Enhanced Community-Based Routing for Low-Capacity Pocket Switched Networks

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By

Khadija Rasul

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Sensor devices and the emergent networks that they enable are capable of transmitting information between data sources and a permanent data sink. Since these devices have low-power and intermittent connectivity, latency of the data may be tolerated in an effort to save energy for certain classes of data. The BUBBLE routing algorithm developed by Hui *et al.* in 2008 provides consistent routing by employing a model which computes individual nodes popularity from sets of nodes and then uses these popularity values for forwarding decisions. This thesis considers enhancements to BUBBLE based on the hypothesis that nodes do form groups and certain centrality values of nodes within these groups can be used to improve routing decisions further.

Built on this insight, there are two algorithms proposed in this thesis. First is the **Community-Based-Forwarding (CBF)**, which uses pairwise group interactions and pairwise node-to-group interactions as a measure of popularity for routing messages. By having a different measure of popularity than BUBBLE, as an additional factor in determining message forwarding, CBF is a more conservative routing scheme than BUBBLE. Thus, it provides consistently superior message transmission and delivery performance at an acceptable delay cost in resource constrained environments.

To overcome this drawback, the concept of unique interaction pattern within groups of nodes is introduced in CBF and it is further refined into an enhanced algorithm known as **Hybrid-Community-Based-Forwarding (HCBF)**. Utilizing this factor will channel messages along the entire path with consideration for higher probability of contact with the destination group and the destination node.

Overall, the major contribution of this thesis is to design and evaluate an enhanced social based routing algorithm for resource-constrained Pocket Switched Networks (PSNs), which will optimize energy consumption related to data transfer. It will do so by explicitly considering features of communities in order to reduce packet loss while maintaining high delivery ratio and reduced delay.

Acknowledgements

I am extremely grateful to my supervisor Dr. Dwight Makaroff for his unparallel support, continuous supervision, effort and review though out my thesis. I also want to express my gratitude to my adviser Dr. Kevin Stanley for ideas and guidance. I appreciate the time and feedback my advisory committee has given me. I want to acknowledge Md. Shaiful Alam Chowdhury for his contribution in community detection and Winchell Qian for his help in creating syntheric datasets. I will also like to thank the University of Saskatchewan Dean's Graduate Scholarship program for funding my research.

Contents

List of Tables

List of Figures

List of Abbreviations

CBF	Community-Based-Forwarding
CBC	Community Betweenness Count
DTN	Delay Tolerant Network
ER	Epidemic Routing
GP	Global Popularity
HCBF	Hierarchical-Cluster-Based-Forwarding
IPN	Inter-planetary Network
LP	Local Popularity
MANET	Mobile Ad-hoc Network
NCF	Nodal Contribution Factor
PSN	Pocket Switch Network
SM	Social Mobile
TTL	Time To Live

Chapter 1

Introduction

A Delay Tolerant Network (DTN) [24] is composed of computing systems known as nodes, for which timeliness of communication is not significant. Eventual delivery is sufficient. The nodes of a DTN are mobile and one node connects to other nodes whenever opportunity arises. Thus, these networks are also referred to as opportunistic networks [79]. An occurrence of nodes connecting to exchange data is referred to as a contact. In a DTN, source and destination nodes might never be connected to the same network at the same time. So hop-by-hop routing is used, rather than end-to-end routing. Due to mobility, hardware failures or any other events, links or connection between nodes in a DTN may be disrupted over time. As a result, whenever a carrier of a message encounters a 'suitable' intermediate node, it forwards the message (destined for the destination node) to that encountered node. These messages will be buffered at each intermediate encountered node, potentially on its non-volatile storage. This enables messages to be stored until a suitable next hop is encountered, which may not happen for a long period of time. In this way message is 'stored, carried and forwarded' by intermediate nodes to the destination node.

There are many situations that DTN technology can provide communication means to that are otherwise impossible or inefficient. One such situation involves remote nomadic societies. In the 1990's the Norwegian government faced the challenge of educating the indigenous people of Scandinavia, the 'Saami' people, who have been recently recognized and protected under the international conventions of indigenous peoples. The challenge was how to provide regular education to the Saami children without hampering their nomadic lifestyle. Bringing Saami children into boarding schools was no longer the solution, as previously it has caused these children - the next generation of Saami - to lose contact with nature and their own culture, thus endangering Saami lifestyle. Moreover, this population does not have reliable wired, wireless or satellite communication capabilities in large parts of their territory. At that time, if affordable mobile communication means existed, then it could be used to connect and enroll the children to regular Norwegian schools via 'long distance' education programs and simultaneously allow them to acquire Saami nature-based education from their tribes. Almost after two decades since this scenario, DTN technology has helped Saami people not only to be educated, but also made way for them to take part in their country's legal decisions and also contribute to the domestic economy [21].

The major challenge a DTN addresses is providing end-to-end service where end-to-end data forwarding paths may not exist between mobile nodes at any given point of time. DTN technology has always been used to bridge communication gaps in extreme environments where standard Internet protocols fail or have poor performance. For example, interplanetary network (IPN) communication [72], monitoring of epidemic spread of diseases[34], wild life tracking application ([45, 32]), providing Internet service in remote villages of South Africa (the Wizzy digital Courier Service) [42] and understanding employees' organizational behaviors in banks [60].

1.1 PSNs

Pocket switched networks (PSNs) are a special case of DTNs, where packets are routed among hand-held devices in an ad-hoc manner based on historical data regarding the non-uniform contact patterns between nodes [39, 37], which scale to inter-community contact patterns [3]. The main purpose of a PSN is to use contact opportunities to allow humans to communicate without using any infrastructure. An example of the usefulness of PSNs is when users of mobile devices are moving between connectivity zones and the current network is weak, expensive or not available (i.e. when they travel through areas that are not covered with Wi-Fi). In such situation, a PSN can be used to provide connectivity between mobile devices and potentially to a wide area network connected to the Internet.

1.1.1 Challenges in PSNs

PSNs have unique challenges due to the environment in which they are built to be used. In order to be used as an every-day's way of communication, PSN need to overcome these challenges. Some of the challenges present in PSNs are the following:

1. **Connectivity:** Standard Internet protocols structures depend solely on the assumption that there will be end to end connectivity. In absence of continuously connected network, finding an end-to-end path has been focus of interest of the MANET community (mobile ad-hoc network) [17]. Nodes in a PSN can be either connected to the Internet (GPRS or Wi-Fi) or be on their own local network. In a PSN, when two nodes belonging to two different networks (i.e. one node uses local and the other node uses the Internet) want to communicate with each other, then the challenge PSN tries to answer is how the node on the Internet will transfer the message to the node not connect to the Internet or vice versa.

2. **Security:** Throughout networking sessions in PSNs, nodes have to depend on their neighbours for data transmission. Thus, security and privacy is a very important issue in the wide deployments of PSNs. Without security and privacy guarantees, people are reluctant to accept such a new networking technology for their hand-held devices.

3. **Energy efficiency:** Devices using PSNs will first try to serve their own communication needs with their existing battery, bandwidth, CPU and memory, and then help neighbours. Thus in PSNs, along

with trusted relationships, nodes cannot be selfish and have to keep provision for neighbours. The solutions for these concerns are still open issues.

4. **User Mobility:** Mobility of the users makes pocket-switch-devices' access to other devices difficult. The communication channel and network quality for PSNs depends on coverage of the infrastructure in which the mobile user is located. Not only so, mobility of nodes makes contact patterns with other devices unpredictable. Therefore, a source node should try to exchange a message with some node that can move around remote parts of the network [64]. Routing methods in mobile opportunistic networks are based on characteristics of participants' movement pattern. In most practical scenarios, the schedules of encounters are not known in advance. Even if the schedules are known to some extent, in case of changes, a routing scheme should be dynamic enough to adapt to changes and still be able to deliver data to the destination. Hence, the mobility model of the nodes is an important parameter that determines how the nodes will encounter one another and an open research area.

1.2 Current Development in PSNs

Like many other applications, PSNs also need 'green' technology for forwarding messages. A scenario that reflects the usefulness of 'green' routing technology for inter-community message forwarding is when the cell phone companies want to make configuration changes and update software patches. Thus for them, being cost effective as well as eventually reaching all the devices (in and out of network) is a must. One can use a DTN to push debugged code throughout the network in a manner that requires eventual consistency [47, 15]. So existence of an energy efficient PSN routing technique will be a natural fit for forwarding messages and upgrading their overall system.

Another use of PSN is in habitat monitoring. The recording of sensor values (many-to-one) for long-term trend analysis can be tolerant of significant delays, particularly in the case of habitat monitoring [45] and in cases where energy consumption is a primary resource constraint, rendering a multicast/broadcast technique such as flooding impractical [66]. A subset of devices detecting a critical measurement change might need to inform another subset of nodes to alter their measurement strategy (many-to-many).

One more area that can make use of energy efficient inter-community routing technology is in the villages of the developing countries, such as India, Bangladesh and Cambodia. In rural parts of the third world countries, 'Internet' is still an unfamiliar word whereas 'Cell phone' is more common term to the mass people living there. These countries' governments have been trying to achieve the goal of "Internet connectivity for everyone" (in Bangladesh this vision is known as 'Digital Bangladesh' [71]). However, due to poor economic conditions and disruptive power supplies in villages, providing permanent infrastructure which supports standard Internet protocols is not being possible. In such scenarios, PSNs can play a major role in accomplishing the goal of 'Internet connectivity'. In this condition by using PSNs, perhaps a village school teacher can send his/her tax forms over e-mails to a government office located in large city with permanent

Internet connectivity, in couple of hours. Such overlay network, using PSNs, can be formed by using the local network of cell phones belonging to the people moving from the village to towns to cities (i.e. inter-community routing). Thus such PSNs will not only provides a way to communicate but also can help low earning villagers save travel expenses. This example is based on the Daknet project [63].

The next question after knowing cost-effective usage of PSNs is, which devices are most cost effective and how to use them? The explosion of portable technology has led to many new devices including smart phones, personal medical monitoring technology and smart badges, which could be used as devices in DTNs. These portable devices have varying levels of software complexity, from multi-process, multithreaded apps built atop an operating system to small, purpose-built, single-process embedded software. These devices can be tasked with environmental or context monitoring activities [32, 33, 45], as well as provide communication services between humans associated with such devices. The sensing methodology and parameters embedded in portable devices, and associated software for performing the measurements may, however, need to adapt in ways impossible to capture in an *a priori* manner. Multiple message generation paradigms can be anticipated that have delay-tolerant properties. Each falls into one of the following four categories: a) one-to-many, b) many-to-one, c) many-to-many, or d) one-to-one. Ideally, to be cost effective, DTN routing protocols have to reduce the numbers of transmission/reception operations performed without compromising on existing delivery ratio or latency. Most optimizations are done keeping 'smart-phones' in mind as the hand-held device. However, smart phones are expensive when it comes to large scale use. A cost effective replacement of smart phones in DTN-type communication is smart badges. Compared to smart phones, smart badges benefits more from optimization and cost effective DTN routing protocols. This is because devices like smart phones have varying computational and networking capabilities, ranging from multiple radio channels with significant CPU power, so smart phones configuration adaptation is relatively straightforward as many communications channels, local short-range and long range, are available. On contrary, smart badges have limited computational capacity and can communicate only using short-range channel so local-only information distribution must be adopted. So from all these scenarios of PSNs, it can be said that when it comes to cost reduction, developing a reliable routing strategy for low capacity PSNs that utilizes resources efficiently is a necessity.

1.3 Motivation and Thesis Contribution

Over past few years, the unique requirements and the usefulness of PSNs have been attracting researchers all over the world. As mentioned in the previous section, among all the challenges, delivering packets reliably in a PSN is a primary challenge. Over past decade, plenty of research has been going on to find optimal routing algorithms. Variety of approaches, starting from flooding [82] to routing based on network coding [84], have been proposed. Since PSNs deal with hand-held devices and people's mobility pattern, a naturally fitting building block for making routing decisions is social relationships. Examples of routing schemes developed on social context are Label [38], BUBBLE-Rap [39] and PROPHET [50]. The principle behind choosing human interaction as a basis for routing decisions in all these above mentioned schemes is because it is less volatile, requires fewer exchanges of control packets and updating of routing table [39] compared to using routing tables built only on mobility as seen in many MANET works [44]. Thus, it is cost effective.

All the above mentioned algorithms have their target to cut down cost by reducing the computational cost needed for updating routing vectors/table, but none of these schemes have any follow-up work for optimizing the cost further. It is known in sensor networks, however, that more energy is consumed in transmission and reception of data than in computation. As a result, even though these algorithms aim for being cost effective, they overlook the main source of cost, which is the energy needed for transmission and reception. Thus, a promising area of PSN research is development of a routing strategy that provides reliability (i.e. high delivery ratio) as well as energy efficiency (i.e. low transmission cost). This thesis aims to contribute in this particular area.

This thesis outlines two proposed solutions, Community-Based-Forwarding (CBF) and Hybrid-Community-Based forwarding (HCBF), for this challenge. The concept behind these solutions is inspired from a previous work known as BUBBLE-Rap [39] developed by Hui *et al.* The two concepts are introduced sequentially to the basic idea of the BUBBLE algorithm are: 1) Inter-community betweenness 2) Mobility of nodes within a community. It is expected that these enhancements will make forwarding more efficient. The message delivery performance of the combined solution should have an upper bound of BUBBLE performance and a lower bound determined by the clique structure of the dynamic contact network. In terms of the latency of message delivery, the performance of combined proposed solution falls between the performance delay of Epidemic routing [82] and that of direct contact message passing [77]. The proposed algorithms provide similar performances in resource rich environments. Moreover due to their more conservative approach, they have consistently superior transmission thus power performance in resource constrained environments as well.

The primary focus of this thesis is inter-community routing. This is because humans are social beings and live in groups/families/clans thus tend to be more in physical contact with neighbours from same group more than with people from other groups. Thus transferring messages inside groups are done more easily than transferring messages between groups, as in the latter case, the probability of direct contact between source and destination is less. In this work, the terms 'group' and 'community' are used interchangeably. The difference in contact behaviour between communities suggests the need for a routing approach that considers inter-community contact patterns in forwarding decisions for message that have to leave their community boundaries. These dynamics also suggest a routing approach that preferentially considers inter-community (inter-clique) contact patterns may be more efficient and give better performance. In a perfect scenario, one solution would be to create dynamic multi-hop routing tables, where nodes would keep track of their dynamic path length to all other nodes. In a real world scenario, such tables lead to an explosion of overhead, however, which rapidly outstrips the capacity of the PSN [50]. Aggregated routing tables, which are small enough to maintain efficiency, but informative enough to provide superior routing performance over naive node-level systems would need to be developed. For PSNs, which are enabled by human mobility, a highly efficient method is to leverage the community structures of the small world networks that naturally arise from human interaction patterns [39]. This thesis takes these observations a step further, and directly leverages the clustered network structure to provide superior routing performance, with fewer resources, providing a particularly compelling routing algorithm for resource-constrained networks.

1.4 Thesis Statement

From the examples given in the previous section, it can be seen that there is need for cost effective inter-community message forwarding method that can be used for communication in 'extreme environment'. The suggested algorithms, **Community-Based-Forwarding (CBF)** and **Hybrid-Community-Based-Forwarding (HCBF)**, fulfill these needs by being the building blocks of a social based forwarding scheme which explicitly considers features of communities, such as group-betweenness and social diversity, in making message-forwarding decisions in order to improve the delivery performance in terms of message transmission, latency and energy use while maintaining same delivery ratio. This is evaluated by carrying out experiments in both resource rich and constrained environments, where either buffer capacity of nodes or TTL of message have been varied while keeping the other constant. Overall, the enhancements have been been done and performance experiments have been conducted via simulation to quantify the extent of message transmission reduction in scenarios of communities and communication paradigm.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents a brief overview of previous work done on routing strategies, related clustering techniques available and similar algorithms developed on Social-based forwarding. Chapter 3 provides details of the datasets and assumptions made about the test environment, describes the design of the simulator used for evaluating the proposed algorithms and quantitatively analyses the datasets to find communities. Chapter 4 focuses on the primary proposed algorithm CBF and quantitatively analyzes results from experiments on sample sets of messages sent between nodes of different communities using CBF and other comparator routing algorithms. Chapter 5 sheds light on how CBF is converted to HCBF to reduce delay. Chapter 6 offers conclusion and potential scope for future work.

Chapter 2

Related Work

The research problem of finding an efficient social based forwarding technique encompasses two major sub-problems. One problem area's focus is how to identify different levels of social connection in a group of nodes (i.e. community detection). The other problem area is the issue of 'Routing' in a delay tolerant network. Both of these issues combine together to form the challenging area of community based routing protocols. This chapter gives a survey of the existing community detection techniques, the basic schemes of DTN routing and the recent similar works done which are related to the proposed algorithms. It also includes a summary that points out how the proposed solution extends similar algorithms.

2.1 Basics of Community Detection

Starting from people in primitive hunting groups to people in multinational business organizations, social communities are ubiquitous in human world. With advancement in technology, we now have electronic databases that can record details of human mobility in a community, giving us novel scope to map and explore the formation behind social and collaboration network. A few application areas that make use of this information are the following: scientific co-authorship and mobile phone communication [62], online social networking sites [81], biological systems [48], Human Resource (HR) departments of banks [60], exchange of media files in peer to peer networks [51], etc.

2.1.1 Basic Terminologies in Social Networks

All of the above examples can be considered Social Networks. Starting from Communication to Biology and Sociology, social network analysis is a popular research topic and an investigation tool that is regularly used. In broad sense, the study of connection between individuals and their effects on the community is referred as 'Social Network Analysis' [83]. Community refers to group of people having similar interest thus locating in a common area, either physical or virtual. Therefore, communities naturally reflect social relationships between people. The most common way to analyze and represent interactions in a social network is by social graphs. A social graph is a virtual representation of contacts between vertices where vertices represent individuals or nodes and edges represent the connection between them. In case of DTNs, if there is an edge present between two nodes, then there is a possibility of opportunistic forwarding. These edges can be weightless or have weight, representing strength of the social tie between the two individuals represented by the nodes. With the aid of social graphs, many different graphical metrics can be calculated. The fundamental social metrics are

1. **Betweenness:** Betweenness [28, 30, 83] of a vertex is a measures of number of shortest path formation that vertex has contributed to. Removal of such vertex increases overall average distance between other vertices in the network. In DTN context, higher betweenness means that a node is a good bridging node in a network as it helps connection between other nodes.

2. **Similarity:** As the name suggests, similarity measures how close two vertices are [18]. In DTNs, one definition could be the number of common neighbors between two nodes.

3. **Centrality:** Centrality [28, 57, 59] refers to importance of a vertex in a network topology. In DTN context it refers to social importance of a person. For example: Hubs in networks have highest centrality values. Degree of centrality is the number of links ending or beginning from a vertex.

2.1.2 Community Detection

Community detection [26] in a social graph is an example of usefulness of these metrics. To find communities, the graph needs to be partitioned into several graphs that share important characteristics determined by the analysts. The measure of quality of this breaking a graph into communities, such that maximum integrity is maintained, is known as Modularity [58].

Whenever any application uses the concept of community, it deals with the structures that exist within the community. In case of community-based message forwarding, the performance is affected most by the mapping from the contact information to the aggregated social graph. Other social metrics are much less influential. Hence, the social graph created out of past contacts should best reflect the underlying (mobility or social) structure generating these contacts, so that nodes can be meaningfully differentiated and edges have predictive value to base decisions on. Thus, community detection techniques are a vital feature of social based routing. Based on the partitioning technique, community detection can be classified into two broad categories as discussed below:

1. **Agglomerative method:** At one extreme of community detection spectrum lies the Agglomerative method. It is a classical method that detects inter-connected groups in a graph hierarchically. It is also known as a 'bottom up' approach. In general, this method starts with a weighted set of edges between nodes and then uses them to form hierarchy of groups and communities. At every step, similarities between clusters are compared and those with maximum similarity are joined until all nonzero similarity clusters are connected. Different agglomerative methods utilize different measures of the similarity between clusters [31, 43]. There are several established approaches which fall under this criteria which are described as follows:

(a) **Self-organizing method:** Using similarity to find clusters is a very old traditional clustering technique that has been used over decades by statisticians. Such techniques, in broader terms, can be referred to as self-organizing clustering. This method operate in two modes: training and mapping. During the 'Training' session, a map is built using the input examples where edges are given weights and based on these weights nodes are clustered. Then the "Mapping" session automatically classifies input into clusters. As it falls under the agglomerative method, a self-organizing technique always starts with single nodes. Then finally ends up with a flat structured partially-connected graph as output. Common examples of self-organizing clustering techniques are K-means Clustering and Neural Network Clustering [29].

(b) **Modularity method:** The next set of methods fall under the Modularity optimization method. In graph theory, Modularity is a measure of the strength of the structure of networks or graphs [58]. Networks with high modularity have dense connections between the nodes within groups but sparse connections between nodes in different groups. There are numerous community-finding algorithms that try to optimize modularity or similarly constructed quality functions in various ways [7, 20, 26]. When modularity optimization algorithms like the ones proposed by Clause *et al.* [16] and Louvain [7] are compared to other well-known clustering algorithms such as K-means or Spectral Clustering [25, 65], the advantage modularity optimization algorithms have over the others is that they do not require any advance knowledge such as final number of communities in a graph. So they are comparatively dynamic. There are many modifications of this method [7, 58].

2. **Divisive method:** At the other end of the community detection spectrum exists the Divisive method, in which one starts with the full graph and then divides it to find sub-communities [74]. Due to its nature, it is also called a 'top down' approach. Given that a network can be divided into non-overlapping components, one can keep partitioning it until a community is composed of a single node (singleton community). This sub-grouping of the nodes can be represented using an hierarchical tree often known as a dendogram [57]. Some community detection approaches which fall under 'Divisive method' are discussed below.

(a) **Centrality-based method:** A popular way of partitioning a network is by Centrality-Based Community Detection, which is first proposed by Michelle Girvan and Mark Newman [56]. If a node or an edge exists in the shortest path of message delivery, it is considered to have high betweenness count. The betweenness of a node or edge quantifies such traffic by considering strictly shortest paths (geodesic betweenness) or densities of random walks (random walk betweenness). An iterative implementation of these methods gives a divisive algorithm for detecting community structure, as it decomposes the initial graph into progressively smaller connected chunks until one obtains a set of isolated groups of nodes based on the "togetherness" measure used. Although this method is intuitively appealing, it is a slow method for large networks.

(b) **Kernighan-Lin (KL) Algorithm:** Another example of divisive method is Kernighan-Lin (KL) Algorithm. In this algorithm, the authors tried to maximize the *quality function* by comparing the number of edges inside a group to the number of edges between different groups. The partitions of networks into communities that are obtained using the KL algorithm depend strongly on the initial partition, therefore it is best used as a supplement to high-quality partitions obtained using other methods.

There are other useful existing methods of community detection which cannot be strictly put under a specific category. One such popular hybrid method for clustering is Spectral Partitioning. The method is used extensively in VLSI CAD applications for load balancing parallel computation ([25, 65]). Other hybrid methods like the Potts method [6] and the Resolution parameter [27] are very well known and domain specific.

2.1.3 Louvain Algorithm

This thesis uses the Louvain algorithm to detect communities. The Louvain algorithm [7] is a heuristic solution that finds out approximate communities without any advanced information. Later, any computation done based on these groupings have been seen to reach performance close to actual known results [64] obtained by direct observation and self declared communities. This Modularity-based method falls under the category of Agglomerative methods.

This thesis employs the Louvain algorithm [7] to find clusters of nodes characterized by frequent, sustained contacts because it is fast, simple to implement and unlike many other algorithms, does not need to predefine the desired number of clusters before performing clustering. Sufficient numbers of contacts must be included for the clustering to be an effective predictor of future contact. This algorithm works in two iterative phases. During the first phase, each node is considered as a separate community. Then during each iteration, the algorithm takes each node p from its own community and puts it in the cluster of each neighbour q of p and measures the gain in modularity of the whole network, where the modularity is defined as the density of links inside a community compared to the links between communities. Finally, the node p is placed into the cluster of the neighbour that will maximize the overall gain of the network and this gain has to be positive. If the gain is negative or zero, node p stays in its original community. This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved and the first phase is then complete. This process is repeated until no further positive gain is possible.

During the second phase, each of the new communities is considered as a single node and the first phase is repeated. This procedure continues until a locally optimal/near optimal point is reached or a prescribed number of iterations is reached. In this phase, the algorithm builds a new graph based on the clusters found in first phase. The edges between the new clusters are weighted by the sum of the weights of the edges linking the two clusters in the previous graph. The algorithm re-iterates using this new graph as input and it stops once the system is stable (i.e. no new clusters can be formed in the new graph). It will leave outliers in a solitary state of single member communities. Besides forming the optimal number of communities, no inappropriate nodes are included in a community when the Louvain algorithm is applied.

2.2 DTN Routing Basics

Routing in DTNs is a major challenge where nodes try to communicate opportunistically with each other in absence of infrastructural support. From a broad perspective, the existing routing solutions in DTN can be categorized into two classes. One of the categories is 'Uninformed' routing, where nodes try to reduce delay by generating multiple copies of the same message. Even though this approach achieves lower delays and higher delivery ratio, it is more resource (energy and buffer space) consuming. The other category can be referred to as 'Informed' routing. In this approach, nodes are aware of the state of the network and they try to minimize usage of buffer by maintaining a minimum number of copies of each message. The potential benefits and trade-offs of DTN routing policies were initially examined using a combination of simple routing strategies which fall in 'Uninformed' category and oracle algorithms [42] which is an extreme case of 'Informed' routing strategy. Some strategies under the latter category utilize the concept of a community and make different routing decisions based on community membership. In this section, a few current approaches under each of these strategies are discussed.

2.2.1 Uninformed Approaches

Under Uninformed Routing strategies, every node broadcasts the received packet to all its neighbours. The earliest works in DTN fall under this family. The basic protocols in this family do not need any information about the network and there is a clear trade-off between cost and performance.

1. **Single-Copy Approach:** Single Copy Approach, such as the Randomized routing algorithm proposed by Spyropoulos *et al.* [77], is the simplest and most naive approach under Uninformed techniques. Many resource-conserving strategies consider the single copy case. In this scheme, either the source directly delivers message to the destination (i.e. direct delivery approach) or it delivers to any node that has an expected lower delivery time, found by using a 'gradient' function, for the message to reach its destination. In the gradient function, each node is assigned a weight that represents its suitability to deliver messages to a given destination. When the carrier of a message contacts another node that has a better metric for the message's destination, it passes the message to the encountered node. This function is called gradient fuction because the message follows a gradient of improving utility function values towards the destination. Overall, this scheme does not make an additional copy of data nor does it consume any additional cost, but it exhibits high latency and potentially poor delivery ratio, particularly when packet TTL indicates delay tolerance but not delay immunity. Also, if proper precautions, like advertising the list of messages already received, are not taken, then there is a chance of forming forwarding loops.

2. **Epidemic routing:** The opposite scheme to the single copy approach is epidemic routing (ER). The basic concept of epidemic routing is to flood the packets, like the virus spreading in an epidemic. Since reliability of message delivery is uncertain in DTNs, the intuition behind this approach is that having more copies of the message increases the probability that one of them will find its way to the destination, and decrease the average delivery time. Each message is sent over all possible paths, delivery of message takes minimum time; thus, it provides optimal delivery times and reliability if there is unlimited buffer capacity in each node. Technically it is performing a parallel depth first search of the dynamic contact tree, rooted at the sending node. Since nodes do not know if the destination has been contacted, multiple receptions of the same message occur. A message is only deleted from a buffer when the destination is encountered assuming infinite buffer space. Unfortunately there are a large number of redundant messages occupying intermediate nodes' buffer space and exchange of messages on every encounter drains nodes of energy and bandwidth. ER has a low delivery ratio if packets must be dropped due to buffer overflow. Overall, this scheme has no knowledge about the network and is likely to cause premature network and node failure due to large number of transmissions. Vahdat and Becker [82] described perhaps the earliest variant of this approach. There are many variations of this approach [67]. In trying to modify ER, researchers have tried to constrain the growth of multiple copies through adaptive limitations on packet time-to-live (TTL) [32] or by only passing additional packets through privileged nodes [4, 35]. While these approaches tend to improve both delivery ratio and latency, it is unclear whether these trade offs are worthwhile, given the delay tolerant properties of the data.

3. **Hybrid Approach:** There are many hybrid approaches existing between single copy and epidemic routing that try to make an optimal trade-off between delay and resource consumption. One such approach under 'Uninformed' criteria is Tree-Based Flooding scheme. An example of this approach is the Spray and Wait algorithm proposed by Spyropoulos *et al.* [75]. In this scheme, source node 'sprays' a number of copies of a message into the network of N distinct nodes and then 'waits' until one of these nodes meets the destination. The scheme is very scalable as node density goes up. However, finding out the 'optimal' number of messages to be sprayed that ensures higher delivery ratio is the challenge here.

   Recently, introducing the concept of network coding [84] in DTN has drawn much attention in the ad-hoc network community as it helps to reduce loss in data transmission. Network coding comes from information theory and can be applied in routing to further improve system throughput. The basic idea of coding method is to encode an original message into a large number of coding blocks and then transmit these blocks to intermediate nodes. Suppose the original message contains n blocks. Using coding schemes, the message is encoded into k blocks ($k > n$) such that if k or more of the n blocks are received, the original message can be successfully decoded. Simulation results show that the packet delivery ratio using network coding is much higher than those schemes that depend on probability for forwarding. The process of determining how many and which messages will be coded together poses significant challenges, especially if this is to be done in a distributed manner.

2.2.2 Informed Routing Approaches

The second category of routing in DTNs is the Informed Routing Approach. Under this approach, to select the best path for forwarding a message to the next hop, routing schemes use network topology and information related to neighbouring nodes. Schemes find optimal paths by making forwarding decision based on metrics that are assigned to nodes or links. By definition, the strategies in this family require some knowledge about the network.

1. **Oracle Based Approach:** At one extreme, a node can make decisions with zero knowledge about the network, except for the contacts which are currently available as in the uninformed approaches and at the other extreme, a node might need to know the complete future schedule of every contact in the network. In reality, the latter case is only possible after experimentation is done and offline data analysis is performed. In the oracle method, wherever in the information spectrum a scheme may be, typically each node sends a single message along the best path, so they do not create any extra copies nor redundantly use resources. Oracle Based Approach was investigated by Jain *et at.* [42]. The authors proposed four routing algorithms (Minimum Expected Delay, Earliest Delivery, Earliest Delivery with Local Queuing and Earliest Delivery with All Queues) based on presence of a knowledge oracle. Under all four algorithms, the DTN is modeled as a directed time-varying multi-graph with link costs based on propagation delay and link capacity. Other schemes' success is measured by comparison with the relevant oracle. The goal of this approach is to maximize delivery ratio and minimize delivery latency, based on incomplete information or heuristic approaches.

2. **History Based Approach:** Another informed routing approach is the 'History based' approach, which uses the 'encounter history' between nodes to make more informed routing decisions. The principle behind this approach is that if a node encounters destination many times, then it is likely to encounter the destination again in the near future. This concept of temporal locality is borrowed from operating system data references. The Zebranet project [45] is one of the earliest attempts to use history of encounters for routing decisions. Lindgren *et al.* [50] proposed PRoPHET as a similar approach to DTN routing, which models nodes future contacts directly from contact history. Overall, the strategy is designed to take advantage of non-random mobility and contact behavior of nodes as in a typical real world scenario. The realistic nature of the work has inspired many researchers to further enhance the scheme.

   Such an enhanced scheme is proposed by Boldrini *et al.* [8]. In this work, each node maintains its encountered neighbour's identity table and then uses this table to learn opportunities of connectivity pattern based on the history of the node relationships and movements among users. Similarly, another work called Context-Aware Adaptive Routing (CAR) [55] forwards packets to nodes with the highest probability of seeing the destination. The node-level bookkeeping required for these algorithms can become prohibitively expensive for large networks or multi-hop routing.

3. **Social Based Routing:**

   Instead of relying on individual node models, some researchers have attempted to model the overall behavior of the dynamic graph by segmenting the graph into cliques based on the social-behavior of the users, cliques are called communities in the rest of this thesis. Such strategies are referred to as social forwarding approaches.

   One of the earliest works in this area was done by Chaintrau *et al.* [13]. This work analyzes the impact of human mobility on the design of opportunistic forwarding algorithms based on six real human mobility traces from four different research groups. After studying six PSN traces, authors of this paper empirically proved that in an opportunistic network inter-contact time is heavy tailed, which is not considered in traditional mobility models for Ad-hoc network. Thus existing mobility models are not sufficient. They also show that the contact process of two people are different and depends on the communities they directly or indirectly belong to. This foundational work is one of the first to considers inclusion of community features in order to improving routing.

   Another work that studies inter-contact time is done by Sastry *et al.* [73]. The authors find that contact distribution between node is not symmetric and contacts can be categorizes into 'rare' or 'frequent'. In this work, 'rare' contacts refer to contacts which occur less than ten times in the entire trace and more than that is considered as 'frequent' contacts. The authors quantitatively analyses and show that considering of 'rare' contacts help improves delivery ratio in PSNs, concluding that the presence of unevenness in contact strengthens connectivity of PSNs. This work is one of the major motivation for this thesis to pursue the scope for improving inter-community routing as inter-community contacts fall under their criteria of 'rare' contacts.

   Another related work motivated from the finding of Chaintreau *et al.* [13] is done by Musolesi *et al.* [54]. This work proposes a community based mobility pattern for mobile ad-hoc research. This scheme groups hosts into communities based on their social relationships and then models their movement which dynamically changes with the change in their social ties.

   An extension of the work of Chaintreau *et al.* [13] and Musolesi *et al.* [54] is done by Boldrini *et al.* [9]. This work also aims to understand the impact of different human mobility patterns on the routing protocol's performance in a delay tolerant network; the authors conclude that the social information-based routing reduce overhead and congestion while improving quality of service.

   In a framework called ContentPlace [10], proposed by Boldrini *et al.*, nodes advertise their objects of interest along with their weighted utility. Simultaneously during that encounter, a node transmits the list of objects it is carrying with its neighbours. Overall, this method exploits the social relationship to transfer messages and the presence of utility values helps nodes decide their priorities. Similar work is done by Ioannidis *et al.* [41] and Boldrini *et al.* [11], where authors of both works have shown how the social network can help dissemination of dynamic contents.

Another upcoming approach in the world of social context is the use of hierarchies. Social-based Hierarchical Routing in Delay-tolerant Networks [70] is an example of that. In this work, distributed K-CLIQUE community detection algorithm is used to uncover the communities of the network, and then construct a hierarchical community structure based on relation graphs in the node and community layer [70]. With this hierarchical structure, the authors developed a new routing protocol that considers intra-community and inter-community routing schemes.

