NETWORK CODING BASED FORWARDING USING HUMAN MOBILITY PATTERN

PREPARED BY

GAURANG P. PRAJAPATI

11MICT13

GUIDED BY

Prof. Zunnun Narmawala

Dept. Of Information Communication & Technology

DEPARTMENT OF INFORMATION COMMUNICATION & TECH.

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2013

NETWORK CODING BASED FORWARDING USING HUMAN MOBILITY PATTERN

Major Project Part - II

Submitted in total fulfilment of the requirements

For the degree of

Master of Technology in Information Communication and Technology

By

GAURANG P. PRAJAPATI

11MICT13

DEPARTMENT OF INFORMATION COMMUNICATION & TECH.

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2013

Declaration

This is to certify that

1. The thesis compromises of my original work towards the degree of Master of Technology in Computer Science & Engineering at Institute of Technology, Nirma University and has not been submitted elsewhere for degree.

2. Due acknowledgement has been made in the text to all other material used.

GAURANG P. PRAJAAPTI

11MICT13

Certificate

This is to certify that the Major Project Report entitled " **Network Coding Based Forwarding Using Human Mobility Pattern** " submitted by **Mr. Gaurang P. Prajapati (Roll No: 11MICT13)** towards the partial fulfilment of the requirements of Master of Technology in the field of Information Communication & Technology of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date:

Prof.Zunnun Narmawala
Guide
Dept. Of Information Comm. & Tech.
Institute Of Technology
Nirma University
Ahmedabad.

Prof. Gaurang Raval
PG-ICT Section Head,
Dept. Of Information Comm. & Tech.,
Institute Of Technology
Nirma University
Ahmedabad.

Dr. Sanjay Garg
Head of the Department,
Dept. Of Computer Sci. & Engg.
Institute Of Technology
Nirma University
Ahmedabad.

Dr.Ketan Kotecha
Director,
Institute of Technology
Nirma University
Ahmedabad.

Acknowledgements

I take this opportunity to express my immense gratitude to **Prof. Gaurang Raval** who always motivated me to do something extraordinary. He provided me with a very friendly environment in the institute. His time particularity and tactics of what and how to learn have helped me a lot to step into the practical world as well as to become a better person. I would like to thank him for providing me all necessary resources, which greatly helped me in completing my project work.

I am grateful to her for his prolonged interest in my work and excellent guidance. He has been a constant source of motivation to me. His unflinching demand for quality and his insistence, I was able to give my best. He has shown me a way to pursue excellence.

It is also a pleasure to mention my good friends for their love, care, support and creating a pleasant atmosphere for me here. I doubt that I will ever be able to convey my appreciation fully, but I owe them my eternal gratitude. A journey is easier when you travel together. Interdependence is certainly more valuable than independence.

I am also thankful to the **Dr. Ketan Kotecha** (Director, Institute of Technology), **Dr. Sanjay Garg** (HOD, CSE Department), **Prof.Gaurang Raval** (Section-Head,PG-ICT) and entire NIRMA family for their co-operation and support. I thank them for providing me such a warm atmosphere to make my stay delightful and memorable.

Last but not least, I would like to pay high regards to my parents for their sincere encouragement and inspiration throughout my research work and lifting me uphill this phase of life. I owe everything to them. Besides this, several people have knowingly and unknowingly helped me in the successful completion of this project.

Gaurang P. Prajapati
11MICT13

Contents

List of Figures

Abstract

Packet Switched Network(PSN)has attracted many researchers because of its growing area of applications. PSN is such a network where traditional MANET algorithms can not work efficiently because there is no end to end connectivity between source and destination. It is a Opportunistic Network where it has to use the opportunity of forwarding data to intermediate nodes so that it can send data to destination/destinations via relay nodes. There are several Forwarding based methods which can be used for PSN. Using social based approach for Forwarding we can improve delivery probability and also reduce delay. Using Network Coding we can make use of broadcasting behavior of Wireless Network. we can reduce Network load and we can improve efficiency. Using Bubble Rap algorithm for social based forwarding and Random Linear Network Coding(RLNC)for network coding a new algorithm is developed. Simulation is done in the ONE simulator. Simulation results shows network coding approach has reduced delay in forwarding.

Keywords:PSN, MANET, Social Based Networking, Forwarding, Network Coding, Bubble Rap, RLNC

Chapter 1

Introduction

1.1 Introduction

In opportunistic mobile networks, routes between nodes cannot be built based on direct observations. An opportunistic forwarding algorithm is required to decide, for each opportunity of data transfer, which nodes and which packet have to be transmitted. which has been conducted inside the Haggle project, has two objectives: (1) understand the feasibility and the constraints of forwarding algorithms in general, and (2) specify and conduct early test for various candidates forwarding algorithms, inside a common taxonomy. As opposed to previous work, we wish to study this problem with real-life measurement, and also assess the impact of several environment factors (density, mobility conditions) as well as networking factors (prior knowledge, partial information on the profile of nodes and the content of packets). As a first step to answer the above questions, we introduce several terms to describe the application context of a pocket-switched network (point to point messaging, interest-based group exchanges, local chat application). We then describe, as a function of the network environment, the assumptions that we can realistically make. We study the direct implication of these assumptions on the elements of a generic architecture: forwarding manager, buffer manager, scheduler. We then study the forwarding problem in a simple case: a source destination message forwarding, with no constraints on bandwidth or memory, and algorithms that use no prior knowledge about nodes of the network. We study this nominal case via the notion of a forwarding paths, that is a sequence of contacts that may be used to transport data to the destination. Several properties of these forwarding paths will be studied, as a first step towards understanding the impact of the environment on the feasibility of opportunistic forwarding. In the third part of this deliverable, we design and analyze multiple forwarding algorithms. These algorithms rely on various properties of PSNs. They use in particular social or historical information to make their forwarding decisions. Using Network Coding we can improve efficiency of the algorithm for PSNs. Network coding in not implemented in Haggle Project so we are planning to implemented Network coding in forwarding algorithm in haggle forwarding approach for that we have to find suitable algorithm for social based routing for forwarding.

1.2 Objective of Work

• To explore wireless network connectivity for Pocket Switched Network Applications.

• To purpose an algorithm using Network coding for social based forwarding.

• To reduce delay in packet transfer in wireless Opportunitistic Network.

1.3 Scope of Work

• The system can work on mobile devices with wireless network connectivity.

• As the network support is for social network application the main parameter to be focused on is Human mobility in different societies.

• Social information is used for making forwarding decisions, for it varies much slowly than the topology of the network.

• Network coding can be used to reduce delay, improve throughput and security in wireless pocket switched networks.

1.4 Motivation

•Now a days most of the users in an organization and in an institutions uses Mobiles/Laptops/PDAs and other hand held mobile devices with wireless con- nectivity. So we can provide them a social network support without using any fixed network infrastructure.

• The applications which are commonly deployed on mobile computing devices (e.g.

email, web browsing), however, are rarely able to fully exploit local wire- less connectivity (Bluetooth,Wi-fi), and instead use it only as a means of acquiring global connectivity via access points. Therefore, there is currently a large amount of wireless bandwidth capacity that remains unused because the current communication paradigm (i.e. the Internet) has not been designed to take advantage of local and intermittent connectivity.

Chapter 2

Literature Survey

2.1 Constraint of Opportunistic Routing

The main aim of an opportunistic network is to enable communication between nodes in a hostile environment where connectivity cannot be ensured. Besides issues existing in more general wireless networks as interference management or channel changes, opportunistic networks have to deal with new challenges that result from lack of connectivity. The solution to deal with these issues is to deepen node cooperation through forwarding. The cooperation behavior in the context of classical routing consists of forwarding received packets to egress ports. Generally, forwarding has to deal with three questions: what to send, to whom and when. To understand forwarding issues, lets describe the answer to these three questions in classical wired networks. The answer to the two first questions comes from routing that decides the egress interface to be used to send a packet. The last question is dealt by scheduling mechanisms that decide when to send packets waiting for transmission. Forwarding in classical networks is based on two unspecified postulates: the existence of a connected topology that can be shared among nodes, and the lack of individuality of a node. The first postulate assumes local knowledge of the neighborhood by a node can be spread in the whole network so that all nodes know how to reach any destination. Routing protocols play the role of spreading this local knowledge in the network, and defining the forwarding cooperative behavior (routing strategy) that ensures packet delivery to the end destinations. The second postulate is the lack of individuality of nodes, meaning that they have to implement the forwarding behavior determined by the routing protocol, and cannot choose their forwarding based on their own interests. Indeed, in classical networks, when a router forwards a packet, he can assume that it will be forwarded in the next hop to ensure its delivery to its final destination. This means that when a node forwards a packet it can release its interest about it and transfer the responsibility of the delivery to the next step. To extend this confidence when a packet has to cross the border of different networks with potentially different policies, the intent of the border node to forward a packet or not is clearly signaled through BGP like protocols. What differentiates mainly the class of scenarios that Haggle project targets from classical networking as described above, is that these two assumptions are not valid anymore. Because of lack of permanent connectivity, local neighborhood knowledge cannot be spread in all nodes and translated to a global routing strategy. In some scenarios, we might even have no access to any local connectivity and neighborhood information. A node cannot therefore defer completely its responsibility in packet forwarding after a single transmission, meaning that packets might have to be stored in memory even the have been forwarded one time. This last point is an essential distinction between a Haggle node where forwarded packets might be stored in memory after forwarding and classical network where packets are removed from memory after a single forwarding.

