is in meters. Before commencing the real message transfer nodes are allowed to move in the world for a certain period of time. The movement of nodes are mimicking according to the average speed of human with the certain amount of pause time. The simulator selects source and destination nodes between 0-400 nodes, randomly. Messages are created throughout simulation time with the time interval of 25-35s.

Table 5.2 Parameters used in RWP scenario

Parameters	**Value**
World's size for Movement Model	8300 X 7300m
Total simulation time	430000s
No. of Hosts	500
Message TTL (time to live)	1344mins
Time to move nodes in the world before real simulation commence	43000s
Nodes speed (pedestrians)	0.5-1.5 m/s
Nodes pause time	1-1500s
Message sizes	500KB-1MB
Message creation interval	25-35s
Air data transmit speed	100kbps
Transmit range	10m
Range of message source/destination addresses	0-400 nodes
Queue sizes	0,20,40,60,80,100,120,140,160,180 in MB
No. of each experiment runs	30

5.2.1 RESULTS

Fig. 5.3 represents the overall delivery ratio for the entire network by dividing total no. of messages received at all destination nodes to the total no. of messages generated by all source nodes. The figure shows that epidemic algorithm performs outstandingly compare to BR and LI. Since the movement of nodes in RWP scenario is random and the epidemic algorithm rely on controlled flooding therefore performance is very good. However, when comparing BR and LI, LI seems to perform better than BR in random scenario. BR relies on the popularity of the node; therefore the current node will not forward messages until it finds a more popular node than the node itself. Whereas, in case of LI which not only depends on popularity but also on lobby index of the node which defines the connectivity of node with its neighbours in current network, as a result LI has more reach in the network and thus shown improvement in message delivery. Therefore, in RWP movement model where movement of nodes are unpredictable, Lobby Influence showed better results compare to the Bubble Rap.

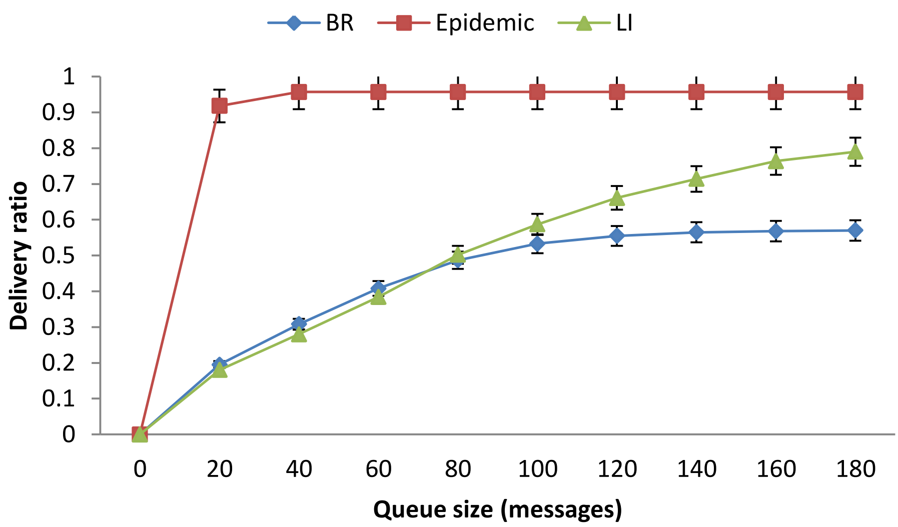

Figure 5.3 Average delivery ratio in RWP mobility scenario

Fig. 5.4 represents average delivery delays caused by different algorithms during communications. Again epidemic seems to be quickest in terms of message delivery in RWP mobility scenario. However, LI proves to be faster compared to BR. By looking at the average latency graph it seems that as queue size increases delay in BR and LI also increases. This phenomenon is due to the fact that with increase in queue size more messages can be stored in buffer for longer period of time, which were previously been dropped due to small queue size. Hence this delay is associated with increase in message delivery.

Fig. 5.5 shows average number of messages forwarded by each algorithm during communication in normalised form. It is evident from the graph that the epidemic algorithm has high overhead and forwarded more messages, because Epidemic routing relies on flooding, thus forwards messages to any node it encounters. Whereas, LI has more overhead compare to the BR, because BR only forward messages to more popular node, whereas LI forward messages to either more popular or well-connected (high lobby index)nodes.