Zhang and Zhou [87] proposed a related routing approach that considers contact probability and duration. Authors of this work have been motivated by the fact that buffer capacity at the receiver's end and duration of contact between nodes might not always be sufficient for data exchange in a PSN. So to improve reliability and latency, it important to design routing protocol which considers contact probability and order of data propagation. By considering the need for managing the order of data dissemination, this work adds a new dimension to social based routing approaches. This work follows the intuition that friends in the real world generally share more common interests than strangers, spending longer periods of time together. This in their work is referred as 'Homophily'. Overall, this work can transmit more information over the same contact opportunities and time. Best part about this work is that it does not conflict with any previously proposed ideas thus can be easily added to enhance other existing routing algorithms. Despite their promising results, there are two potential drawbacks. First, for most DTNs, the relationship among the mobile nodes cannot be known in advance. Second, self-reported friendship networks might not always reflect the actual contact patterns of the nodes [5, 64].

There is a whole paradigm of work done under social context routing where people or nodes are categorized under different labels and the forwarding of messages are carried out. One such work by Spyropoulos *et al.* [76] states that traditional DTN mobility model do not hold in human society due to presence of nodes having heterogeneous natures and based on this statement, authors of several follow-up works have proposed models which are adaptive to changes existing in PSNs. One such work is conducted by Miklas *et al.* [52], where people are grouped under the categories of either friends or strangers. When routing is carried out, it resulted in efficient protocol with higher delivery and effective security. Similar work is done by Hui *et al.* [38] proposes the Label Algorithm. The motivation behind Label is reduction of congestion in PSNs by using 'tagging' system. In this work it is seen when communities are identified using tags or 'Labels', efficiency in data dissemination increases. As next hops for messages, this algorithm always chooses only members identified as members of destination group. The strength of this algorithm is its cost efficiency and ease of its implementation. However, in PSNs, self identified communities do not always reflect the geographical connections, which is needed for data delivery, thus messages routed via the Label algorithm have a chance of experiencing more than expected delay.

Another related work done by Yoneki *et al.* [86] is an improvement over Miklas *et al.*[52]. This work proposes a visualization tool, which can imply different detection algorithms to detect communities from mobility traces. The authors claim that such a tool is important as it helps to gives insight into social behavior which leads to formation of communities and also helps to visualize network states that affects human interactions, which in turn has influence on the data dissemination process. This work categorizes nodes in four categories, named: friend, familiar, familiar stranger and stranger. All these works focus on the heterogeneity, which exists in human community, through which they are able to capture real-life mobility features in forwarding schemes.

SimBet [18] employs betweenness and similarity metrics to identify 'bridge' nodes to route packets. In this work, nodes who can bridge other nodes in the network have higher betweenness value and nodes which have more common neighbours have higher similarity value. Both of these values are calculated and updated dynamically in the DTN. This work uses both metrics to identify suitable carriers in different situations. This work points to the fact that use of multiple social metrics enhances social based algorithms' performance. This work is also a motivation for this thesis to explore more into small world characteristics of social networks. Features characterizing the small world networks are chosen because PSNs are examples of small world networks [14]. Betweeness is used by Hui *et al.* [39] to develop BUBBLE, which is a routing algorithm that also greatly emphasizes on social characteristics. As BUBBLE is the background work behind this thesis, the following section specifically describes this routing strategy in detail.

Fabbri and Verdone proposed a routing algorithm [23] which utilize various metrics for sociability of individual nodes, intended for use in a vehicular network scenario. This work is based on the idea that nodes contacting others frequently are good carriers for DTN routing. This algorithm targets those nodes which have the largest number of encounters with their neighbours. They distinguish two types of sociability, one in which a node has many contacts with a single other node, and one in which a node visits many different locations in a short time. They are both useful for popularity ranking in a highly dynamic environment. During calculation of sociability of a node, the impact of presence of social neighbour is considered as a positive gain.

2.3 Community Based Routing

Several works in the field of DTN have concentrated on the details of community structure and social behavior of nodes to forward packets opportunistically through out a PSN. Closely related works that base their forwarding decision on clustering, not necessarily inter-community and intra-community structure, include the following:

1. **Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Network:** Dang and Wu's protocol [19] lets each mobile node learn otherwise unknown and possibly random mobility parameters and then allowed the nodes join together in a cluster with other mobile nodes that have similar mobility patterns. The nodes in a cluster can then interchangeably share their resources for overhead reduction and load balancing in order to improve overall network performance. Due to the lack of continuous communications among mobile nodes and possible errors in the estimated nodal contact probability, convergence and stability become major challenges. So routing tables are used. Here the *gateway* nodes are the most globally popular nodes which are used for transferring messages between communities. Due to the use of routing tables, there is a continual need for updating them in a dynamic environment. As a result, for large scale use of this algorithm will not be cost effective and will not scale well.

2. **Friendship Based Routing in Delay Tolerant Mobile Social Networks** Friendship-based routing [12] proposes a new metric, called social pressure metric (SPM), to detect contact relations between nodes accurately based on friendships. Then, a new routing algorithm is used in which each node forwards their messages to nodes that contain the destination node in their friendship communities. The authors define better friendship (or link) based on frequency, longevity and regularity of friendship. The calculation of social metric SPM needs entire history of contact of each node, which clearly is a drawback for PSNs in resource constrained environment. Also it uses self-declared communities to identify nodes and geographic contacts might not always coincide with the self declared ones leading to higher delay. This problem is also seen in Label [38].

3. **Cluster-based Forwarding in Delay Tolerant Public Transport Networks** Although Cluster-based Forwarding in Delay Tolerant Public Transport Networks [2] belongs to the MANET research area, rather than to PSN, but the concept is close to the Community Based Forwarding (CBF) approach designed in this thesis. In this paper, authors have demonstrated how clustering of nodes in a DTN can enhance the packet forwarding performance. They have placed public buses into groups as per their frequency of contact and then used those groupings to perform simple forwarding. Their routing algorithm is a simple flooding strategy and they do not make use of any social natures of the nodes to make the forwarding decisions.

4. **Lobby Influence: Opportunistic Forwarding Algorithm based on Human Social Relationship Patterns** Like this thesis, Lobby-Influence [46] is an enhancement work over Bubble-rap [39]. In this work, the influence of the clique structure derived from the dynamic network dominates the routing protocol employed. Similar to this thesis, Lobby-Influence [46] highlights the twin problems of clique-based routing through dynamic contact networks: clique definition and routing policy deployment to exploit these cliques. They also try to balance load by considering 'Popularity' [39] and the 'Lobby Index' [1] together during each forwarding decisions. The strength of this algorithm is that

nodes in this work can belong to more than one community, but in a large network, it will definitely increase need of storage in a resource constrained PSN devices and also complexity of algorithm in the long run. This work did not consider dynamic changes in communities, which the proposed algorithms have considered. Moreover, like BUBBLE, they detect communities using K-CLIQUE, so problems of community detection discussed in Section 2.4.2 also persist in this work. Moreover, they did not use real social community traces to test their experiments, whereas this thesis uses three different PSN traces to evaluate performance.

5. **PeopleRank** Another closely related work, known as Peoplerank [53], uses similar centrality and node-ranking concepts based on social behaviour for their forwarding algorithm, motivated by Google's page rank algorithm. Though the structures are similar to those in the proposed algorithm, Community based forwarding (CBF), their datasets are obtained from a social network and an online multiplayer game. Overall contact patterns have been used to detect smaller communities within the user population [64]. These communities are correlated with participants' social behaviour. Higher contact-rate, lower contact-duration nodes play a major role in efficiently forwarding data. The Louvain clustering algorithm is similar, but employ different community measures; and CBF employs different routing policies.

2.4 BUBBLE-Rap

2.4.1 BUBBLE-Rap Algorithm

One work that looks at both intra-community and inter-community message forwarding is proposed by Hui *et al.*, called BUBBLE [39]. This work is motivation behind the proposed algorithms in this thesis. The BUBBLE algorithm assumes that a node or person is a member of at least one community and has a ranking in each community to which it belongs according to the number of contacts with other members of that community. This scheme takes the advantage of the fact that the network is partitioned into communities and interaction of people inside and between communities is dependent on their popularity. By popularity, they refer to how interactive a person is within its own community (Local centrality) and also with members of entire communities (Global centrality). So BUBBLE first attempts to identify cliques of individuals, then attempts to route packets through highly connected individuals both between and within cliques. In the BUBBLE algorithm, for inter-community message forwarding, if source wants to send a message to a node located in another community, it forwards its message to only those encountered nodes that have higher global centrality than itself. Similarly in case of intra-community communication, local centrality is considered.

An illustration of BUBBLE is provided in Figure 2.1. To elaborate with an example, suppose node A in Figure 2.1 has a message for destination node B. So if node A uses BUBBLE for routing then at first it will bubble the message up toward higher global centrality, until the message reaches a member of destination group C_b. In this process once when the message arrives the destination community C_b at node E, Bubble Rap shifts to the second phase of its forwarding scheme, which uses members of destination group as the carrier. This forwarding strategy continues to bubble the message up, this time using higher local popularity of the members until the destination is reached.

The algorithm for BUBBLE at one time step t is shown in Algorithm (1). The input parameters are the carrier node, destination node and the list of encountered node (encountered by the carrier node in time epoch t) having a size of 'n'. Shortcomings of BUBBLE were that the algorithm imposed a disproportionate burden on highly connected nodes and nodes with higher social ranking do not pass messages to nodes of lower ranking, even if they interact more with the destination nodes.

The authors of BUBBLE-Rap [39] has enhanced this work in their extended work [40]. In the extended work they have kept BUBBLE algorithm same but compared it with one more established algorithm called Simbet [18], analyzed how performance of BUBBLE changed when it is allowed have to multiple-copies of same messages and provided more details on how the Haggle Emulator is used for simulating the experiments. In their extended work, no changes has been brought to the original BUBBLE algorithm. As this thesis is enhancing BUBBLE's routing capabilities thus throughout this thesis the original work of BUBBLE-Rap [39] is used as a motivation and basis for enhancement. Inclusion of multi-copy approach and comparison with Simbet is beyond the scope of current state of work and can be considered as a potential future work.

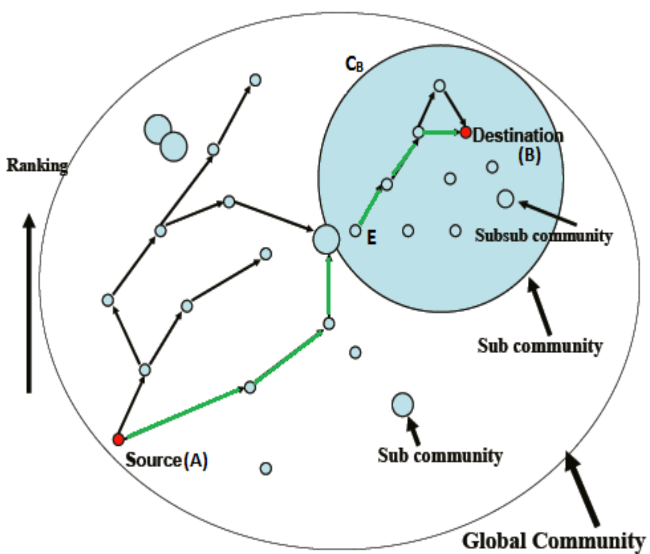


Figure 2.1: Illustration Of BUBBLE [39]

2.4.2 Scope for Improvement

As many other researchers have noted, the human contact patterns have a non-uniform structure when aggregated over time. The contact probability network formed by summing time in contact between nodes in the graph tends to form networks with small world properties [14, 36]. Many researchers have attempted

Algorithm 1 BUBBLE-Rap (Node carrier, Node [ ] en, Node dest)

for i=1 to n **do**
    if $Group(carrier) == Group(destination)$ **then** // *Intra-community forwarding*
        if $((Group(en[i]) == Group(destination))\ AND$
            $((LP(en[i]) > LP(carrier)))$ **then**
            $en[i].addMessageToBuffer(message)$
        end if
    else
        if $((GroupID(en[i]) == GroupID(destination))\ OR$
            $(GP(en[i]) > GP(carrier)))$ **then** // *Inter-community forwarding*
            $en[i].addMessageToBuffer(message)$
        end if
    end if
end for

to use this structure to increase routing efficiency in PSNs [35, 39, 73] while keeping in mind that the representation is aggregate, and the instantaneous contact pattern is dynamic and stochastic.

In particular, the tendency for small world networks to form highly connected clusters, characterized by short paths connecting every pair of nodes, through privileged highly connected nodes has been investigated as a promising means of improving routing [39]. The problem of determining routing heuristics can then be reduced to two separate performance problems: determining the most appropriate cluster structure for the graph, and determining which nodes should be chosen for inter-cluster communications. BUBBLE achieved this by directly exploiting high-betweeenness individuals in the graph, creating an algorithm which routed the packet to evermore popular individuals until the target community was reached. While this algorithm has shown significant promise, it has a highly asymmetric load distribution, reflecting the power law distributions that typify betweenness distributions in PSNs. BUBBLE also has the tendency to ignore lower-betweenness individuals who have small degree but strong inter-cluster ties (i.e. bridging nodes). While these nodes typically have lower betweenness values than social nodes with broader inter-cluster ties, packets will be preferentially passed to higher-betweenness nodes, disregarding the bridging capability.

This inclination to route through specific nodes is particularly problematic in resource constrained networks. High centrality nodes will bear a disproportionate energy cost, potentially causing these nodes to fail completely. For systems that are characterized by buffer memory restrictions, this congestion could also lead to lower delivery ratios as packet drops due to buffer overflow increase.

The primary intuition behind this thesis relates to exploiting intermediate paths within the small world network to avoid the high-betweenness paths preferred by BUBBLE. These opportunistic paths tend to form between nodes of moderate betweenness, which bridge only one or two communities. Routing algorithms can exploit bridging nodes by utilizing paths providing fewer hops to the destination community, and disperse packets more broadly across the network, as all bridging nodes, so that not just high betweenness nodes are utilized.

As a corollary, it is necessary to establish *bridging communities* as well as *bridging nodes*, to ensure that complete paths can be formed. For example, consider routing a packet from community A to community C. A does not contain any nodes that have strong connections to C, therefore its community-connection to C is low. However, if community B has nodes with strong connections to both A and C, then it could serve as an intermediate community (given that intra-community routing tends to be simpler and more efficient than inter-community routing). By keeping track of individual nodes' inter-community connections, as well as each community's inter-community connections, intermediate betweenness can be exploited further to provide fairer and more reliable PSN routing in resource-constrained networks.

BUBBLE can be further improved by using the robust community detection techniques rather than K-CLIQUE. One such precise approach to community detection is the Louvain algorithm and that is why it is used by the proposed algorithms. The K-CLIQUE algorithm, used by BUBBLE, requires that the number of communities must be specified prior to forming the communities. This is a considerable obstacle for a contact network without metadata (such as labels) or an unstructured dataset. Moreover, the locations of the centroids in K-CLIQUE are significantly affected by outliers [85]. Nodes with abnormally low contact probability with others should not be included as members of a well-constructed community; this is generally not the case when employing K-CLIQUE. It is better to consider these kinds of nodes as isolated communities of one or a very small number of outlying nodes, so that more efficient routing decisions can be made.

Overall, BUBBLE can be easily extended to give more accurate result using the enhanced technique developed in this thesis. The proposed algorithms are similar to BUBBLE, but employs different community measures and routing policies. In this thesis, the initial version of the enhancement is named Community-Based-Forwarding (CBF), which uses pairwise group betweenness and an individual node's contribution during inter-community routing. CBF also uses features of the community structure more elaborately. Details of the algorithm and results of message transmission experiments are given in Chapter 4. Further enhancements made to reduce latency are proposed in Hybrid Community-Based-Forwarding (HCBF), the design and evaluation of HCBF are provided in Chapter 5.

2.5 Summary

Routing in DTNs can be categorized in Uninformed and Informed routing. Among all DTN routing schemes, social based routing is a fairly new field to be explored and this scheme is a natural fit for low capacity PSNs. In PSNs, higher contact-rate, lower contact-duration nodes play a major role in efficiently forwarding data. From the related work of community based routing, as it appears that success has been achieved in exploiting social connectedness between nodes, it is important to have clear understanding of techniques developed in detecting communities and measuring the level of connectedness. As community detection is a multidisciplinary field, many approaches exist. The Louvain Algorithm has the flexibility to be dynamic and accurately detects communities.

The bodies of research discussed in the previous sections have a lot in common with the proposed algorithm, but there exist some major differences. During making forwarding decisions, unlike most of the related work the final version of the proposed routing algorithm, HCBF has encompassed a more accurate community detection strategy, it considers both pairwise community interaction as well as individual node's interaction with communities during making forwarding decisions and it takes diversity existing in nodes' interaction pattern with its group members into account. To be cost efficient nodes in the proposed algorithms are not required to maintain any routing tables and through out routing process only a single copy of message is kept in the network.

Chapter 3

Community Detection, Simulator and Experimental Design

There is no doubt that utilizing the social structure of humans for routing messages is a very promising research area [13, 18, 86, 87]. Knowledge of the social structure can enable devices in PSNs to be the bridge between the disconnectedness of real network. As seen from previous chapters, it does become clear that accurate community formation is a prerequisite to employ a community-based routing technique. Prior to social based routing, it is necessary to detect communities in datasets used in simulation for analyzing performance of the proposed forwarding algorithms. This chapter is dedicated to the setup needed prior to the experiments conducted on message delivery and packet forwarding.

3.1 Community Detection in Datasets

3.1.1 Dataset Overview

To compare the performance of BUBBLE, CBF and HCBF, three different datasets were used: Flunet [34], St. Andrews (Sassy) [5] and (SHED1) [33]. The first dataset, Flunet [34], contains contacts information for thirty six participants. The data was collected over a period of three months, from November, 2009 to February, 2010. The participants were graduate students from seven different labs of computer science students at the University of Saskatchewan, as well as staff and undergraduate students associated with those labs. The participants were requested to carry a wireless sensor mote (MicaZ), within that measurement period. Approximately 70,000 contact records were collected.

The second data set is St. Andrews (Sassy) [5]. It contains contact information of twenty two undergraduate students, three postgraduate students and two staff members of the University of St. Andrews. Three base stations were deployed across two Computer Science buildings in this institution. The measurement period was seventy nine days (similar to Flunet) and had 113,000 contacts collected.

The last data set - the Saskatchewan Human Ethology Dataset 1 (SHED1) [33] - was collected over a period of five weeks in 2011 with thirty nine participants and includes accelerometer, GPS, Bluetooth contacts, as well as WiFi contacts and battery state information. The participants were primarily CS graduate students from several laboratories, as well as technical and administrative staff. For this thesis, Bluetooth contact records between participants have been used to build the dynamic contact graph. Overall, the dataset accumulated over 45 million records, among which 22721 distinct contacts between participants were collected. A dataset summary is given in Table 3.1.[1]

Table 3.1: Dataset Overview

	Dataset		
	Flunet [34]	**Sassy [5]**	**SHED1[33]**
Participants	36	25	39
Period(days)	92	79	35
Setting	Grads & Undergrads	Mostly Undergrads	Mostly Grads
Communities	5	5	6
Device	MicaZ mote	T-Mote	Android phone
Local Encounters	72.73%	78.9%	62.1%
Global Encounters	27.27%	21.1%	37.9%

The data sets ranged from extremely clustered to more isolated individuals, as described above. The main difference between datasets is that SHED1 contains participants who primarily interacted within their own community and had fewer interactions with other communities, likely because graduate students tend to primarily reside in their labs. Flunet had a combination of both undergrad and grads, so formation of communities is less prominent than in SHED1. Sassy had mostly undergrad students where every student had their own class schedule and separate sets of friends, presumably many whom were nonparticipants. This is shown in detail for each node in Figure 3.1. On average, in case of Flunet 72.73%, Sassy 78.9% and SHED1 62.1% of the contacts were within communities. Significant differences in community contact patterns have been observed in university communities for which datasets are available for analysis [5, 33, 34].

3.1.2 Community Determination

As previously mentioned, detecting stable communities is a vital part and pre-requisite of any social based routing. For this thesis Louvain algorithm has been used and the accuracy of community detection technique of this algorithm has been varied with the social network analysis tool called Gephi.[2] Gephi uses 'Resolution Parameter' proposed by Lambiotte *et. al* [49] to verify the precision of network decomposition into communities by an algorithm. During verification with Gephi, the Resolution parameter has been kept to maximum

[1]The reason for not including the datasets used in the original Bubble-rap [39] work ain this thesis is because they were either too short or lacked details needed for community detection.
[2]https://gephi.org. Date of last access: September 16, 2013

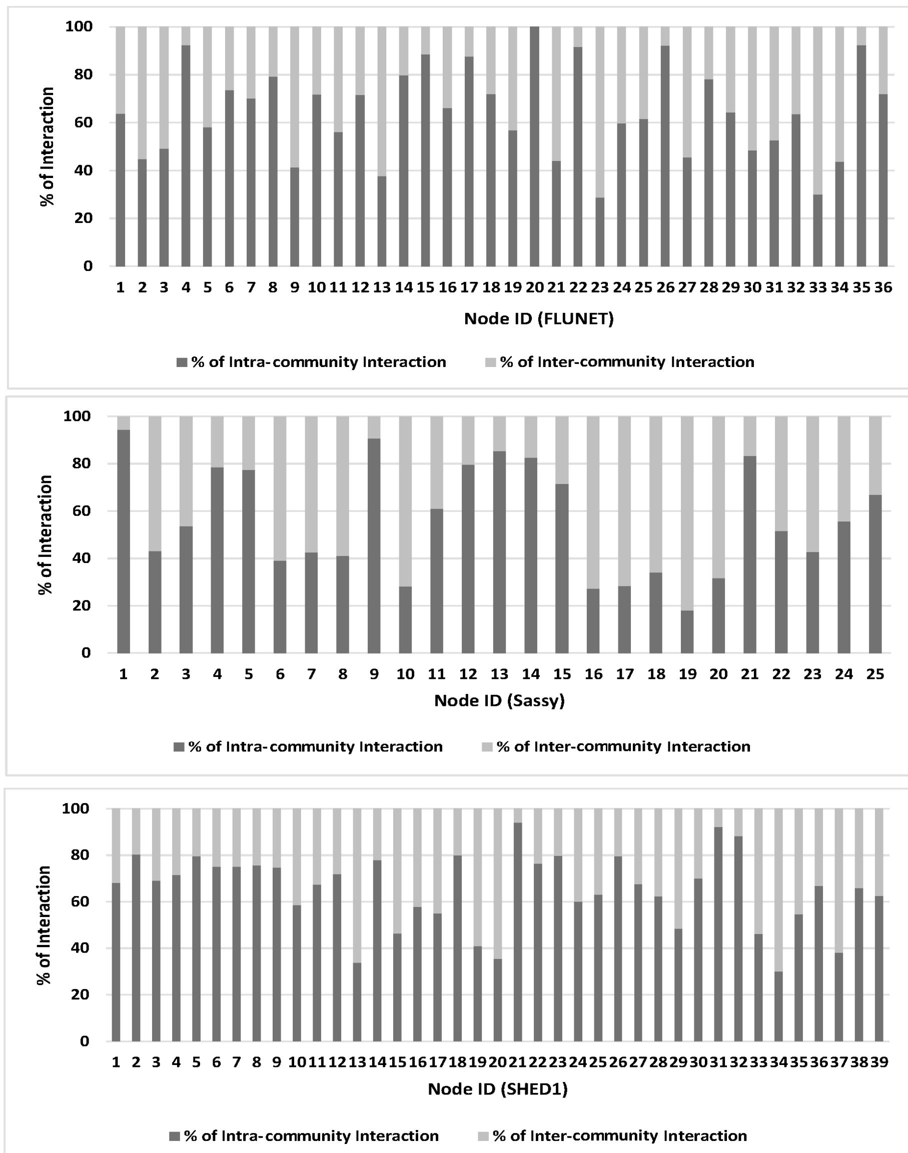


Figure 3.1: Distribution Of Interaction Between Nodes

and several trials were performed for each time frames. Result obtain were consistent indicating that communities for were stableand accurate. Moreover, for the Flunet dataset, self reported community data was available and results obtained from Louvain Algorithm were cross referred with those. It was observed when routing of message involved larger clusters, results were very similar. Later post-analysis and comparison shows that slight difference in results were due to self-reported single member communities, who had shown more collaborations during data collection.

Moreover, the selection of the community formation period is a challenge. It is necessary to give time to stabilize node interaction pattern. So that stable communities can be detected from the aggregated social graph. At the same time, sufficient numbers of contacts must be included for the clustering to be an effective predictor of future contact. A period that is too short (for example: one hour) is inappropriate for DTNs. This is because it is unlikely that all the mobile nodes will communicate with the sink node to be updated about the current community structure within that time. Likewise, forming communities based on a long period (for example: one month) of contact patterns is unfeasible, especially when the total measurement period is comparatively short (i.e. up to three months). Overall, in an attempt to find communities, it is essential to break the dataset in such a way that most stable communications are detected for formation of clusters. As mentioned previously, for this thesis, communities have been detected using Louvain Algorithm. The work of finding embedded time frames for respective datasets so that stable communities are detected has been completed in collaborative work [68], which is a partial contribution of this thesis.[3] Quantitative analysis regarding time frame of community detection shows that, when the trace data from each of the three datasets (i.e. Flunet, Sassy and SHED1) is broken down into subsets of seven days, and those subsets are used in Louvain algorithm, only then the most stable communities appear. For the datasets of Flunet, Sassy and SHED1, seven days is the most suitable cluster formation time as it is neither too long or too short, includes weekdays and the weekend, thus intuitively corresponds to a common unit of time. Based on this conclusion [68], in this thesis, seven days is chosen as the time frame for finding most stable communities; thus each week's routing decisions are done on the communities found in the previous week.

When clustering is done with seven day period then the number of times that nodes change groups through out the period of data collection is shown in Figure 3.2. SHED1 has the shortest duration, but shows substantially higher stability than the other two datasets. Flunet and Sassy are fairly similar in duration, but over 70% of the nodes in Flunet change groups two or fewer times in twelve weeks, whereas 20% of the Sassy nodes change groups five times in eleven weeks.

[3] Md. Shaiful Alam Chowdhury contributed the implementation and analysis of the communication and epoch calculation using the Louvain Algorithm.

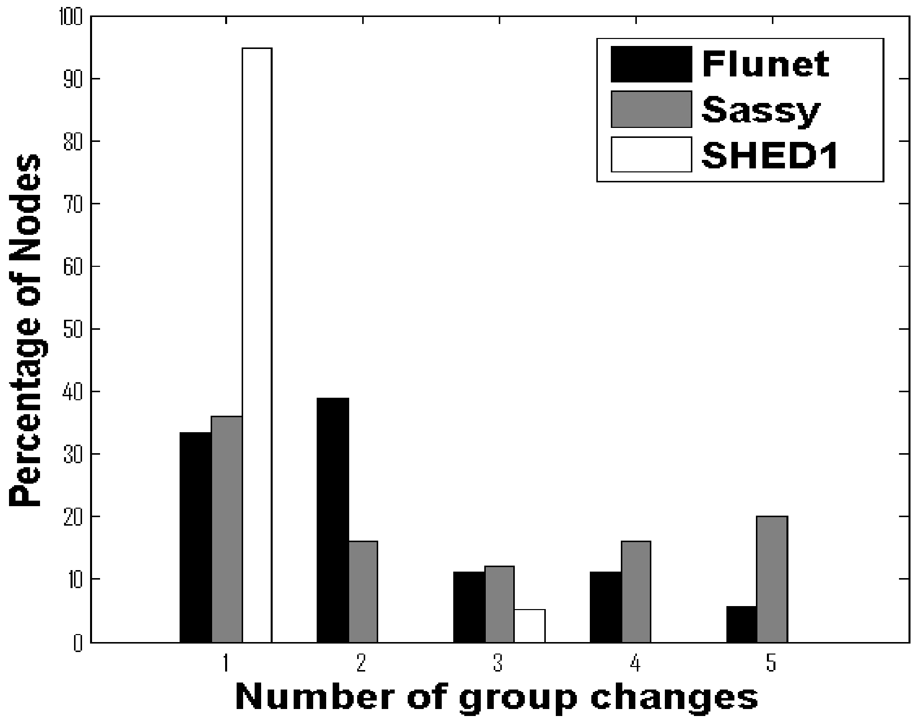


Figure 3.2: Nodes' Group Changes

3.2 Routing Simulation Assumptions

Due to time constraints and information limitation, during simulation it was not possible to integrate all the information needed in real life for a DTN routing. The proposed algorithms have been designed subject to the following set of limiting assumptions relating to the application area, properties of networks, and behavior of the system. As noted below, these assumptions are consistent with those made in the literature.

1. **Resource Limited:** Devices employing PSNs are likely resource limited in memory and computational power.

2. **Delay Tolerant:** Packets possess a delay tolerant property encoded in TTL. That is, as long as the packet is successfully delivered prior to TTL expiry, packet delivery was successful. Hui and Crowcroft [38] and de Olivera and de Albuquerque [61] share this definition of successful delivery.

3. **Human Mobility:** The system represents a PSN on human subjects that have non-degenerate mobility patterns [50, 61].

4. **Socially Connected:** Nodes are sufficiently socially connected to form quasi-stable cliques [73], which follows almost directly from the previous assumption.

5. **Fixed Message Size:** The contact duration has no effect on the ability to exchange metadata concerning the content of each others buffer and the messages themselves [14], simplifying the final analysis.

3.3 Simulator

All the routing experiments were done in a custom simulator written in Java. Given the size of the network, performance of the simulation was not an issue. During the experiments, different input combinations for CBF and for HCBF were examined in a full block structure and the resulting delivery ratio, packet delay, total number of transmissions and number of messages dropped per node were recorded. Unless otherwise specified, for the experiments done in Chapter 4 for evaluating performance of CBF, 100K, 80K and 48K messages were generated for the Flunet, Sassy, and SHED1 datasets respectively. The values assigned to each parameter are described in Table 3.2. Once CBF's performance is evaluated over a large scale and results have been analysed (in Chapter 4), further enhancement on CBF, that is HCBF, is tested over an equal range for all three datasets. Thus for experiments in Chapter 5 on HCBF, 50K, 40K and 48K messages were generated for the Flunet, Sassy, and SHED1 datasets respectively. The values assigned to each parameter are described in Table 3.3. For both CBF and HCBF, twenty experimental runs were performed for respective input-parameters combinations to quantify the stability of the results and standard deviation was calculated to observe the variation in results from the experimental runs. Each experiment was run for all three datasets. All simulations were run on a machine with a 2.3 GHz dual core i5 processor and 4GB RAM.

Table 3.2: Ranges of Input Used in Different Types of Experiments in CBF

Input	Limited Resource		
	Flunet	Sassy	SHED1
TTL(days)	0-15	0-15	0-7
Buffer Capacity	10% of total msg	10% of total msg	10% of total msg
Messages	0-80000	0-100000	0-480000
	Unlimited Resource		
	Flunet	Sassy	SHED1
TTL(days)	0-180	0-180	0-90
Buffer Capacity	10-200000	10-200000	2-100000
Messages	10-80000	10-100000	5-48000

For both the algorithms, the first set of experiments addresses the unlimited resource scenario where one parameter was varied and the others set to inexhaustible values. This is done to observe how the algorithms perform without resource constraints. The second set of experiments portray the the realistic scenario of scarcity of resources. The limited resource case is when any two experimental parameters were constrained and the third one varied within a limited range. Also in all datasets for both CBF and HCBF, in all experiments with a fixed size, 10% of the number of messages generated was used as an acceptable threshold for buffer capacity by trial and error. Increasing the buffer size beyond 10% did not provide a proportionate increment in delivery ratio for either BUBBLE or CBF. Similarly, TTL values were set to one sixth (i.e. 15 days) of the simulation period for Flunet and Sassy and one fourth (i.e. 7 days) of the simulation period for SHED1 to provide meaningful results. The number of messages generated was varied to match the duration and contact density of the datasets.

Table 3.3: Ranges of Input Used in Different Types of Experiments in HCBF

Input	Limited Resource		
	Flunet	Sassy	SHED1
TTL(days)	0-15	0-15	0-7
Buffer Capacity	10% of total msg	10% of total msg	10% of total msg
Messages	0-40000	0-50000	0-480000
	Unlimited Resource		
	Flunet	Sassy	SHED1
TTL(days)	0-180	0-180	0-90
Buffer Capacity	40-40000	50-50000	48-48000
Messages	0-40000	0-50000	0-480000

A block diagram of the components in the simulator is shown in Figure 3.3 and the activity diagram illustrating its working scheme is shown in Figure 3.4. The three major components of the simulator are the DataAnalyzer, MessageGenerator and Forwarder. The simulator requires four inputs. They are the following: a dataset contact pattern, maximum node buffer capacity, maximum number of messages to be generated and TTL. At the start of an experiment, original trace files are divided into discrete sequential time periods and fed into the DataAnalyzer as inputs. Using the Louvain Algorithm [7], the DataAnalyzer created communities. Communities potentially change their members every time period. The first time period's data is used as a training session; packets are forwarded starting in the second time period.

From the start of the second time period, the MessageGenerator starts to produce messages. Every second there may or may not be a message produced, depending on the probability ratio given in Equation 3.1. The number of messages generated was varied to match the duration and contact density of the datasets. In the simulator, message generation is distributed such that the last message is produced when the simulation nears

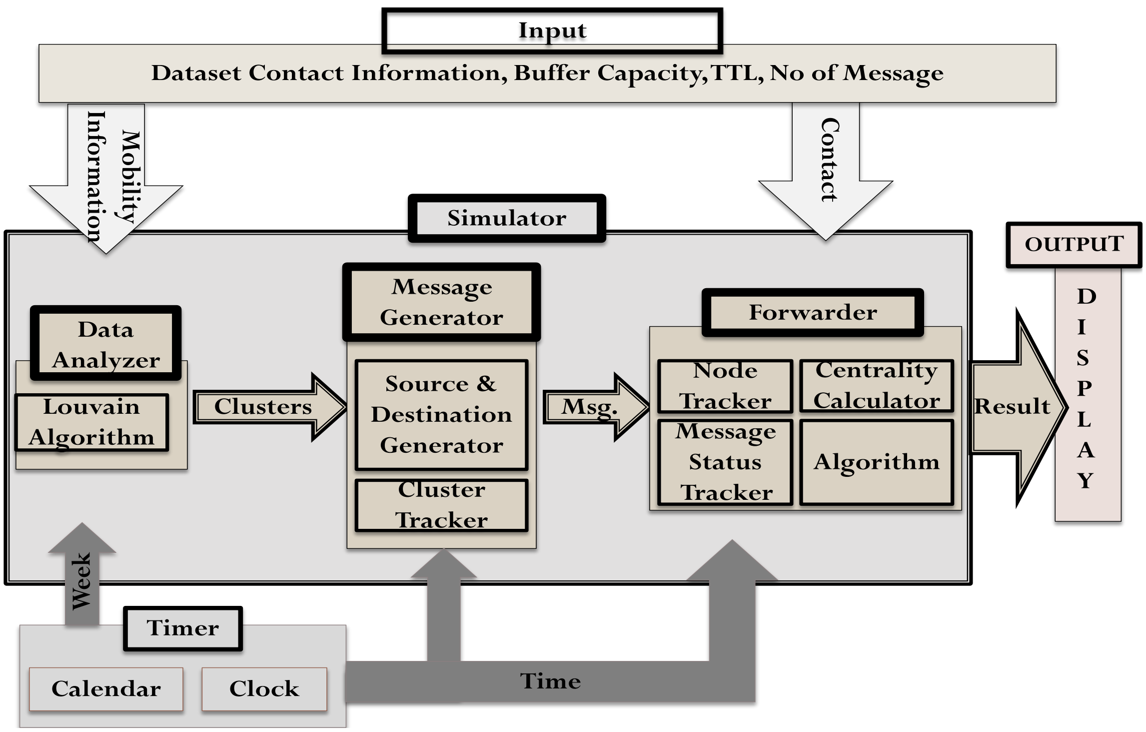


Figure 3.3: Simulator Block Diagram

completion. Message generation over the span of the experimental time frame increases the robustness of the simulator by sampling many different dynamic graphs but depresses the delivery ratio for all algorithms due to messages created with a very low probability of delivery near the end of the simulation. The impact can be quantified by examining the unlimited/ER case, which sets the bound for maximum possible deliveries.