2.2 Routing in Opportunistic Network

Routing in opportunistic networks is surely one of the most compelling challenge. The design of efficient routing strategies for opportunistic networks is generally a complicated task due to the absence of knowledge about the topological evolution of the network. Routing performance improves when more knowledge about the expected topology of the network can be exploited ([1]). Unfortunately, this kind of knowledge is not easily available, and a trade-off must be met between performance and knowledge requirement. A key piece of knowledge to design efficient routing protocols is information about the context in which the users communicate. Context information, such as the users' working address and institution, the probability of meeting with other users or visiting particular places, can be exploited to identify suitable forwarders based on context information about the destination. In the following of this section we classify the main routing approaches proposed in the literature based on the amount of knowledge about the context of users they exploit. We specifically identify three classes, corresponding to context-oblivious, partially context-aware, and fully context-aware protocols. Approaches in the context-oblivious class neglect the context information, and thus work in the very same way independently of the context. Oblivious approaches do not assume any knowledge about the environment or the user behaviour, and look at optimisations of dissemination schemes such as Epidemic routing. The other approaches assume that some knowledge about the environment and/or the users' behaviour can be learnt by nodes themselves through autonomic features, and exploit this knowledge to drive the forwarding process. The ones in the partially context-aware class use only part of the context information, thus for some aspects can benefit from the knowledge of the context information, but for several other aspects they behave like the approaches of the context-oblivious class. The approaches in the fully context-aware class exploit the context information as much as possible.

2.2.1 Context Oblivious Routing

Routing techniques in this class basically exploit some form of flooding. The heuristic behind this policy is that, when there is no knowledge of a possible path towards the destination nor of an appropriate next-hop node, a message should be disseminated as widely as possible. Protocols in this class might be the only solution when no context information is available. Clearly, they generate a high overhead (as we also highlight in the performance evaluation section), may suffer high contention and potentially lead to network congestion ([2]). To limit this overhead, the common technique is to control flooding by either limiting the number of copies allowed to exist in the network, or by limiting the maximum number of hops a message can travel. In the latter case, when no relaying is further allowed, a node can only send directly to the destination when (in case) it is met. The most representative protocol of this type is Epidemic Routing (Epidemic for short) ([**?**]). Whenever two nodes come into communication range they exchange summary vectors that contain a compact unambiguous representation of the messages currently stored in the local buffers. Then, each node requests from the other the messages it is currently missing.

2.2.2 Partially Context aware Routing

Partially context-aware protocols exploit some particular piece of context information to optimise the forwarding task. The main difference with fully context-aware protocols is the fact that the latter usually provide a full-fledged set of algorithms to gather and manage any type of context information, while the former are customised for a specific type of context information. Probabilistic ROuting Protocol using History of Encounters and Transitivity (PROPHET [4]) is one of the most popular examples of protocols falling in this class. PROPHET is an evolution of Epidemic that introduces the concept of delivery predictability. The delivery predictability is the probability for a node to encounter a certain destination. The delivery predictability for a destination increases when the node meets the destination, and decreases (according to an ageing function) between meetings. A transitivity law is also included in the algorithm, such that if node A frequently meets node B, and node B frequently meets node C, then nodes A and C have high delivery predictability to each other. The PROPHET forwarding algorithm is similar to Epidemic except that, during a contact, nodes also exchange their delivery predictability to destinations of messages they store in their buffers, and messages are requested only if the delivery predictability of the requesting node is higher than that of the node currently storing the message.

2.2.3 Context aware Routing

Fully context-aware protocols not only exploit context information to optimise routing, but also provide general mechanisms to handle and use context information. Indeed, these routing protocols can be used with any set of context information, thus allowing the system to be customised to the particular environment it has to operate in. To the best of our knowledge, three protocols only fall in this category, i.e., Context-Aware Routing (CAR, [5]), History Based Opportunistic Routing (HiBOp) and Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (Propicman [6]).

2.3 Social based Forwarding in PSN

Social network analysis (SNA) [7], [8] has attracted a significant attention in many research areas such as anthropology, biology, communication studies, economics, information science, computer science and engineering. SNA mainly focuses on studying relationships among social entities and the patterns and implications of these relationships. With the increasing popularity of online social networks and new information technologies (such as mobile computing, E-commerce, distributed systems, and smart sensing), SNA becomes a more powerful tool to study the relationships and ties among users, and thus may guide the design of new policies, protocols, or applications for different information systems. In this section, we briefly review some recent advances in SNA and its applications in information systems, with a focus on four major aspects: community detection, information propagation, recommendation system, and security and privacy. Some of these advances will become the basis of social-based approaches for DTN routing.

2.3.1 Social properties of DTN

In this section, we introduce some social properties related to DTNs routing. They can be categorized into positive properties which benefit the relay selection (e.g. community, centrality, similarity and friendship) and negative properties which hurt the network performance (e.g. selfishness). Many of these social properties have been recently studied in social network analysis.

Social graph and contact graph

A social graph is an intuitive source for many social metrics such as community and friendship. Unfortunately it is not always available (due to either privacy or security reasons) or hard to be obtained via disclosed social data. However, with new networking technology, we can study relationships among people by observing their interactions and interests over wireless networks. Building a contact graph is a common way to study the interactions among people in a network and thus analyse their relationships and estimate the social metrics among them. In DTNs, each possible packet forwarding happens when two mobile nodes are in contact (i.e., within transmission range of each other). By recording contacts seen in the past, a contact graph can be generated where each vertex denotes a mobile node (device or person who carries the device) and each edge represents one or more past meetings between two nodes. An edge in this contact graph conveys the information that two nodes encountered each other in the past. Thus the existence of an edge intends to have predictive capacity for future contacts. A contact graph can be constructed separately for each single time slot in the past, or it can be constructed to record the encounters in a specific period of time by assigning a set of parameters to each edge to record the time, the frequency and the duration of these encounters. From the observation that people with close relationships such as friends, family members, etc. tend to meet more often, more regular and with longer duration, we can extract DTN nodes' relationships from the recorded contact graph, estimate their social metrics, and use such information to choose relays with higher probabilities of successful forwarding.

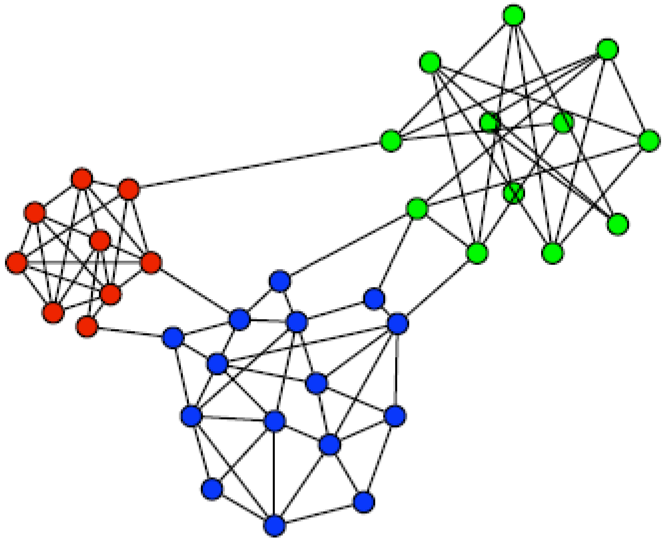

Figure 2.1: Social graph and contact graph [9]

Community

Community is an important concept in ecology and sociology. In ecology, a community is an assemblage of two or more populations of different species occupying the same geographical area. In sociology, community is usually defined as a group of interacting people living in a common location. Community ecologists and sociologists study the interactions between species/people in communities at many spatial and temporal scales . It has been shown that a member of a given community is more likely to interact with another member of the same community than with a randomly chosen member of the population. Therefore, communities naturally reflect social relationship among people. Since wireless devices are usually carried by people, it is natural to extend the concept of social community into DTNs to explore interactions among wireless devices. It is believed that devices within the same community have higher chances to encounter each other. Figure 1 illustrates a example of three community structures in a social/contact graph. Therefore, the knowledge of community structures could help a routing protocol to choose better forwarding relays for particular destinations, and hence improve the chance of delivery.