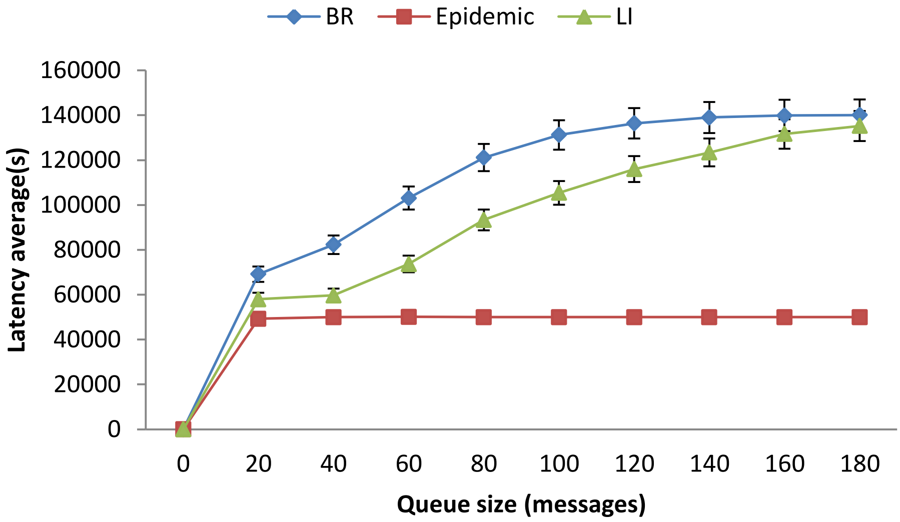

Figure 5.4 Average latency in RWP mobility scenario

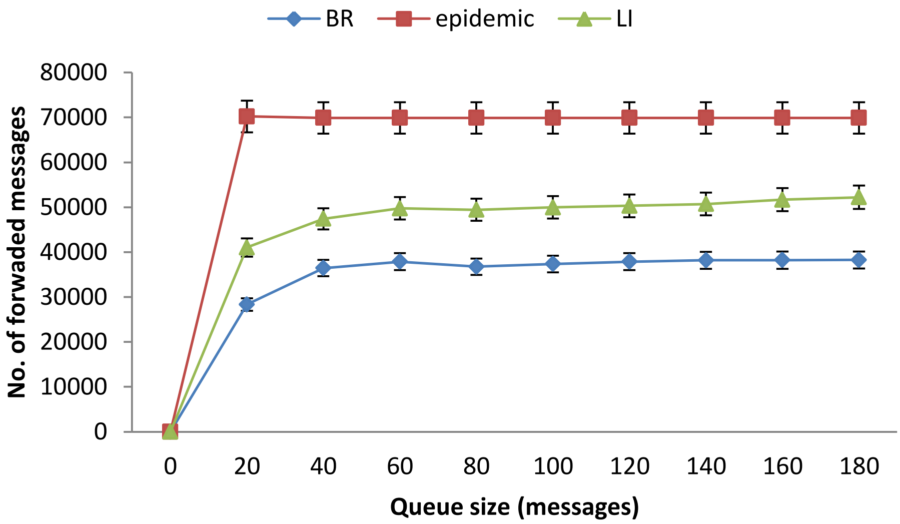

Figure 5.5 Average no. of forwarded messages in RWP mobility scenario

5.3 HELSINKI CITY SCENARIO (HCS)

This scenario uses simple map-based movement model as describe in [77]. Six trams are used which follows predefined path. Two thirds of nodes represent pedestrians and one third as cars. Cars and trams are running at different speed with different pause times. Pedestrians and cars use the shortest paths to reach to their randomly chosen destination. Cars are running on roads and pedestrians are walking on footpaths. A screen shot of HCS movement model is shown in Fig. 5.6.

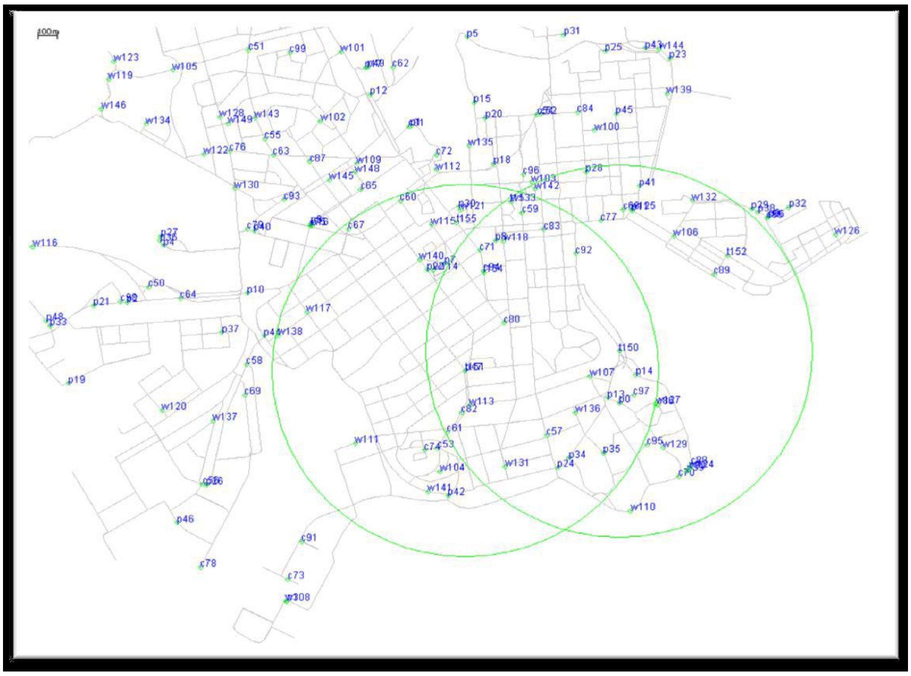

Figure 5.6 Screen shot of HCS in ONE simulator

Table 5.3 summarize the parameters used in this experiment. The simulator selects source and destination nodes between 0-155 nodes, randomly. Messages are created throughout simulation time with the time interval of 5-15s.

Table 5.3 Parameter used in HCS scenario

Parameters	**Value**
World's size for Movement Model	4500 X 3400m
Total simulation time	430000s
No. of Hosts [pedestrians, cars, trams]	[100, 50, 6]=156
Message TTL (time to live)	1344mins
Time to move nodes in the world before real simulation commence	43000s
Nodes speed [pedestrians, cars, trams]	[0.5-1.5, 2.7-13.9, 7-10]m/s
Nodes pause time [pedestrians, cars, trams]	[0-120, 0-120, 10-30]s
Message sizes	500KB-1MB
Message creation interval	5-15s
Air data transmit speed	250kbps
Transmit range	10m
Range of message source/destination addresses	0-155 nodes
Queue sizes	0,20,40,60,80,100,120,140,160,180 in MB
No. of each experiment runs	15

5.3.1 RESULTS

Fig. 5.7 represents the overall delivery ratio in HCS mobility scenario by dividing total no. of messages received at all destination nodes to the total no. of messages generated by all source nodes. It shows that Lobby Influence outperforms Epidemic and Bubble Rap in situation where nodes follows a specific path and there is always good chance that two nodes can encounter each other more frequently or may meet more than once. Thus such kind of mobility scenarios has a pattern which helps to locate destinations more quickly. Epidemic showed good delivery response when queue size becomes 100 or more. This means that at lower queue size packet loss is very high since epidemic relies on flooding in order to accommodate new packets it has to drop older ones.

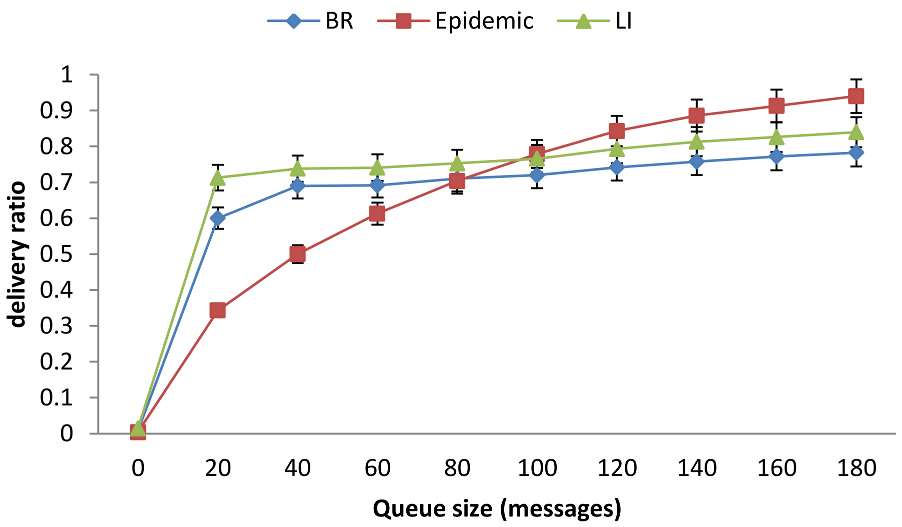

Figure 5.7 Average delivery ratio in HCS mobility scenario

Fig. 5.8 shows average delays experienced during message delivery. Epidemic has a quickest response since it uses flooding rather than some criteria as used by BR and LI. By comparing BR with LI, LI proves to be faster. Unlike BR which relies on more famous nodes, LI keep on forwarding messages either to more famous nodes or nodes having high lobby index.