Probability of message generation:

$$P(Packets/s) = \frac{Desired \quad Number \quad of \quad messages}{Length \quad of \quad data \quad collection} \tag{3.1}$$

For experiments when CBF is compared to BUBBLE, source and destination are chosen randomly from different groups to examine the impact on inter-group communication as BUBBLE and CBF have identical intra-community routing policies. Data structures have been used to keep track of messages so that the pattern of message generation is unique and not repetitive between time frames and trials. The same set of messages were used to evaluate the performance between the algorithms. In case of experiments when HCBF is compared to BUBBLE, source and destination of a message can belong to any community as HCBF has its own set of scheme, developed for both inter and intra community routing, which are different from BUBBLE's.

Once a message is generated, it is pushed to the Forwarder. The main task of the Forwarder is to relay a message based on the specified algorithm. Every encounter leads to an exchange of labels and then using the algorithms, forwarding decisions are made. Only the cost of transferring a message is considered, but not the cost of exchanging labels required for this transfer because of the marginal increase in overhead between BUBBLE and proposed algorithms. Every encounter leads to an exchange of labels and then using the algorithms, forwarding decisions are made. For most systems, the label will be less than 8 bits. Using source coding, the transmission cost of sending control labels can be further merged into the cost of data transmission. This details are beyond scope of this work. The Forwarder uses the traces of particular data sets to provide the encounters between nodes, which guides the mobility of nodes for all the algorithms used in this thesis.

All existing messages are relayed to the next suitable carrier. The Forwarder module is comprised of the NodeTracker, MessageStatusTracker, CentralityCalculator and Algorithm modules, which operate as follows:

1. **NodeTracker:** Nodes contain the following attributes: node id, weekly group id, weekly Local Popularity (LP) (Equation 4.1), weekly Global Popularity (GP) (Equation 4.2), weekly Group Betweenness Count(GBC) (Equation 4.3), weekly Nodal Contribution Factor (NCF) (Equation 4.4), weekly Unique Interactions (UI) (Equation 5.1) and buffer size. This are computed from entire week's contact. Based on the contact pattern, NodeTracker decides the contact model for the nodes. It also manages all the nodes, synchronizing dynamic attribute updates. Each node has a maximum buffer space or queue size referred as Buffer capacity. A message arriving at a full queue causes the oldest message to be dropped. After the transmission of a message, a node deletes its copy.

2. **CentralityCalculator:** Centrality measures were needed to make routing decisions. For quantifying the current week's forwarding policies, CentralityCalculator measures LP, GP, GBC, NCF and UI of the nodes, based on their contact patterns in the previous week.

3. **MessageStatusTracker:** Messages have a specified TTL, source ID and destination ID. With every clock tick, all TTL counters are updated. The message is deleted if the TTL reaches 0 or it reaches the destination.

During end of the simulation period, the MessageGenerator produces its last message and triggers the Forwarder to stop and copy all the information stored in the Forwarder to the Output module. The stored information has two types. First type is from the perspective of messages and the second types is from the perspective of nodes. For each message, its ID, source, destination, list of carriers, time of generation, number of inter hop and intra-hop and total delay is recorded. For each node, its buffer capacity, total number of packet drop, total number of packet forwarded, total number of encounter, its centrality values, list of packets it carried are logged. Information regarding how often individual centrality values are used is also recorded. Based on the information relayed, the Output module post processes and calculates the delivery ratio, transmission cost, packet drop, delivery latency and displays these results to users by writing them in output log files.

The trace-driven simulator supports only the network layer, since objective of this thesis is to evaluate routing protocols. Details about the underlying layers like error correction, error detection, collisions, and re-transmissions are not implemented, so messages are only discarded due to buffer overflow or TTL expiry.

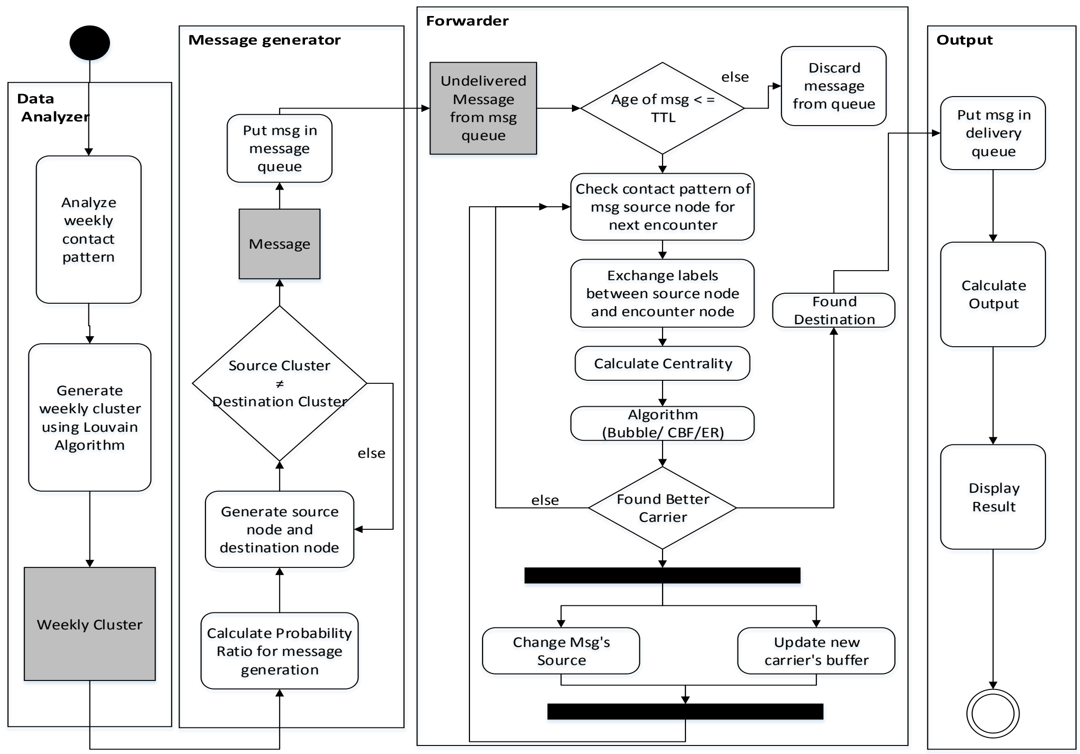


Figure 3.4: Simulator Activity Diagram

3.4 Mobility Model

When evaluating the performance of PSNs, the use of simulated packet transfer, allows comparison of multiple algorithms against exactly the same contact patterns. The dynamic contact patterns underlying the simulations are typically generated in three ways: 1) directly from sensed contact patterns from in-house [34] or publicly available [36] datasets; 2) inferred from higher-level mobility data such as class [78] or bus [4] schedules; or 3) from synthetic contacts generated directly from theoretically grounded, simulated mobility patterns [69].

Direct sensing of contact patterns is the most accurate representation of the dynamic contact network, but empirical patterns can be difficult and expensive to obtain, and are therefore usually limited in number of nodes (dozens) and experimental duration (weeks or months). Schedules can be useful proxies, but they miss finer grained interactions within the bus or classroom, and cannot account for phenomenon like absenteeism. Simulated systems are easier to deploy, and can be scaled as far as available computing infrastructure allows, but currently only offer stylized representations of human mobility and contact, which may create problems for cluster-based routing systems. Su *et al.* [80] and Daly *et al.* [18] have tested the feasibility of using contact patterns as the basis for mobility model and concluded that user mobility can be used to form a network. In this simulator, the same concept has been used for modeling mobility of nodes.

3.5 Comparator Algorithms

An oracle algorithm is a routing scheme that has the complete knowledge of the future, thus can deliver the packet along the most desirable path to the destination, depending on the desired performance metric. There are many version of oracle algorithm as mentioned by Jain *et al.* [42]. In this thesis, oracle algorithms are used to compare performance of proposed algorithms against BUBBLE, as an implementation of a context-aware forwarding approach. Two comparator algorithms are used in this thesis: Fastest Oracle and the Minimum-cost Oracle. Additionally, the metrics of interest in determining how proposed algorithms and BUBBLE perform in comparison to the oracle algorithms which are optimal with respect are message transmission cost (number of forwarding operations), delivery success (delivery of a packet prior to TTL expiry) and delivery latency. Details on the oracles used in this thesis are as follows:

1. **The Fastest Oracle:** The *Fastest Oracle* is an oracle algorithm that routes packet on a path with shortest delay. Here the time delay is referred as the time period elapsed from generation of message to delivery of message. Unfortunately, finding the fastest, shortest or guaranteed route through a dynamic graph requires exact knowledge of the graph's future structure; while this is functionally impossible, it can be post-computed using oracle algorithms or functionally approximated statistically using historical data [50].

   In other words, in the *Fastest Oracle*, the fastest path is found by back tracing first successful message delivered by ER with unlimited buffer. If any node that has been given a copy of the message contacts the destination, the transfer is complete and this time is recorded. This aggressive approach is not a practical implementation alternative.

   In this thesis during simulation of the fastest oracle, messages had no TTL restriction and nodes were assigned unlimited buffer capacity. Through out the simulation period, such messages are generated and flooding is performed in such environment. After simulation period is over, information regarding delivered messages are post processed and output values for those copy of messages that had arrived the destination earliest are record, analyzed and displayed to the user as output of Fastest oracle.

2. **The Minimum-cost Oracle:** *Minimum Cost Oracle* is an oracle algorithm that transmits messages on a route that uses the fewest hops. By flooding approach, messages with shortest route are only considered to have met this criteria. If time-to-live is made sufficiently large, the fewest hops will be one for most of the messages in our datasets because the aggregated graph of contacts is almost completely connected (i.e. almost every node meets every other node before the end of the measurement period). For moderate TTLs, the shortest path transmission containing the fewest hops is saved at the destination.

During the simulation, for the Minimum-cost oracle messages are flooded and copies of messages that arrives their respective destinations in fewest hops are selected by post-processing the output log and results regarding transmission cost, delay, etc are presented to the user.

The performance of CBF is compared with two implemented algorithms, BUBBLE [39] and Epidemic Routing [82] as well as Oracles that optimally selects routing according to one's performance measure. ER is commonly considered as the performance benchmark algorithm in most previous DTN work ([39], [50],[53],[55],[61]). Due to a lack of knowledge about the future, the flooding approach and post-processing of logs were performed to calculate the performance metrics of the Oracle algorithms. Details from simulation show that in real life, even if the buffer capacity was available, the amount of energy expended by having both nodes exchange the difference of their entire buffers on every contact would be prohibitive, and contacts of short duration in reality will not be sufficient for the data transfer, contradicting previous assumptions.

3.6 Summary

This chapter describes three different datasets used for testing performance of proposed routing algorithms. It also shows the impact of datasets' nature on community detection and on determination of community formation period. Furthermore, it elaborates on different modules of the custom simulator designed for the performance evaluation of CBF and HCBF, emphasizes on how the simulator models certain PSN routing algorithms and talks about to the compactor algorithms which are used as benchmarks. Also, this chapter briefs on the mobility model used by the simulator and focuses on the assumptions made during routing decisions.

Chapter 4

Community Based Forwarding: An Inter-Cluster Based Forwarding Algorithm

In this chapter, an enhanced routing strategy called Community Based Forwarding (CBF) is introduced. This routing approach explicitly targets the long, 'rare' and more challenging task of inter-community message forwarding. The chapter begins with the description of the motivation behind this work followed by description on design and implemented of CBF and also evaluates CBF's performance through simulation.

4.1 Motivation

From chapter 2, it is seen that most of the PSN routing algorithms [2, 12, 19, 46, 53] use social-based routing schemes. Although these works have based their schemes on the essence of community, but other than from BUBBLE, most of the works use the concepts of modularity and betweenness only for finding the communities and not for making routing decisions. Probability of contact is used thereafter for making forwarding decisions. This is because, in general, during data analysis it is often seen that nodes within a community (intra-community) are likely to see each other more often than nodes in different communities (inter-community) [87]. From the respective datasets' traces, in the case of Flunet [34] 45.46% more interactions happen within the community than outside the community, in the case of Sassy [5] there are 57.8% more and in the case of SHED1 [33] 24.2% more. This is previously shown in Figure 3.1 and in the last 2 rows of Table 3.1. Inter-community interaction is more likely to be the focus during development of routing strategies due to the higher number of contacts. However, fewer contacts between members of different groups also means that forwarding packets opportunistically outside one's own community is more difficult than within a community, in a social based routing. Thus, it is a more challenging task.

Moreover, in a realistic extreme environment, a node is more likely to occasionally contact unfamiliar nodes, which is a typical scenario of DTN accordingly as well so inter-community routing becomes more important. Not only extreme environment but in regular life, inter-community interaction has importance too. With the passage of time, nodes may change their community, so a local encounter may become global in the future or vice-verse; having obsolete local community information does not help a social-based routing algorithm perform better. The proposed algorithm, CBF, targets the longer inter-community paths. This intuition to optimize inter-community message forwarding to improve routing performance is inspired from the work done by Sastry *et al.* [73]. The authors empirically demonstrated that PSN networks connectivity depends crucially on the less frequent contacts. In all our datasets, when global encounters are compared with the local encounters, global encounters can be treated as the somewhat *'rare'* contacts. In order to handle such situations, the decision making process for forwarding messages between communities and within a community should be looked from a different perspective. As not many researchers have covered this precise feature of PSNs, it remains an unexplored area. BUBBLE [39] emphasizes the fact that treating the pairwise inter and intra-group contacts separately can improve forwarding decisions. Similar to BUBBLE, this thesis also uses this difference in contact patterns as a building block, not only for community detection but also for forwarding decisions.

4.2 Forwarding Algorithm: Community Based Forwarding (CBF)

The notion of communities is employed to manage the routing table size and community affiliations, instead of individual connectedness to determine whether a message is to be forwarded through an encountered node or retained. CBF employs the same 'intra-community' routing policies as BUBBLE, namely *local popularity* (Eq. (4.1)); therefore it has the same performance when routing within the same community. For 'inter-community' routing, *global popularity* (Eq. (4.2) in BUBBLE) is replaced with *community betweenness count* (CBC) and *nodal contribution factor* (NCF) defined by Equations (4.3) and (4.4), respectively. CBC counts the number of interactions between 2 communities and NCF measures an individual node's interaction with other communities. In all equations, $g(x, y, k) = 1$ if an encounter between 2 nodes x and y occurs in the k^{th} measurement interval, and 0 otherwise.

Local Popularity:

$$\forall(x) \quad LP_x = \sum_{y \in C_x} \sum_{k=0}^{K} g(x, y, k) \tag{4.1}$$

Global Popularity:

$$\forall(x) \quad GP_x = \sum_{y \notin C_x} \sum_{k=0}^{K} g(x, y, k) \tag{4.2}$$

Community Betweenness Count:

$$\forall(C_x, C')_{s.t.(C_x \neq C')} \quad CBC_{C_x, C'} = \sum_{x \in C_x} \sum_{y \in C'} \sum_{k=0}^{K} g(x, y, k) \tag{4.3}$$

Nodal Contribution Factor:

$$\forall(x)\forall(C')_{s.t.(x \notin C')} \quad NCF_{x, C'} = \sum_{y \in C'} \sum_{k=0}^{K} g(x, y, k) \tag{4.4}$$

In the equations above, C denotes corresponds to the community, k denotes a particularly temporal aggregation epoch within the week and K denotes the epoch duration. In particular C_x represents the community of which node x is a member. The calculations in epoch e are used in forwarding decisions in epoch $e + 1$.

Epoch duration depends on the nature of the dataset and the amount of latency tolerated. Networks which consider social communication as a measure of forwarding are prone to changes [39]. Dynamic contacts in a social network change as nodes change their contact patterns, leading to change in edge weights in a traditional graph representation of the contacts. For measuring centrality values of nodes, the weights on edges due to the contact pattern have to stabilize over a certain time period of time. The epoch needed for the datasets is 7 days [68].

Packet forwarding is accomplished through a set of heuristics. CBF never forwards outside the community when source and destination belong to the same community. When a carrier node encounters another node that neither belongs to its own community nor to that of the destination, the proposed algorithm passes the packet to the node *only* if it has a higher CBC with the destination community. When the encountered node belongs to the same community, but not the destination community, CBF algorithm only passes the message if it possesses a higher NCF with the destination community than the carrier node. These heuristics are designed to reduce both transmission and computation costs while maintaining BUBBLE's performance as a performance bound.

Figure 4.1 shows how CBF makes its forwarding decision. In this scenario, node S has a message for node D. In step 1, node S does not encounter node D, rather it encounters node E and node I. In this step node E's community, Group P, has the higher CBC with destination group, indicating that members of Group P are more likely to meet members of the destination group. Thus message is given to node E. Now node E is the carrier of message as shown in step 2. Among its interacted nodes, E picks node M as the carrier of the message. This is because node M has a higher NCF than node E, showing that it has higher probability of encountering a member from destination group. Then in the last step, step 3, message reaches a destination group member and that node bubbles the message up along the gradient of local popularity in the hope that message will finally reach the destination node D. The formal representation of CBF is shown in Algorithm 3. The input parameters are the carrier node, destination node, the list of encountered nodes (encountered by the carrier node in time epoch t) having a size of 'n', nodes which have the maximum LP, maximum NCF and maximum CBC from the encountered list of nodes.

4.3 Behavioral Study of Communities under CBF

The overall internal structure of the networks for each dataset is examined by generating 5000 inter-community messages with randomly selected source and destination nodes in unlimited resources environment. During this experiment, the number of messages transmitted by individual nodes through out the simulation period has been recorded to understand the details of how each of algorithms performs load balancing in the different datasets.

In all three datasets, nodes undergo community changes in shorter time periods, so globally aggregated membership of the nodes are used in Figure 4.2, Figure 4.3 and Figure 4.4 to give readers an overall idea

Algorithm 2 Community-Based-Forwarding (Node carrier,Node [] en, Node enMaxLP, Node enMaxNCF, Node enMaxCBC, Node dest)

for i=1 to n **do**

// Destination encountered

  if $(en[i] == dest)$ **then**

    $en[i].addMessageToBuf(message)$; **break** ;

//Member of destination group encountered

  else if $((C(en[i]) == C(dest))$ **and** $(C(carrier) \neq C(dest)$ **then**

    $en[i].addMessageToBuf(message)$; **break** ;

  end if

end for

// Intra-community routing

if $(C(carrier) == C(dest)$ **and** $(C(enMaxLp) == C(dest))$ **and** $(LP(carrier) < LP(enMaxLP))$ **then**

    $enMaxLP.addMessageToBuf(message)$;

// Inter-community routing

else

//Carrier & encountered node in same group

  if $((C(enMaxNCF) == C(carrier))$ **then**

    if $(C(enMaxNCF) \neq C(dest))$ **and** $(NCF_{[carrier][C(dest)]} < NCF_{[enMaxNCF][C(dest)]})$ **then**

      $enMaxNCF.addMessageToBuf(message)$;

    end if

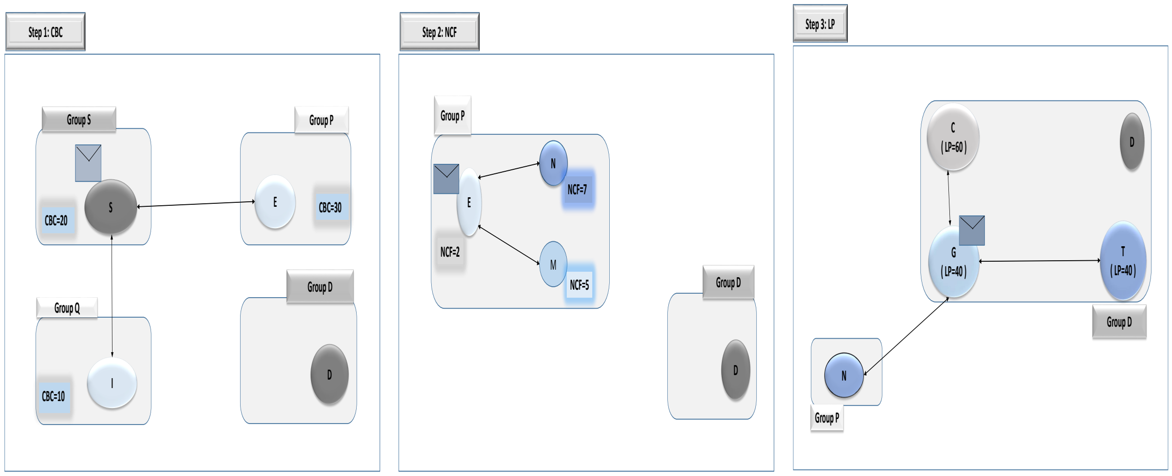
// Carrier, encountered and destination nodes in different groups

  else if $(CBC_{[C(carrier)][C(dest)]} < CBC_{[C(enMaxCBC)][C(dest)]})$ **then**

    $enMaxCBC.addMessageToBuf(message)$;

  end if

end if



(a) Use of CBC **(b)** Use of NCF **(c)** Use of LP

Figure 4.1: Illustration of CBF Routing Algorithm

on the network structure under each of the algorithm.[1] For example, weekly data for Sassy gives rise to five communities in each week, whereas aggregated data over the entire simulation period produces only two communities.

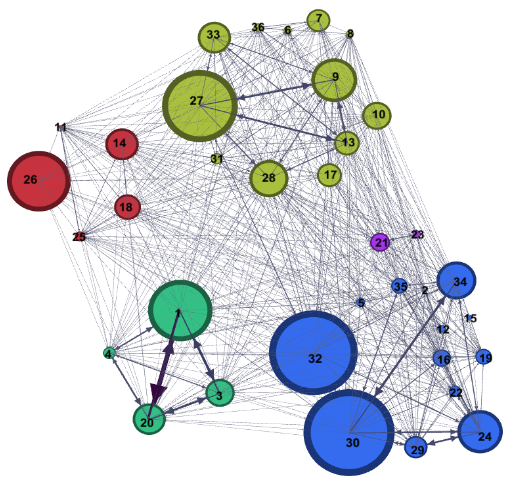
Figure 4.2, Figure 4.3 and Figure 4.4 show the underlying network structure, the contact patterns and the message forwarding behaviour. For both BUBBLE and CBF, network connectivity remains the same. Encounters are represented as edges in the graph. The thickness of an edge is proportional to the number of encounters between two nodes and node size is proportional to the total number of packets forwarded by that node.

For all three datasets, nodes using CBF for forwarding have more uniform sizes, whereas in BUBBLE, there are nodes that shoulder a disproportionate burden, as indicated by their size. In SHED1, most of the globally popular nodes belong to one community and when using BUBBLE, the most social node (for example node 16) has a very high responsibility for forwarding. On the other hand, CBF in SHED1 distributes the packets among the nodes more evenly.

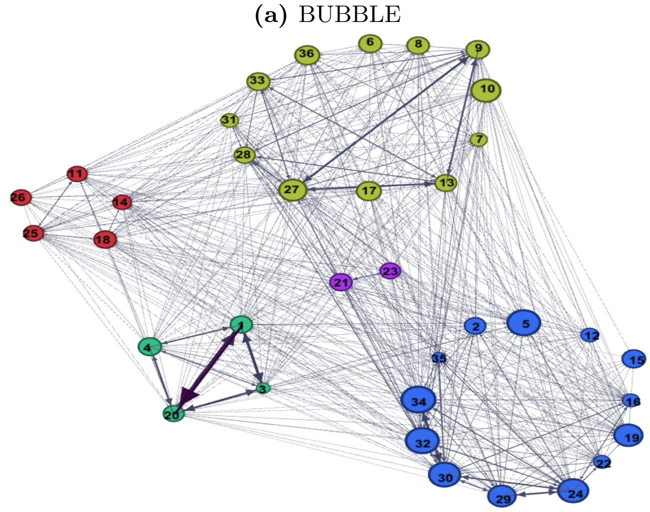
For further analysis on load balancing, a boxplot of the total messages forwarded by each node during the length of simulation is visualized in Figure 4.5. A boxplot is a statistical tool which can graphically represent degree of dispersion in data. It gives a compact view of maximum and minimum in data, median and quartiles, all in one glance. The top and the bottom of the box represent the first and the third quartile respectively whereas the line in the box represent the median. The range is denoted by the whiskers stretched out from the box.

For BUBBLE with the Flunet and SHED1 datasets, the few nodes which are social forward a higher

[1]In Figure 4.2, Figure 4.3 and Figure 4.4, for compact view and ease of representation, the communities shown were computed from the contact probability network aggregated over the entire dataset. For all experiments, weekly membership of the communities is used. Thus, the actual community membership and number of communities used during experiment will not exactly synchronize with the aggregated membership of nodes shown in the figures.

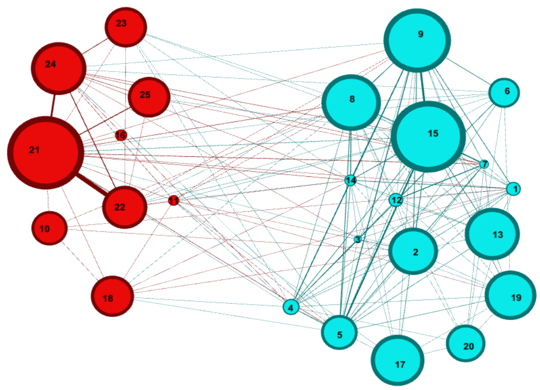


(a) BUBBLE

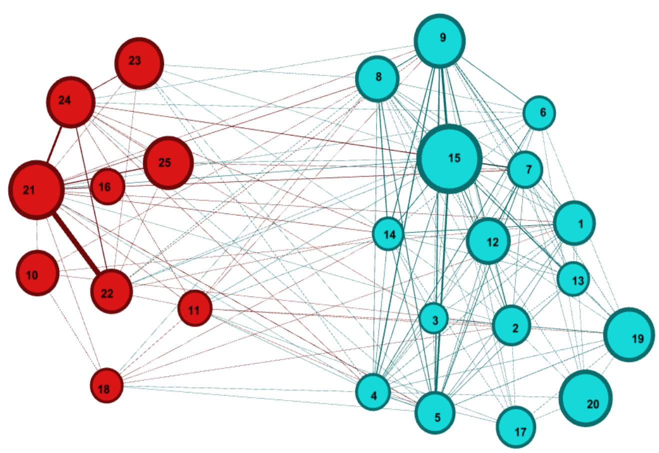


(b) CBF

Figure 4.2: Internal Network Structure: Flunet

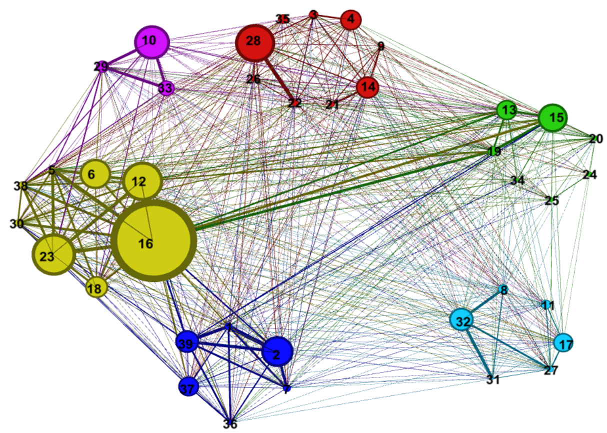


(a) BUBBLE

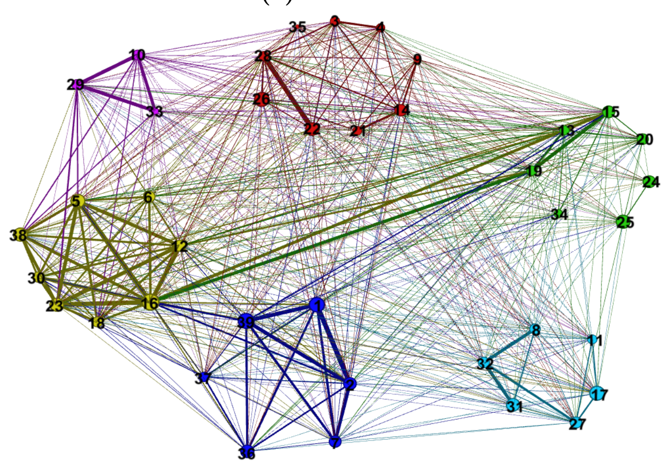


(b) CBF

Figure 4.3: Internal Network Structure: Sassy



(a) BUBBLE



(b) CBF

Figure 4.4: Internal Network Structure: SHED1

percentage of the packets and most of the nodes contribute to a lesser degree. From the relatives sizes of quartiles in boxplots of Figure 4.5, in the case of BUBBLE, nodes with number of transmissions above the median are more widely spread than nodes with number of transmissions below the median. In the same figure, CBF has a more symmetric distribution than BUBBLE. On average, in Flunet, nodes in CBF transmits 8668 fewer packets and the range is smaller by 68164 packets. In Sassy, CBF transmits 6668 fewer packets and the range is smaller by 33779 packets. Finally in SHED1, CBF transmits 6458 fewer packets than BUBBLE and the range is short by 73952 packets. Both the median transmissions per node and the range of messages transmitted help to strengthen the fact that CBF forwarding scheme performs load balancing; thus its transmission is more uniform than that of BUBBLE. The average transmissions per nodes is a factor of the efficiency of the algorithm in routing packets and does not address the dispersion in distribution of load between the nodes.

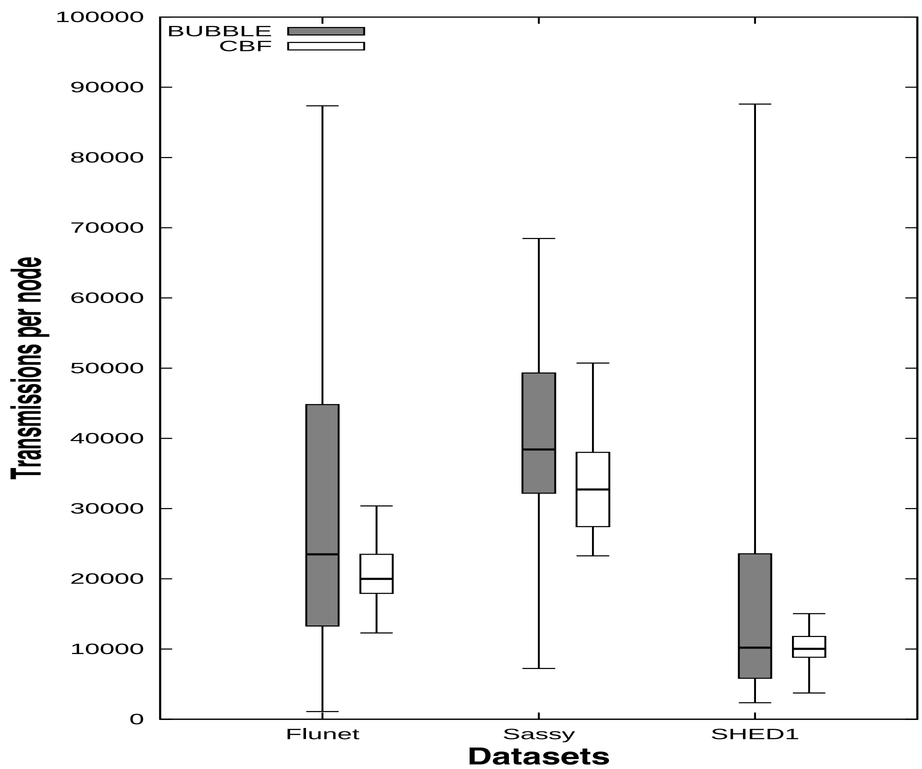


Figure 4.5: Node-Specific Message Forwarding: Unlimited Resources

4.4 Forwarding Results

The first set of experiments were executed to observe how the algorithms performed without resource constraints. For unlimited resource simulations, one parameter was varied and the others were set to inexhaustible values. The second set of experiments addressed the limited resource case where any two experimental parameters were constrained within a limited range. For this particular scenario, in both sets of experiments, source and destination of messages' are chosen randomly from different communities to examine the impact on inter-community routing as BUBBLE and CBF have identical intra-community routing policies.

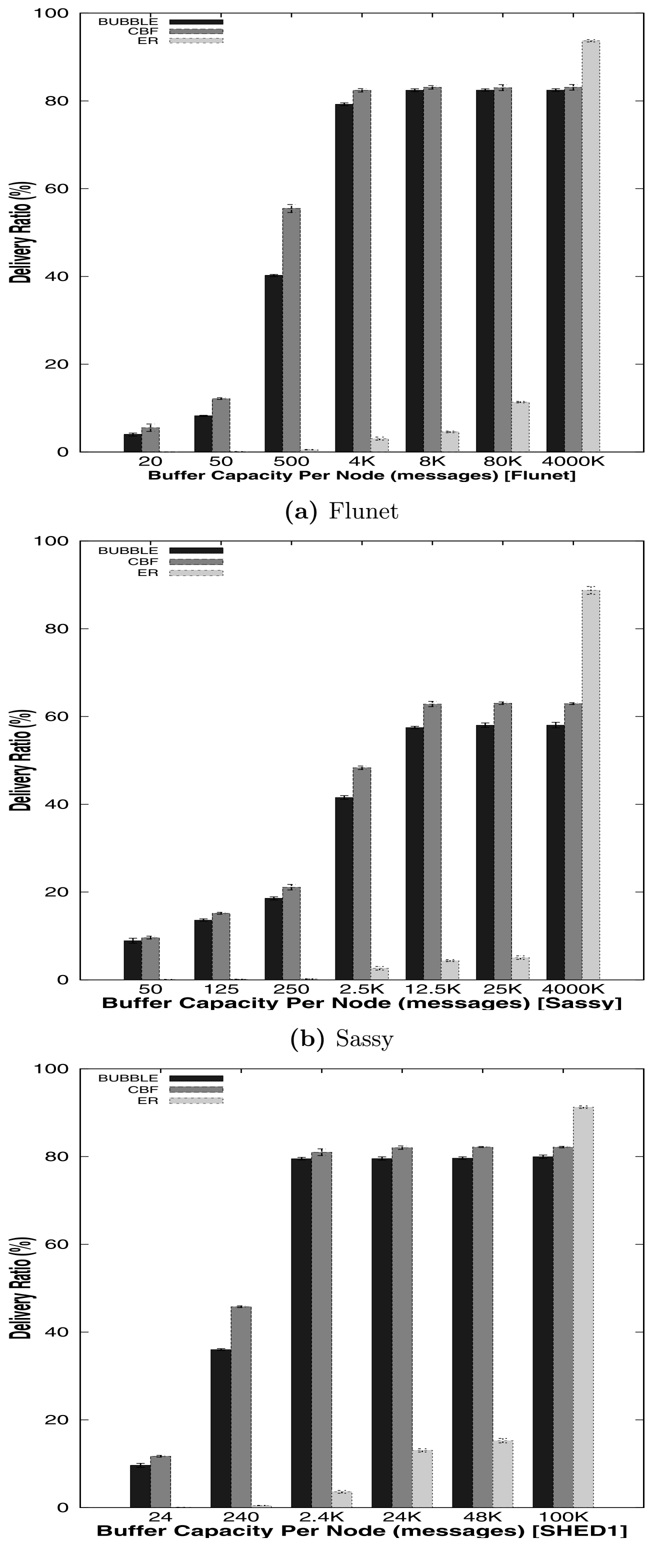
4.4.1 Case Study 1: Unlimited Resources

The first set of experiments is meant to serve as a baseline demonstrating the relative performance of algorithms under ideal unconstrained conditions and the impact of community modeling during routing. If community modeling was providing the savings, similar performance differences between BUBBLE and CBF in the limited and unlimited case should be seen. As shown in Figure 4.6, the overall differences between the algorithms' delivery ratios were not significant for unlimited buffer sizes (at 80%) for Flunet and SHED1. On average, CBF outperformed BUBBLE by 3.68% for Flunet, 3.83% for Sassy and 3.45% for SHED1. However, for some limited buffer sizes, CBF substantially outperforms BUBBLE for the Flunet and SHED1 datasets. Epidemic Routing (ER) is viable only when nearly unlimited buffer space is available. The delivery ratio of ER with unlimited buffer and TTL is the maximum number of deliverable packets and is shown only for comparison purpose. During this experiment, buffer size was varied, but TTL was unlimited and the maximum number of messages was generated as indicated in Table 3.2.