Centrality

In graph theory and network analysis, centrality is a quantitative measure of the topological importance of a vertex within the graph. A central node, typically, has a stronger capability of connecting other nodes in the graph. In a social graph, the centrality of a node describes the social importance of its represented person in the social network. There are several ways to define centrality in a graph. Three common centrality measures are degree centrality, betweenness centrality, and closeness centrality. Degree centrality is the simplest centrality measure which is defined as the number of links (i.e., direct contacts) incident upon a given node. For example, in Figure 2, the degree centrality of node a is 3 while those of the other nodes are 1. A node with a high degree centrality is a popular node with a large number of possible contacts, and thus it is a good candidate of a message forwarder for others (i.e., a hub for information exchange among its neighbourhood). Betweenness centrality measures the number of shortest paths passing via certain given node. For example, the betweenness centrality of node a in Figure 2 is 6, since every shortest path passes through it. But for the other nodes, their betweenness centralities are 3. Nodes that occur on many shortest paths between other nodes have higher betweenness than those that do not. A node with high betweenness centrality can control or facilitate many connections between other nodes, thus it is ideal for a bridge node during message exchange. The closeness centrality of a node is defined as a the inverse of its average shortest distance to all other nodes in the graph. If a node is near to the centre of the graph, it has higher closeness centrality and is good for quickly spreading messages over the network. For the example in Figure 2, the closeness centrality of node a is 1 since its average shortest distance to all others is 1. For any of the other nodes, the closeness centrality is 0:6 since their average shortest distance is 5/3.

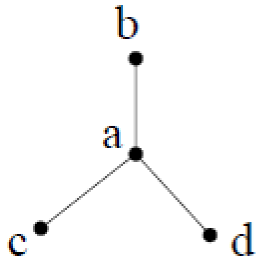

Figure 2.2: Centrality [9]

Similarity

Similarity is a measurement of the degree of separation. It can be measured by the number of common neighbors between individuals in social networks. Sociologists have long known that there is a higher probability of two people being acquainted if they have one or more other acquaintances in common. In a network, the probability of two nodes being connected by a link is higher when they have a common neighbor. When the neighbors of nodes are unlikely to be in contact with each other, diffusion can be expected to take longer than when the similarity is high (with more common neighbors). In addition, there are other ways to define the similarity beyond common neighbors, such as similarity on user interests and similarity on user locations.

Friendship

Friendship is another concept in sociology which describes close personal relationships. In DTNs, friendship can be defined between a pair of nodes. On the one hand, to be considered as friends of each other, two nodes need to have long-lasting and regular contacts. On the other hand, friends usually share more common interests as in real world. In sociology, it has been shown that individuals often befriend others who have similar interests, perform similar actions and frequently meet with each other.

Selfishness

Selfishness has been well-studied in sociology and economics, and has recently been considered in design of computer networks. In DTNs, selfishness can describe the selfish behaviors of DTN nodes controlled by rational entities. Selfish nodes can behave selfishly at individual level and aim to only maximize their own utilities without considering system-wide criteria. They can also behave selfishly in a social sense and are willing to forward packets for nodes with whom they have social ties but not the others. A selfish DTN node may drop others' messages and excessively replicate its own messages to increase its own delivery rate while significantly degrading other users' performance or even cause starvation.

2.4 Data Forwarding

There are two types of data forwarding.

1. Context-oblivious procedures
2. Context Informed Routing

2.4.1 Context-oblivious procedures

The first procedure, called self-limiting epidemic forwarding, is using TTL field in a non-conventional manner to optimize broadcast. This can be used to disseminate information about the context of nodes. The second procedure, called delegation forwarding, describes a simple rule to adapt in an online way the replication of packets as new nodes with different transmission quality are met. It is shown to reduce drastically the cost of any forwarding, including context aware one.

Self Limiting Epidemic Forwarding

A common feature here is that there is no practical bound on the number of users (unlimited network), and contact times may be short and unpredictable. In practice, implementing multihop broadcast in such settings poses a number of practical challenges, which, if not correctly addressed, may lead to very poor performance. Such challenges are: the impossibility to set the TTL correctly at all times, the poor performance of multiple access protocols in broadcast mode, flow control when there is no acknowledgement and scheduling of multiple concurrent broadcasts. Our design, called "Self Limiting Epidemic Forwarding" (SLEF), automatically adapts its behavior from single hop MAC layer broadcast to epidemic forwarding when the environment changes from being extremely dense to sparse, sporadically connected. A main feature of SLEF is a non-classical manipulation of the TTL field, which combines the usual decrement-when-sending to many very small decrements when receiving.

Delegation Forwarding

Successful delivery of messages at low costs and delays in such networks is thus challenging. Most forwarding algorithms avoid the cost associated with flooding the network by forwarding only to nodes that are likely to be good relays, using a quality metric associated with nodes. However it is non-trivial to decide whether an encountered node is a good relay at the moment of encounter. Thus the problem is in part one of online inference of the quality distribution of nodes from sequential samples, and has connections to optimal stopping theory. Based on these observations we develop a new strategy for forwarding, which we refer to as delegation forwarding. We analyse two variants of delegation forwarding and show that while naive forwarding to high contact rate nodes has cost linear in the population size, the cost of delegation forwarding is proportional to the square root of population size. We then study delegation forwarding with different metrics using real mobility traces and show that delegation forwarding performs as well as previously proposed algorithms at much lower cost. In particular we show that the delegation scheme based on destination contact rate does particularly well.

2.4.2 Context Informed Routing

The motivation behind the three approaches (Bubble Rap, HiBOp, Propicman) is similar. Naive protocols (such as Epidemic Routing) can easily saturate network and nodes' resources pretty fast. This not only reduces the network efficiency, but also has detrimental effects on the delivery performance (e.g., it has been shown that taking into consideration realistic CSMA MAC protocols, approaches such as Epidemic achieve extremely high delay and message loss, simply because they generate too much traffic in the network [?]). A mobile network has a dual nature: it is both a physical network and at the same time it is also a social network. A node in the network is a mobile device, and is also associated with a mobile human. Many MANET and some Delay Tolerant Network (DTN) routing algorithms [12] [?] provide forwarding by building and updating routing tables whenever mobility occurs. We believe this approach is not cost effective for an opportunistic network, since mobility is often unpredictable, and topology changes can be rapid. Rather than exchange much control traffic to create unreliable routing structures, we prefer to search for some characteristics of the network which are less volatile than mobility. An opportunistic network is formed by people. Those people's social relationships may vary much more slowly than the topology, and therefore can be used for better forwarding decisions. The main idea of context-aware forwarding is looking for nodes that show increasing match with known context attributes of the destination. High match means high similarity between node's and destination's contexts and, therefore, high probability for the node to bring the message in the destination's community (possibly, to the destination). We also consider that people are not likely to move around randomly. Rather, they move in a predictable fashion based on repeating behavioral patterns at different timescales (day, week, and month). If a node has visited a place several times before, it is likely that it will visit this location again in the future. Therefore, Bubble Rap, HiBOp, and Propicman exploit additional information about the context users are embedded in to make the forwarding function more efficient. Bubble Rap assumes that relationships among users follow a precise model of social structure. Specifically, they assume that nodes are clustered in cliques (representing social groups), and that nodes social connectivity degrees (within each clique) are highly non homogeneous. In other words, the number of social links each node has towards other nodes in its clique is highly variable, and is distributed according to power laws (which have been observed in real social networks). Cliques can communicate thanks to shared members (i.e., users being part of different social groups). The main idea behind Bubble Rap is automatically inferring the parameters of the underlying social structure, and exploiting the structure properties to select paths. To this end, Bubble Rap dynamically identifies users' communities, and ranks the nodes "sociability" (measured as the number of links) inside each community. These building blocks allow Bubble Rap to derive a model of the users' social structure, which is then exploited to forward data. Messages are pushed up in the starting community towards higher-rank nodes (i.e., more sociable users), until a contact with the destination's community is found. Pushing messages up in the senders' community ranking stores messages in more popular nodes, that have more chances to get in touch with the destination community. Among the three protocols, Bubble Rap is the only one that assumes and explicitly derives a model of the users' social network structure. HiBOp and Propicman infer, as a side effect, social relationships between nodes from context information dynamically gathered at each node. However, they do not include an explicit representation of the underlying social network. Propicman stores at each node a profile of the node's user. Profiles are exploited upon contact opportunities to forward messages. The main idea is to look for increasing matches between the profile of the destination user, and the profile of encountered users. When a user showing a higher match is encountered, the message(s) addressed to the destination are handed over to that node. A distinctive features of the Propicman approach with respect to Bubble Rap and HiBOp is exploiting decision trees to select next hops. Propicman idea is thus looking for users with increasing commonality (based on users' profiles) with the destination. The HiBOp approach is partly similar. However, with respect to Propicman, HiBOp manages context information quite differently. HiBOp distinguishes between different contexts (the context of the node, the context of its neighbourhood, the historical context). To select next hops, HiBOp looks (as Propicman does) at context information related to the user of the encountered node, but also at historical context information related to that user. In other words, HiBOp is able to understand if a node is a good forwarded based also on the similarity between the destination, and the context (the set of nodes) the encountered user is typically in touch with. This allows HiBOp not only to forward through users similar to the destination, but also through users often in touch with users similar to the destination. Thanks to this feature, HiBOp infers social relationships between users and users' communities, and implicitly learns the network social structure defined by the users' habits. Finally, HiBOp includes mechanisms to control the messages' replication rate, by dynamically selecting the number of copies in the network. With respect to Bubble Rap, HiBOp and Propicman do not assume a precisely defined model of the social relationships among users. The underlying structure is automatically learnt based on the historical context information gathered by nodes . While the Bubble Rap approach is more efficient when the model of the social relationships match the real social structure, HiBOp and Propicman are more flexible, as they are able to learn generic structures of social relationships between users. Finally, as far as security and privacy issues, Propicman includes self-defined security/privacy mechanisms, while both Bubble Rap and HiBOp rely on the solutions developed within other parts of the Haggle Project's activities.