Fig. 5.9 shows average number of messages forwarded by each algorithm to reach to the destinations in normalised form. Epidemic proves to be the most costly algorithm because it floods messages to any node it encounters. The sudden dip in Epidemic curve shows when queue size is small, so many messages are discarded. Thus a new node encounter could lead to a new copy of same data being exchanged again, leading to an increase in number of messages forwarded. When messages are not being discarded, these spurious additional forwards are not seen anymore. Whereas, LI overhead is little higher than BR, this trend justifies the fact that LI has to forward more messages based on its criteria i.e. forwarded messages to more popular or well-connected (high lobby index) nodes.

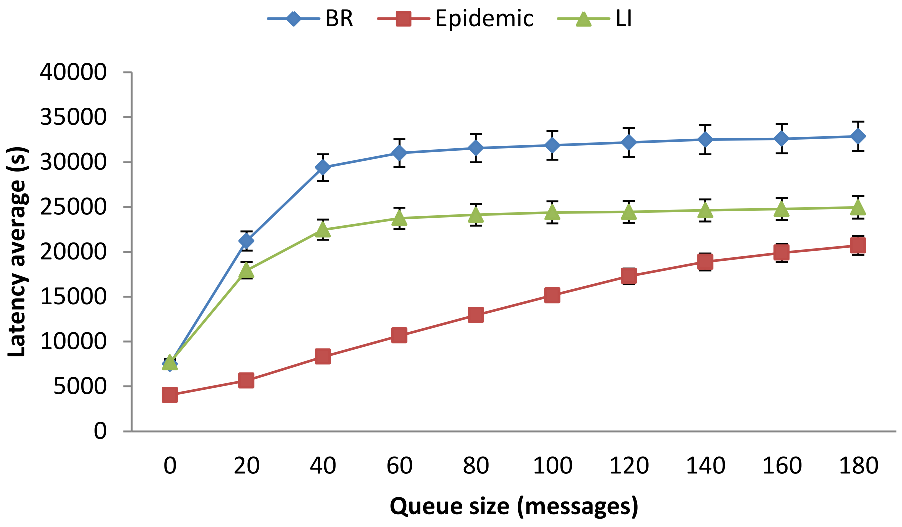

Figure 5.8 Average latency in HCS mobility scenario

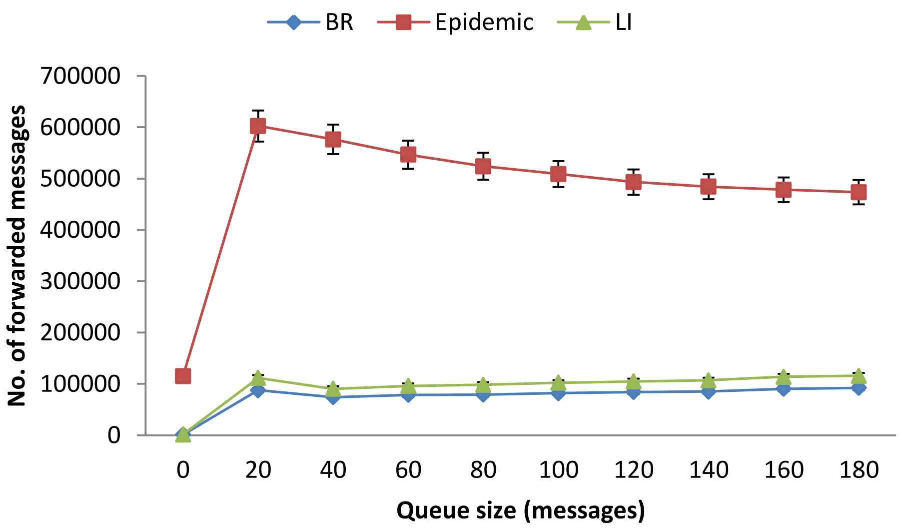

Figure 5.9 Average no. of forwarded messages in HCS mobility scenario

5.4 WORKING DAY MOVEMENT MODEL (WDM)

Average humans tend to live in society they adopt specific living patterns based on their daily life. Human have daily routines where they go to office in the morning, go for the shopping and go back to the home in the evening, this represents reality of the life. BR and LI based on the analogy of human behaviours; therefore it is very important that performances of these algorithms should be tested against the model that represents real life patterns. To approach realistic model, a model known as working day movement [78] is adopted in this simulation. In this model a map of Helsinki city is presented where its central areas are divided into 4 artificial districts A,B,C and D, please see [78] for more details. For this simulation, this thesis considers districts A, B and C. District A is the busiest district which is connected with districts B and C through overlay districts E and F respectively. Different offices, shopping areas and meeting spot are present in those districts. Nodes go to these meetings point through their own car or by taking buses. Thus represents a real world scenario where different nodes have chance to meet with each other and the nodes which are far apart from each other may be connected through intermediate nodes. Following table shows the assignment of nodes, offices and meeting spots to the respective districts.

Table 5.4 District settings

District	Nodes	Offices	Meeting Spots
A	30	30	4
B	30	10	1
C	30	20	2
E(A & B)	30	20	2
F(A & C)	30	30	4

Table 5.5 summarize the parameters used in this experiment. The simulator selects source and destination nodes between 0-159 nodes, randomly. Messages are created throughout simulation time with the time interval of 15-25s. Nodes transmit messages throughout simulation time period (from start to the end of simulation) between message creations intervals defined in simulation settings.