In resource-rich environments, CBF performance is close to that of BUBBLE for delivery ratio because CBF employs a conservative routing decision metric, and preferentially holds a message until a node with better CBC is found, reducing the amount of local bubbling that occurs. This decreases the total number of transmissions consistently over all buffer sizes tested. This is shown in Figure 4.7.[2] Notice that there is a consistent difference in transmissions between the two algorithms over all buffer sizes tested. In case of Flunet, CBF had 22.3% to 37.8% fewer transmissions, in case of Sassy 19% to 28% fewer transmissions and in case of SHED1 19% to 34% fewer transmissions.

Then the impact of routing decision on the buffers of the nodes is observed (i.e. how the two algorithms induce packet drops). Figure 4.8 shows both the algorithms tendency to drop packets when nodes have unlimited buffer. CBF's reduction in transmission decreases the packet drops at globally popular nodes, as seen in Figure 4.8. In general, CBF spreads traffic more evenly and therefore, there are fewer packet drops. With large buffers, neither algorithm drops any packets. Since the delivery ratio performance of BUBBLE and CBF converge as buffer size increases, it can be concluded that it is the packet drop due to congestion at popular nodes, rather than the effectiveness of the group model that dominates the performance when resources are constrained. It is worth noting that CBF always outperforms BUBBLE in terms of delivery ratio, albeit by diminishing margins, reinforcing the contention that CBF's delivery performance is bounded by BUBBLE, an intuition derived from the algorithm's design.

[2] In graphs on transmission and packet drop results from ER are omitted, because the values are substantially higher, rendering any comparison between the other algorithms hard to observe.

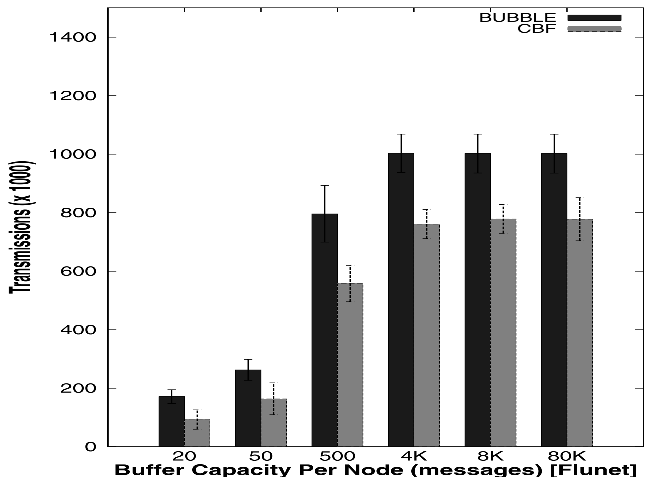


(a) Flunet

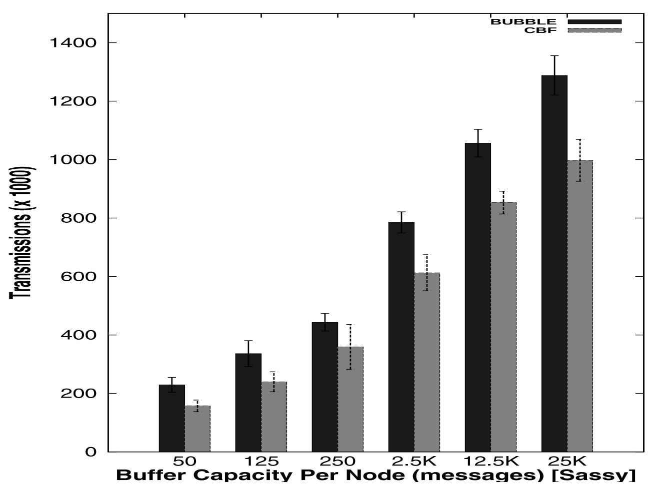
(b) Sassy

(c) SHED1

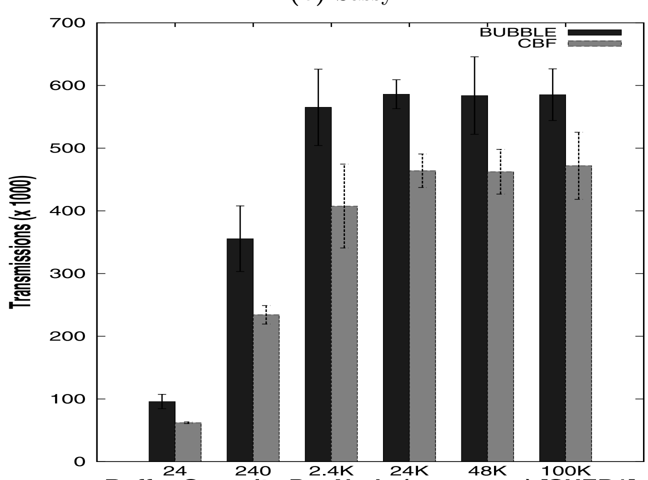
Figure 4.6: Delivery Ratio: Unlimited Resource



(a) Flunet

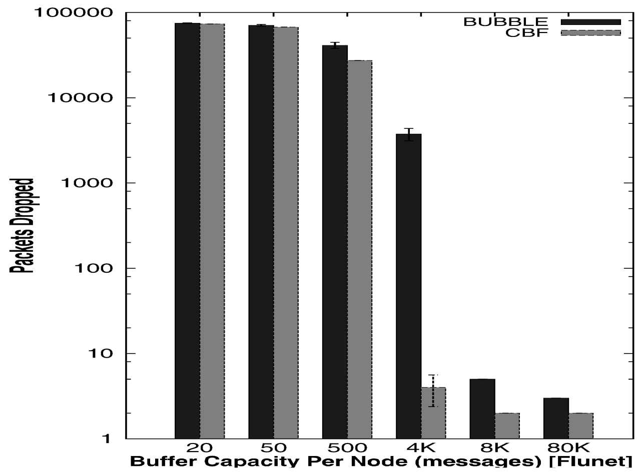


(b) Sassy

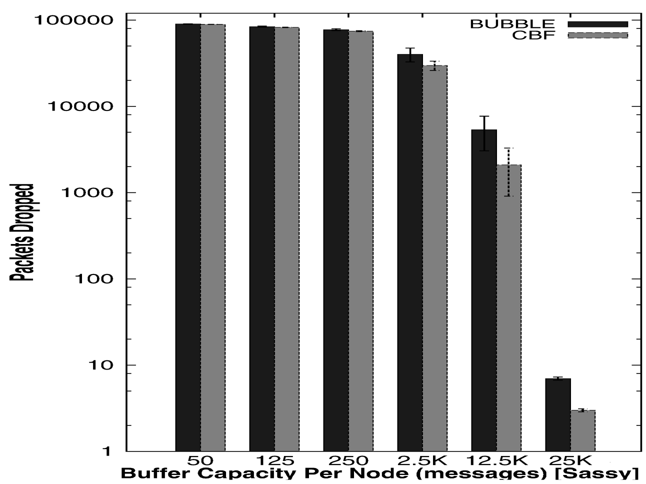


(c) SHED1

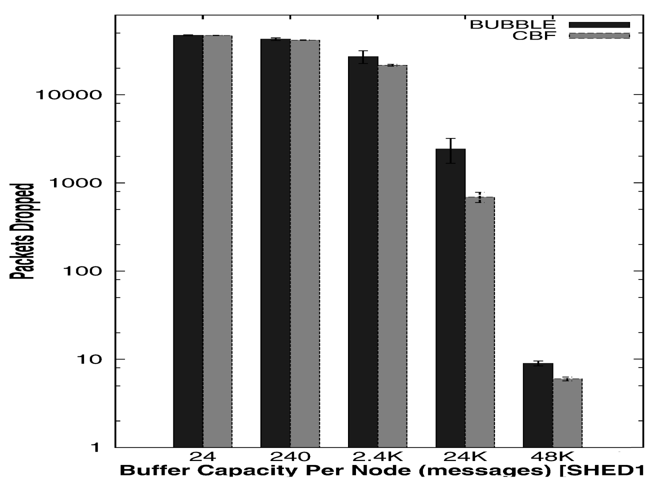
Figure 4.7: Transmission: Unlimited Resource



(a) Flunet



(b) Sassy



(c) SHED1

Figure 4.8: Packet Drop: Unlimited Resource

4.4.2 Case Study 2: Messages with Limited Life Span

In the first limited resource experiment, TTL was varied while generating a fixed number of messages and constraining the buffer as specified previously. Figure 4.9 shows the delivery ratio achieved by each algorithm on the various datasets. On average, CBF had 8.12% more deliveries than BUBBLE in Flunet, 4.89% more deliveries in Sassy and 12.98% more deliveries in SHED1. Increasing TTL in ER decreases delivery ratio, likely because with higher TTL, fewer packets expire, causing network congestion and potentially deliverable packets to be dropped due to buffer overflow. The instability of the community associations in the Sassy dataset manifests itself as poor performance for both BUBBLE and CBF under constrained resources. In the limited environments considered here, ER quickly saturates all node buffers and its delivery ratio plummets. Both BUBBLE and CBF show greater delivery ratios with increasing TTL values.

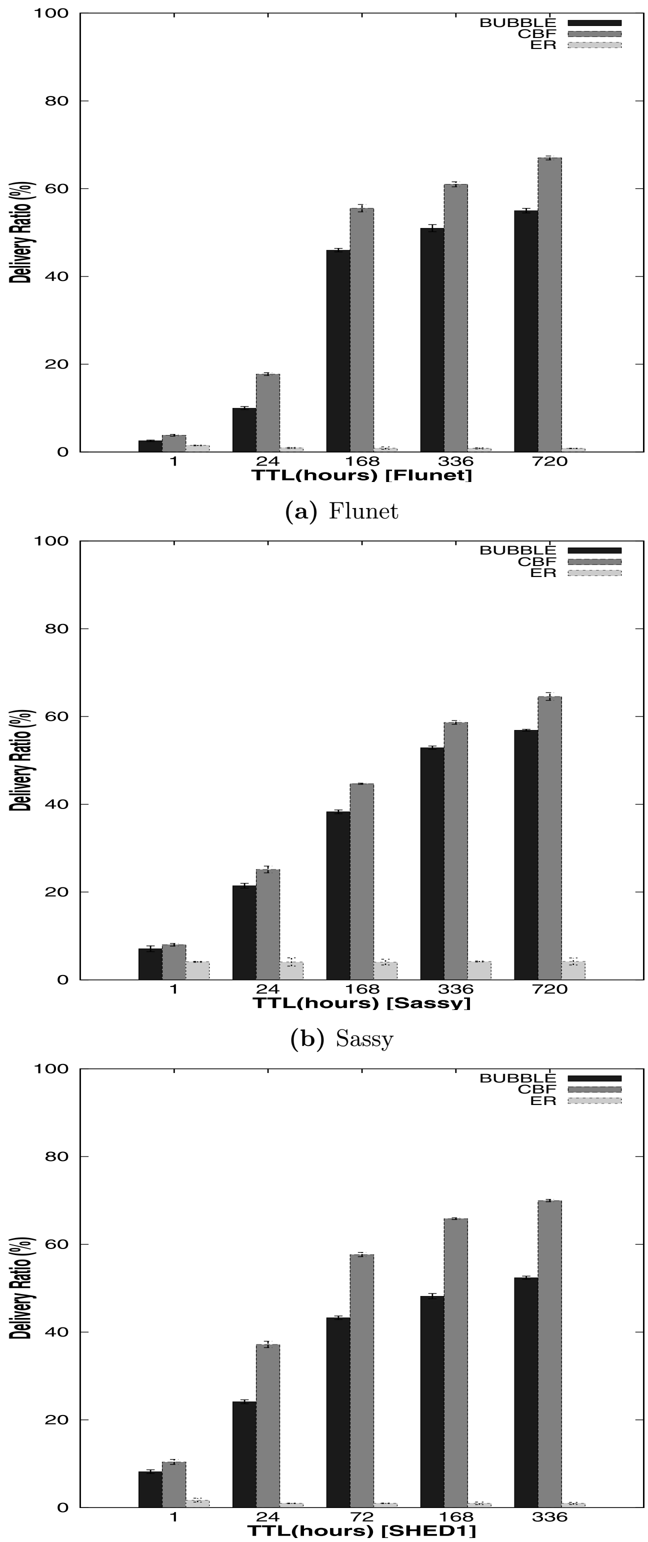
The end result is that, at moderate TTL values, CBF has higher delivery ratio with an absolute difference in delivery ratios of between 10% and 15% when compared to BUBBLE. For more stable datasets, higher TTL messages create a greater performance difference between the two algorithms.

The previous experiment with unlimited resource environment, indicates that CBF provides more equitable resource utilization. By increasing TTL, the lifespan of messages were increased, allowing them undergo more exchanges before getting dropped, shown in Figure 4.10. For both algorithms, in the case of the three-month-long datasets (Flunet and Sassy), the number of transmissions stabilizes somewhat within first fifteen days, and in the case of SHED1, after seven days. The stability of the dataset influences the growth pattern of transmissions. As TTL increases, the gap between the two algorithms reduces. For all three data sets, CBF outperforms BUBBLE by transferring more messages with fewer transmissions. Results using the SHED1 dataset shows the maximum difference whereas the least difference is seen in Sassy. Under all conditions, CBF transmits fewer packets. On average, it transmits up to 16.5% fewer messages in Flunet, up to 16.5% fewer messages in Sassy and up to 26.8% fewer messages in SHED1, respectively.

Figure 4.11 quantifies this effect on the datasets when TTL is varied. As TTL is increased, packets have a longer life span. Under both the algorithms, nodes drop older packets to make space for newer messages. In a scenario with limited buffer space, social nodes experience a high number of packet drops. On the other hand, CBF diffuses packets based more broadly to reach respective destinations. With varying age of packets, the CBF forwarding decision helps nodes to smartly manage their own limited buffer. For TTL longer than one day, CBF had 37% to 56.7% fewer packet drops in Flunet, 29% to 41.53% fewer packet drops in Sassy and 39% to 47% fewer packet drops in SHED1. TTL of one day shows almost an order of magnitude difference in packet drops, but this is not a realistic value to be used in a deployment of DTN.

4.4.3 Case Study 3: Nodes with Limited Buffer Capacity

From case study 1, it can be expected that CBF will perform better than BUBBLE by having fewer packet deletions, especially due to buffer overflow. To analyze this fact further, limited resource experiments, starting

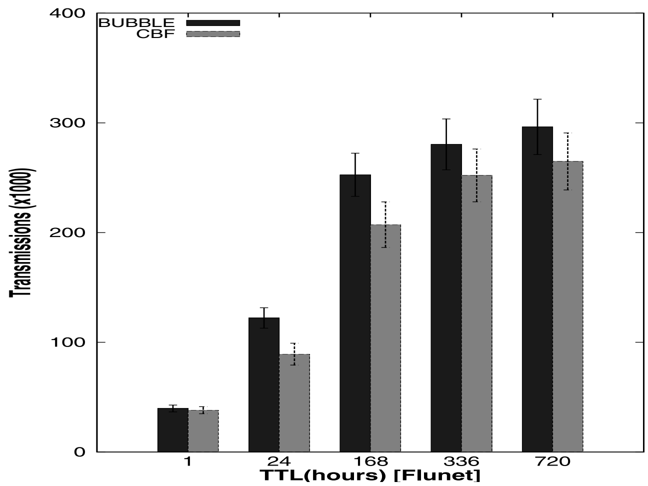


(a) Flunet

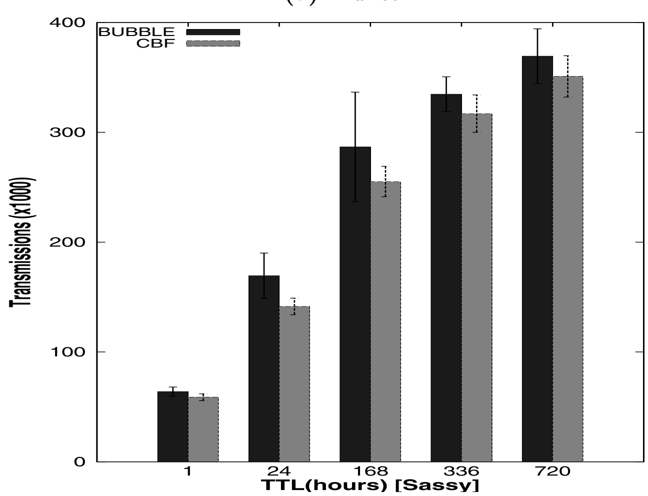
(b) Sassy

(c) SHED1

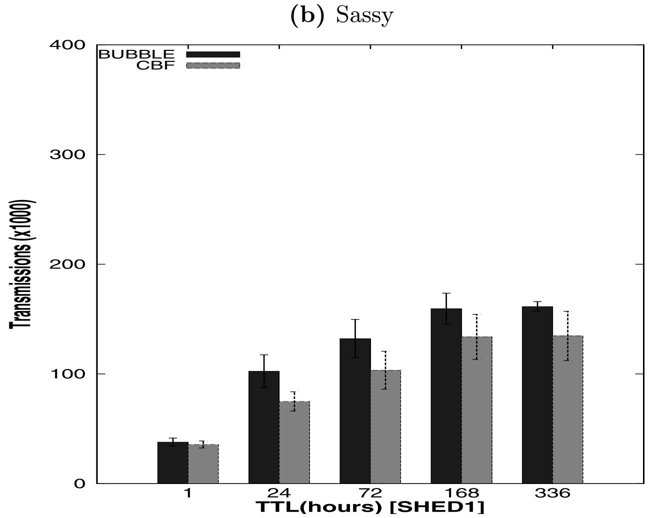
Figure 4.9: TTL-Modulated Delivery Ratio: Limited Resources



(a) Flunet

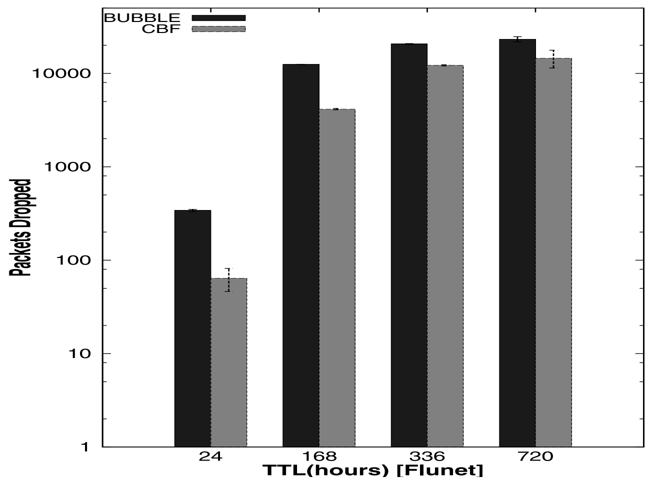


(b) Sassy

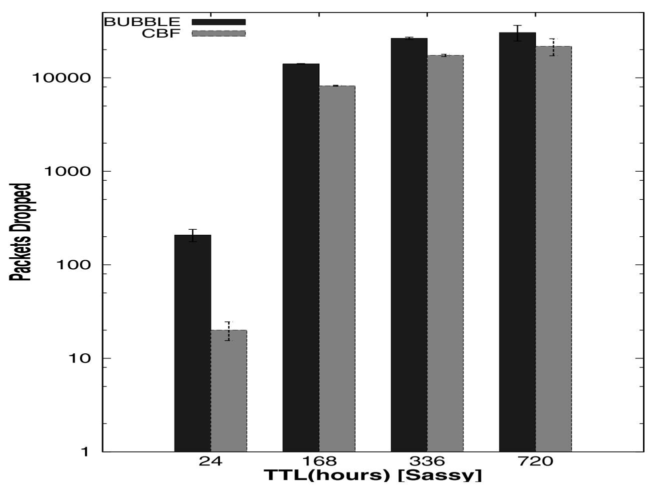


(c) SHED1

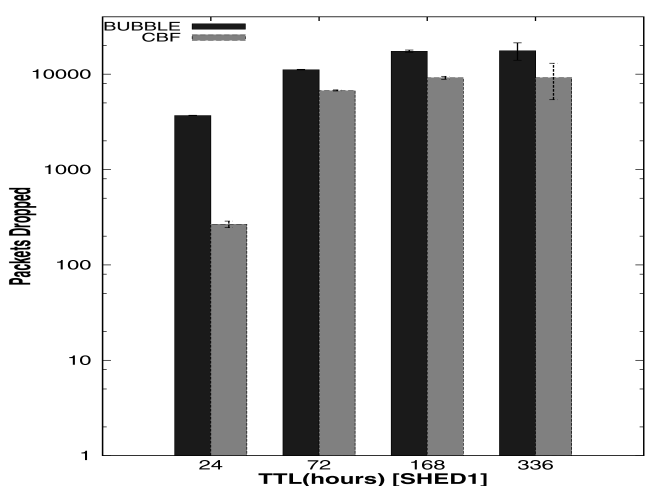
Figure 4.10: Transmissions: Limited TTL



(a) Flunet



(b) Sassy



(c) SHED1

Figure 4.11: Packets Dropped: Limited TTL

with limited buffer, were carried out. This set of experiments examined the impact of varying buffer size keeping TTL and number of messages constant. Figure 4.12 shows delivery ratio as a function of buffer space. Ignoring buffer size lower than 100, CBF delivered up to 15.11% more packets than BUBBLE in Flunet, up to 6.10% more in Sassy and up to 18.97% more in the case of SHED1.

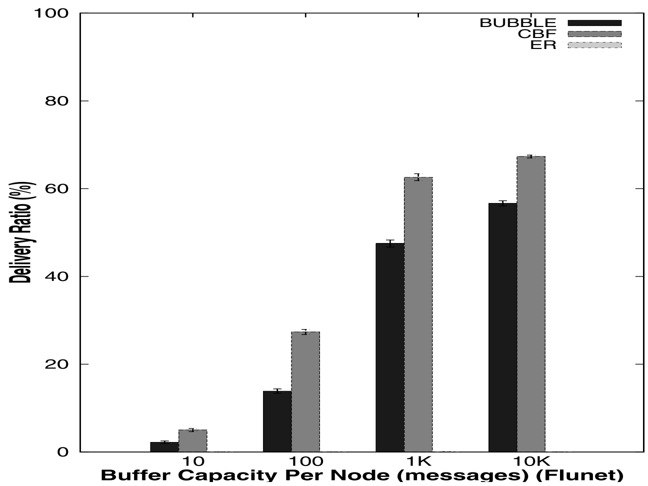
After delivery ratio, forwarding cost between the two algorithms in a realistic, resource-constrained environment is compared. Figure 4.13 shows transmissions as a function of limited buffer space. Because BUBBLE is a greedy algorithm, with larger buffer space, nodes using BUBBLE transfer messages towards the social nodes irrespective of how this affects those messages' route or age. Figure 4.13 clearly shows that there exist absolute differences of transmissions between CBF and BUBBLE for larger buffer sizes. In Flunet, CBF had 16% to 38% fewer transmissions when compared with BUBBLE, 7% to 11% fewer transmissions in Sassy and 19% to 33% fewer transmissions in SHED1, respectively.

Figure 4.14 shows packets dropped as a function of limited buffer space. In both algorithms, a larger buffer results in fewer packet drops. The absolute differences of packet drops between CBF and BUBBLE, when TTL is fixed, for larger buffer sizes is reduced. To attain better readability of diagram, Figure 4.14 shows only those conditions that led to packet drops greater than 100 packets. Ranging over various buffer sizes and the numbering of packets dropped by CBF in Flunet is 21% to 39% fewer than BUBBLE, in Sassy, 18% to 26% fewer and in SHED1, 40% to 53% fewer.

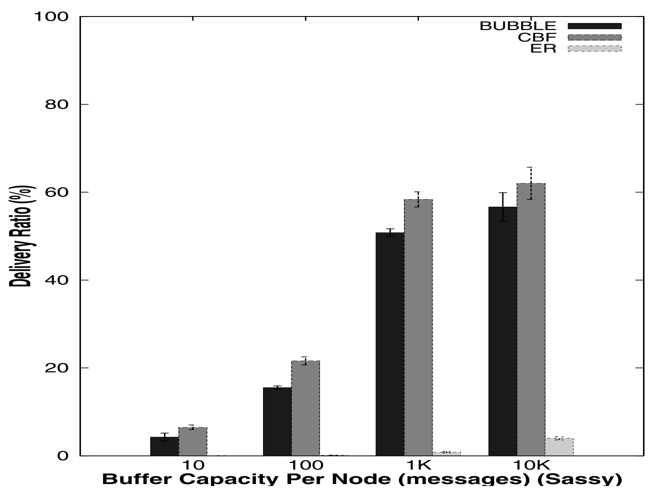
Overall, as buffer capacity increased, the performance gap between the two algorithms was reduced. With greater buffer space, BUBBLE's globally popular nodes were able to successfully relay more messages before overflowing, reducing the absolute differences of transmissions and packet drops for larger buffer sizes.

4.4.4 Load Balancing and Global Popularity

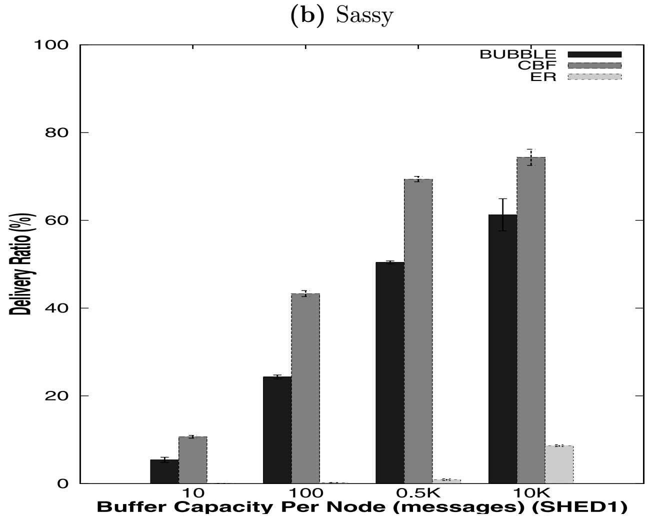
In the case of SHED1, where the top four social nodes belong to the same community, there is higher packet drop behaviour. For all three datasets, Pearson's correlation coefficient is calculated to see the impact of global popularity of a node on the amount of packets dropped by it during simulation period. The result is shown in Table 4.1. Table 4.1 shows packets dropped by nodes with limited buffer capacity when 5000 messages were generated having limited TTL. Table 4.1 depicts that, in all three datasets, packet drops have a high positive correlation with global popularity when BUBBLE is used to forward packets. When the same nodes with same global popularity use CBF, packet drop has a low positive correlation with global popularity, especially in the Flunet and SHED1 datasets. By comparing the Pearson's correlation coefficients, it can be concluded that, in a realistic limited resource environment, the dependency of BUBBLE on the most social nodes becomes its major drawback rather than its strength.



(a) Flunet

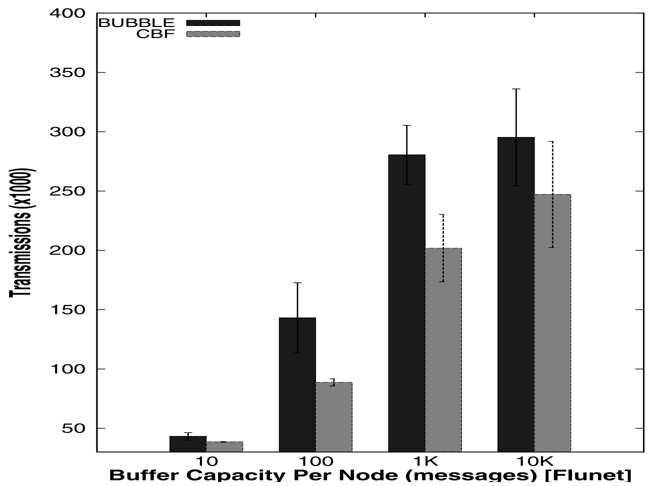


(b) Sassy

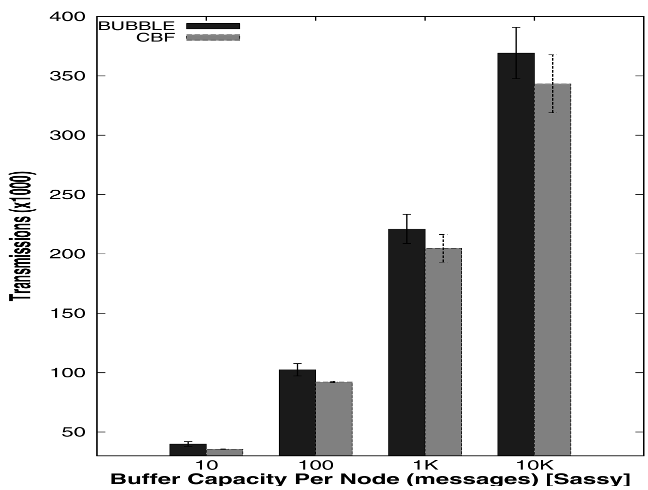


(c) SHED1

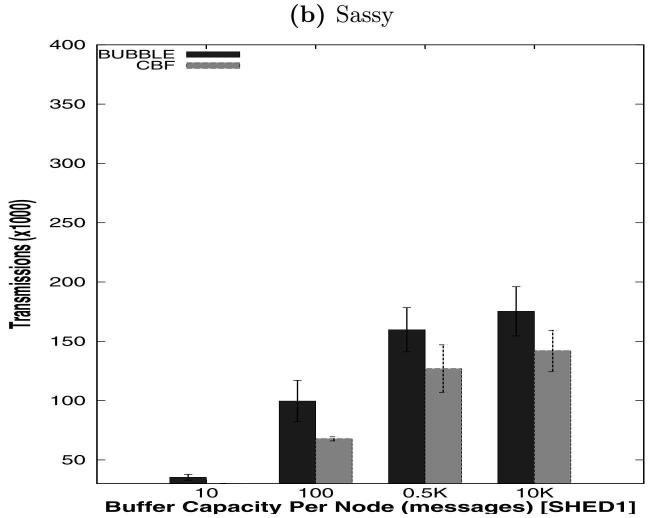
Figure 4.12: Delivery Ratio: Limited Buffer Size



(a) Flunet

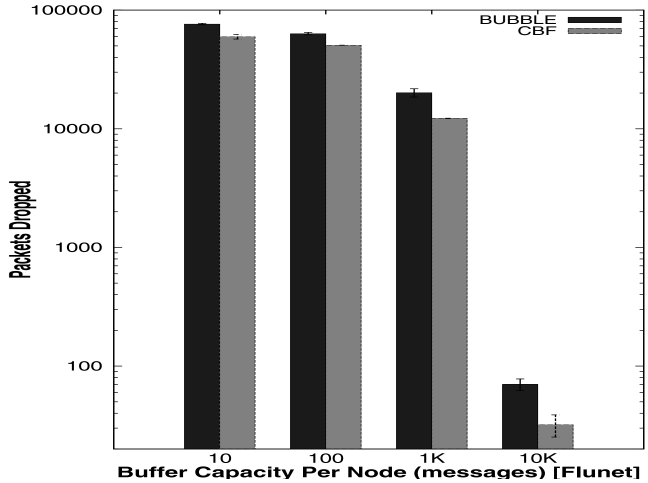


(b) Sassy

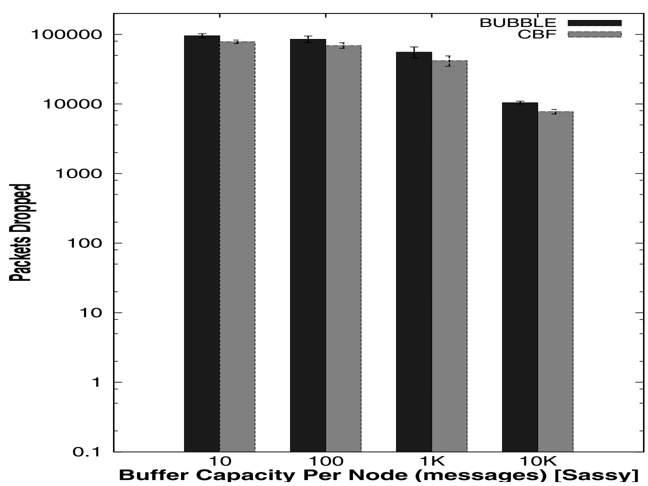


(c) SHED1

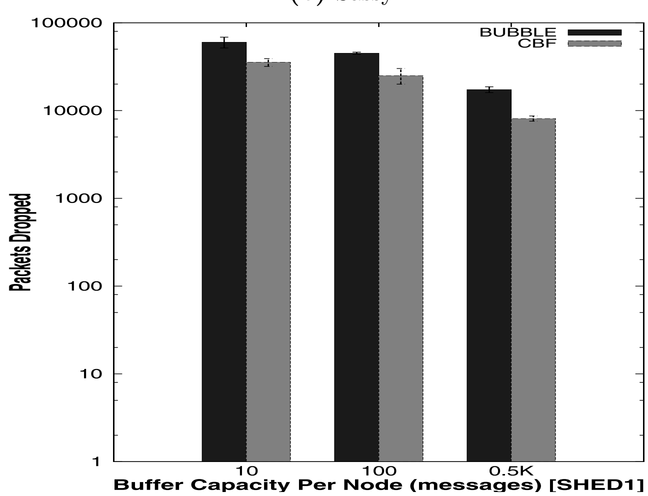
Figure 4.13: Transmissions: Limited Buffer Size



(a) Flunet



(b) Sassy



(c) SHED1

Figure 4.14: Impact of Buffer Size on Packet Drop under Limited Resources

Table 4.1: Pearson's Correlation Coefficient: Global Popularity and Packets Dropped

Algorithm	Flunet	Sassy	SHED1
BUBBLE	+0.7	+0.85	+0.90
CBF	+0.3	+0.41	+0.26

4.5 Comparison with Minimum-cost Oracle

Transmission performance of BUBBLE and CBF is compared with the *Minimum-cost Oracle* by measuring the total number of transmissions required by each of the algorithms for delivering the exact same set of messages with no resource constraints. Figure 4.15 shows that to reach a destination node, each message under CBF needs on average two more hops than the Oracle, whereas BUBBLE needs on average three more hops than the Oracle. With neither algorithms' transmission ability faring well against the Oracle, there is definitely room for further improvement in both the algorithms, a potential scope for future work.