HiBOp: a History Based Routing Protocol for Opportunistic Networks

HiBOp is a fully context-aware routing protocol completely described in [?] [15]. HiBOp includes mechanisms to handle any type of context information. As a particular instance, [?] [15]in the context is assumed to be a collection of information that describes the community in which the user lives, and the history of social relationships among users. At each node, basic data used to build the context can be personal information about the user (e.g. name), about her residence (e.g. address), about her work (e.g. institution), etc. In HiBOp nodes share their own data during contacts, and thus learn the context they are immersed in. Messages are forwarded through nodes that share more and more context data with the message destination. Note that HiBOp message can contain any payload, and context information is only used to identify good carriers (pretty much like the destination address is used in legacy IP networks). Since users of HiBOp have possibly to share personal information, privacy issues should be considered. Privacy management in opportunistic networks is – in general – a topic still largely not addressed, and it is not the target of this section to provide complete privacy solutions for HiBOp. Some privacy solutions are given in Section 4.4. It should be noted that the set of information that is considered in [?] [?] (and that we also consider hereafter) is equivalent to personal information people advertise on their public web pages (e.g., the working institution and address) which are, therefore, not perceived as sensitive information from a privacy standpoint. Designing complete privacy solutions for HiBOp is one of the main subjects of future work. HiBOp can be used for a number of applications of opportunistic networks in particular, and pervasive networking more in general. Beside simple messaging applications, it can be used for targeted advertisements (identifying groups of interested users through common context information, as shown in [?]), file sharing and data dissemination in general, and even emergency scenarios [?]. More in detail, HiBOp assumes that each node locally stores an Identity Table (IT), that contains personal information on the user that owns the device . Nodes exchange ITs when getting in touch. At each node, its own IT, and the set of current neighbours' ITs, represent the Current Context, which provides a snapshot of the context the node is currently in. The current context is useful in order to evaluate the instantaneous fitness of a node to be a forwarder. But even if a node is not a good forwarder because of its current location/neighbors, it could be a valid carrier because of its habits and past experiences.

Under the assumption that humans are most of the time "predictable", it is important to collect information about the context data seen by each node in the past, and the recurrence of these data in the node's Current Context. To this end, each context attribute seen in the Current Context (i.e., each row in neighbours' ITs) is recorded in a History Table (HT), together with a Continuity Probability index, that represents the probability of encountering that attribute in the future (actually more indices are used, as described in [13] [14]).

Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (Propicman)

Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks (Propicman) [26] also belongs to the fully context-aware routing protocols class. In Propicman, the carrier's information, called node profile, plays an important role in predicting the mobility of the nodes, and describe the social environment of the users and their relationships. We consider that people have repeating behavioral patterns at different timescales (day, week, and month). If a node has visited a place several times before, it is likely that it will visit this location again in the future. For example, because this person likes pasta and the place is an Italian restaurant. Thus, relevant information can be deducted by the node profile. Based on the node profiles of its two-hop neighbor nodes, a sender can select, as forwarder(s), the node(s) with the highest probability of delivering the message toward the destination (delivery probability). The delivery probability can be considered quasi-static information, hence the mobility of a node does not really affect to the validity of Propicman selection. Our results show that Propicman exploits effectively the mobility, as well as reduces significantly the number of nodes involved in the forwarding process. Thus the network overhead is significantly low in comparison with the other recent dissemination-oriented algorithms, such as Epidemic or Prophet. Propicman also takes into account information privacy from design. Most of the solutions proposed to MANET routing force users to share and exchange their information during the routing process, but it is unlikely that people accept that. This is one of the main barriers to MANET acceptance. In our solution nodes can share their information in a "hidden" format, but this information can still be used for routing. Furthermore, only the destination can read and understand the message content, while this content is hidden from the intermediate nodes.

Chapter 3

Bubble Rap: A Social Based Forwarding

3.0.3 Bubble Rap

Bubble Rap combines the knowledge of community structure with the knowledge of node centrality to make forwarding decisions. There are two intuitions behind this algorithm. Firstly, people have varying roles and popularity in society, and these should be true also in the network – the first part of the forwarding strategy is to forward messages to nodes which are more popular than the current node. Secondly, people form communities in their social lives, and this should also be observed in the network layer – hence the second part of the forwarding strategy is to identify the members of destination communities, and to use them as relays. Together, we call this Bubble Rap forwarding. For this algorithm, we make two assumptions:

1. Each node belongs to at least one community. Here we allow single node communities to exist.

2. Each node has a global ranking (i.e.global centrality) across the whole system, and also a local ranking within its local community. It may also belong to multiple communities and hence may have multiple local rankings.

Forwarding is carried out as follows. If a node has a message destined for another node, this node first bubbles the message up the hierarchical ranking tree using the global ranking, until it reaches a node which is in the same community as the destination node.

Then the local ranking system is used instead of the global ranking, and the message continues to bubble up through the local ranking tree until the destination is reached or the message expires. This method does not require every node to know the ranking of all other nodes in the system, but just to be able to compare ranking with the node encountered, and to push the message using a greedy approach. In order to reduce cost, we also require that whenever a message is delivered to the community, the original carrier can delete this message from its buffer to prevent further dissemination. This assumes that the community member can deliver this message. We call this algorithm Bubble Rap, using the metaphor of bubble for a community. The forwarding process fits our intuition and is taken from real life experiences. First you try to forward the message via surrounding people more popular than you, and then you bubble it up to well-known popular people in the wider-community, such as a postman. When the postman meets a member of the destination community, the message will be passed to that community. The first community member who receives the message will try to identify more popular members within the community, and bubble the message up again within the local hierarchy, until the message reaches a very popular member, or the destination itself, or the message expires. Figure 1. illustrates the Bubble Rap algorithm and Algorithm 1 summarise the operations in a flat community (not hierarchical) space.

3.1 Community detection

In PSN, the social network could map to the computer network since people carry the computing devices. To answer the question whether communities of nodes are detectable in PSN traces community detection algorithm is needed.

• K-CLIQUE Algorithm

In real human societies, one person may belong to multiple communities and hence it is important to be able to detect this feature. The K-CLIQUE method satisfies this requirement. It is designed for binary graphs, thus the edges of the contact graphs must be threshold in the mobility traces to use this method.[1]

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

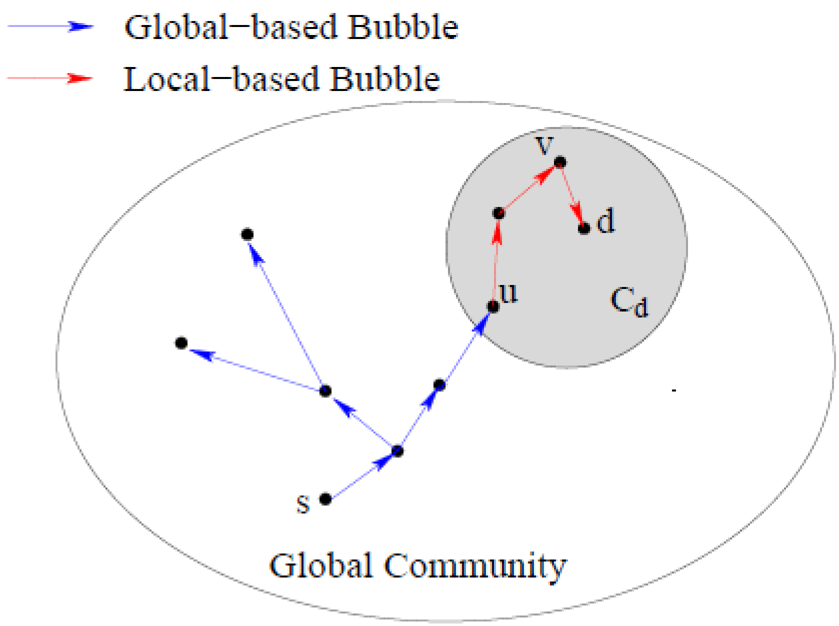

Figure 3.1: Bubble Rap [17]

these nodes exceeds a configurable parameter (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.