Table 5.5 Parameters used in WDM scenario

Parameters	**Value**
World's size for Movement Model	10000 X 8000m
Total simulation time	57000s
No. of Hosts [pedestrians, buses]	[150, 10]=160
Message TTL (time to live)	960 mins
Time to move nodes in the world before real simulation commence	7200s
Nodes speed [pedestrians, buses]	[0.5-1.5, 7-10]m/s
Nodes pause time [pedestrians, cars, trams]	[0-0, 10-30]s
Message sizes	500KB-1MB
Message creation interval	15-25s
Air data transmit speed	250kbps
Transmit range	10m
Working day length	28800s
Probability to go shopping after work	0.5
Own car probability	0.5
Range of message source/destination addresses	0-159 nodes
Queue sizes	0,20,40,60,80,100,120 ,140,160,180 in MB
No. of each experiment runs	10

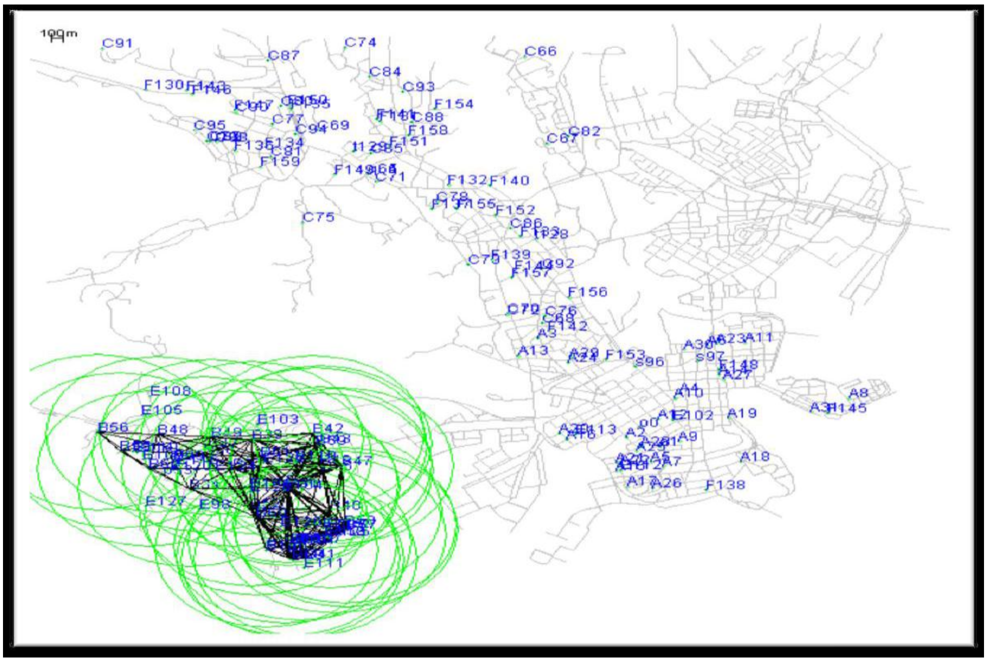

Figure 5.10 Screen shot of WDM in ONE simulator

This simulation considers activities of nodes for one day. The length of the day is approx. 16 hours because after that a node assumes to be at home anyway. The working day length is 8 hours and probability to go out for even activity is 0.5. Every district has its own route with 2 buses each and half of the nodes can use cars. A snap shot of WDM mobility scenario is shown in fig. 5.10. Table 5.5 summarize the parameters used in this experiment.

5.4.1 RESULTS

Fig. 5.11 shows the overall delivery ratio in WDM mobility scenario by dividing total no. of messages received at all destination nodes to the total no. of messages generated by all source nodes. The graph shows that LI outperforms BR in terms of message delivery. Whereas, epidemic has poor efficiency because it uses flooding and drops many packets when queues are full or TTL of messages expires. LI not only relies on more famous node but it can also exploit those nodes which have high connections in the network, thus proved as more efficient.

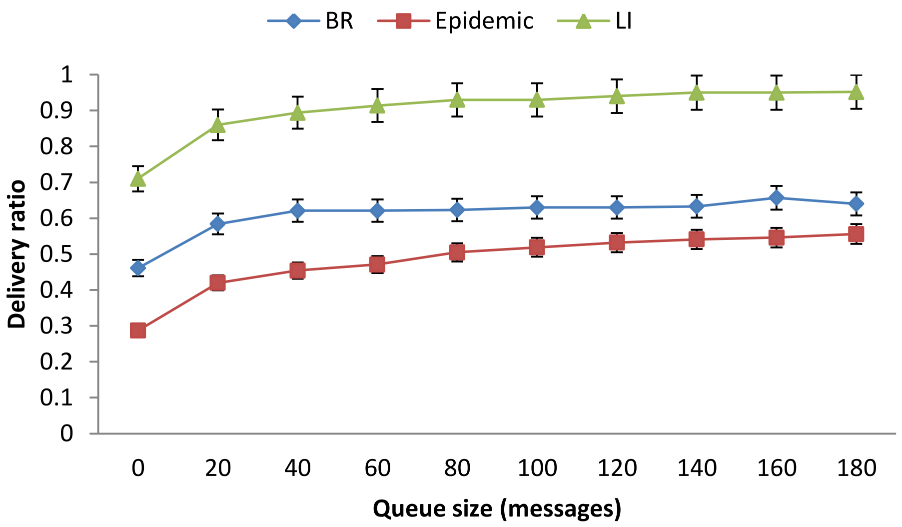

Figure 5.11 Average delivery ratio in WDM mobility scenario

Fig. 5.12 shows average delays experienced by the messages when using different algorithms. Epidemic has the highest delays; this is due to the fact that nodes are doing different activities this means that they are spending sometime in a particular spot for longer period of time therefore messages may be expired before reaching to its destination. Secondly most of the nodes exhausted their queue size by accepting unnecessary messages due to lack of message forwarding criteria as a result more messages dropped. LI again proves to be the quickest one in this scenario, this is simply because it is keep on forwarding message towards destination by exploiting more famous nodes or influential nodes with high connections.

Fig. 5.13 shows the number of messages forward during the simulation in normalised form. The graph shows that epidemic proves to be very costly compare to BR and LI. However, LI has more overhead than BR this is simply because it is now forwarding messages to more famous nodes as well as more influential nodes which are well-connected in the network.