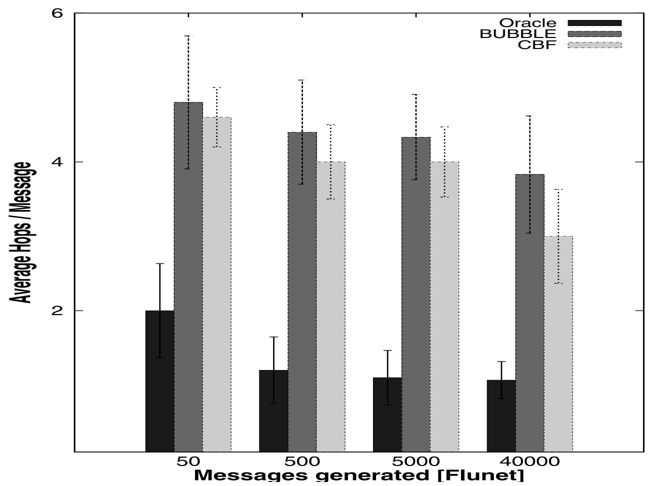
The hop count has a consistent pattern for all the algorithms in each of the datasets irrespective of the number of messages generated, because hop count is dependent on algorithm and the contact pattern, both of which remain consistent within a dataset's history.

4.6 Latency

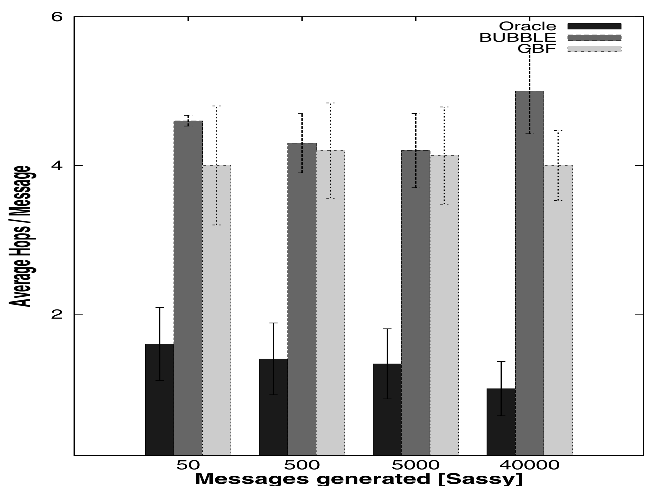
4.6.1 Delay During Routing

On average, CBF's delay always exceeds BUBBLE's. A combined view of delay for all experiments is shown in Figure 4.16. Under limited conditions, when TTL was varied, CBF produced 22.56%, 36.56% and 9.05% higher delay than BUBBLE for the respective datasets. Finally, when buffer space was varied, CBF produces 22.39%, 23.58% and 5.81% higher delay than BUBBLE. The stability of SHED1 increases the delay by a smaller amount. In the resource-rich environment, CBF had 13.48%, 33.89% and 4.44% more latency than BUBBLE, across the three datasets respectively.

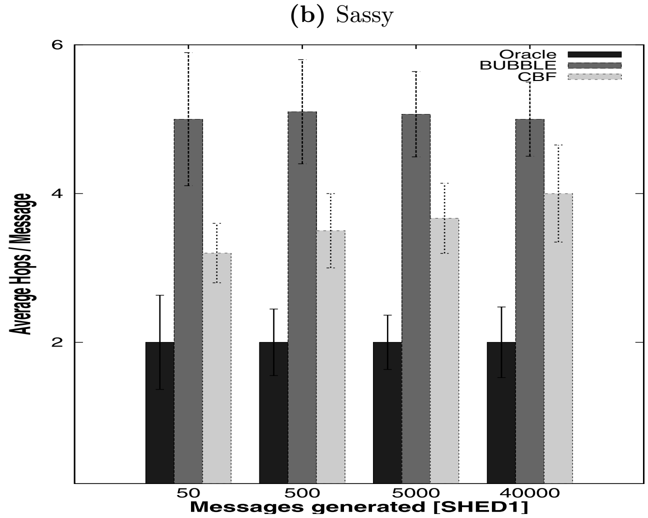
CBF propagates messages more slowly because of the choice to use a more efficient path, whereas BUBBLE uses its greedier approach to transfer packets to respective destinations faster at the cost of more buffer space and more transmissions. While this trade-off may not always be appropriate, under Assumption 2 in Chapter 3, it is valid to deprecate this performance difference because it is already assumed that the degree of delay tolerance was encoded in its TTL. Any packet delivered successfully prior to TTL expiry is considered a valid delivery and any packet with an expired TTL is considered a packet drop, both of which manifest in the delivery ratio, the most relevant performance measure under the assumptions of Chapter 3.



(a) Flunet

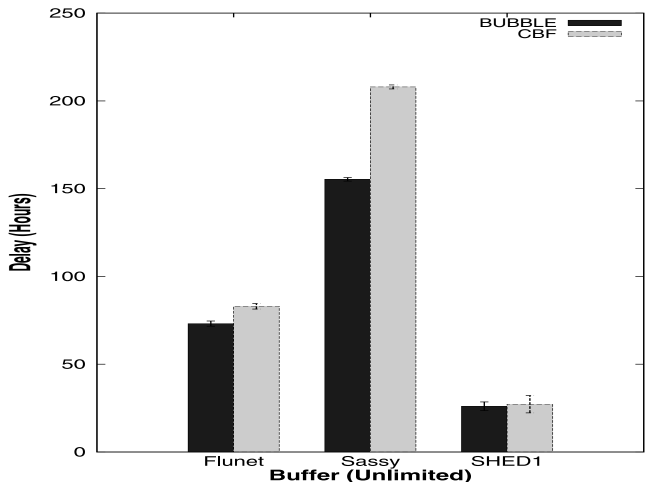


(b) Sassy

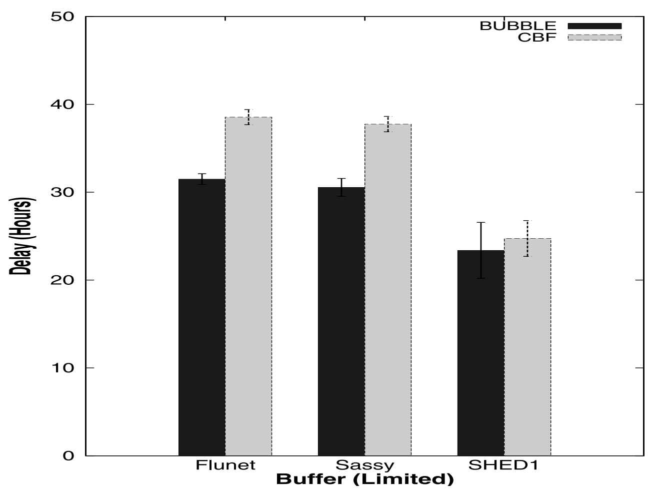


(c) SHED1

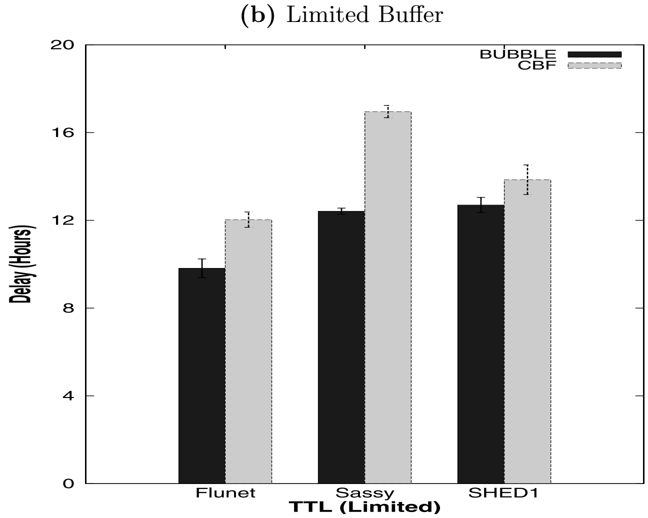
Figure 4.15: Hop Count: Limited Resource Environment



(a) Unlimited Resource



(b) Limited Buffer



(c) Limited TTL

Figure 4.16: Average Delivery Delay

4.6.2 Comparison with Fastest Oracle

In this section, the latency performance of BUBBLE and CBF is compared to the *Fastest Oracle*. As load is increased, social based routing algorithms performance approach the Oracle's, as shown in Figure 4.17. As mentioned in Section 3.5, buffer space of nodes in Oracle, Bubble and CBF is set to unlimited in order for packets to find the fastest route without facing buffer congestion. In all three datasets, with increases in number of messages produced, the Oracle algorithm delivers more messages closer to TTL, showing an increasing trend in the latency profile.

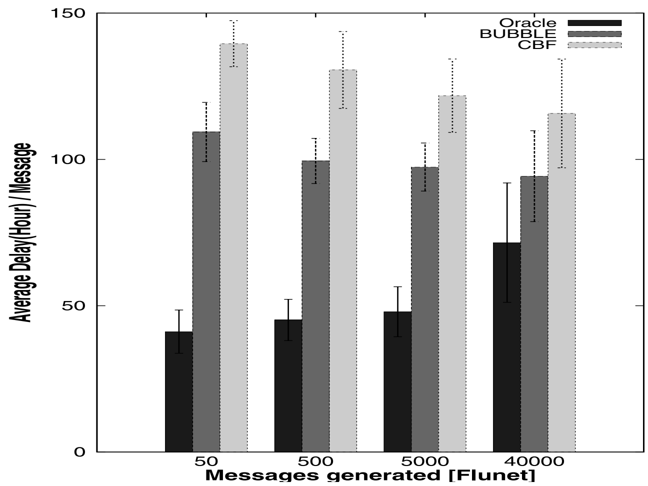
In consecutive experiments ranging from 50 messages to 40000 messages, increasing the total number of messages over the same length of the simulation period increases concentration of live packets present at a time in a network thus allowing more packets to find suitable carriers that can carry them to their respective destination faster. With more stable datasets, CBF makes a better decision in finding a suitable carrier node, thus latency reduces at a faster rate compared to BUBBLE. With neither algorithm's latency reaching close to that of the oracle, there is definitely scope for further improvement in latency.

4.7 Analysis of Inter-Community Routing Decisions

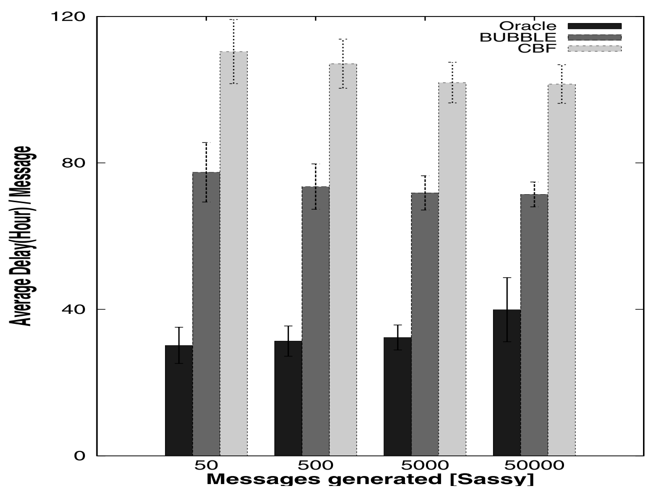
To illustrate the underlying mechanics which lead to the trends observed, an analysis of an indicative case is presented here. A fixed number of messages (80000 for Flunet, 100000 for Sassy and 48000 for SHED1), and TTL (15 days for Sassy/Flunet and 7 days for SHED1) and 10% buffer capacity (all datasets) was employed. Table 4.2 shows the use of inter-community routing factors (GP for BUBBLE, CBC and NCF for CBF). CBF achieves savings over BUBBLE in each dataset. CBC and NCF enabled decisions based on the individual relationship status of a node with other communities as a whole rather than on the connectedness of a node in the entire system. The previous results (Figures 4.13 and 4.10) show that total packet transmission in CBF is less than BUBBLE. For this reduced number of transmissions, a higher proportion of passes (12.34% - Flunet, 5.87% - Sassy, 17.91% -SHED1) were made based on inter-community factors in CBF than in BUBBLE, indicating that CBF selects an appropriate node along a likely path rather than a globally popular node more often.

Table 4.2: Data Transfer Factor Percentages

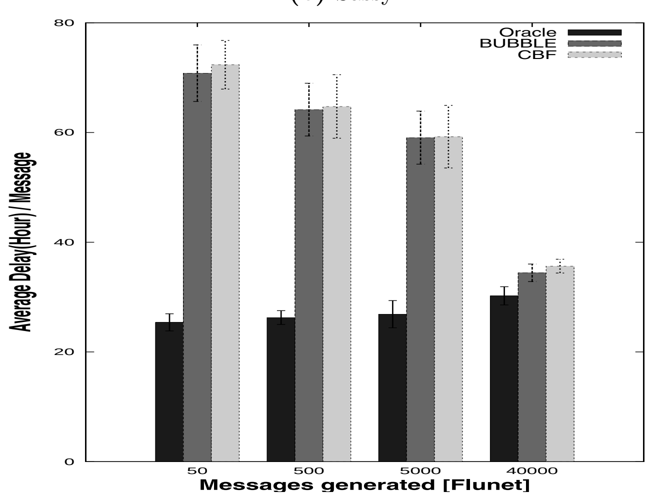
Dataset	**Routing Factor**		
	in **BUBBLE** due to	in **CBF** due to	
	GP	**CBC**	**NCF**
Flunet	38.3	18.1	32.6
Sassy	34.0	13.7	26.2
SHED1	50.2	26.2	41.9



(a) Flunet



(b) Sassy



(c) SHED1

Figure 4.17: Message Latency: Limited Resource Environment

4.8 Summary:

Sastry *et al.* [73] have shown that connectivity in PSNs are crucially dependent on the 'rare' contacts. In this work through quantitative analysis of different types of datasets, it is shown that inter-community interactions between nodes falls in the minor portion of total interactions, thus they can be treated as 'rare' contacts. Over a decade of research on PSNs, not many works have focused on developing routing strategy which mostly employs these rare contacts. This thesis is proposing an enhanced routing algorithm, named Community based forwarding (CBF), which incorporates rare contacts interactions like pairwise-community interactions and node-to-community interactions to find paths comprised of fewer hops towards the destination group than that of comparator algorithms. Total transmission decreases without decreasing delivery ratio. In this way, CBF ensures energy efficiency which makes it suitable as a routing algorithm for resource constrained PSN devices. CBF is an enhancement over BUBBLE, thus most of the experiments in this chapter compared CBF to BUBBLE.

In a resource constrained PSN system, particular attention must be paid to the appropriate utilization of available resources. CBF which attempts to balance the use of resources with the likelihood of delivery by adopting a more conservative forwarding scheme preferentially chooses nodes with a greater probability of being in the target community. The result of the simulation experiments conducted to quantify the performance difference between CBF and competitor algorithms show that CBF transmits fewer messages in all cases, making it more energy-efficient and its delivery ratio is bounded above by BUBBLE. More stable communities are, stronger the social ties between and within communities. As CBF is using social metrics for packet forwarding, experiment results show that stronger social ties have positive impact on forwarding capability of CBF.

Despite all these probable useful applications of energy efficient CBF, its delay exceeds that of BUBBLE's. This is the only trade-off seen so far in CBF. As CBF is an algorithm for delay tolerant systems, higher delay is not considered as a major drawback. However, how to lower delay and make CBF perform better than BUBBLE in every aspect of routing in social context, is the next challenge this thesis explores in Chapter 5.

Chapter 5

Hybrid Community Based Forwarding: A Complete Energy Efficient Algorithm for PSNs

The previous chapter confirms that in inter-community message forwarding, the performance of routing schemes like BUBBLE can be improved for different types of datasets, varying from extremely clustered (SHED1) to individual (Sassy), by changing the nature of forwarding heuristics, from greedy (i.e. Popularity rankings) to conservative. The major trade-off of CBF is additional delay. So the next challenge is to find more optimized paths for messages so that they can reach their respective destinations faster, without increasing transmission or reducing the delivery ratio. The focus of this chapter is to find way of overcoming the extra delay introduced by CBF decision making, without losing out the benefits by CBF over BUBBLE, gained so far.

5.1 Delay Reduction Options

There are two proposed approaches by which improvements in delay can be achieved. One of them is to completely change the way of looking at the connection between communities. This can be done by modeling the communities as hierarchical rather than the flat structure it has now. One way to do this is by finding dynamic community-based-tree-structure existing among communities and then use that to find possible available shortest route to the destination. This approach involves more into finding better clustering strategy than enhancing existing routing scheme. Another requirement for the proper evaluation of this approach is that datasets for modeling need to have a sufficient number of nodes so that hierarchies among communities can be found or heterogeneity among social behavior of nodes can be detected. There are very few of these datasets available as the length of time and amount of effort needed to create such a dataset is immense. An initial steps for looking at this approach is provided as future work of this thesis in Chapter 6.

The second way to improve delay is based on finding improved method for intra-community forwarding. So far, the improvement in CBF is due to changes in the way inter-community packets are routed. However, previous results in Chapter 3 show that in three of the datasets used, more than half the encounters occur between nodes in the same community (72.73% of Flunet, 78.9% of Sassy and 62.1% of SHED1). The fact that a major fraction of total interactions happen inside communities suggests the existence of more community features remaining unexplored, which can be used to further enhance the performance of CBF. How a message is being forwarded within all three sending, intermediate and destination communities has an impact on both intra and inter-community routing. Once messages reach the destination group, the existing method of intra-community message passing is same for both CBF and BUBBLE. The current approach in BUBBLE finds the the most locally popular node based on how often each of them interact with members in same community and then selects that node to bubble up the message towards the destination. In addition to local centrality, if nodes with more diverse interaction patterns are considered, then the most cooperative nodes will be selected as carriers and the presence of such nodes in forwarding scheme will help to reduce latency. For this thesis, the second approach has been considered. How the second approach gets along with other metrics of CBF is elaborated further in the following sections.

5.2 Mobility

Let us consider a scenario. Suppose two nodes belonging to same community increase their 'Local Popularity' by only contacting each other and no one else in that community, then should they be labeled as the 'most social nodes' in that community? Based on BUBBLE's way of finding most social node, BUBBLE will rank these two nodes as the most social, but how realistic is this approach? There is no doubt that these nodes are not actually social as their genuine social influence is very limited. To avoid such misguided nodes from being selected as carriers, some other behaviours of nodes have to be considered. One such feature is 'mobility'. The term mobility in this thesis has nothing to do with geographical movement, rather it refers to diversity in the number of unique nodes which are encountered by a particular node. When a node is socially popular as well as has a diverse interaction pattern, its probability of meeting a variety of nodes increase and so does its ability to deliver packets becomes faster and more reliable. For intra-community forwarding apart from popularity, mobility is an important factor [64] and motivational point for an enhanced protocol to overcome current drawback of high latency.

CBF's target was to provide more reliable and energy efficient inter-community paths for messages to reach their destination group. After a message arrives at the destination group, due to the nature of each node's local popularity metric, messages might fall in the loop of greediness, finally get stuck in the highly popular stationary node or get trapped into false maximas; both cases increase the total latency, degrading the overall gain in performance attained so far. In order to continue CBF's nature of avoiding unnecessary transmission of messages and to reduce delay due to stationary local maximas or false maximas, it is necessary to add mobility along with 'social'-ness in the calculation of node ranking. This helps decision making of the enhanced forwarding scheme remain as firm and controlled as it was in CBF's inter-community routing. The high contact rate nodes [64] are mostly responsible for efficient content dissemination and thus ensure higher delivery ratio. A method to measure the contact rate given the characteristics of the data is presented in the next subsection.

5.2.1 Unique Interaction

Unique interaction (UI) (Eq. 5.1) of a node calculates how many neighboring nodes, within its own community, it has interacted with, over a period of time marked by epochs. In the equation below, $g(p, q, r)$ refers to an encounter between 2 nodes, node p and node q on every r^{th} epoch. As in the previous experiments, the epoch duration is seven days.

Unique Interaction:

$$\forall(x \in C) \quad UI_x = |S_x| \qquad where\ S_x = \bigcup_{y \in C_x} s.t.\ \ \exists g(x, y, k) = 1,\ k \leq K \tag{5.1}$$

As both UI and LP are based on interaction between nodes within the same group, they might appear to related to each other. However, when these two values are compared for every node in the three datasets, no correlation has been found. For example, a small community might have a pair of highly interactive nodes which are interacting mostly within themselves, thus the LP values of those nodes will be very high, but due to the small size of the set of interacting members, the unique interaction values for those individual nodes will be low. Figure 5.1 confirms that there exists no relationship/dependency between LP and UI. In this thesis, the words UI, mobility and diversity are used inter-changeably.

Before introducing UI into CBF, the distribution of nodes' UI values for each node in each of the datasets was studied. Figure 5.2 shows the cumulative distribution function (CDF) of the average mobility of nodes when they are ranked in descending order of mobility. Another way of viewing the data in Figure 5.2 is to notice that 50% of the sum of the UI values in the Sassy dataset is contributed by the top 4 nodes. 18 nodes are needed to contribute 50% of the mobility in SHED1. If UI is used as a routing factor, the top 4 nodes in Sassy will be selected more often and could become bottlenecks, potentially reducing the advantage of introducing UI in the first place. In comparison to Sassy, a larger number of nodes in SHED1 nodes fall in the socially mobile criteria because nodes in SHED1 are the members of most stable groups; thus they have a higher probability of meeting more community members weekly. The average values of UI for each dataset is calculated and shown in Table 5.1.

Table 5.1: Average UI Values

Dataset	**Average weekly UI**
Flunet	3.32
Sassy	2.39
SHED1	4.63

5.2.2 Algorithm: Hybrid Community Based Forwarding (HCBF)

A formal representation of how UI is included into CBF algorithm is shown in Algorithm **??**. The formal representation of CBF is shown in Algorithm 3. The input parameters are the carrier node, destination node,

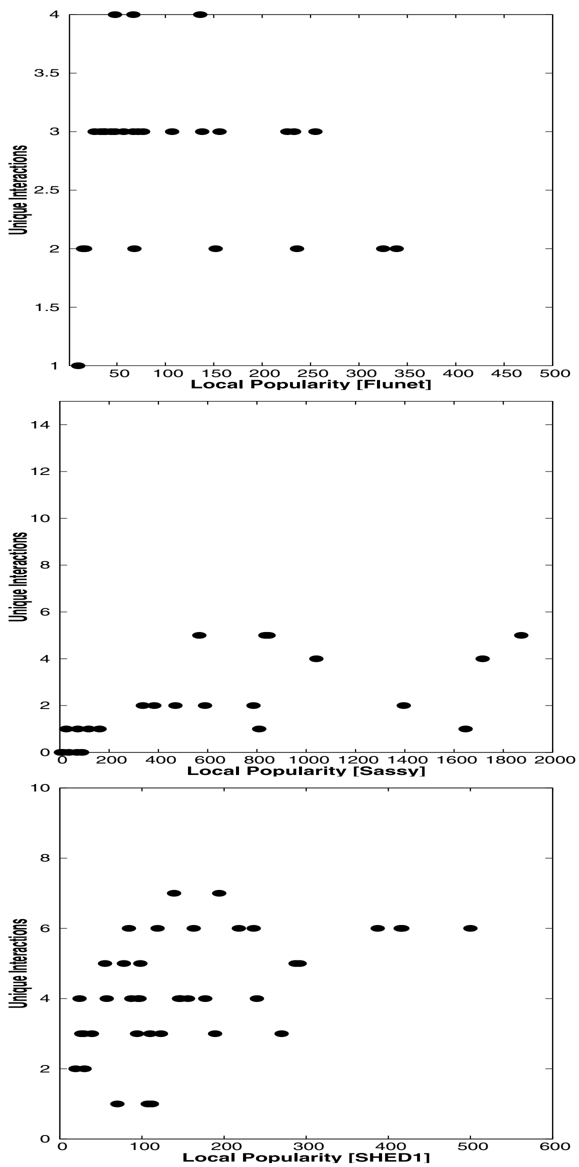


Figure 5.1: Relation Between LP and UI

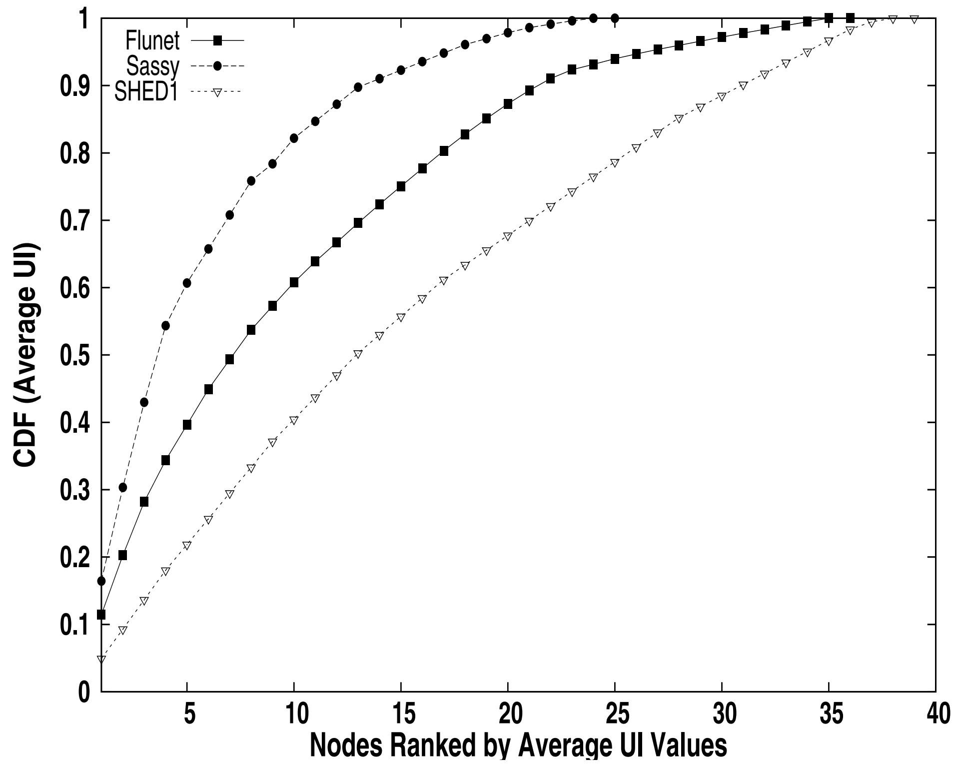
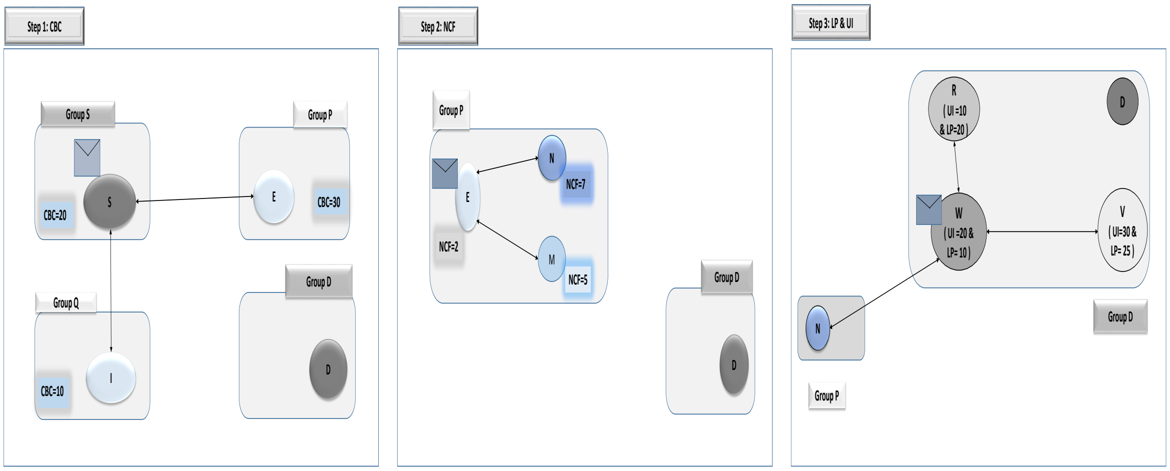


Figure 5.2: CDF of UI

the list of encountered nodes (encountered by the carrier node in time epoch t) having a size of 'n', nodes which have the maximum LP, maximum NCF, maximum CBC and maximum UI from the encountered list of nodes. This reinforced algorithm is called Hybrid Community Based Forwarding (HCBF). HCBF is CBF with an extra encapsulation of mobility, thus inheriting all the benefits of CBF. One part of its decision making heuristics are conservative and holds messages back, yet at the same time the other part is designed to deliver messages fast within communities to get to the right labeled node. As the new proposed algorithm is a combination of such features, it is called the 'Hybrid Community Based Forwarding' (HCBF). The Intra-community routing is the only part of the algorithm that is different from CBF. In an intra-community interaction, HCBF initially tries to find the most socially mobile (SM) node. If found, then message is forwarded to that node. If not found, then message is passed to a node that is equally mobile but more social. So it is keeping the advantage of BUBBLE while adding mobility, which makes it efficient. In this decision making, the 'stationary' nodes and 'stationary and unpopular' nodes, which are potential sources of delay and unnecessary transmission are avoided, so only the social mobile nodes are targets of CBF and nodes.

During message delivery in HCBF, messages are forwarded towards destination communities using CBC and NCF and once they arrive at the destination communities, nodes with higher LP and UI are selected to carry them faster towards their respective destinations. The only disadvantage of CBF over BUBBLE in Chapter 4 was its delay, which HCBF partially overcomes by using UI. Figure 5.3 is an example of how



(a) Use of CBC **(b)** Use of NCF **(c)** Use of UI

Figure 5.3: Illustration of HCBF Routing Algorithm

messages are routed using HCBF. In this scenario, like Figure 4.1 in Chapter 4, node S has a message for node D. Step 1 and step 2 are exactly the same as CBF, as in these steps, inter-community routing is performed. The enhancement lies in the last step where intra-community forwarding of message occurs. In this step, the message carried by node N enters the destination group via node W. Node W encounter node U and node V. Even though node U has higher local popularity, it is not selected due to its lower diversity (i.e. UI value). Node V is selected instead, because node V has both UI and LP greater than that of node W. In this way, by selecting the diverse and popular nodes, messages are pushed faster towards their respective destinations. The next section elaborates on performance analysis of forwarding capability of HCBF along with its comparisons to CBF.

5.3 Impact of Nonsocial Mobile Nodes on HCBF

As can be seen from the algorithm, HCBF considers only socially mobile nodes as potential forwarder. Nodes that have been avoided are the unpopular non-mobile nodes and the unpopular mobile nodes. In Pietiläinen and Diot [64], it is the nonsocial high contact rate nodes that are mostly responsible for efficient content dissemination and their presence ensures higher delivery ratio. In their work [64], contact rate refers to number of contacts per unit time and social node refers to a node that spends significant portion of its contact time in a 'temporal community' (i.e. hierarchically aggregated community). If the time unit is considered to be same, then high contact rate nodes can be *considered* to be highly mobile or diverse nodes. In that case, HCBF should give preference to diversity and not at all to sociability of nodes. To decide whether or not to include nonsocial mobile nodes, experiments were carried out. In order to distinguish HCBF proposed from the HCBF which only favors mobile nodes, the former is named HCBF and latter is named HCBF$_{nonstationary}$.

Algorithm 3 Community-Based-Forwarding (Node carrier,Node [] en, Node enMaxLP, Node enMaxUI, Node enMaxNCF, Node enMaxCBC, Node dest)

for i=1 to n **do**

// Destination encountered

  if $(en[i] == dest)$ **then**

    $en[i].addMessageToBuf(message)$; **break** ;

//Member of destination group encountered

  else if $((C(en[i]) == C(dest))$ **and** $(C(carrier) \neq C(dest)$ **then**

    $en[i].addMessageToBuf(message)$; **break** ;

  end if

end for

// Intra-community routing

if $(C(carrier) == C(dest))$ **and** $(C(enMaxLp) == C(dest))$ **and** $(enMaxLP == enMaxUI)$

    and $(LP(carrier) < LP(enMaxLP))$ **and** $(UI(carrier) \leq UI(enMaxUI))$ **then**

    $enMaxUI.addMessageToBuf(message)$;

// Inter-community routing

else

//Carrier & encountered node in same group

  if $((C(enMaxNCF) == C(carrier))$ **then**

    if $(C(enMaxNCF) \neq C(dest))$ **and** $(NCF_{[carrier][C(dest)]} < NCF_{[enMaxNCF][C(dest)]})$ **then**

      $enMaxNCF.addMessageToBuf(message)$;

    end if

// Carrier, encountered and destination nodes in different groups

  else if $(CBC_{[C(carrier)][C(dest)]} < CBC_{[C(enMaxCBC)][C(dest)]})$ **then**

    $enMaxCBC.addMessageToBuf(message)$;

  end if

end if

To test this method, about 50000 inter-community messages were generated with fixed TTL and 10% buffer capacity for each of the three datasets. Then in similar environment, turn by turn nodes routed messages using BUBBLE, CBF, HCBF and HCBF$_{nonstationary}$ and results of performances were recorded. The results are shown in Figure 5.4.

It has been seen that that delivery ratio of HCBF$_{nonstationary}$ improved compared to that of CBF by 2.41% in Flunet, 1.5% in Sassy and 1.37% in SHED1 and delay reduces by 11 hours, 10 hours and 14 hours per message. When compared to the performances of CBF, the proposed HCBF has slightly higher delivery ratio than HCBF$_{nonstationary}$ and lower delay, whereas packet drop is slightly less and transmission is far less than that of HCBF$_{nonstationary}$. The improvement in delay comes with a tradeoff of immense transmission cost which exceeds BUBBLE's transmission cost by 60.2%, 75.8% and 93.8%. So while reducing delay and increasing gain delivery ratio further, HCBF$_{nonstationary}$ negates the benefits which CBF originally achieves in transmission costs. From this analysis, it can be concluded that only considering mobile (or high contact rate) nodes as forwarders is not a good decision metric in a social context. This is the reason why the proposed HCBF considers only social mobile nodes as potential forwarders.