3.2 Centrality Detection

In human society, people have different levels of popularity: salesmen and politicians meet customers frequently, whereas computer scientists may only meet a few of their colleagues once a year. Here, heterogeneity in popularity is employed to help design more efficient forwarding strategies: it is preferred to choose popular hubs as relays rather than unpopular ones.

Two types of centrality are computed

- Global Centrality : Along the whole network.
- Local Centrality: Along each Community.

The Centrality measure in BUBBLE Rap is carried out by DEGREE metric which is as follows.

DEGREE: The total no of unique nodes seen by a node throughout the experiment is the node degree.So, the node which meets more no of nodes throughout the whole experiment

has the high degree and hence high centrality value.
BUBBLE is a combination of LABEL and RANK forwarding algorithm. It uses RANK to spread out the messages and uses LABEL to identify the destination com- munity.
LABEL: Explicit labels are used to identify forwarding nodes that belong to the same organization. Optimizations are examined by comparing label of the potential relay nodes and the label of the destination node.This is in the human dimension, although an analogous version can be done by labeling a k-clique community in the physical domain.
RANK: The forwarding metric used in this algorithm is the node centrality. A message is forwarded to nodes with higher centrality values than the current node. It is based on observations in the network plane, although it also reflects the hub popularity in the human dimension.

3.3 BubbleRap Algorithm

Forwarding is carried out as follows. If a node has a message destined for another node, this node would

rst bubble this message up the hierarchical ranking tree using the global ranking until it reaches a node which has the same label (community) as the destination of this message. Then the local ranking system will be used instead of the global ranking and continue to bubble up the message through the local ranking tree until the destination is reached or the message expired. This method does not require every node to know the ranking of all other nodes in the system, but just to be able to compare ranking with the node encountered, and to push the message using a greedy approach.This algorithm is called as BUBBLE[1], since each world/community is like a bubble. The psuedo code is as follows [19],

begin

$foreach_EncounteredNode_ido$

$if(LabelOf(currentNode) == LabelOf(destination))then$

$if(LabelOf(EncounteredNode_i) == LabelOf(destination))$

and

$(LocalRankOf(EncounteredNode_i) > LocalRankOf(currentNode))$

$then$

$ncounteredNode_i.addMessageToBuffer(message)$

$else$

$if(LabelOf(EncounteredNode_i) == LabelOf(destination))$

or

$(GlobalRankOf(EncounteredNode_i) > GlobalRankOf(currentNode))$

$then$

$ncounteredNode_i.addMessageToBuffer(message)$

end

Chapter 4

Random Linear Network Coding

4.1 Introduction

Much work in network coding has concentrated around a particular form of network coding: random linear network coding. Random linear network coding was introduced in [62] as a simple, randomized coding method that maintains "a vector of coefficients for each of the source processes," which is "updated by each coding node". In other words, random linear network coding requires messages being communicated through the network to be accompanied by some degree of extra information—in this case a vector of coefficients. In today's communications networks, there is a type of network that is widely-used, that easily accommodates such extra information, and that, moreover, consists of error-free links: packet networks. With packets, such extra information, or side information, can be placed in packet headers and, certainly, placing side information in packet headers is common practice today (e.g., sequence numbers are often placed in packet headers to keep track of order).

The second Definition is coding at a node in a packet network (where data is divided into packets and network coding is applied to the contents of packets), or more generally, coding above the physical layer. This is unlike network information theory, which is generally concerned with coding at the physical layer. We use this definition in this book. Restricting attention to packet networks does, in some cases, limit our scope unnecessarily, and some results with implications beyond packet networks may not be reported as such. Nevertheless, this definition is useful because it grounds our discussion in a concrete setting relevant to practice.

The output flow at the given node is obtained as a linear combination of its input flows. The coefficients selected for this linear combination are completely random in nature, hence the name Random Linear Network Coding (RLNC). The node combines a number of packets it has received or created into one or several outgoing coded packets.
The encoding process involves linearly combining the native / original packets with randomly selected coefficients. The coefficients are independently and randomly selected from a finite field called Galois Field (GF). The coefficients of this combination form a coding vector. The encoding, re-encoding and decoding operations are implemented via matrix operations. The re-encoding process is almost similar to the encoding process with the exception that the coding vector of the re-encoded packet is calculated by the arithmetic operation between the newly generated coefficients at that node and the original coefficients of the received coded packets. which is depicted in below figure.

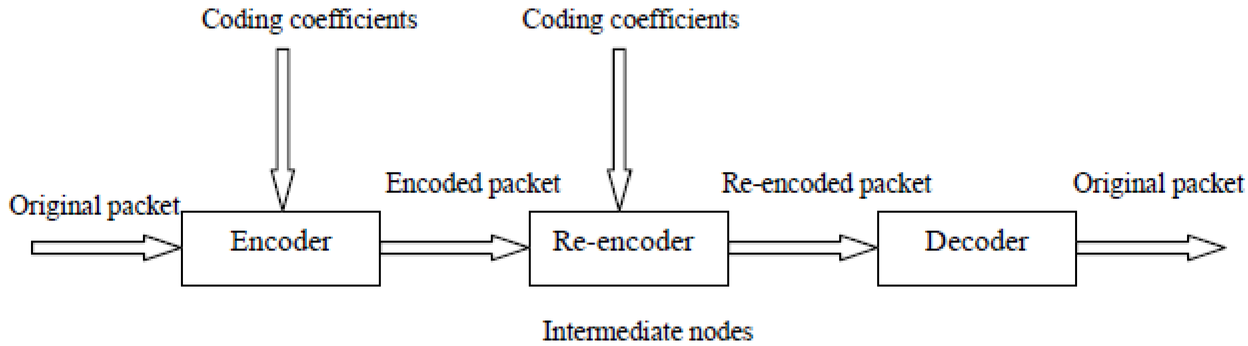

Figure 4.1: RLNC:Process

- **Generation**

It is important to limit the size of the matrix that is used for encoding and decoding. For that purpose the packets are grouped together in blocks. Each block is called a Generation. Only packets of the same generation can be encoded and later decoded. It is shown that the size and composition of the generation has significant impact on the performance of network coding.

- **Dependency**

It is shown that with RLNC, there exists a probability of selecting linearly dependant combinations, which depends upon the size of the GF, i.e., the range of possible coding coefficient values. However, it is shown through simulations that, even choosing a small field size, this probability becomes negligible.

- **Rank of a Matrix**

The rank of a matrix is the maximum number of independent rows (or the maximum number of independent columns) of a matrix.

- **Innovative Packet**

A packet is said to be innovative if it increases the rank of a matrix.

RLNC Algorithm

The specific coding scheme we consider is as follows [20]. We suppose that, at the source node, we have K message packets w1,w2, . . . ,wK, which are vectors of length λ over some finite field Fq. (If the packet length is b bits, then we take $\lambda = \frac{b}{\log q}$ The message packets are initially present in the memory of the source node. The coding operation performed by each node is simple to describe and is the same for every node: received packets are stored into the node's memory, and packets are formed for injection with random linear combinations of its memory contents whenever a packet injection occurs on an outgoing link. The coefficients of the combination are drawn uniformly from Fq. Since all coding is linear, we can write any packet u in the network as a linear combination of w1,w2, . . . ,wK, namely, $u = \sum_{k=1}^{k} \gamma kwk$ We call γ the global encoding vector of u, and we assume that it is sent along with u, as side information in its header. The overhead this incurs (namely, K log2 q bits) is negligible if packets are sufficiently large. Nodes are assumed to have unlimited memory. The scheme can be modified so that received packets are stored into memory only if their global encoding vectors are linearly-independent of those already stored. This modification keeps our results unchanged while ensuring that nodes never need to store more than K packets. A sink node collects packets and, if it has K packets with linearlyindependent global encoding vectors, it is able to recover the message packets. Decoding can be done by Gaussian elimination. The scheme can be run either for a predetermined duration or, in the case of rateless operation, until successful decoding at the sink nodes. We summarize the scheme in Figure 4.4. The scheme is carried out for a single block of K message packets at the source. If the source has more packets to send, then the scheme is repeated with all nodes flushed of their memory contents.