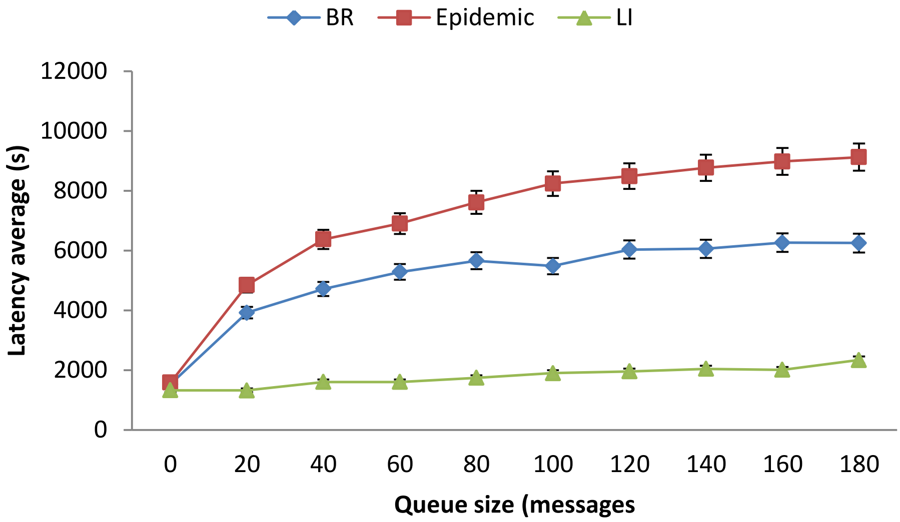

Figure 5.12 Average latency in WDM mobility scenario

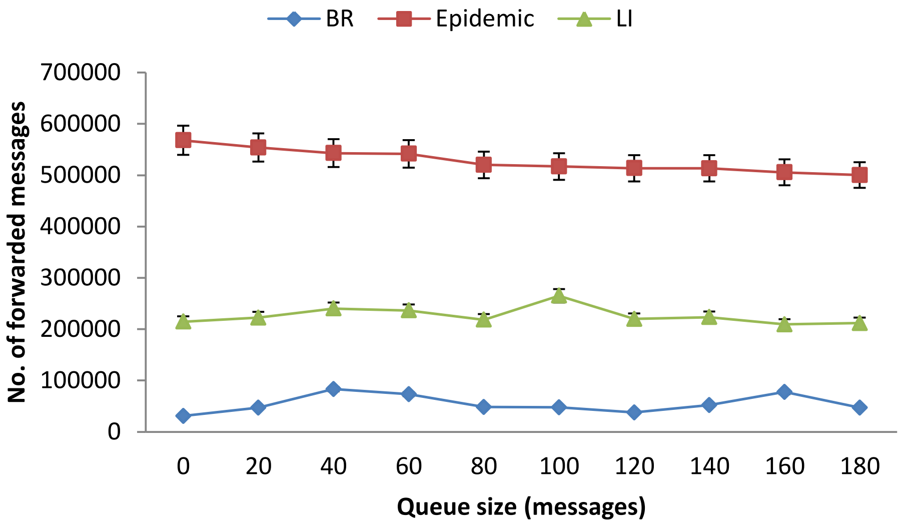

Figure 5.13 Average no. of forwarded messages in WDM mobility scenario

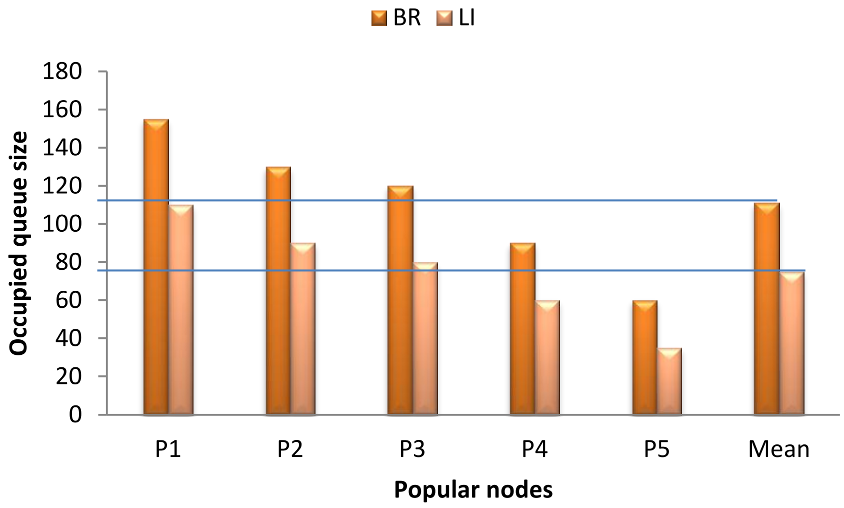

Figure 5.14 Message load on top five popular nodes in WDM scenario

Figure 5.14 shows the message load on top five popular nodes and their mean by the end of simulation time. The graph shows that LI has significantly reduced the load on popular nodes compared to the BR algorithm. This is because in LI, nodes can forward messages to either more popular nodes or more influential nodes; their queues are available to accept more messages that needs to be forward. In case of BR, nodes forward messages only to the more popular nodes; this will creates pressure on popular nodes and their queues. Queues fill more quickly as a result message loss could increase due to less space in the queue.

5.5 IMPACT OF THE MOBILITY MODELS ON PROTOCOL PERFORMANCE

It is clear from the fig. 5.15 that mobility models have a great impact on the probability of message delivery in an intermittent communication environment. In RWP movement model, nodes are randomly moving; there are very rare chances that same node encounters each other again. Thus, the pattern they generate are random, in such a scenario Epidemic proved to be the most effective. However, lobby influence provides better results than Bubble Rap, because LI transfers messages to the neighbouring nodes with which the node is directly connected. In HCS, nodes such as trams are moving continuously on the routes specified in the movement model and nodes use these trams for travelling purposed. Thus the chances of meeting with same nodes are very high, and as a result LI outperforms BR and epidemic in terms of delivery messages to the destinations. WDM resembles to the real life of humans, nodes move in a specific patterns, frequently meeting with some nodes as they go in offices or spending time in some evening spot. In such kind of scenario, LI again proves to provide the best results compared to BR and epidemic routing protocols.