Although Pietilänen *et al.* [64] is the motivation for this thesis to include node mobility, the difference in result of this thesis (i.e. HCBF) from the one stated by Pietilänen *et al.* [64] (i.e. HCBF$_{nonstationary}$) is because of the following reasons:

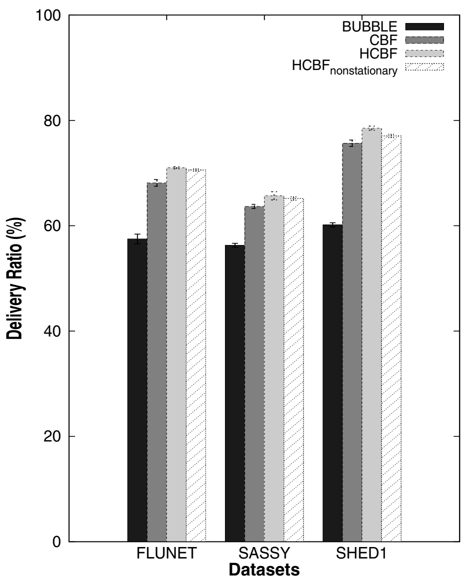
1. Different community formation methods: In both works, communities are detected using Louvain algorithm but in the work of Pietilänen *et al.*, they further hierarchically aggregate the communities into 'temporal' communities.

2. Different definition of terminologies: The difference in meaning of terminology leads to difference in results. In this thesis and in BUBBLE-Rap [39], social node refers to the nodes' interaction capability whereas as per Pietilänen *et al.* social refers to total time spent in a temporal community.

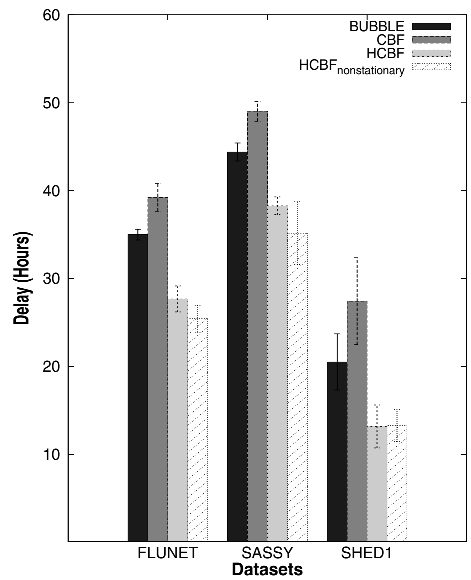
3. Different time period: In Pietilänen *et al.* [64], the authors deal with length of contact and inter-contact time lengths. However, in this thesis we have assumed all contact times to be uniform prior to experiments.

5.4 Inter-community Routing in Resource Constrained Environment

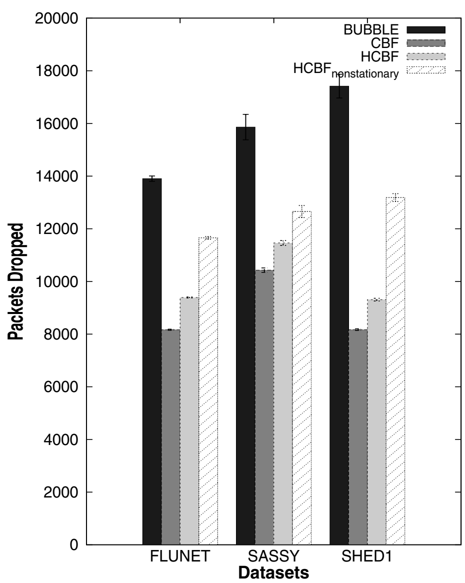
In this section, a comparative study on performance analysis of HCBF is done. The sequence of experiments is kept similar to previous experiments in Chapter 4 and the performance metrics measured are delivery ratio, transmission, packet drop and latency. This section has two parts. The first part compares how much improvement in performance is achieved by HCBF over CBF, in inter-community routing. It is seen from Chapter 4 that CBF achieve its full strength under limited resource condition, so this section of comparative



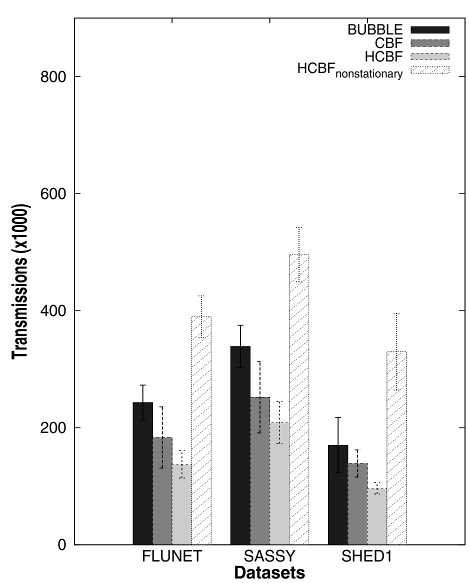
(a) Delivery Ratio Under Limited Resources



(b) Delay Under Limited Resources



(c) Packet Drop Under Limited Resources



(d) Transmission Under Limited Resources

Figure 5.4: HCBF$_{nonstationary}$ Performance Measures

analysis is done in the limited resource environment. Only at first, buffer capacity of nodes is varied in limited environment condition and then similarly message TTL is varied. Then the next part of this section, examines the overall changes in performance introduced by HCBF compared with BUBBLE's during both inter and intra-community forwarding. This part comprises of limited and unlimited resource experiments.

5.4.1 Case 1: Messages with Limited Lifetime

This section of results focuses on the performance of inter-community forwarding of HCBF in limited resource environment. Messages are generated such that their source and destination belong to different communities.

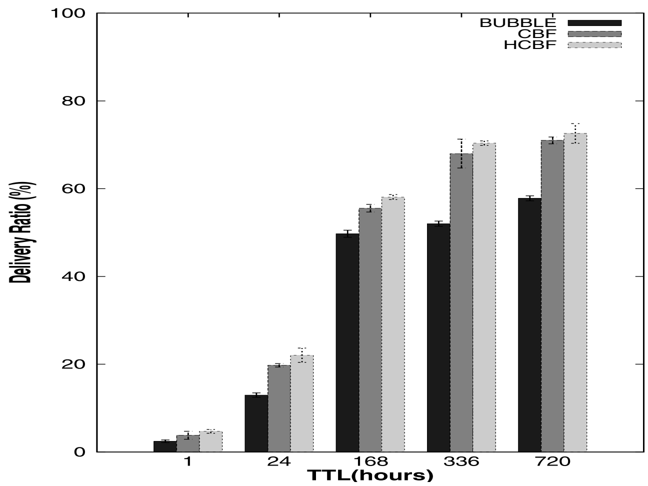
The first experiment for examining performance difference in inter-community message forwarding between HCBF and CBF is the experiment with messages having limited TTL. Comparison of reliability, expressed in terms of delivery ratio, is shown in Figure 5.5. Overall, performance of HCBF over CBF in term of delivery ratio is seen to improve by 2% to 11.6% for Flunet, 1.5% to 9.67% for Sassy and 1.5% to 7% for SHED1. Usually in ad-hoc networks when messages have longer life span, it is a blessing for routing algorithms. However, when nodes use HCBF and messages age span over 15 days, then higher TTL values degrade system performance in terms of packet drops, transmission and delivery ratio. This is because older messages persist longer in the resource constrained nodes having limited buffer thus inducing congestion in SM nodes, which in turn drops packets from the network. When message have TTL 7 days or less, with 10% buffer capacity nodes deliver maximum packets. This can be seen in Figure 5.5.

When messages have longer lives, older undelivered messages are dropped out of node's limited buffer to make space for newer ones, thus the total number of packet drops in the network increase, reducing total number of transmission for all algorithms in the three datasets. However for TTL of 15 days or more, HCBF drops more than CBF because SM nodes' buffers undergo heavy congestion as they shoulder the major burden of local communication. The results of drops can be seen in Figure 5.6.

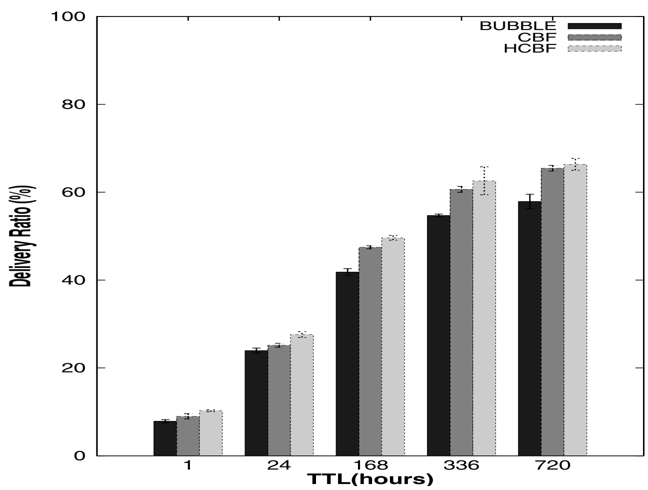
Having to transmit messages with longer TTL increases the total delay of all three algorithms. Drops cannot, however, impact HCBF's improvement in delay. With the introduction of the concept of mobility, HCBF's delay is always less than CBF's and close to or less than BUBBLE's. The results of measured output are shown in Figure 5.6. To be specific, if TTL is 24 hours or longer, HCBF delay is less than that of BUBBLE. The delay gap between HCBF and BUBBLE increase up to 7 days, after which the increment happens at a decreasing rate. This again points to the congestion created by older messages with long lives.

Change in TTL also has an impact on transmission of all the algorithms. Under CBF, SM nodes uses their popularity and diversity to find most suitable carrier in a shorter period of time than nodes using BUBBLE for intra-community forwarding. Thus, at lower TTL values, the maximum performance gap is observed. Figure 5.8 shows the results of transmission. It is seen that, HCBF saves 15.98% to 25% of transmissions in Flunet, 10% to 17% transmissions in Sassy and 23% to 32% transmissions in SHED1.

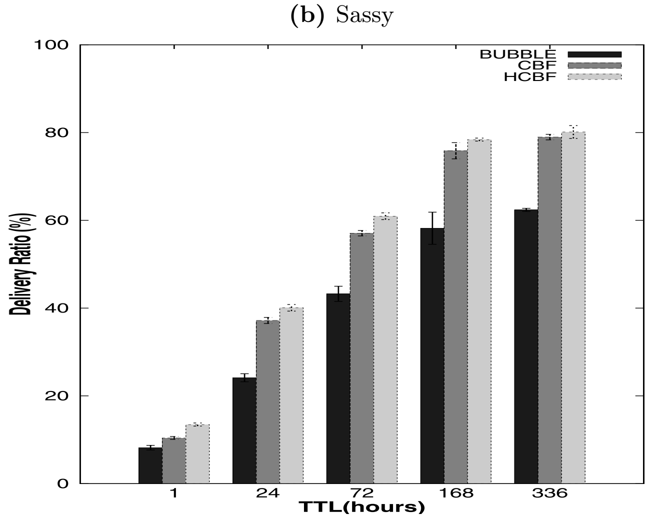
The faster nodes deliver messages, the faster they make space within themselves to carry next set of messages. Due to this nature, SM nodes can utilize limited resource as efficiently as possible. Overall, in



(a) Flunet

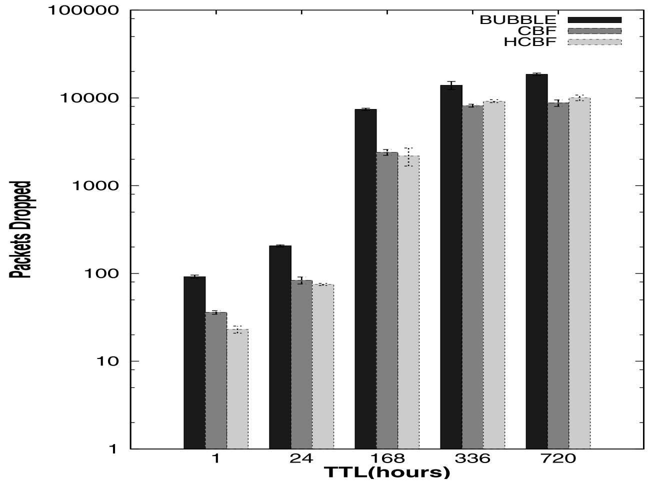


(b) Sassy

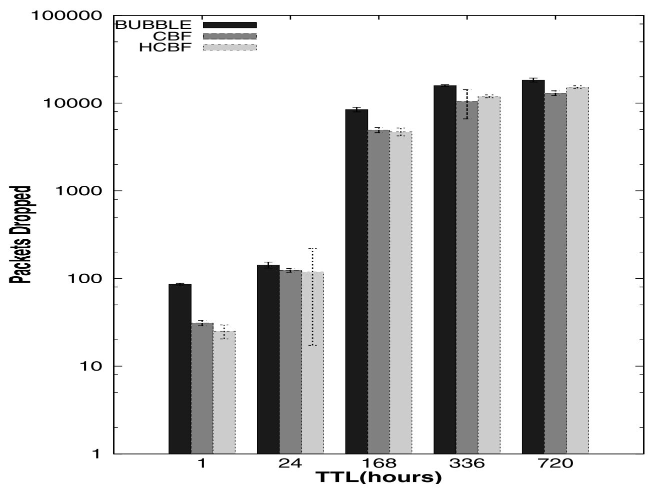


(c) SHED1

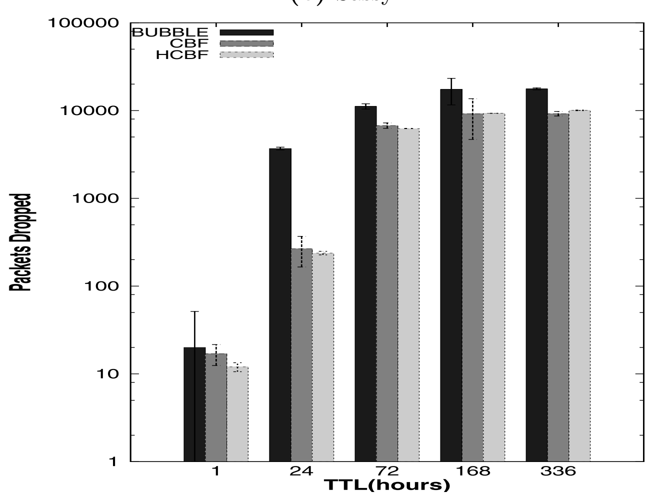
Figure 5.5: Delivery Ratio: Limited TTL (Inter-community Messages)



(a) Flunet

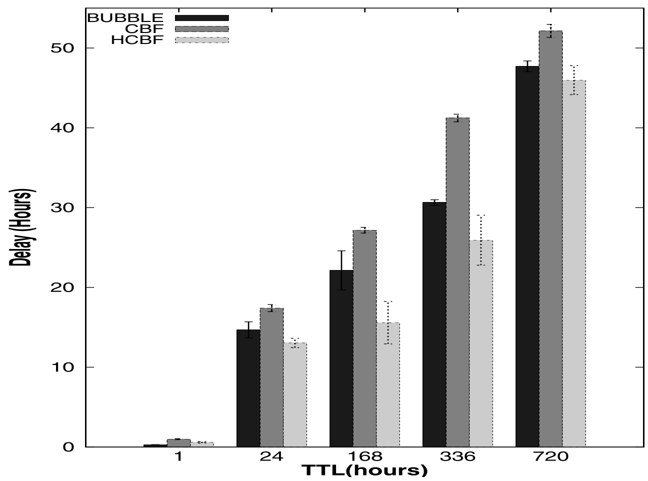


(b) Sassy

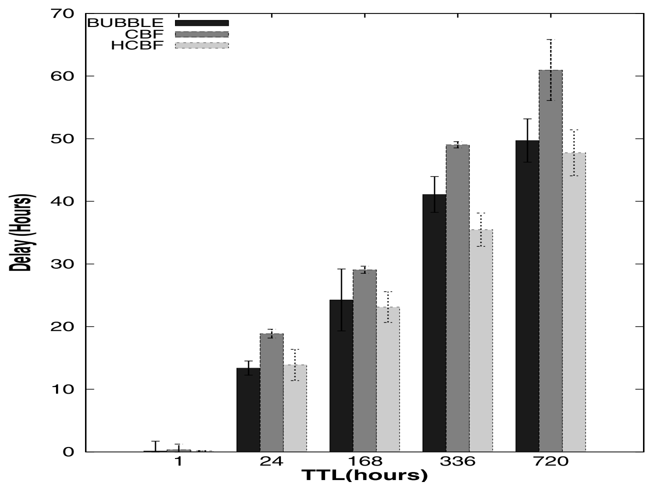


(c) SHED1

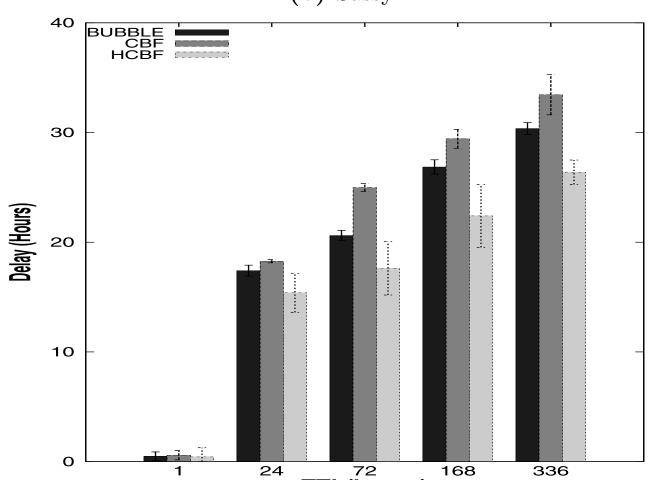
Figure 5.6: Packets Dropped: Limited TTL (Inter-community Messages)



(a) Flunet

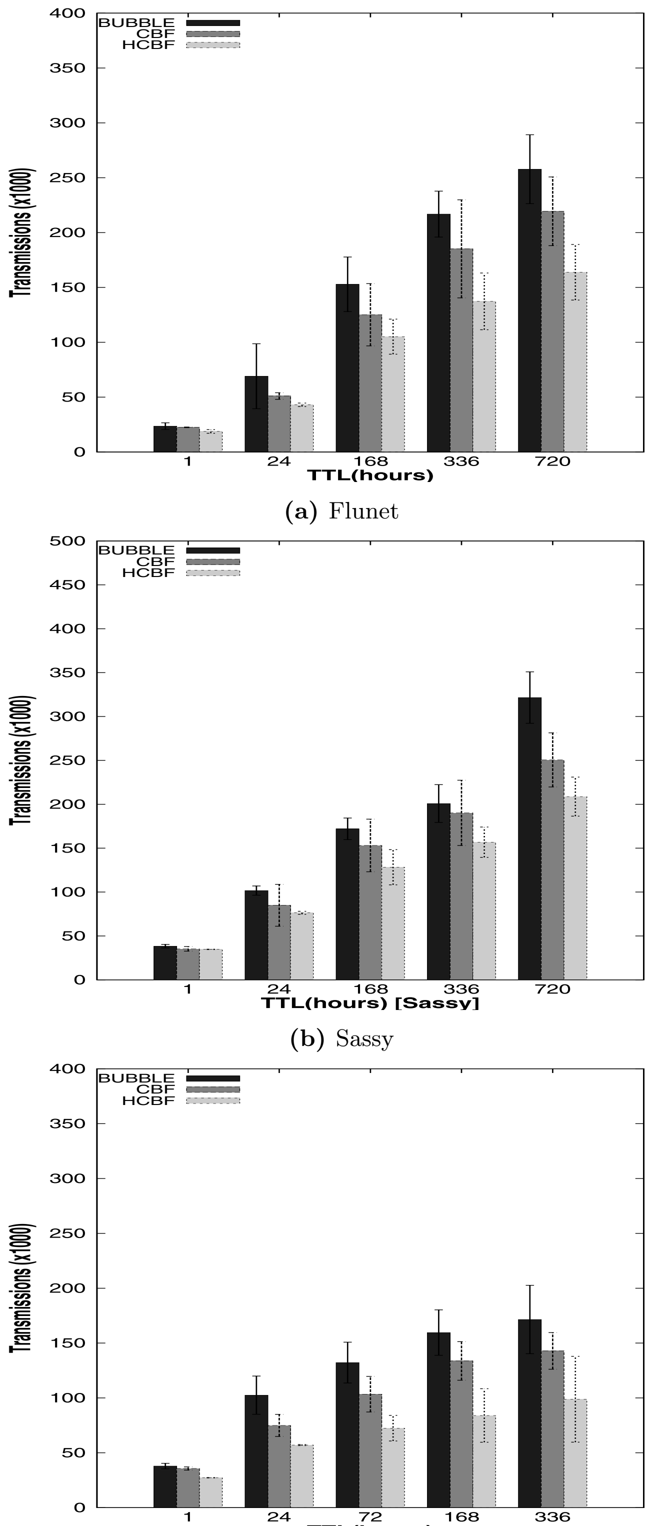


(b) Sassy



(c) SHED1

Figure 5.7: Latency: Limited TTL (Inter-community Messages)



(a) Flunet

(b) Sassy

(c) SHED1

Figure 5.8: Transmissions: Limited TTL (Inter-community Messages)

terms of delivery ratio, transmission, packet drop and delay CBF performs better when TTL is within one fourth (7 days) to one sixth (15 days) of simulation period. Beyond this range, performance gain of CBF degrades.

5.4.2 Case 2: Nodes with Limited Buffer Capacity

The last experiment for examining performance difference in inter-community message forwarding between HCBF and CBF is the experiment with nodes having limited buffer. Those nodes that are highly mobile and have high social ranking are defined to be social mobile (SM) nodes. As a result all other nodes want to pass their messages to them. During limited buffer capacity environment, most SM nodes become potential points of packet drops due to their constrained low buffer. This changes are shown in Figure 5.9. However, with increase in buffer size, overall performance gap increases slightly. This is because congestion inside social nodes reduces hence their performance as well as delivery ratio of the entire network improves. The increment in delivery ratio is not proportionate to the increment in buffer size. This points to the fact that in inter-community routing SM nodes comparatively leverages performance of those messages brought forward by CBF to reach destination faster and in fewer hops more than helping additional new messages find the destination node. Under all conditions, HCBF's performance is above that of CBF. HCBF exceeds CBF by 0.5% to 2.88% for Flunet, 2.21% to 3.39% for Sassy and 2.43% to 3.63% for SHED1.

The major improvement of HCBF, brought by social mobile nodes, is not delivery ratio, rather it is the gain in delay. In general, with increase in buffer capacity, older messages can be stored and delivered to respective destinations. This is the reason why in general for all three algorithms increase in buffer gives an increase in delay. Use of higher LP helps HCBF to have the advantages of CBF and on top use of diverse nodes as carrier (i.e. UI) along with LP in intra-community forwarding helps to avoid social yet very stationary nodes. These are the major scenarios where messages can get stuck until destination is met and if the destination node is not encounter by the social stationary node then in a resource constrained environment the messages are dropped out from the network. By avoiding them, HCBF has lower delay than CBF and when sufficient buffer capacity is provided, its delay reaches that of BUBBLE or becomes even less. Results of delay is shown in Figure 5.10. HCBF saves 17.01% to 23.13% latency in Flunet, 21.93% to 33% latency in Sassy and 14.18% to 43.93% latency in SHED1 than CBF.

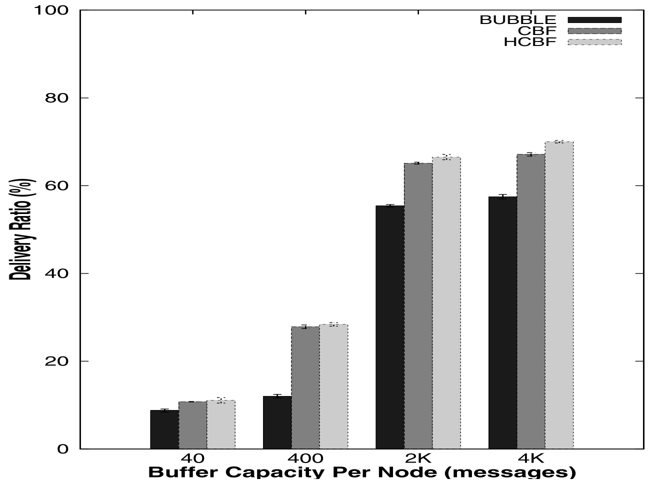
The next experiment evaluates transmission cost. The result from this experiment is shown in Figure 5.11. Under all conditions, by having to satisfy more social network analyzing metrics, HCBF transmits fewer packets within a community than CBF. The HCBF algorithm is designed such that a suitable carrier has to be more social but at least as mobile as the current carrier. This restriction automatically cuts down available next-probable carriers, thus helping nodes to strictly avoid unnecessary transmissions. With increment in buffer size, SM nodes can transfer more messages quicker and further toward their goal. As more messages are generated, transmission cost is further reduced, increasing performance gap between CBF and HCBF. In the case of Flunet 10.2% to 20.3% fewer messages, in case of Sassy 12.05% to 17.08% fewer messages and in case of SHED1 28.38% to 32.69% fewer messages are transmitted. This ranges do not include buffer size smaller than 0.01% as such small buffers are unrealistic but were included for the purpose of completion of experiment.

However, all these advantages come at the cost of higher rate of packet drops. Figure 5.12 depicts that nodes using HCBF in Flunet drops at most 12.07% more messages, at most 13.18% more messages in Sassy and at most 5.31% more messages in SHED1 than CBF. Nodes in SHED1 drop the least number of packets because of larger number of nodes having higher value of UI. Under all conditions, CBF has fewer packet drops than HCBF, but the inherited advantage from CBF allows HCBF to have fewer drops than BUBBLE. Despite such high packet drops, when there is about 10% buffer capacity or more available, with the help of SM nodes messages are delivered faster and in fewer exchanges thus improvement in delivery ratio of HCBF persist over CBF and BUBBLE. This also means that at the end of the simulations, nodes under HCBF have fewer undelivered messages left in their buffers than when they use CBF.

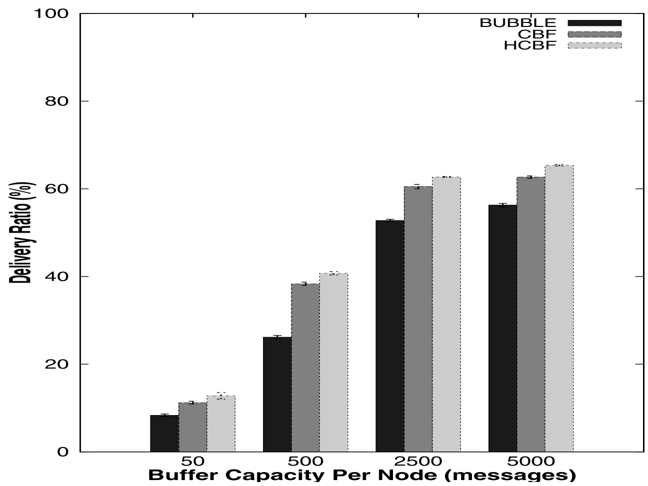
During this experiment, to get an insight on improvement in transmission cost of HCBF over BUBBLE, inter-hops and intra-hops have been traced and post-processed. For all delivered messages, CBF has either intra-community routing or inter-community routing in common with the other two algorithms thus it has been avoided in this comparison. For performance benchmark, messages are routed using the Minimum-cost Oracle algorithm and similar information is recorded. Figure 5.13 shows the distribution of hop types.

Since messages are performing inter-community forwarding and Minimum-cost oracle helps them to find the shortest route, under all three datasets there exist cases when the oracle's minimum intra-hop count is zero. As a result in Figure 5.13, all the datasets Oracle's intrahop count starts from 0. In Flunet, there are more messages that take shorter path than the median intra-hop value when they use HCBF than when messages are routed using BUBBLE. Moreover, messages having path length greater than median intra-hop, under HCBF need at least 2 hops less than when they use BUBBLE. HCBF allows range of intra-hop of messages to be shorter and also allow messages on difficult paths to have fewer hops. In case of messages in Sassy, range of intra-hops need by messages using HCBF is at least three hops smaller than the range of intra-hop when they use BUBBLE. Median value of HCBF's intra-hop count is also lower than BUBBLE's. When compared to other datasets, messages in SHED1 using HCBF have the tightest intra-hop distribution pattern with the shortest range. This is because SHED1 has more nodes with higher UI values. Due to community stability in SHED1, the intra-hop performances of both algorithms are similar for those messages that take fewer than the median number of hops but the difference in performance is made by those messages whose number of intra-hop requirement is higher than the median intra-hop value. Also HCBF, with same extreme messages taking fewer hops, has a shorter range difference than BUBBLE in SHED1. In general for all datasets, as both BUBBLE and HCBF depend on social nodes irrespective of their mobility, it is the messages which have path inside community longer than the median whose transmission makes the difference on the performance measures.

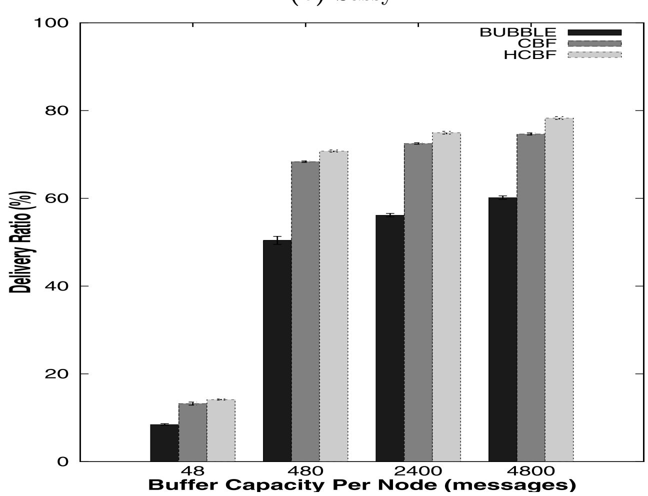
As seen from Chapter 4, using CBF for inter-community forwarding reduce the number of hops by at



(a) Flunet



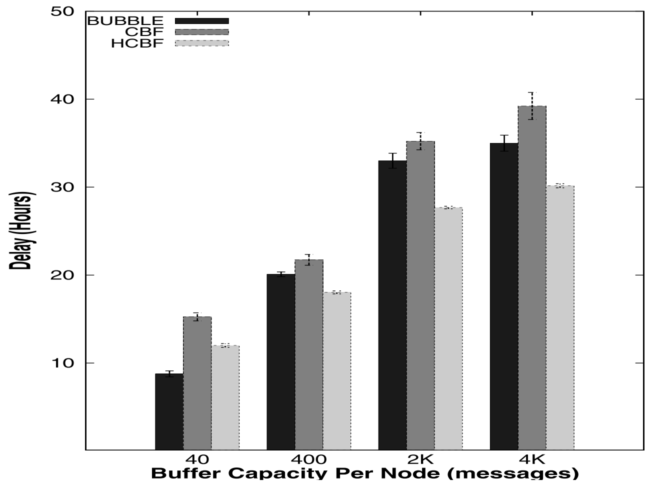
(b) Sassy



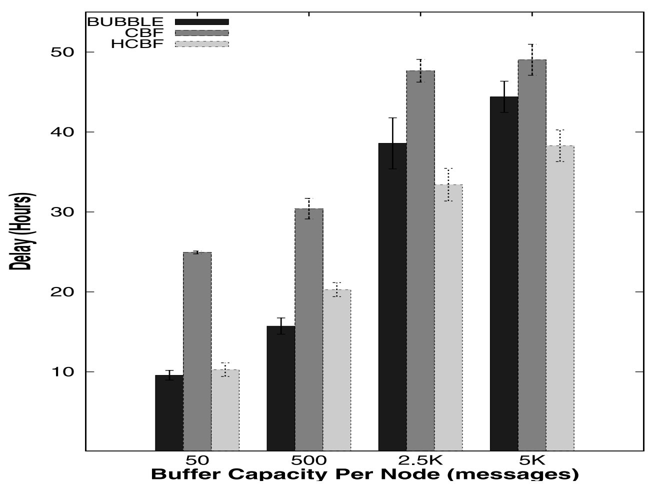
(c) SHED1

Figure 5.9: Delivery Ratio: Limited Buffer (Inter-community Messages)

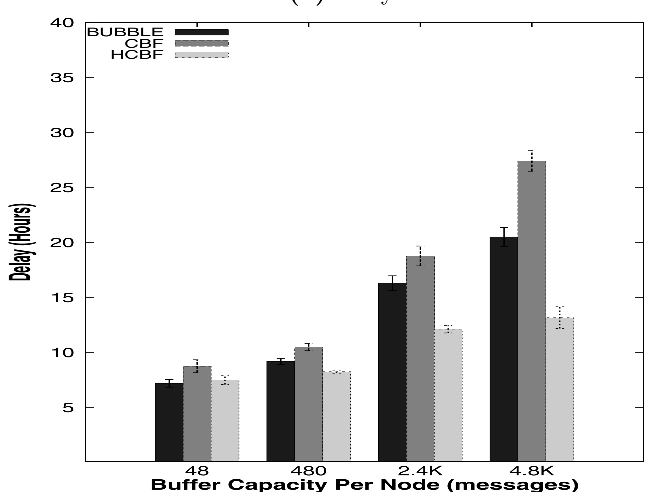
]



(a) Flunet

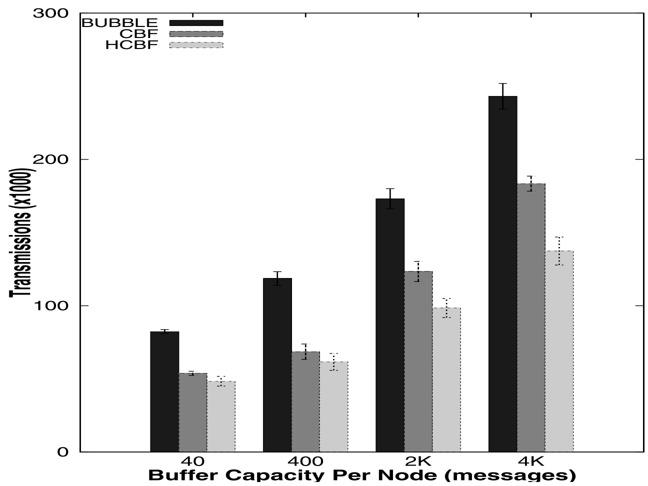


(b) Sassy

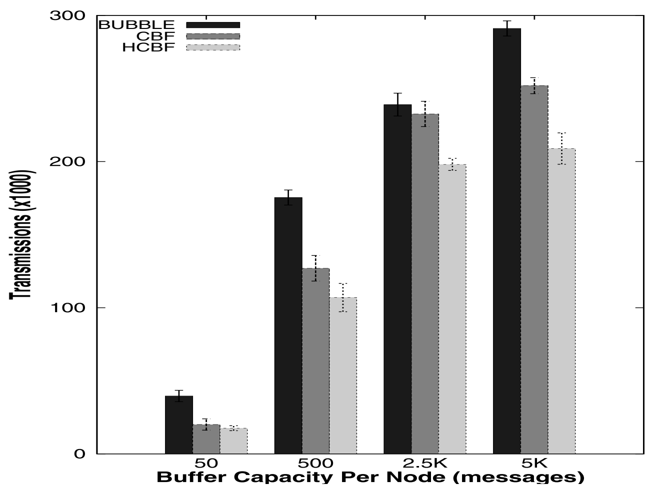


(c) SHED1

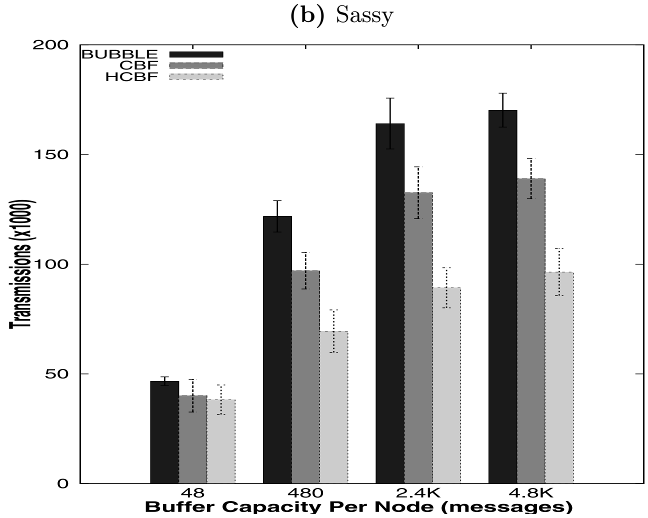
Figure 5.10: Latency: Limited Buffer (Inter-community Messages)