Initialization

- The source node stores the message packets w1,w2, . . . ,wK in its memory.

Operation

• When a packet is received by a node, the node stores the packet in its memory. When a packet injection occurs on an outgoing link of a node, the node forms the packet from a random linear combination of the packets in its memory. Suppose the node has L packets u1, u2, . . . , uL in its memory. Then the packet formed is

$$uo := \sum_{l=1}^{L} \alpha lul$$

where αl is chosen according to a uniform distribution over the elements of Fq. The packet's global encoding vector γ, which satisfies

$$u = \sum_{k=1}^{k} \gamma kwk$$

Decoding Each sink node performs Gaussian elimination on the set of global encoding vectors from the packets in its memory. If it is able to find an inverse, it applies the inverse to the packets to obtain w1,w2, . . . ,wK; otherwise, a decoding error occurs.

What is Network Coding Good For?

1. **Throughput**

   we can code two or more packets together and send them so we can improve the throughput of forwarding strategy by forwarding two or more packets together in single instance. Here we are going to focus only on how to minimize delay and increase throughput.

2. **Reliable**

   Some of the main advantages of NC include higher reliability and robustness, especially in case of mobile and lossy networks, where other FEC or ARQ schemes do not show good performance. By encoding the packets into a single packet, we are ensuring that a single packet loss does not necessarily require retransmissions. If the complete set of coded packets of the same generation can be received from any node, decoding can be successful and all the packets can be recovered. Similarly, the concept of partial decoding is also provided in the literature where partial packets can still be recovered even if all the required encoded packets are not received.

3. **Complexity**

   Overall, NC works by solving the set of equations linearly combined together in polynomial time. The decoding is performed using Gaussian elimination methods. These methods are simple in computation and utilize the cheap computational power to improve network efficiency.

4. **Distributed Nature**

   With NC, there is no need to have global knowledge of the network. Especially in case of RLNC, we even do not care what the neighbour has received. It is highly distributed in nature. Due to this property it is well suited for wireless networks which are also distributed in nature.

5. **Mobility**

   In mobile environments, the network topology changes over time and a main difficulty for many routing protocols are the frequent route updates and gathering new topological information. NC can address this uncertainty and alleviate the need for exchanging route updates.

6. **Security**

   Sending the linear combination of the packets instead of un-coded packets offers a natural way to take advantage of multipath diversity for security against wiretapping attacks in wireless networks

Chapter 5

Proposed Algorithm

5.1 Proposal of Network Coding Based Forwarding Algorithm

here we are proposing the new algorithm using BubbleRap algorithm for social based forwarding and RLNC algorithm for network coding.

1. The purposed algorithm is described as given below.

2. which uses Bubble Rap as forwarding algorithm and RLNC for Network coding. The flow of the approached algorithm is as described in figure.

3. Bubble Rap will forward packet based on local and global community detection.

4. Source node forwards an encoded packet intermediate node. Intermediate node first check if the packet is for itself or not. if intermediate node is the destination node then it will decode the packet using decoding vector.

5. If it is not the destination node for that packet then will first reencode the packet using other packet and adds key to encoding vector and forward to suitable node.

6. Finally packet reaches to the destination node it will decode the packet using values encoding vector.

   Flow of the purposed algorithm is described in below figure.

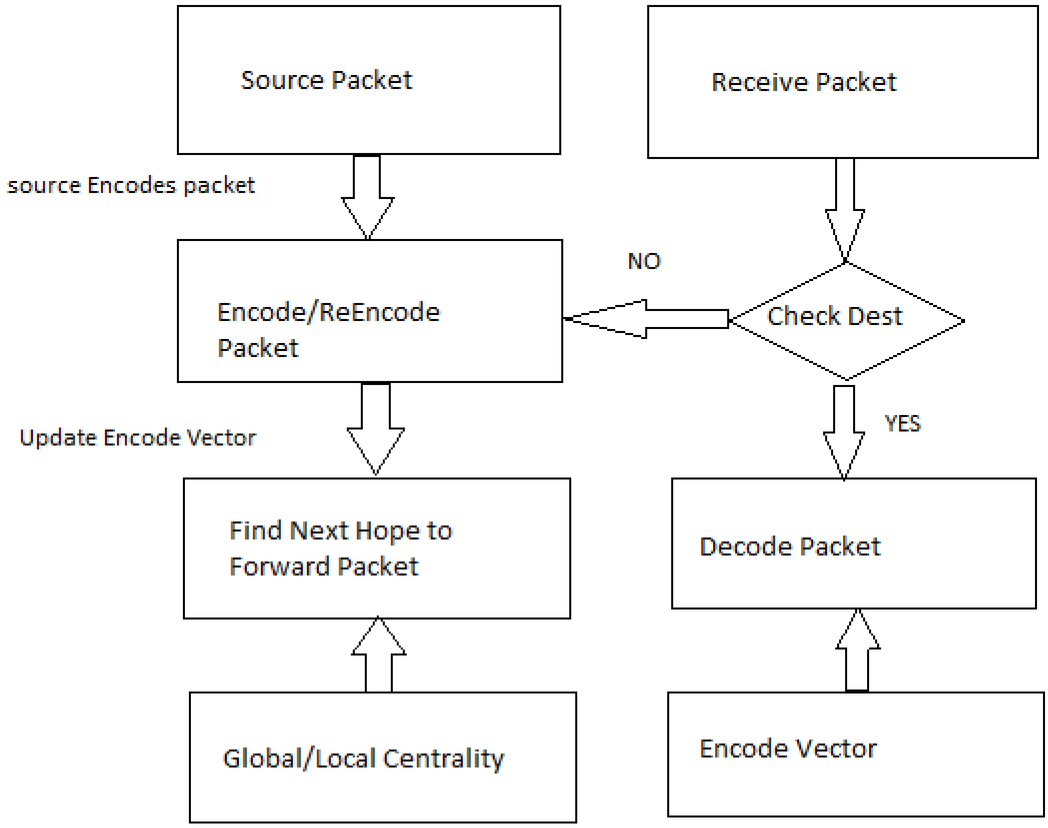

Figure 5.1: Network Coding Based Forwarding

Chapter 6

Implementation in The ONE Simulator

Delay-tolerant Networking (DTN) enables communication in sparse mobile ad-hoc networks and other challenged environments where traditional networking fails and new routing and application protocols are required. Past experience with DTN rout- ing and application protocols has shown that their performance is highly dependent on the underlying mobility and node characteristics. Evaluating DTN protocols across many scenarios requires suitable simulation tools.

In this thesis work Opportunistic Network Environment (ONE)simulator is used to observe the effectiveness of the proposed architecture.ONE simulator speci

cally de- signed for evaluating DTN routing and application protocols. It allows users to create scenarios based upon different synthetic movement models and real-world traces and offers a framework for implementing routing and application protocols (already includ- ing six well-known routing protocols). Interactive visualization and post-processing tools support evaluating experiments and an emulation mode allows the ONE sim- ulator to become part of a real-world DTN testbed. For more details about ONE refer README.txt in ONE.

6.1 Bubble Rap in ONE

BUBBLE Rap is the algorithm which combines the knowledge of both centralities of nodes and community structure, to achieve forwarding performance improvements in

PSN. Messages bubble up and down the social hierarchy, based on the observed community structure and node centrality, together with explicit label data. Bubbles represent a hybrid of social and physically observable heterogeneity of mobility over time and over community.
PSN is temporal network or time evolving network which is a kind of weighted network. The centrality measure in traditional weighted networks may not work here since the edges are not necessarily concurrent (i.e., the network is dynamic and edges are time-dependent).
Basically BUBBLE Rap measures the centrality value of a node by using DEGREE algorithm as seen in the previous chapter. But for practical applications, BUBBLE must be implemented in a distributed way. To achieve this, each device should be able to detect its own community and calculate its centrality values. The next step is to ask how can each node know its own centrality in a decentralized way, and how well past centrality can predict the future.

6.2 Centrality Calculation

As PSN is a decentralized network hence the total degree seen by a node throughout the experiment is not a good approximation for node centrality. Instead the degree per unit time (for example the number of unique nodes seen per 2 hours) and the node centrality have a high correlation value. That means in PSN how many people you know does not matter too much, but how frequently you interact with these people does matter. However, the average unit-time degree calculated throughout the whole experi- mental period is still difficult for each node to calculate individually. So here the degree for the previous unit-time slot (slot window)is considered such that when two nodes meet each other, they compare how many unique nodes they have met in the previous unit-time slot (e.g. 2 hours). This approach is called as single window (S-Window).
Another approach is to calculate the average value on all previous windows, such as from yesterday to now, then calculate the average degree for every 6 hours. This approach cumulative window (C-Window).