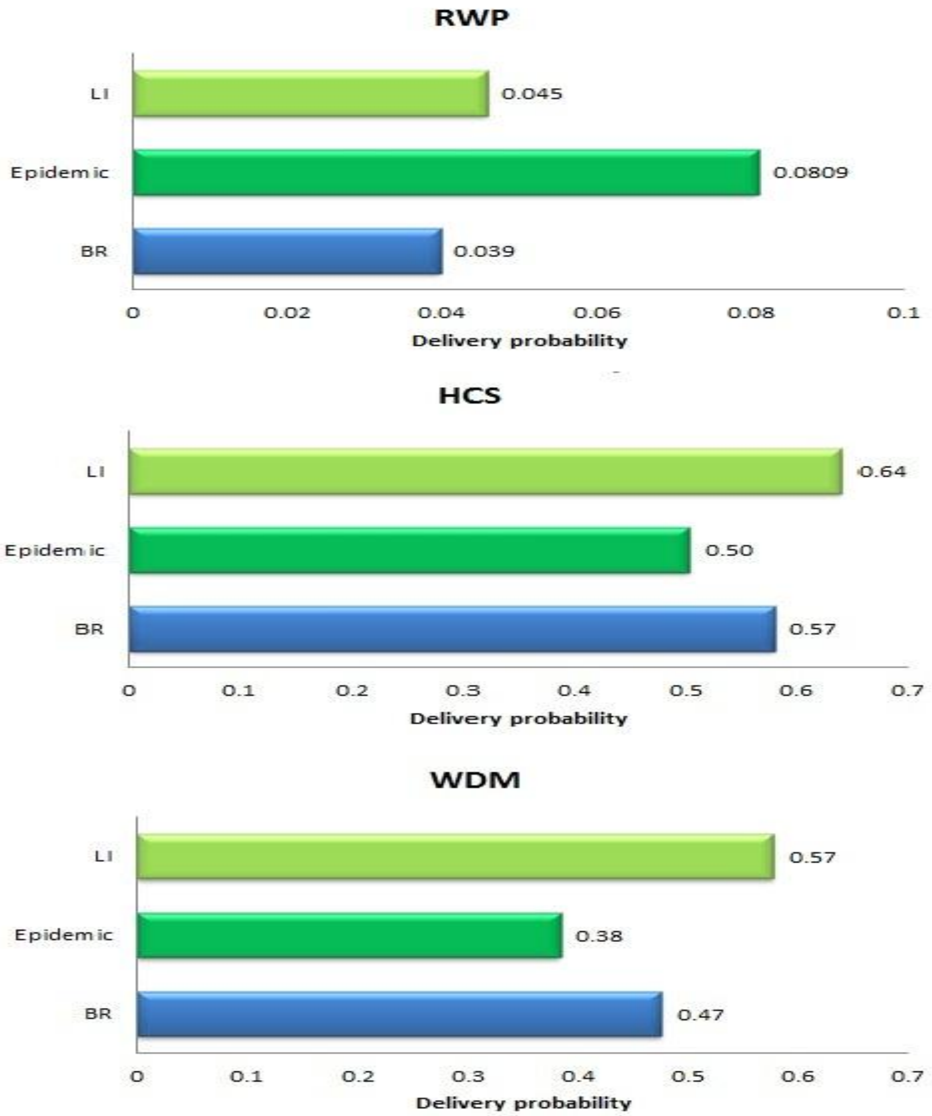

Figure 5.15 Impact of mobility models

5.6 Chapter Summary

This chapter presented different comparison graphs of Lobby Influence algorithm with other two well-known algorithms Bubble Rap and Epidemic. These algorithms are tested in different mobility models such as RWP, HCS and WDM in ONE simulator. The result show that the observations based on which LI forward messages proves to be successful, nodes not only transfer messages to popular nodes but un-poplar nodes with popular neighbours as a result high message delivery and low latency is observed at intended destinations.

The next chapter will summarize and conclude the thesis work and give the future direction of this work.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter presents concluding discussion on the Lobby Influence algorithm in section 6.1 and section 6.2 highlights the future direction of this thesis work.

6.1 CONCLUSIONS

This thesis describes routing/forwarding techniques for information dissemination in DTN. Specifically, it investigates algorithms that use social relationship patterns. Improvements in message routing/forwarding in opportunistic networks is achieved by exploiting nodes that reveal a high level of popularity or influence within the network. The Lobby Influence algorithm allows un-popular nodes to take responsibility and forward messages to neighbouring nodes with high degree of influence or connections in the network, thus providing fairer distribution to alleviate saturation on individual nodes. The Lobby Influence algorithm shows improvement in terms of message delivery and speed over Bubble Rap and Epidemic routing algorithms. Although, LI has a slightly higher communication cost compared to BR, it significantly reduces load on the most popular nodes in the network without approaching the communication cost of Epidemic routing algorithm.

6.2 FUTURE WORK

In this research, LI has been implemented in synthetically developed scenarios. In future, the next step would be to test this algorithm against real world mobility traces such as MIT reality mining or Haggle project to further establish the robustness of this algorithm for practical use. There is huge potential in this area; many new strategies based on human social relationships can be devised to further enhance the performance of message delivery while keeping the communication cost minimum. For instance, human social relationship patterns can be combined with geographical information based routing protocols where nodes have some knowledge of its location and neighbouring nodes with in the network. This kind of opportunistic forwarding can be very useful for delay tolerant applications where cost effective communication is priority. Furthermore, such kind of communication can be very useful in disaster situation where typical telecommunication infrastructure may not be available.

.

REFERENCES

[1] http://www.ubiq.com/weiser/ (accessed 03/09/2009)

[2] http://www.ubiq.com/hypertext/weiser/UbiHome.html (accessed 12/09/2009)

[3] http://www.facebook.com (accessed 10/11/2009)

[4] http://www.orkut.com (accessed 10/11/2009)

[5] P. Hui et al,. (2008), *"Bubble Rap: social-based forwarding in delay tolerant networks"*, Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, pp.241-250.

[6] Newman M. E. J., & Girvan M., (2004), "*Finding and evaluating community structure in networks*", Physical Review E, vol. 69, no. 2, id. 026113.

[7] Lancichinetti A., Fortunato S., & Kertész J., (2009), "*Detecting the overlapping and hierarchical community structure in complex networks*", New Journal of Physics, vol. 11 033015, pp.18.

[8] Euler L., (1736), "*Solutio problematis ad geometriam situs pertinentis. Commentarii Academiae Petropolitanae"*, vol. 8, pp. 128-140.

[9] Wasserman S., Faust K., (1994), "*Social Network Analysis: Methods and Applications*", Cambridge University Press, Cambridge (UK).

[10] Scott J. P., (2000), "Social Network Analysis", Sage Publications Ltd., London (UK).

[11] Girvan M., & Newman M.E. J., (2002), "Community structure in social and biological networks", Proc. Natl Acad. Sci. USA, vol. 99, no. 12, pp. 7821-7826.

[12] Fortunato S., & Castellano C., (2009), *"Encyclopedia of Complexity and System Science"*, Ed B Meyers (Heidelberg: Springer) (arXiv:0712.2716).

[13] Lusseau D., & Newman J., (2004)*,"Identifying the role that animals play in their social Networks"*, Proc. R. Soc. B 271 S477.

[14] Adamic L., & Glance N., (2005), Proc. 3rd Int. Workshop on Link Discovery (Information Sciences Institute, University of Southern California, Los Angeles) p36.