(a) Flunet

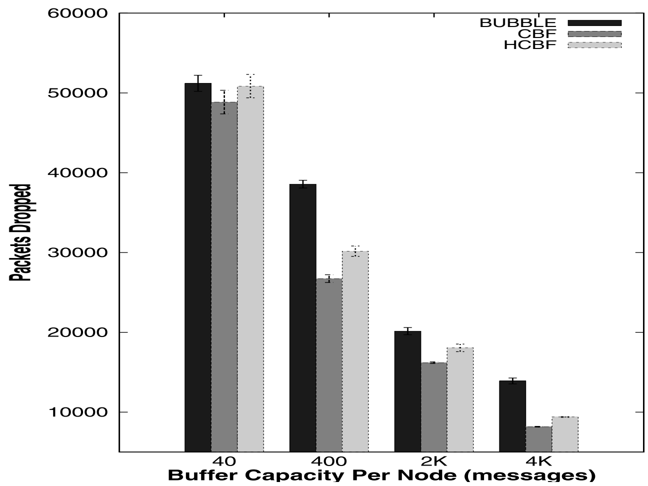


(b) Sassy

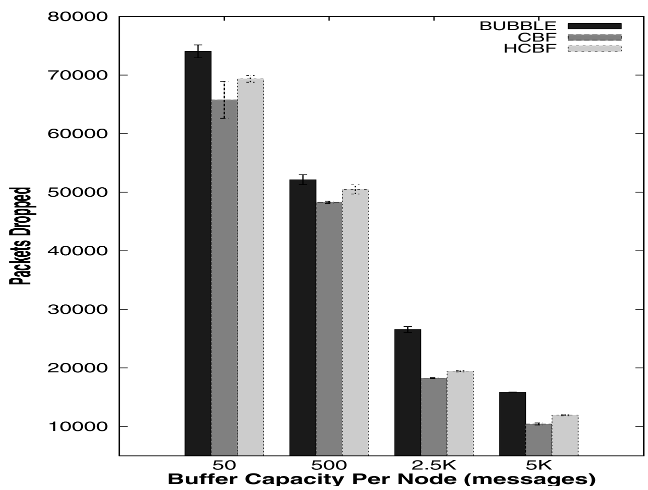


(c) SHED1

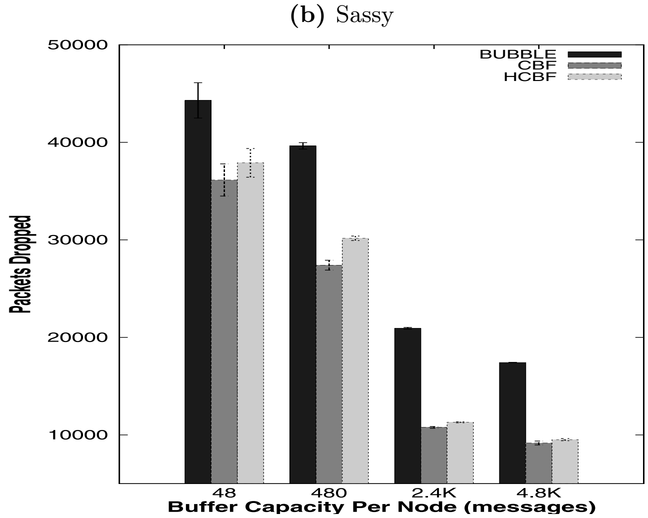
Figure 5.11: Transmissions: Limited Buffer (Inter-community Messages)



(a) Flunet

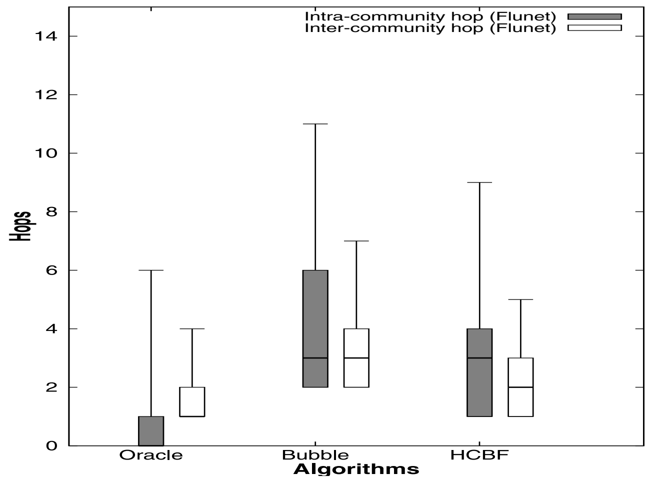


(b) Sassy

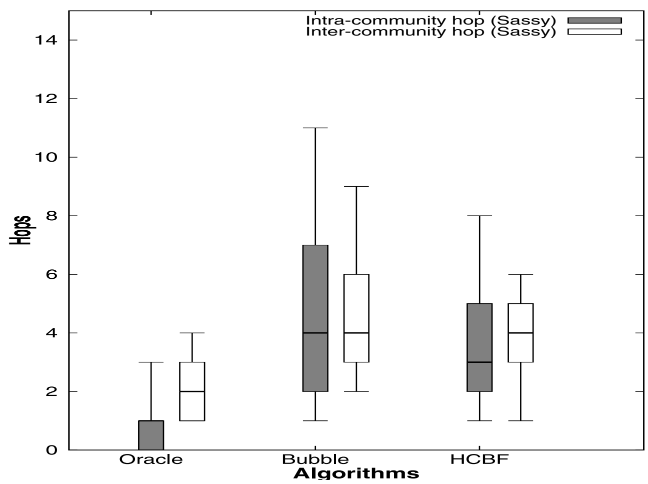


(c) SHED1

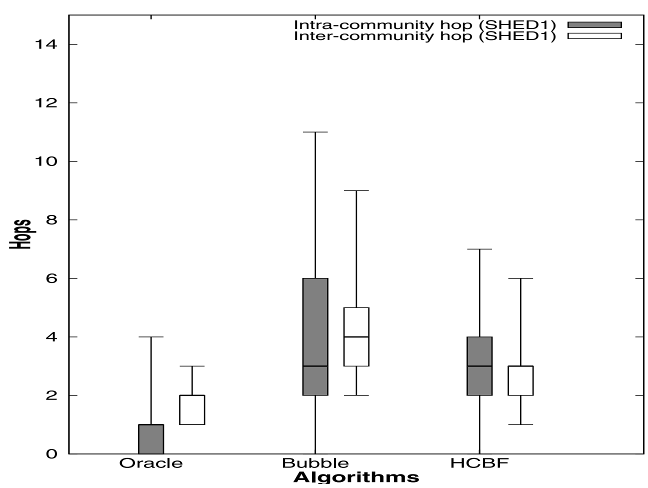
Figure 5.12: Packets Dropped: Limited buffer (Inter-community Messages)



(a) Flunet



(b) Sassy



(c) SHED1

Figure 5.13: Hop Type: Limited Buffer (Inter-community Messages)

least one compared to BUBBLE. This is also depicted by the inter-community hops in Figure 5.13, where due to the stable dataset SHED1 performs the best, followed by Flunet and due to fragile communities Sassy is tailing at the end. In all cases, intra-community routing of HCBF out-performs that of BUBBLE. In all datasets, it is not the intra-hops of HCBF which brings in the major difference, rather it is the conservative nature of CBF which strongly controls inter-community hops, allowing overall transmission cost to reduce.

5.5 Inter-community and Intra-community Routing

In this section, performance of HCBF is compared to that of BUBBLE for both inter and intra community message forwarding. Messages are no longer forced to be destined for other communities. This restriction is removed so communications within community along with outside communities can be evaluated. Under such conditions, a set of experiments in both resource-rich and resource-constrained environment are carried out and comparisons between HCBF and BUBBLE are presented and analyzed below.

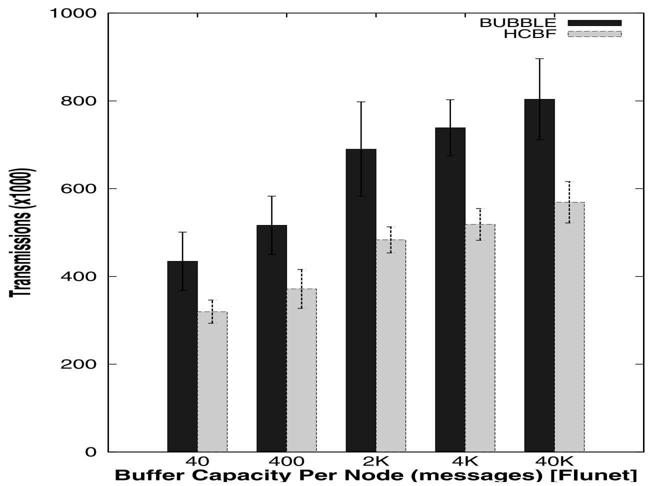
5.5.1 Case 1: Unlimited Resource

In order to compare HCBF's performance to BUBBLE, the first set of experiments are done in an ideal condition. For first experiment, messages were generated in unlimited resource environment and performance metrics were recorded. Like previous experiments in Chapter 4, the TTL value for messages was set to six months for Sassy and Flunet and messages in SHED1 had TTL set to three months. These values of TTL are double the period of simulation, thus can be treated as an unlimited resource. In this environment, 40000 messages for Flunet, 50000 messages for Sassy and 48000 messages for SHED1 were generated and their delivery ratio, latency, total transmission, total packet drops by nodes with varying sizes of buffer capacity for each node ranging from 10% to 100% of message generated were recorded.

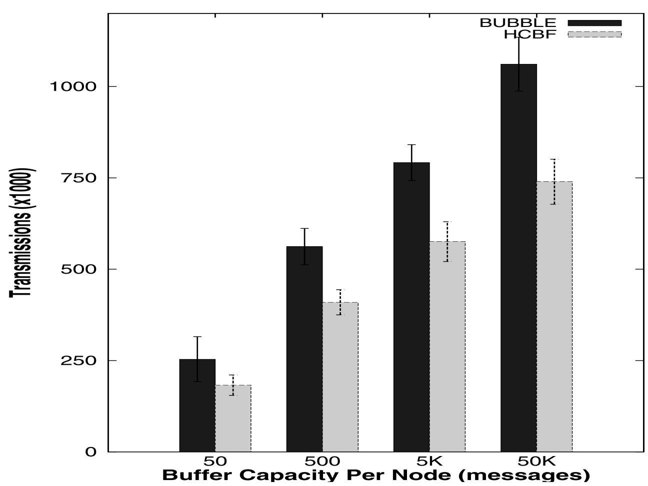
During this experiment, due to the use of SM nodes, messages are transferred with fewer hops than CBF and even fewer than BUBBLE. The results when compared to BUBBLE are shown in Figure 5.14. With increment in buffer, the performance difference increases. This is because SM nodes can contribute better with higher buffer capacity. HCBF saves 26% to 29% hops in Flunet, 27% to 30% hops in Sassy and 27% to 36% hops in SHED1. If the dataset is more stable, CBC and NCF can be used more accurately for message forwarding.

Along with transmissions, HCBF can also save on message latency in intra-community messages. With increase in buffer, delay of both algorithms increase as older messages can be transferred successfully. The results are shown in Figure 5.15. If 1% or more buffer is provided, HCBF outperforms BUBBLE in terms of delay. In Flunet, HCBF has up to 6.1% less latency, 3.4% less latency in Sassy and 12.78% less latency for SHED1 than BUBBLE.

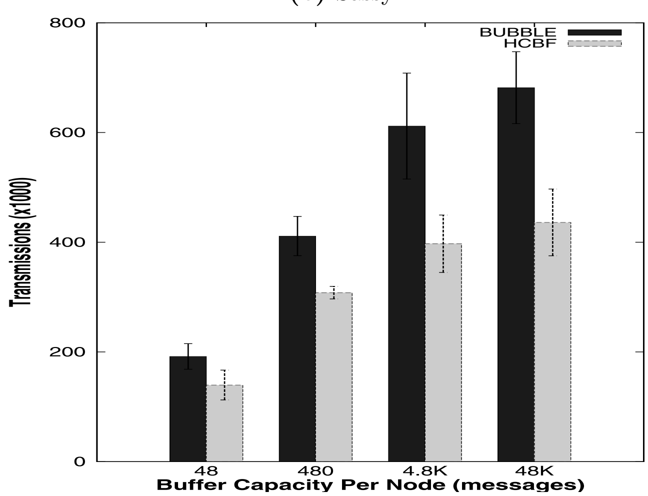
In the next experiment, performance in terms of delivery ratio of HCBF is compared to that of BUBBLE. At smaller buffer sizes, TTL of messages as high as 6 months and 3 months create unnecessary traffic in



(a) Flunet

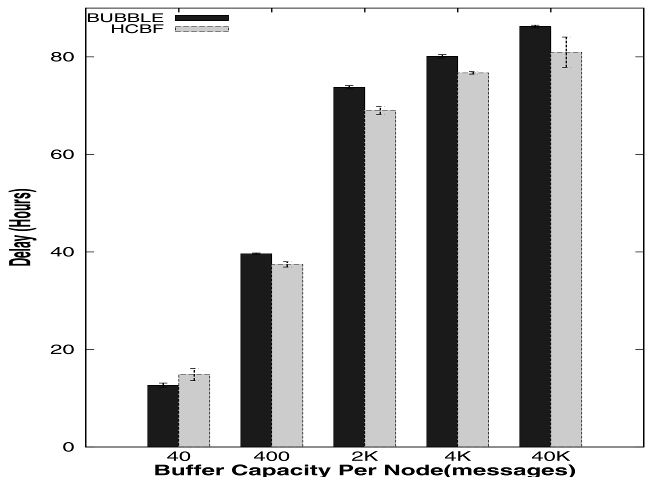


(b) Sassy

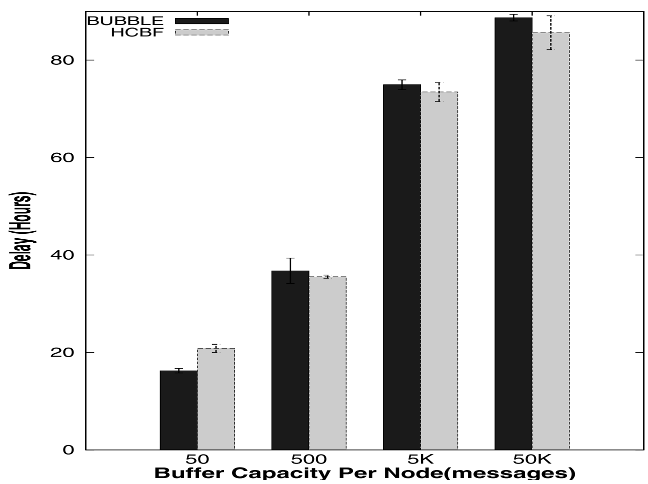


(c) SHED1

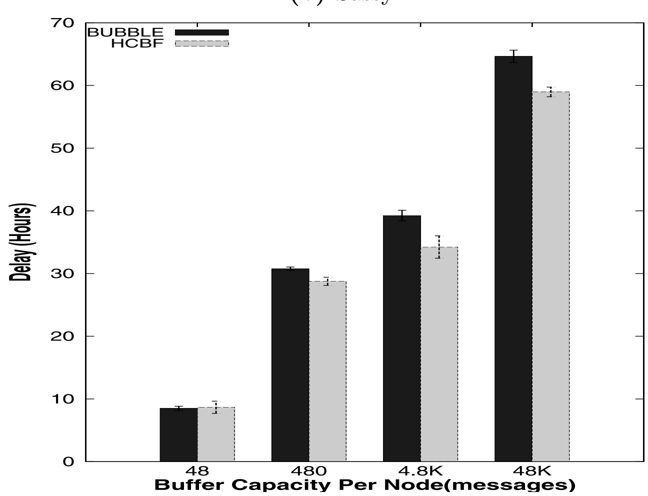
Figure 5.14: Transmissions: Unlimited Resource (Mixed Messages)



(a) Flunet



(b) Sassy



(c) SHED1

Figure 5.15: Message Latency: Unlimited Resource (Mixed Messages)

network, thus not much performance difference between CBF and HCBF is observed. Moreover nodes in BUBBLE can utilize the longer life span in their greedy approach to gain higher delivery ratio. At higher buffer capacity, CBC, NCF and UI work together to overcome congestion due to unlimited TTL and thus create performance improvements. The results are shown in Figure 5.16. The gap in delivery ratio ranges from 9.98% to 15.02% in Flunet, 6.05% to 8.59% in Sassy and 15.39% to 19.95% in SHED1. This increment is because of HCBF's ability to explore and encompass more features of small world network into its forwarding scheme than BUBBLE.

Unexpectedly, even after such savings in transmission and latency, delivery ratios of HCBF is not much higher than BUBBLE's. This is because in the resource-rich environment, messages with lives as long as 3 month to 6 months create congestion in the most social mobile nodes, thus leading to increased packet drops, which turn do not let delivery ratio to raise to the level it would have been otherwise. The results for packet drop shown in Figure 5.17. At 1% buffer, packet drops of both algorithm are very close, but after that with increment in buffer capacity, gap between the two algorithms regarding packet drop increases. We can ignore the scenario at buffer capacity as low as 1% and state that with larger buffer SM nodes come to their full potential so HCBF performance improves. HCBF drops up to 41% fewer packets in Flunet, 38% fewer packets in Sassy and 32.23% fewer packets in SHED1 than BUBBLE.

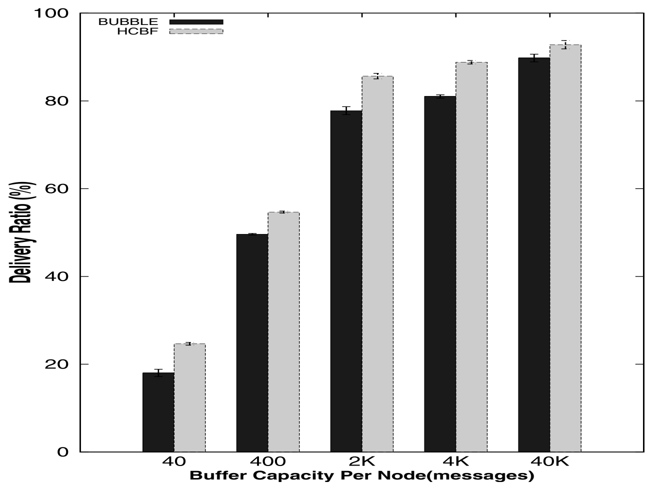
5.5.2 Case 2: Nodes with Limited Buffer Capacity

The next set of experiments are carried out when nodes have limited buffer capacity. Approximately 50000 messages with TTL of 15 days (or 7 days for SHED1) were generated and different buffer sizes were used to see how the performance of the routing algorithms varied. Higher buffer capacity reduces congestion and saves packets from dropping out of network due to buffer overflow.

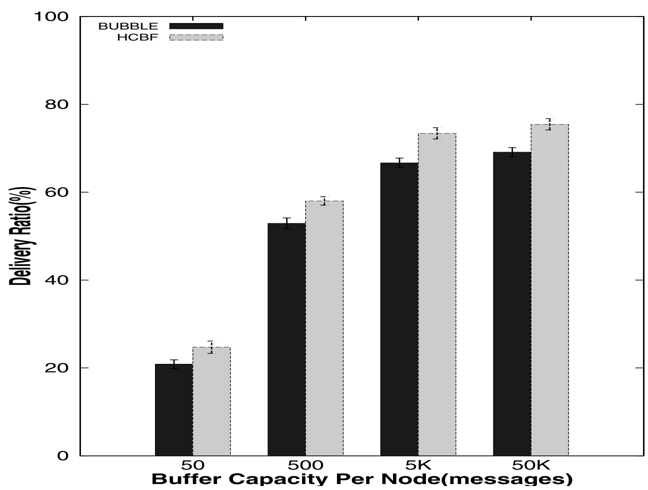
In Figure 5.18, HCBF delivers up to 15.02% more messages in Flunet, 8.59% messages in Sassy and 22.28% more messages in SHED1 than BUBBLE. In all cases, in terms of packet delivery ratio HCBF outperforms BUBBLE. It has been already separately shown in previous sections that HCBF saves on transmission in both Intra and Inter-community routing. The result for limited buffer capacity of nodes is shown in Figure 5.19. HCBF saves 20.7% to 31.21% transmissions in Flunet, 16.3% to 19.9% transmissions in Sassy and 21% to 30% transmissions in SHED1. Behind such success of energy efficiency is the contribution of conservative metrics like CBC, NCF and UI. Use of conservativeness along with social diversity helps HCBF deliver more packets to respective destination.

Due to use of diverse nature of SM nodes, with availability of 1% or more buffer, delay of HCBF is less than that of BUBBLE. For every message, HCBF saves 2.60 hours of delivery time in Flunet, 1.25 hours of delivery time in Sassy and 2.96 hours of delivery time in SHED1. This is shown in shown in Figure 5.20. Overall in limited environment situation HCBF provides better performance in terms of delivery ratio, transmission, latency and packet drop than BUBBLE.

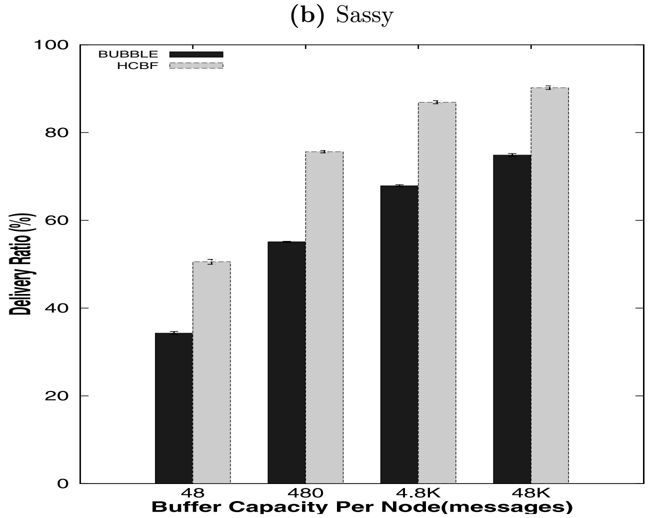
In a mixed communication type environment, HCBF has fewer packet drops than BUBBLE as a result



(a) Flunet

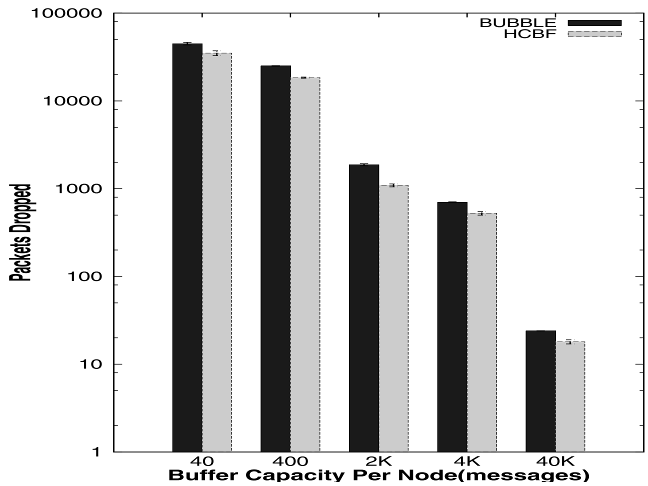


(b) Sassy

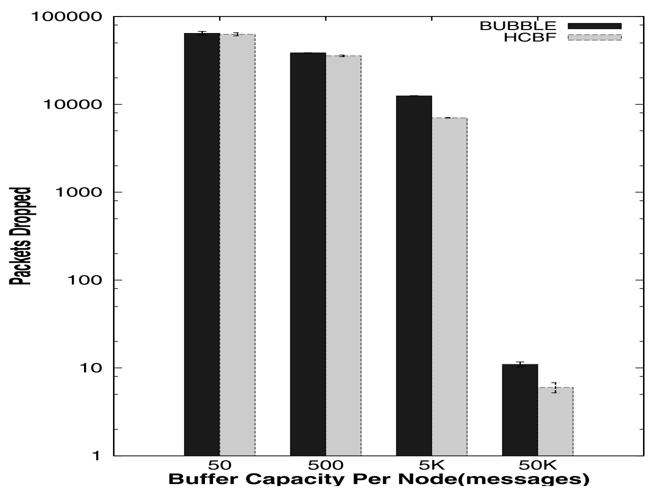


(c) SHED1

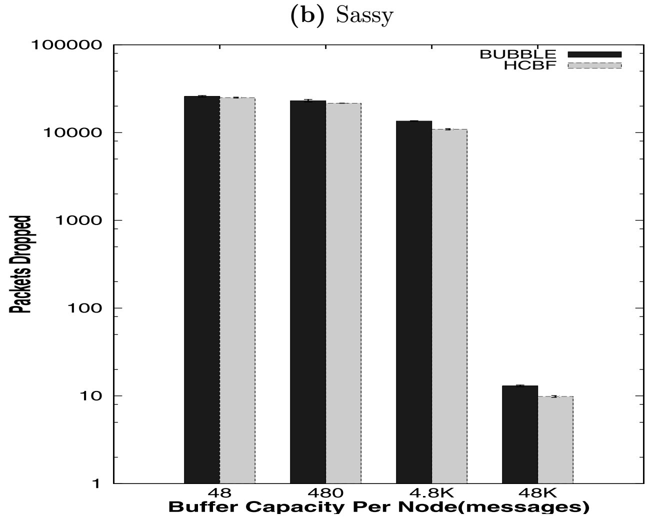
Figure 5.16: Delivery Ratio: Unlimited Resource (Mixed Messages)



(a) Flunet

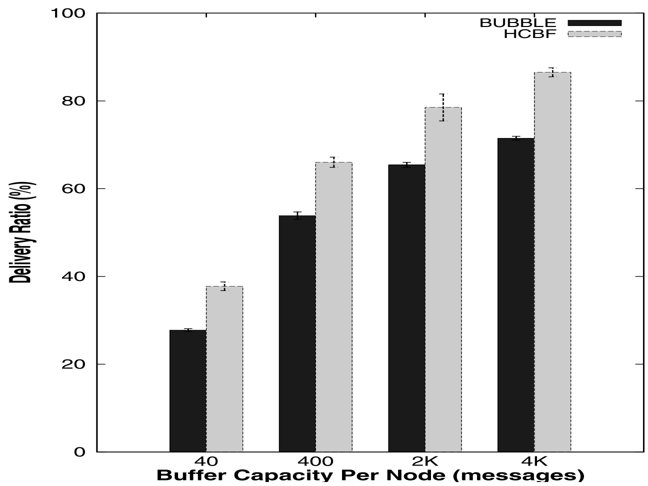


(b) Sassy

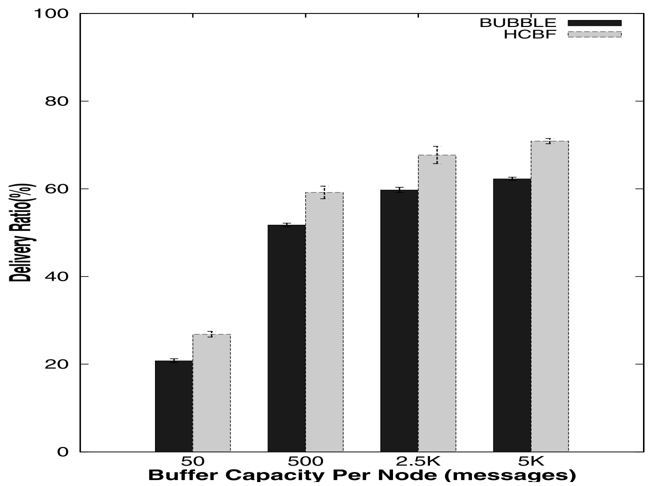


(c) SHED1

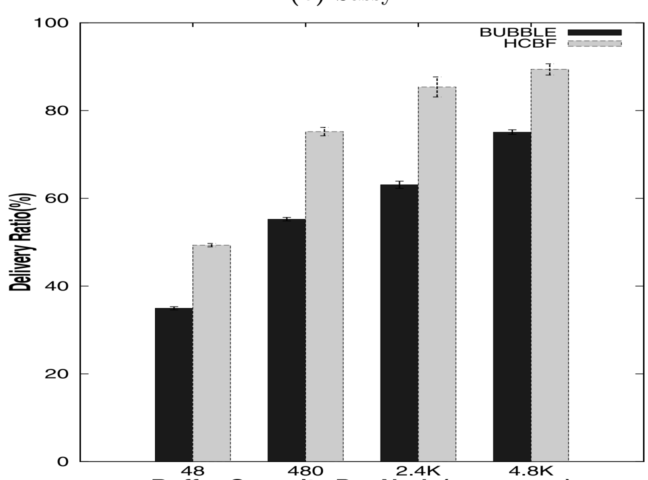
Figure 5.17: Packets Dropped: Unlimited Resource (Mixed Messages)



(a) Flunet

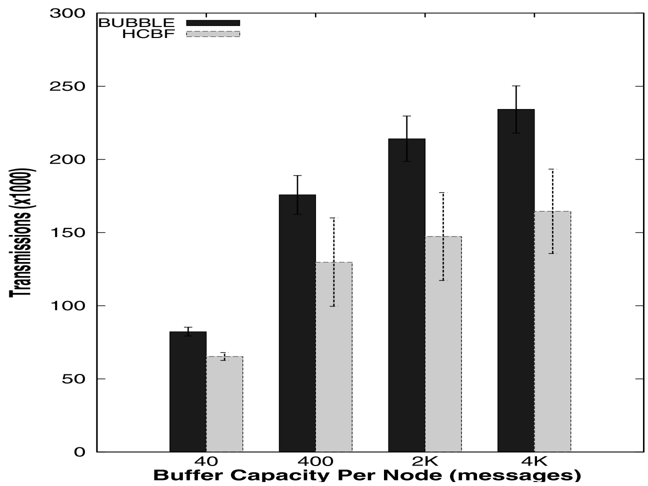


(b) Sassy

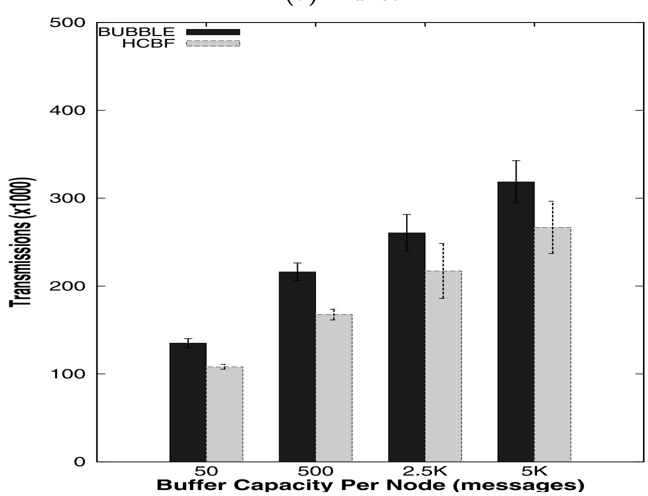


(c) SHED1

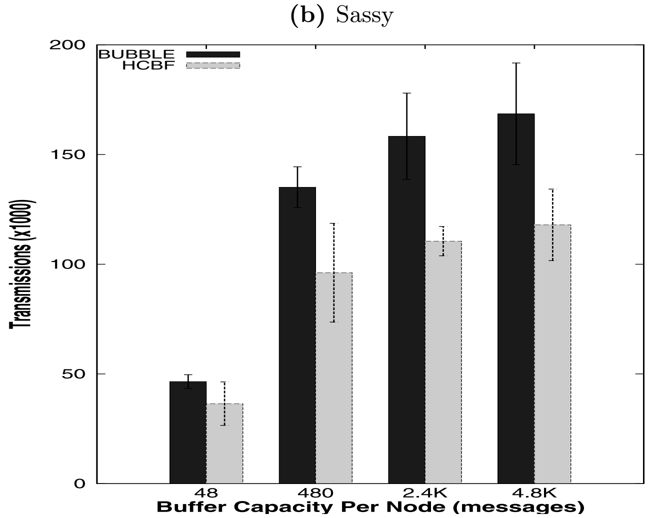
Figure 5.18: Delivery Ratio: Limited Buffer (Mixed Messages)



(a) Flunet

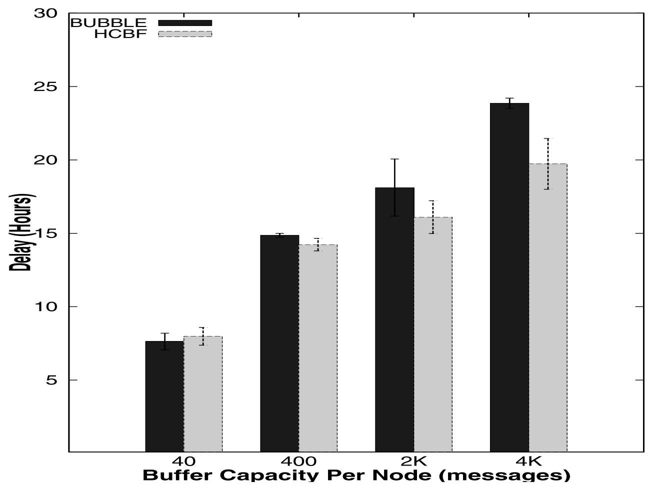


(b) Sassy

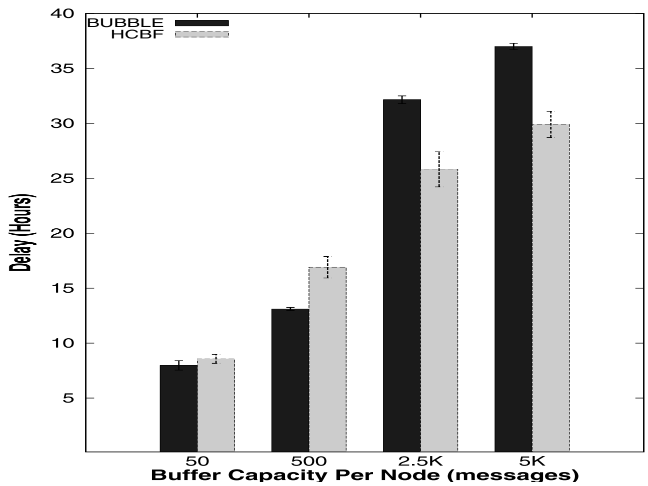


(c) SHED1

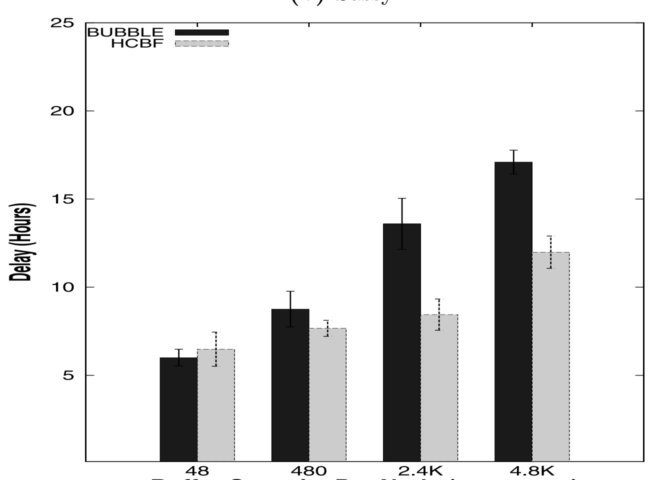
Figure 5.19: Transmissions: Limited Buffer (Mixed Messages)



(a) Flunet



(b) Sassy



(c) SHED1

Figure 5.20: Delay: Limited Buffer (Mixed Messages)

of the conservative load balancing performed by CBF. This is shown in Figure 5.21. HCBF has up to 33.06% fewer drops than BUBBLE in Flunet, 29.31% fewer drops in case of Sassy and 41.65% fewer drops in SHED1. Benefit of HCBF is its local decision making capability which is based on metrics the evaluates the underneath social structures that allows load balancing during data forwarding, thus giving it the edge of intelligent decision power in limited resource environment which, due to lack of resource, globally greedy approach of BUBBLE cannot achieve.