6.3 Implementation

All the above approaches,such as DEGREE, S-Window and C-Window provide us with a decentralized way to approximate the centrality of nodes in the system, and hence help us to design appropriate forwarding algorithms.

Combining these approximate methods and the distributed community detection, the BUBBLE can be put into reality. Here in this thesis work the BUBBLE Rap algorithm is implemented in ONE simulator along with traces given to the external mobility model. Here a network of mobile users spanning an entire city is made in which each device can detect its own local community or its social graph. At the same time, it also counts its own global and local centrality values. Its global ranking can be approximated as its 2-hour-averaged degree for all nodes and its local ranking can be approximated as its 2-hour-averaged degree only for nodes inside its community. With all these metrics, each node can forward messages using BUBBLE rap K-Clique community detection.

6.4 Infocome 2006 Traces

Dataset are downloaded form CRAWDAD project. This Experiment is carried out on real devices. Information about this experiment is [21].

Location: Princesa Sofia Gran Hotel, Barcelona

Date: Monday, April 24th to Thursday April 27th, 2006

Duration:

Devices distributed on Sundary April 23rd, between 7:00 and 9:00 pm.

Devices collected back starting from April 26th and on April 27th during the day.

Origin of time:

All times indicated in the files corresponds to the seconds elapsed since the motes were turned on. The motes had been all turned on on Sunday April 23rd at 5:01pm (which is 61260 seconds after midnight on that day).

6.5 Simulation Parameters

Parameter	Value
World size	70008500m2
Simulation time per run	7200s(2hr)
Interface	Bluetooth
Bluetooth Interface transmit Range	10 m
Total number of node groups in the scenario	5
Nodes mobility model	External Mobility Traces
Message size range	500KB
Total simulation time	500ks
Movement model	External Movement Model

Figure 6.1: Simulation Parameters

Chapter 7

Simulation Results and Analysis

To perform model evaluation, a series of simulations have been performed in ONE Simulator. A single simulation time is 2hrs. The warm-up period of movement model is 43000s, that is half a day which is sufficient due to the periodic nature of the mobility model. The scenario is evaluated for the following parameters:

7.1 Simulation Parameter

1. Network Parameter
   - No of Nodes

2. Protocol Parameter
   - Bubble Rap with RLNC

3. Performance Parameter
   - Delay
   - Throughput

7.2 Simulation Results

Here seen in below figure Bubble Rap algorithm with Wdm with network coding is better than bubble rap with WDM. It is because network coding decreases delay in forwarding. It also enhances Throughput of the Algorithm.

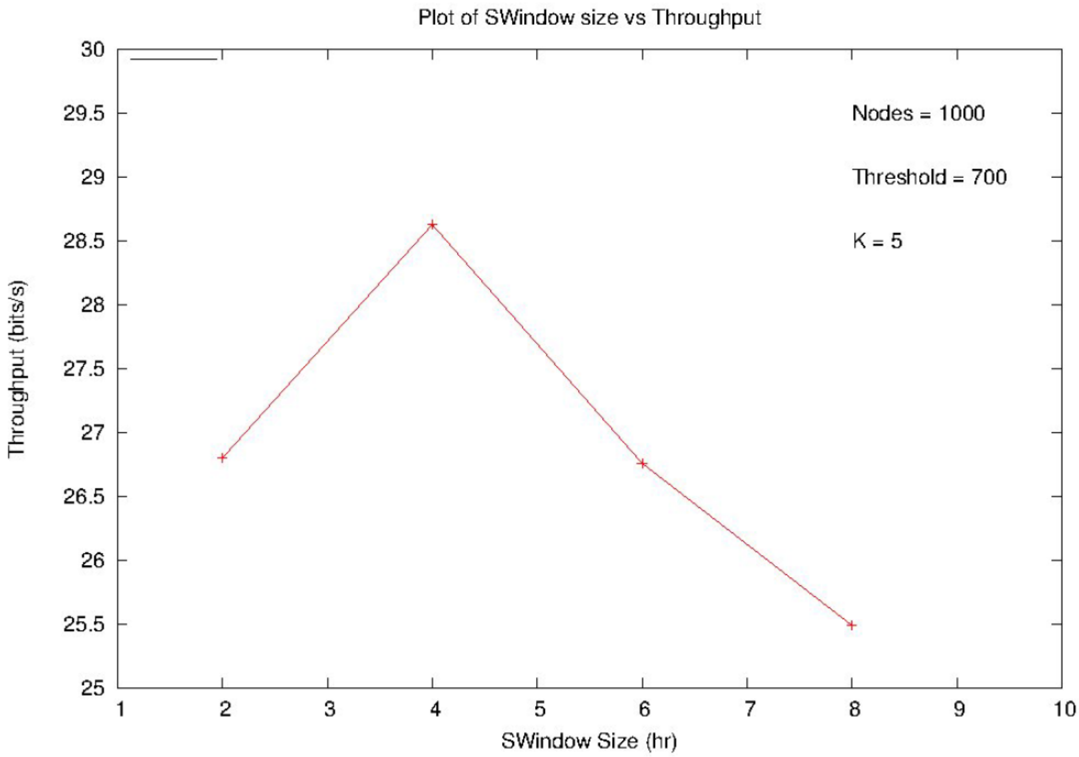

Figure 7.1: Bubble Rap with WDM Throughput

7.2.1 Scope of Project

Using Network coding in social based forwarding we can improve efficiency of the algorithm exploiting social community techniques and reduce network overhead and energy resources which are very crucial in PSN.

7.2.2 Future Work

Bubble Rap forwards data in uni-cast manner so it is not using the full power of wireless medium if Bubble Rap is implemented for multicasting them algorithm efficiency can increase.

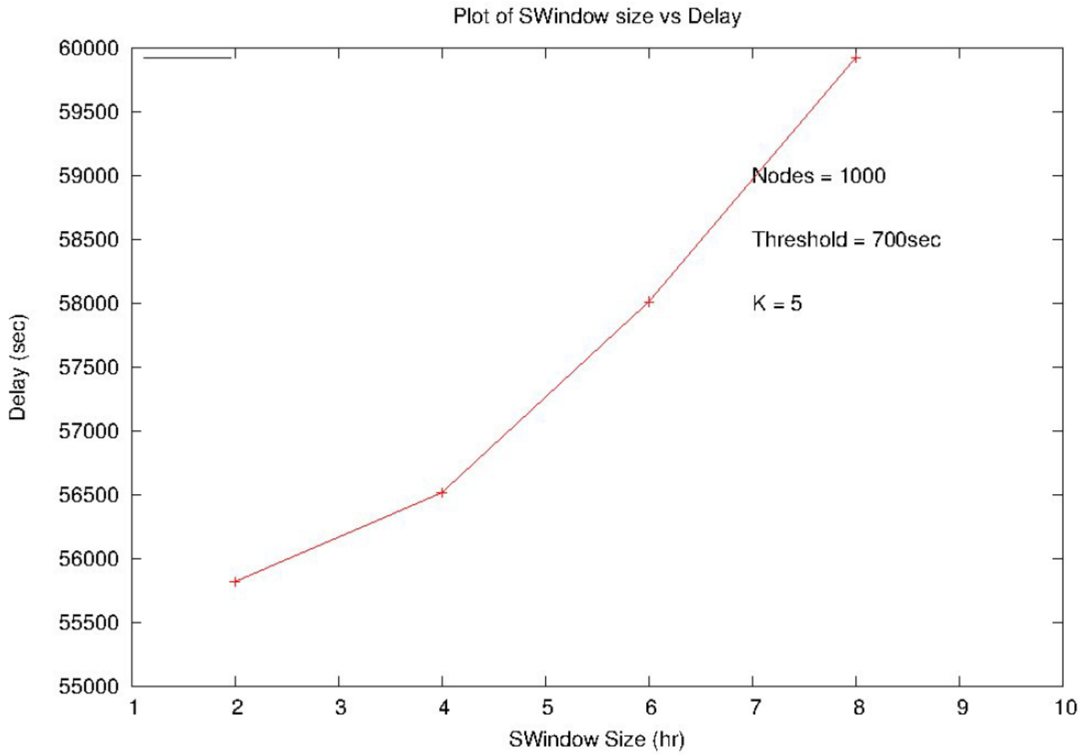

Figure 7.2: Bubble Rap with WDM Delay

7.2.3 Limitations

Only limitation i can see here is all the mobile devices can not have same processor and RAM. Here we are computing using same capacity.