[15] Flake G W., Lawrence S., Lee Giles C., & Coetzee M., (2002), "*Self-Organization and Identification of Web Communities*",IEEE Computer, vol. 35, pp. 66-71.

[16] Guimerà R., Mossa S., Turtschi A., & Amaral L., (2005), "*The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles",* Proc Natl Acad Sci U S A, vol. 102, no.22, pp. 7794-9.

[17] Pimm S L., (1979), "*The structure of food webs*", Theor. Popul. Biol., vol. 16, pp. 144-158.

[18] Krause A E., Frank K A., Mason D M., Ulanowicz R E., & Taylor W W., (2003), "*Compartments revealed in food-web structure*", Nature, vol. 426, pp. 282.

[19]Lusseau D., Schneider K., Boisseau J. O., Haase P., Slooten E., & Dawson M. S., (2003), "*The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations",* Behavioral Ecology and Sociobiology, vol. 54, no. 4, pp. 396-405, doi: 10.1007/s00265-003-0651-y.

[20] Farber S., Costanza R., Childers L. D., Erickson J., Gross K., Grove M., Hopkinson C., Kahn J., Pincetl S., Troy A., Warren P., & Wilson M., (2006), "*Linking Ecology and Economics for Ecosystem Management*", BioScience, vol. 56, pp. 121–133, doi:10.1641/0006-3568(2006)056[0121:LEAEFE]2.0.CO;2.

[21] Clauset A., Moore C., & Newman J., (2007) Lect. Notes Comput. Sci., vol. 4503, pp. 1-13.

[22] Palla G., Derényi I., Farkas I. & Vicsek T., (2005), "Uncovering the overlapping community structure of complex networks in nature and society", Nature, vol. 435 pp. 814.

[23] Baumes J., Goldberg M., Krishnamoorty M., Magdon-Ismail M., & Preston N., (2005), "Discovering hidden groups in communication networks", In: IADIS (Eds.: Guimaraes, N. and Isaias, P. T.), pp. 97–104.

[24] Nicosia V., Mangioni G., Carchiolo V., & Malgeri M., (2008), "*Extending the definition of modularity to directed graphs with overlapping communities*", Journal of Statistical

Mechanics: Theory and Experiment, vol. 2009, no. 03, pp. 22, doi: 10.1088/1742-5468/2009/03/P03024.

[25] Lancichinetti A., Fortunato S., & Kertész J., (2009), "*Detecting the overlapping and hierarchical community structure in complex networks",* New J. Phys., Vol. 11 id.033015, pp. 18.

[26] www.cs.ucf.edu/courses/cot3100.fall00/section2/.../lec1003.doc (accessed 05/10/2009)

[27] Fortunato S., & Castellano C., (2007), "*Community Structure in Graphs",* eprint, arXiv:0712.2716.

[28] Newman J., & Girvan M., (2004), "*Finding and evaluating community structure in networks*", Physical Review E vol. 69, id. 026113.

[29] Gfeller D., Chappelier J. C., & Rios P., (2005), "*Finding instabilities in the community structure of complex networks*", Physical Review E, vol. 72, id. 056135.

[30] Newman J., & Girvan M., (2004), "*Finding and evaluating community structure in networks*", Physical Review E, vol. 69, id. 026113.

[31] Girvan M., & Newman J., (2002), *"Community structure in social and biological Networks"*, Proceedings of the National Academy of Science of the USA, vol 99, no. 12, pp. 7821–7826.

[32] Newman J., & Girvan M., (2004), "*Finding and evaluating community structure in networks"*, Physical Review E, vol. 69, id. 026113.

[33] Brandes U., (2001), "*A faster algorithm for betweenness centrality"*, Journal of Mathematical Sociology, vol. 25, No 2, pp. 163–177

[34] Tyler R., Wilkinson M., & Huberman A., (2003), "*Email as spectroscopy: automated discovery of community structure within organizations"*, Proceeding of the First International Conference on Communities and Technologies, Huysman M, Wenger E and Wulf V, Eds., Kluwer Academic Press, Amsterdam.

[35] Wilkinson M., & Huberman A., (2004), "*A method for finding communities of related genes"*, Proceedings of the National Academy of Science of the USA, vol. 101, suppl. 1, pp. 5241–5248 77.

[36] Fortunato S., Latora V., & Marchiori M., (2004) "*Method to find community structures based on information centrality",* Phys. Rev. E, vol. 70, id. 056104.

[37] Latora V., & Marchiori M., (2001), "*Efficient behavior of small-world networks*". Physical Review Letters, vol. 87, id. 198701.

[38] Zhou H., (2003), "*Network landscape from a Brownian particle's perspective*", Physical Review E, Vol. 67, id. 041908.

[39] Palla G., Der´enyi I., Farkas I., & Vicsek T., (2005), "*Uncovering the overlapping community structure of complex networks in nature and society"*, Nature, vol. 435, pp. 814-818.

[40] Palla G., Barab´asi L., & Vicsek T., (2007), "*Quantifying social groups evolution"*, Nature, vol. 446, pp. 664–667.

[41] Farkas I., bel D., Palla G., & Vicsek T., (2007), "*Weighted network modules*", New Journal of Physics, vol. 9, pp. 180.

[42] Palla G., Farkas J., Pollner P., Der´enyi I., & Vicsek T., (2007), "*Directed network Modules"*, New Journal of Physics, vol. 9, pp. 186.

[43] Palla G., Der´enyi I., Farkas I. & Vicsek T., (2005), "*Uncovering the overlapping community structure of complex networks in nature and society*", Nature vol. 435, pp. 814**.**

[45] Newman J. (2004),"*Analysis of weighted networks*", Physical Review E, vol. 70, id. 056131.

[46] Hui P., & Crowcroft J., (2008), "*Predictability of Human Mobility and Its Impact on Forwarding"*, Communications and Networking in China, ChinaCom 2008, pp. 543 – 547.

[47] http://www.dtnrg.org/docs/specs (accessed 01/11/2009)

[48] http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf (accessed 5/11/2009)

[49] Pelusi L., Passarella A., & Conti M., (2006), "*Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks*", IEEE Communications Magazine, pp. 134 -141.

[50] Fall K. (2002), "*A Delay-Tolerant Network Architecture for Challenged Internets",* Intel Research. http://www.dtnrg.org/docs/papers/IRB-TR-03-003.pdf

[51] http://reality.media.mit.edu/ (accessed 15/11/2009)

[52] http://www.haggleproject.org (accessed 15/11/2009)

[53] http://cordis.europa.eu/ist/fet/comms-sy.htm (accessed 17/11/2009)

[54] http://www.princeton.edu/~mrm/zebranet.html (accessed 18/11/2009)

[55] Juang P., Oki H., Wang Y., Martonosi M., Peh L., & Rubenstein D., (2002), "*Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with ZebraNet*", ACM SIGPLAN Notices, vol. 37, pp. 96-107.

[56] Small T., & Haas Z., (2003) "*The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way)",* In proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, MD, USA, June 1-3.

[57] Pentland A., Fletcher R., & Hasson A., (2004), "*DakNet: Rethinking Connectivity in Developing Nations"*, IEEE Computer, vol. 37, no. 1, pp. 78-83.

[58] Doria A, Uden M., & Pandey D., (2002), "Providing connectivity to the saami nomadic community," In proceedings of the 2nd international conference on Open Collaborative Design for Sustainable Innovation, Bangalore, India.

[59] Goodman D., Borras J., Mandayam N., & Yates R., (1997), "*INFOSTATIONS: A new system model for data and messaging services"*, in IEEE VTC'97, vol. 2, pp. 969–973.

[60] Jain S., Shah R., Brunette W., Borriello G., & Roy S., (2006), "*Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks"*, ACM/Kluwer Mobile Networks and Applications (MONET), vol. 11, no. 3, pp. 327-339.

[61]Zhao W., Ammar M., & Zegura E., (2004), "*A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks"*, Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (Mobihoc), ACM Press, pp. 187–198.

[62] Chatzigiannakis I., Nikoletseas S., & Spirakis P., (2001), " *Analysis and experimental evaluation of an innovative and efficient routing protocols for ad-hoc mobile networks*", Lecture Notes in Computer Science, vol. 1982, pp. 99-111.

[63] Chatzigiannakis, Nikoletseas S., Paspallis N., Spirakis P., & Zaroliagis C., (2001), " *An experiment study of basic communication protocols in ad-hoc mobile networks*", Lecture Notes in Computer Science, vol. 2141, pp. 159-169.

[64] Vahdat A., & Becker D., (2000), "*Epidemic routing for partially connected ad hoc networks*", Tech. Rep. CS-2000-06, Department of Computer Science, Duke University, Durham, NC.

[65] Burns B., Brock O., & Levine B., (2005), "*MV Routing and capacity building in disruption tolerant networks*", Proceedings of the IEEE INFOCOM 2005, Miami, FL.

[66] Lindgren A., Doria A., & Schelèn O., (2003), "*Probabilistic routing in intermittently connected networks*", Mobile Computing and Communications Review, vol. 7, no.3.

[67] Widmer J., & Boudec J., (2005), "*Network Coding for Efficient Communication in Extreme Networks*", Proceedings of the ACM SIGCOMM 2005 Workshop on delay tolerant networks, Philadelphia, PA, USA, August 22–26.

[68] Pelusi L., Passarella A., & Conti M., (2006), "*Encoding over the Network: Techniques and Challenges*", IIT-CNR Tech. Rep., online available at http://bruno1.iit.cnr.it/~bruno/techreport.html

[69] Musolesi M., Hailes S., & Mascolo C., (2005), "*Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks*", Proceedings of the 6th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005).

[70] Leguay J., Friedman T., & Conan V., (2006), "*Evaluating Mobility Pattern Space Routing for DTNs*", Proceedings of the IEEE Infocom 2006, Barcelona, Spain.

[71] Bavelas, A., (1948), "*A mathematical model for group structures*", Human Organization, vol. 7, pp. 16-30.

[72] Borgatti S. P., (1997), "*Network Analysis of 2-mode data"*, Social Networks, vol. 19, pp. 243-269.

[73] Sabidussi, G., (1966), "*The centrality index of a graph*". Psychomatrika, vol. 31, pp. 81-603.

[74] Holme P., & Ghos G., (2009), "*The Diplomat's Dilemma: Maximal Power for Minimal Effort in Social Networks* "Adaptive networks: Theory, models and applications, Thilo Gross and Hiroki Sayama, eds., Springer, pp. 269-288.

[75] Korn A., Schubert A., & Telcs A., (2009). "*Lobby index in networks*". *Physica A: Statistical Mechanics and its Applications*, vol. 388, no.11, pp. 2221 – 2226.

[76] Hui P., & Crowcroft J., (2007), "*How small labels create big improvements*", In proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, pp.65-70, doi:10.1109/PERCOMW.2007.55.

[77] Ari K., Jorg O., & Teemu K., (2009), "*The One simulator for DTN protocol evaluation*", SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Belgium.

[78] Ekman F., Keranen A., Karvo J., & Ott J., (2008), "*Working day movement model*". In proc. 1st ACM/SIGMOBILE workshop on Mobility Models for Networking Research, doi:10.1145/1374688.1374695.

[79] Chen D., Deng J., & Varshney P., K. (2007), "*Selection of a Forwarding Area for Contention-Based Geographic Forwarding in Wireless Multi-Hop Networks*". IEEE Transactions on Vehicular Technology, vol. 56, no. 5.

[80] Yang X., Yin J., & Yuan S. (2009), "*Location-Aided Opportunistic Routing for Mobile Ad Hoc Networks*". In proc. of the 5th international conference on Wireless Communications, Networking and Mobile Computing (WiCom '09).

[81] Yang S., Yeo K. C., & Lee S. B. (2010), "*Robust Geographic Routing with Virtual Destination based Void Handling for MANETs*". In the Proc. of the IEEE 71stVehicular Technology Conference (VTC 2010-Spring).

[82] Yan Y., Zhang B., Zheng J., & Ma J. (2010), "*CORE: a Coding-Aware Opportunistic Routing Mechanism for Wireless Mesh Networks*". IEEE Wireless Communications.

[83] Daly M. E., & Haahr M. (2007), "*Social network analysis for routing in disconnected delay-tolerant manets*". In MobiHoc'07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing.

[84] Mtibaa A., May M., Diot C., & Ammar M. (2010), "*PeopleRank: social opportunistic forwarding*". In INFOCOM'10: Proceedings of the 29th conference on Information communications.

[85] Brin S. & Page L. (1998), "*The anatomy of a large-scale hyper textual web search engine*". *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107–117.

[86] Clauset A., Moore C., & Newman M.E.J (2008), "Hierarchical structure and the prediction of missing links in networks," Nature, pp. 98-101, doi: 10.1038/nature06830

[87] Dillon P. J., Software contribution to the ONE simulator, [Online] Available: www.cs.pitt.edu/~pdillon/one