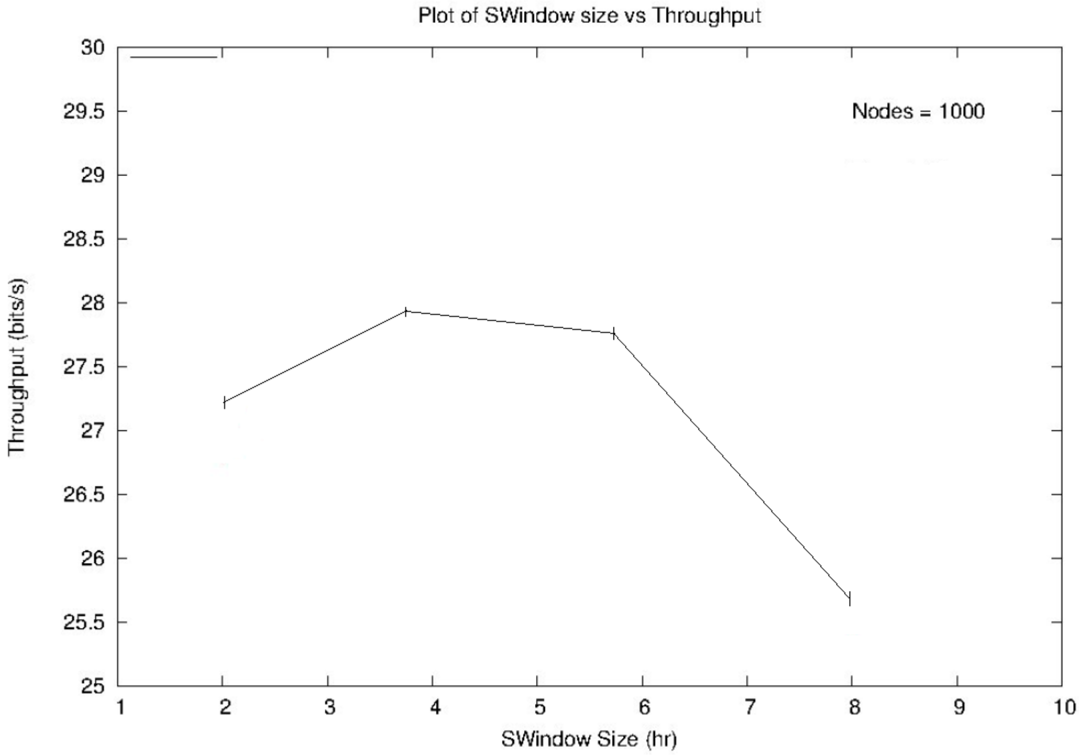

Figure 7.3: Bubble Rap with WDM using Network coding Throughput

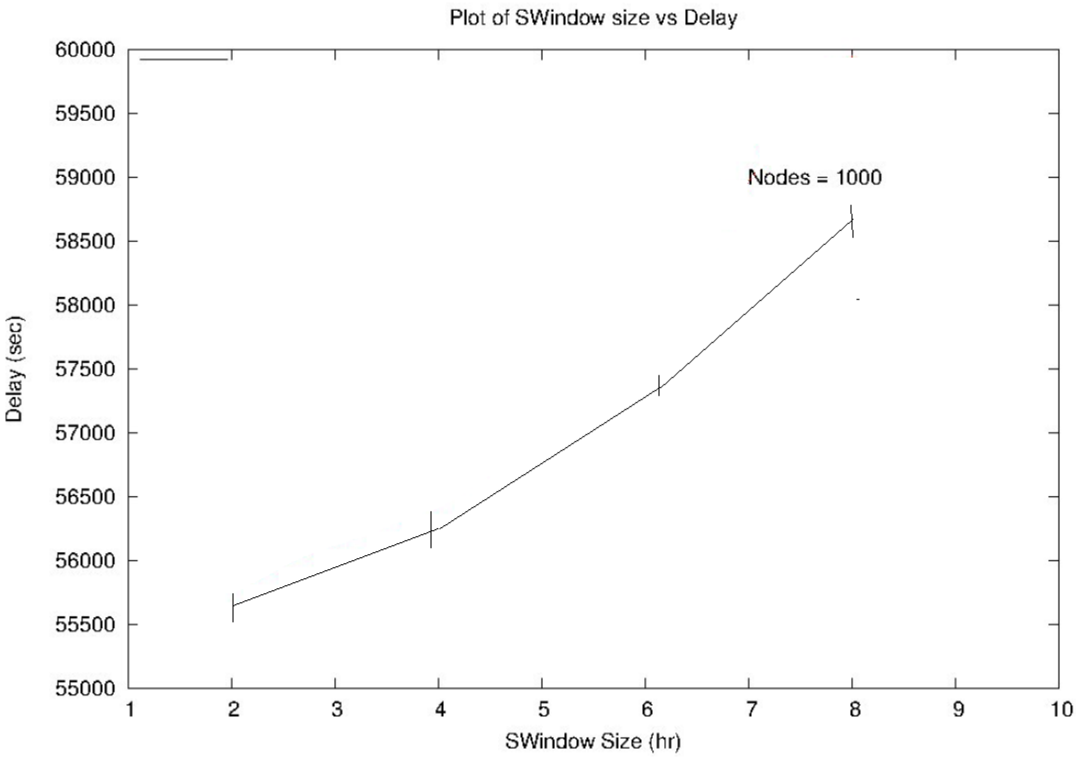

Figure 7.4: Bubble Rap with WDM using Network coding Delay

Chapter 8

Bibliography

Bibliography

[1] C. Liu and J. Wu, "Scalable routing in delay tolerant networks," in MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, pp. 51–60, 2007.

[2] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective networks," in MobiHoc '09: Proceedings of the 10th ACM international symposium on Mobile ad hoc networking and computing, 2009.

[3] A. Chaintreau, P. Fraigniaud, and E. Lebhar, "Opportunistic spatial gossip over mobile social networks," in WOSN '08: Proceedings of the first ACM Workshop on Online Social Networks, 2008, pp. 73–78.

[4] F. Chierichetti, S. Lattanzi, and A. Panconesi, "Gossiping (via mobile?) in social networks," in DIAL M-POMC '08: Proceedings of the 5th ACM international workshop on Foundations of mobile computing, pp. 27–28, 2008.

[5] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han, "WhozThat? evolving an ecosystem for context-aware mobile social networks," IEEE Network, vol. 22, no. 4, pp. 50–55, 2008.

[6] J. Scott, Social Network Analysis: A Handbook. Sage Publications, 2000.

[7] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.

[8] http://www.haggleproject.org/deliverables

[9] Ying Zhu, Bin Xu,"A Survey of Social-based Routing in Delay Tolerant Networks: Positive and Negative Social Effects", "IEEE Transaction", 13,pp.288-290,2011.

[10] Dijiang Huang,"Network Coding And Its Security Application for Mobile Ad Hoc Networks", "Arizona State University"

[11] T. Spyropoulos and K. Psounis. "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility". Proc. of IEEE Percom International Workshop on Intermittently Connected Mobile Ad hoc Networks, March 2007.

[12] J. Leguay, T. Friedman, and V. Conan. "Evaluating mobility pattern space routing for dtns". Proc. of the 25th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2006), pages 1–10, 2006.

[13] A. Lindgren, A. Doria, and O. Schelen. "Probabilistic routing in intermittently connected networks." ACM Mobile Computing and Communications Review, 7(3):19–20, 2003.

[14] C. Boldrini, M. Conti, I. Iacopini, and A. Passarella. HiBOp: "A History Based Routing Protocol for Opportunistic Networks." Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), pages 1–12, 2007.

[15] C. Boldrini, M. Conti, and A. Passarella. "Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution." Elsevier Pervasive and Mobile Computing, 2008.

[16] R. Bruno, M. Conti, and A. Passarella. "Opportunistic networking overlays for ICT services in crisis." In Proc. ISCRAM, May 2008.

[17] $haggleproject.org/deliverables/D2.2_final.pdf$. "Final specification of forwarding paradigms in Haggle"

[18] $haggleproject.org/deliverables/D5.4v2_final.pdf$. "Final test-bed for repeatable experiments and evaluation of mobile Haggle nodes"

[19] Pan Hui, Jon Crowcroft, Eiko Yoneki. "BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks" MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

[20] Tracey Ho, Desmond Lun "Network Coding: An Introduction"

[21] $http : //crawdad.cs.dartmouth.edu/meta.php?name = cambridge/haggle/imote/infocom$ "Infocom 2006 Traces"