5.5.3 Case 3: Messages with Limited TTL

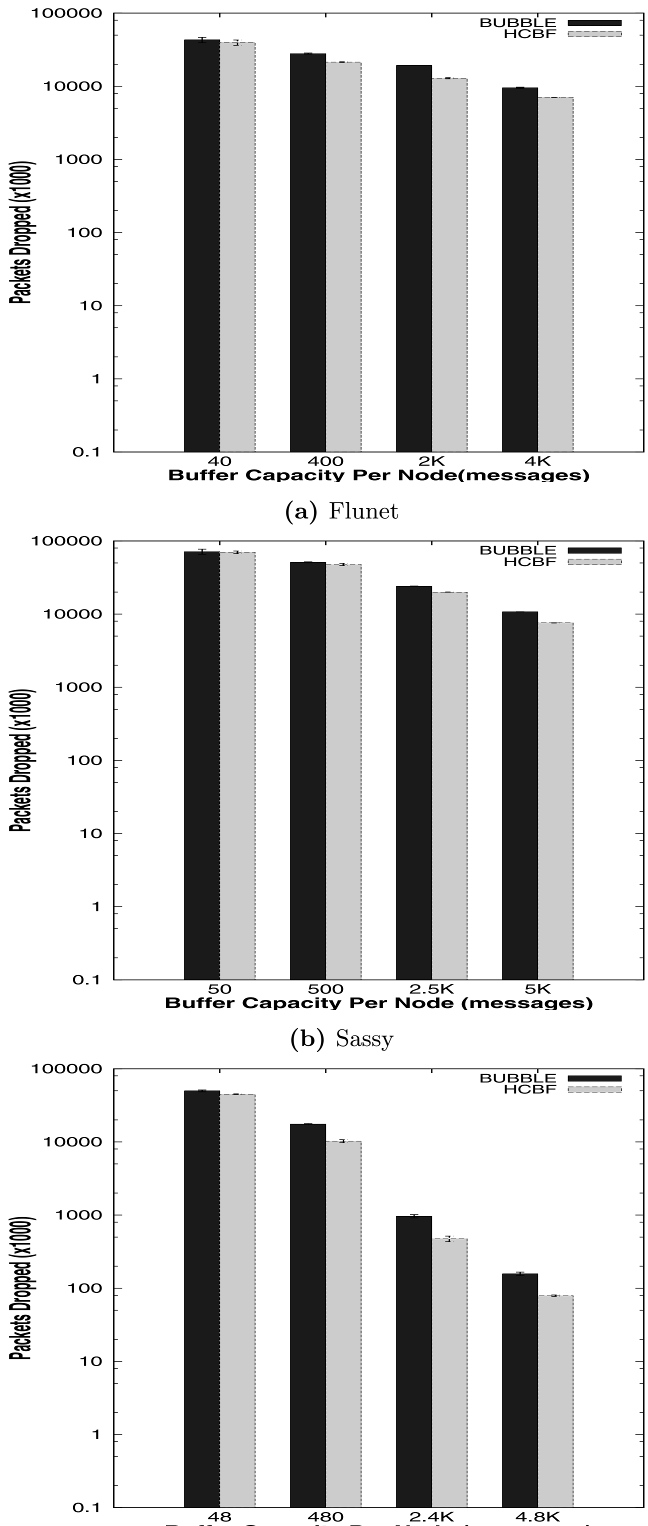
In this last set of experiments, messages have a limited life span and nodes buffer size can accommodate up to 10% of messages produced. As TTL increase, both algorithm have the chance to deliver otherwise difficult-to-deliver packets. At the same time in such scenarios, when the TTL of messages increases, the network suffers from higher congestion in both algorithms. But even in such situations, HCBF's delivery ratio exceeds that of BUBBLE by 10% to 11.56% in Flunet, 5.21% to 6.65% in Sassy and 10.50% to 15.56% in SHED1. This can be seen in Figure 5.22. The credit of increase in reliable delivery in a limited period of time goes to the CBF and NCF which balances the network load in such a way that higher number of messages can reach up to destination groups without getting dropped from the network and from then on social yet diverse nodes quickly carries them to respective destination nodes. Performance difference in delivery ratio between BUBBLE and HCBF is greatest when TTL is at 7 days.

When TTL is varied transmission in HCBF is reduced up to 31.32% in case of Flunet, 16.5% in case of Sassy and 26.88% in case of SHED1. This is shown in Figure 5.23. This reduction is because of use of social metrics, such as NCF and CBF, which prevents messages from using bad paths, hence reducing unnecessary transmission and use of UI helps message take fewer hops within community to reach the destination.

The use of UI value avoids unnecessary waiting at local maximas, thus reducing delay of HCBF of every message by 4.06 hours in Flunet, 2.85 hours in Sassy and 4.51 hours in SHED1 as shown in Figure 5.24. With increment in TTL, delay increases at a decreasing rate.

Moreover faster delivery, shorter route and load balancing helped HCBF have fewer drops. Packet drop in HCBF is at least 11.97% less in Flunet, 15% less in Sassy and 25.3% less in SHED1 respectively. This is shown in Figure 5.25. The credit of load balancing solely goes to CBF and NCF conservative nature. Over all as TTL increases, performance of HCBF degrades but in all conditions all its performance are better than BUBBLE's.

After quantitatively analyzing the six case studies in this chapter, it can be concluded that SHED1 has the best performance gain in both inter and intra-community forwarding of all three datasets. Performance gain in SHED1 is from CBC and NCF is due to SHED1's stable clusters and its gain from UI is because its nodes have the highest mobility rate. Next is the performance of Flunet. Flunet makes most of its gain from inter-community routing by using CBC and NCF. Its gain from UI is not as great as that of SHED1 because its nodes are less mobile. Last position is held by Sassy. Nodes in Sassy belong to small communities which

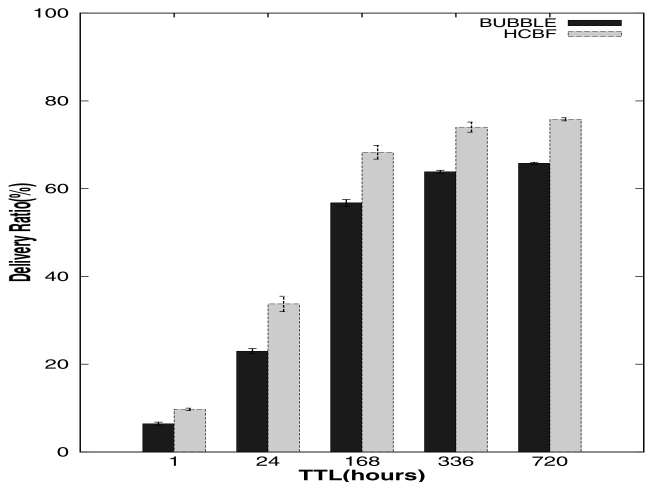


(a) Flunet

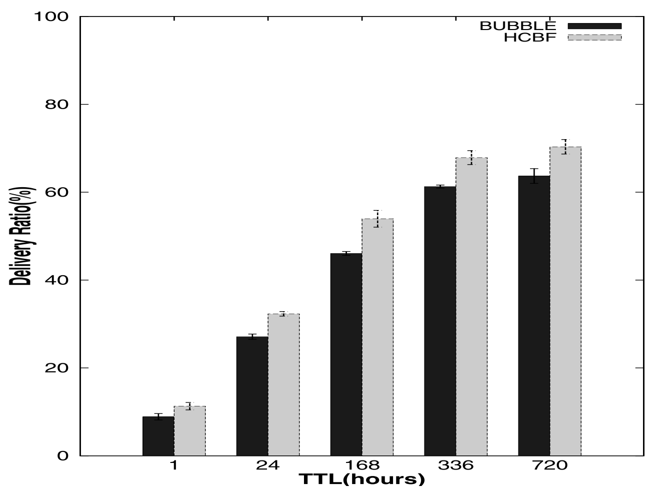
(b) Sassy

(c) SHED1

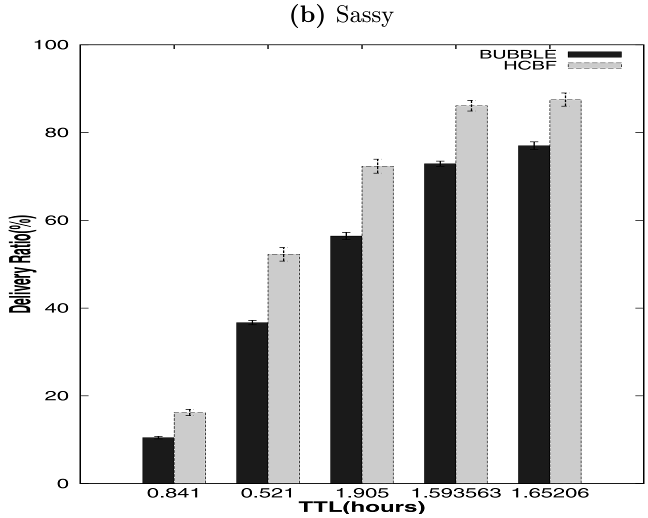
Figure 5.21: Packets Dropped: Limited Buffer (Mixed Messages)



(a) Flunet

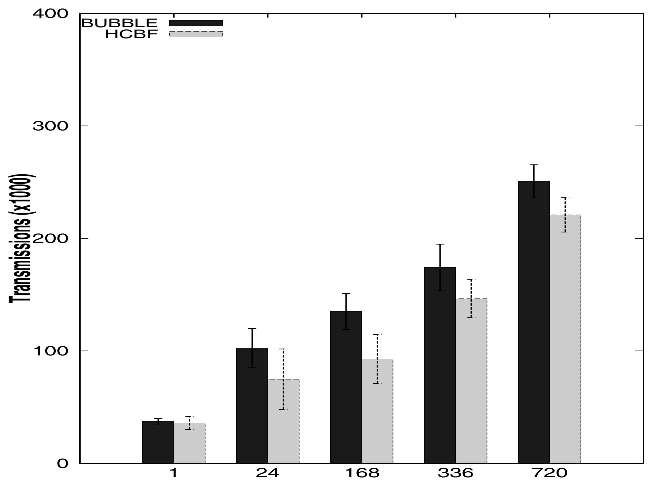


(b) Sassy

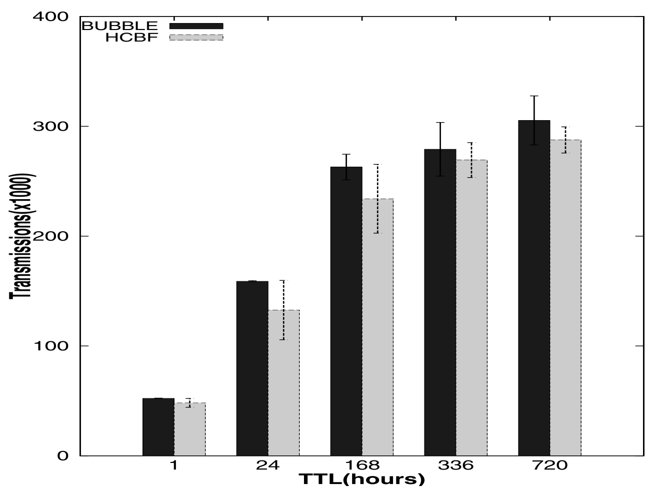


(c) SHED1

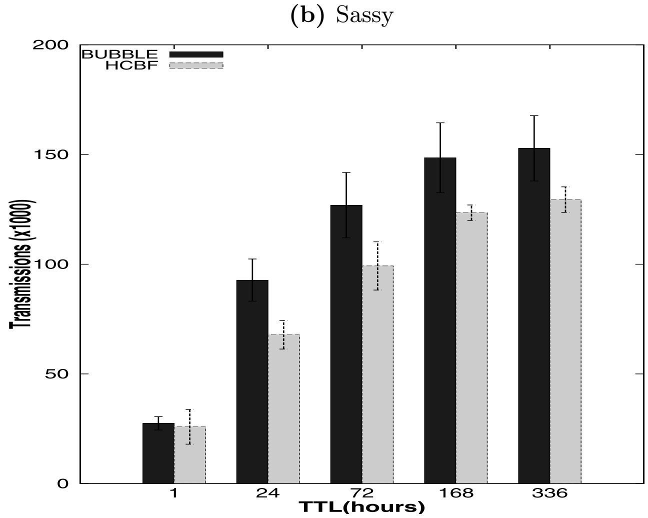
Figure 5.22: Delivery Ratio: Limited TTL (Mixed Messages)



(a) Flunet

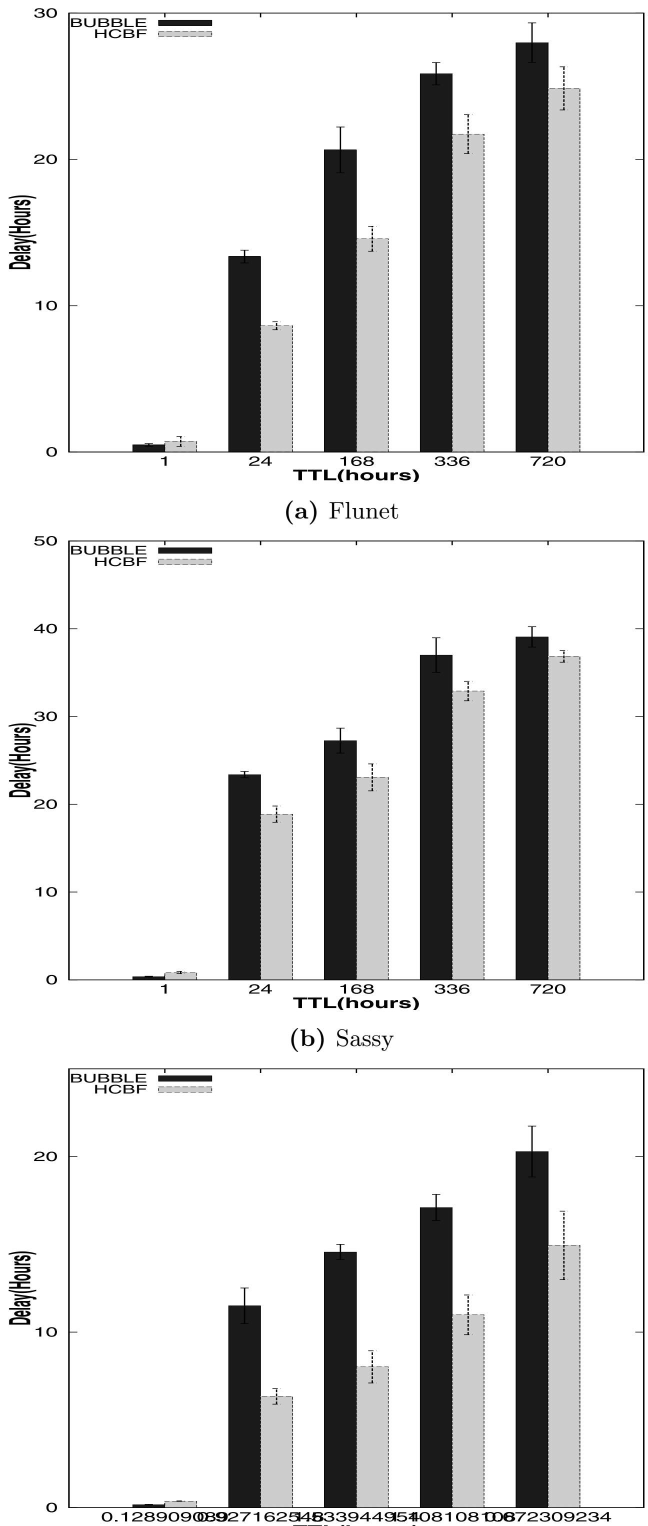


(b) Sassy



(c) SHED1

Figure 5.23: Transmissions: Limited TTL (Mixed Messages)

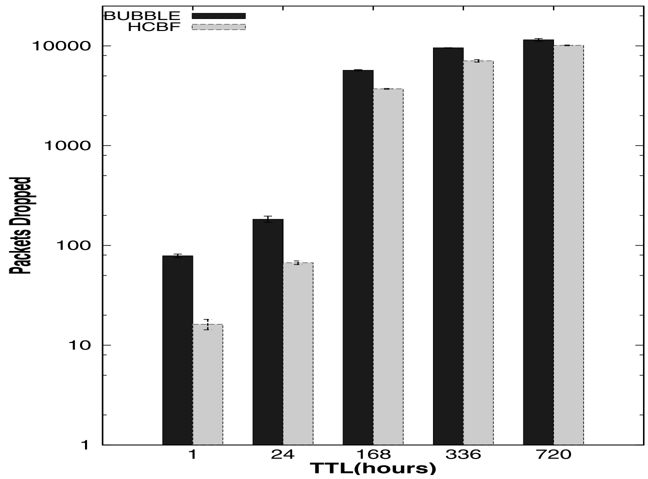


(a) Flunet

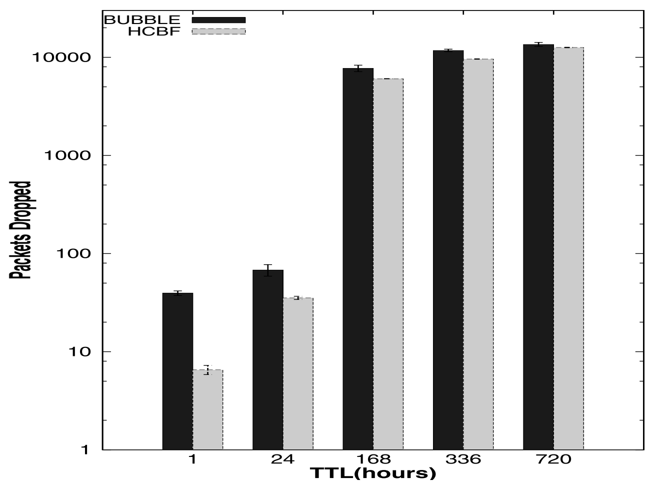
(b) Sassy

(c) SHED1

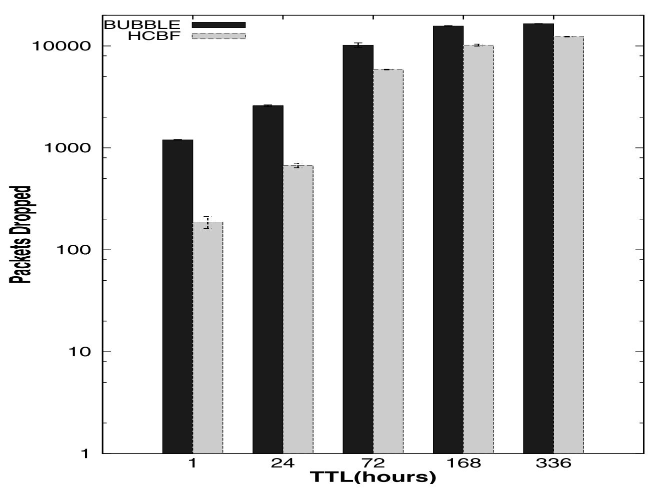
Figure 5.24: Delay: Limited TTL (Mixed Messages)



(a) Flunet



(b) Sassy



(c) SHED1

Figure 5.25: Packets Dropped: Limited TTL (Mixed Messages)

do not remain stable for long. Although instability of Sassy does not help its performance much in inter-community routing, its nodes are mobile in general, so intra-community forwarding shows good performance. In the mixed message environment, Sassy's mobile nodes help improve its performance compared to BUBBLE. Overall when contribution of HCBF, except reduction in latency, are compared to those of CBF, it becomes clear that it is CBF which is carrying the major responsibility for the making the performance difference with BUBBLE.

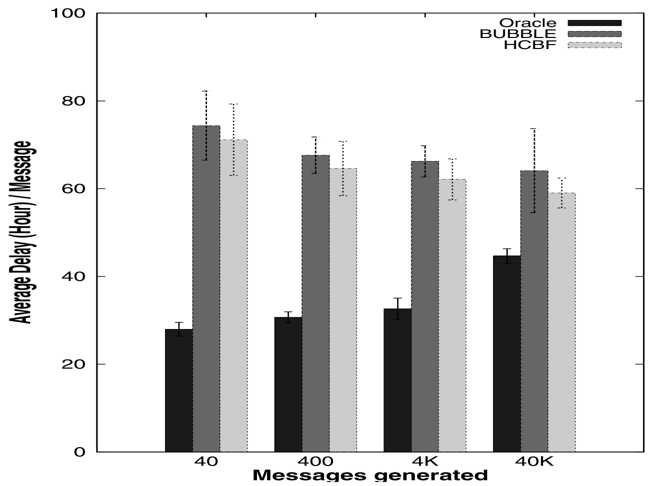
5.6 Oracles

Further analysis of HCBF and BUBBLE's performance is done by comparing them with the Oracles. For these comparison, mixed messages targeting inter and intra community routing were produced. Fastest oracle experiments were done with unlimited resource so now messages of any age can reach the destination, hence with increment in number of message production, total delay increases. On the other hand in case of BUBBLE and HCBF, TTL was set to limited. As a result in such scenario, higher number of messages also meant more difficult to deliver messages getting dropped, thus total latency reduces. The Fastest Oracle's performance is far superior to either algorithm showing that there is potential scope for further development. This comparisons can be seen in Figure 5.26 in Chapter 4. HCBF reaches closer to performance of oracle by 3.86 hours in Flunet, 1.63 hours in Sassy and 3.25 hours in SHED1 than BUBBLE. This was not observed when CBF was compared to BUBBLE and Fastest Oracle in Figure 4.17.

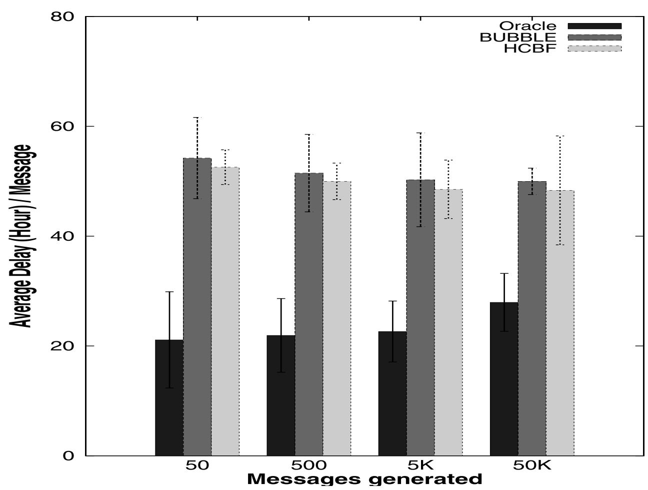
As only the number of messages were varied and all other conditions remained the same, thus the performance gap was consistent. As both the algorithms make decisions on social context, thus Figure 5.26 shows that messages in the stable community of SHED1 reaches closest to that of Fastest Oracle. But use of UI in HCBF reduces the gap further than that of BUBBLE in all three dataset.

The number of transmissions from both algorithms in comparison with the Minimum-cost Oracle is shown in Figure 5.27. In case of hop counts, in Flunet nodes using HCBF need 1.36 more hops whereas nodes using BUBBLE need 2.54 more hops than Minimum-cost Oracle. In Sassy nodes using HCBF needs 1.51 more hops where as nodes using BUBBLE need 2.16 more hops than Minimum-cost Oracle. In SHED1, nodes using HCBF need 1.25 more hops where as nodes using BUBBLE need 2.21 more hops than shortest route Oracle. In all three dataset, HCBF has closer performance to Oracle's. This is mostly because of conservative nature of CBF which is induced into HCBF.

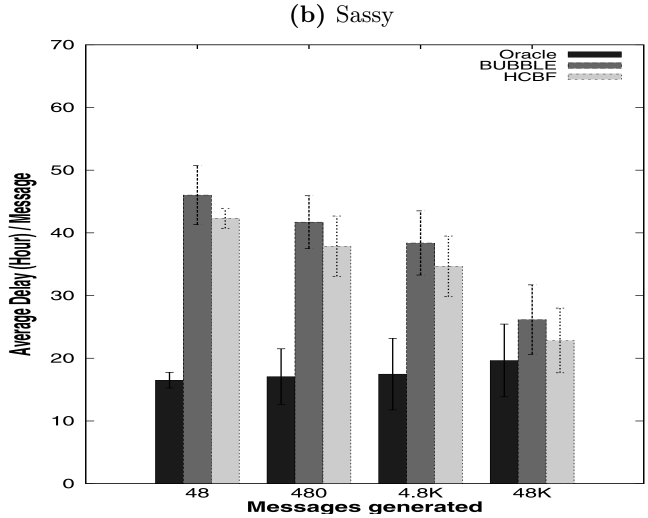
Overall when compared to the Oracles, both in terms of latency and transmission, HCBF performs closer to Oralce than BUBBLE. In case of latency, diversity of nodes in intra-community communication helps makes it a faster scheme. And in case of transmission, load balancing capability of NCF and CBC makes it energy efficiency.



(a) Flunet

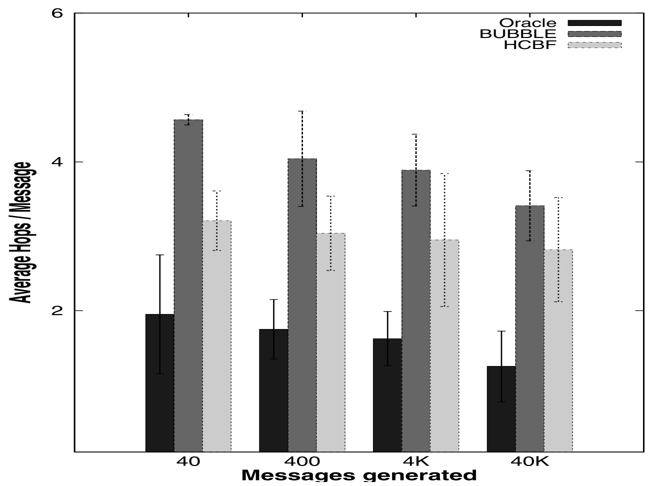


(b) Sassy

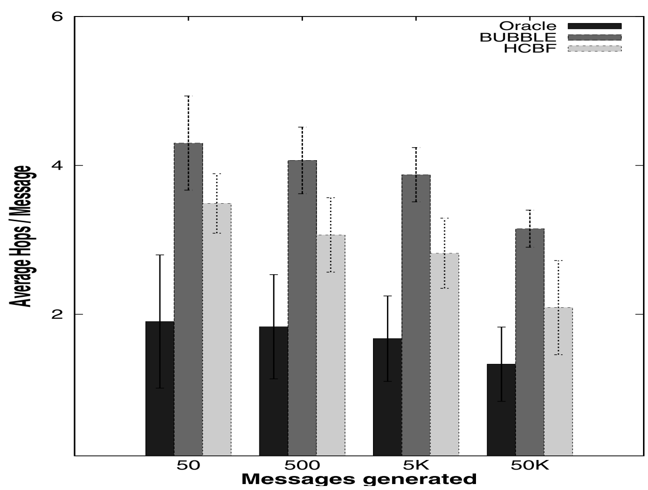


(c) SHED1

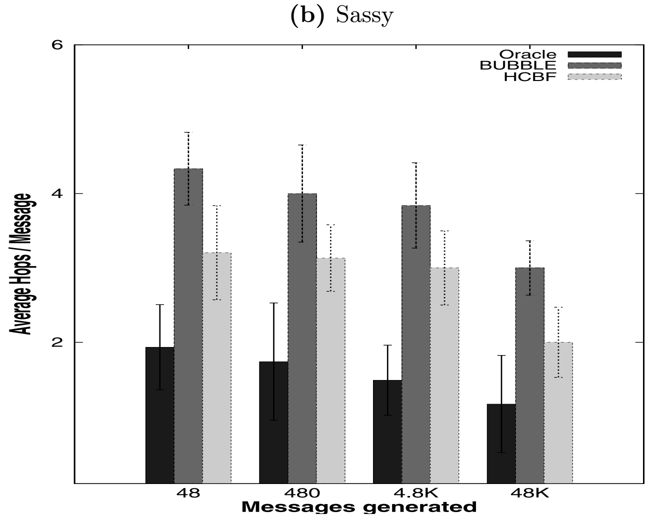
Figure 5.26: Message Latency: Fastest Oracle



(a) Flunet



(b) Sassy



(c) SHED1

Figure 5.27: Message Hop: Minimum-cost Oracle

5.7 Summary

In this chapter, a unique method for reducing delay in DTN has been introduced. This approach focuses on improving on intra-community routing as intra-community routing affects both local and global communications. So in this chapter, the CBF algorithm is further strengthened into HCBF. In HCBF, during message forwarding within community, a new metric called Unique Interaction is considered along with Local Popularity for reducing delay. CBF's goal is to be conservative and balance load with the use of CBC and NCF. HCBF further modifies CBF by reducing delay with the use of UI. Overall, HCBF is CBF with an encapsulation of mobility. Comparison with Pietilänen and Diot [64] shows that inclusion of social popularity along with diversity during forwarding reduces delay as well as keeps the routing scheme energy efficient. The experiments in this chapter confirm this observation for the datasets studied. This chapter quantitatively evaluates that when social and diverse nodes are selected, delivering packet becomes faster and more reliable as social mobile nodes have higher chance of meeting wider range of other nodes. As HCBF has more restricted criteria for node selection than CBF, its transmission cost is further reduced. In the process of depending on the social mobile nodes of a community, however, more load is given to these nodes; thus HCBF has packet drop performance that is bounded by BUBBLE from above and CBF from below. When compared to Oracle algorithms, HCBF reaches closer to Oracles' performance than BUBBLE.

In general, HCBF can attain better performance in terms of delivery ratio, packet drop, latency and transmission than BUBBLE. Not only so, unlike CBF, HCBF use is not only limited to inter-community routing. Just like BUBBLE, it can be used in in both intra and inter-community message forwarding. HCBF has shown better performance in datasets where communities were more stable. Moreover, its inclusion of socially mobile nodes allows even fragile communities like Sassy to upgrade their performance. Therefore, HCBF performs better than BUBBLE in all types of datasets used in this thesis.

Chapter 6

Conclusion

6.1 Contributions and Summary

Social networks often exhibit traits of small world networks. For efficient routing in social contexts, forwarding algorithms need to consider routing heuristics that encompass small world network properties. Moreover, in a resource constrained PSN system, particular attention must be paid to the routing algorithm's appropriate utilization of available resources. The primary contribution of this thesis are the proposed algorithms, CBF and HCBF, which use more nuanced information about the structure of the social networks arising from human mobility patterns to make more judicious forwarding decisions. As an ancillary benefit, this thesis also provides an analysis of the role of network during personal inter-connectivity in PSN systems. While well connected-people may span several communities, it is often advantageous to find members closer to the message destination which bridge relevant communities. This is not only more effective in reducing packet drop in resource constrained systems, but also reduces the asymmetric resource utilization on globally popular nodes observed in node-centric algorithms such as BUBBLE [39] and PRoPHET ([50]).

By employing datasets with varying degrees of community stability, this work has established that even under varying community membership, CBF (and by extension HCBF), can provide superior performance to BUBBLE. Obviously, there are limits to the extent to which this applies, degenerate systems, such as contact networks at immigration points where few people will ever see each other again, will require routing systems with weaker assumptions. Finally, by comparing the constrained and unconstrained cases, this work is able to quantify the degree to which resource constraints impact the performance of PSN algorithms. Severe performance degradation for greedier schemes was noted, as expected.

Specifically, on average, for inter-community messages CBF increases the delivery ratio by 10% to 15% as the number of messages and TTL were varied in the limited resource situation. As well, there was a consistent improvement in transmissions and packet drops. When buffer capacity was varied, delivery ratio and transmissions were consistently better for CBF than BUBBLE, but the difference in packet loss diminished with larger buffers. Delay was increased by as much as 38% in the limited resource conditions.

Later, with the introduction of HCBF, delivery ratio is further increased by 3% at most for both inter and intra-community messages. But delivery ratio is not the major benefit HCBF provides. It is the further reduction in transmission and the reduction in the delay of CBF to the level of Bubble or less that makes contribution of HCBF significant. In doing so, improvement in packet drop by CBF has been forgone but the current packet drop of HCBF is still less when compared to that of BUBBLE. This trade-off in packet drop indicates there may still be pivotal decision making points that can be improved and thus locating them are potentially fruitful avenues for next step.

Summary of the contributions are:

1. **Algorithm:** Two new routing algorithms have been proposed which provides superior performance to standard PSN algorithms in constrained environments.

2. **Dataset Analysis:** Quantification of the impact of community stability on routing performance on three datasets.

3. **Impact of Resource Constraints:** Quantification of the impact of constraining different key node resources on the resource utilization and routing performance.

4. **Guideline for future work:** A detail guideline for future work has been provided, which is fairly a new technique that has not been much explored in the field of PSN yet.

In conclusion, this work describes two proposed algorithms, CBF and HCBF, which attempts to balance the use of resources by adopting a more conservative forwarding scheme, and employs decision making metrics that can evaluate arising social structure of a network, in turn enabling them to preferentially choose nodes with a greater probability of being in the target community. When compared to BUBBLE, CBF's and by extension HCBF's transmission performance reaches closer to that of the Minimum-cost oracle and in all cases they transmits fewer messages, making them more energy-efficient. Quantitative analysis shows that more stable networks reap the benefits of the proposed algorithm enhancements to a greater degree. The delivery ratios of the algorithms are bounded above by BUBBLE and HCBF reduced latency to the level of BUBBLE or less.

In real life, forwarding schemes like CBF can be useful for families geographically apart to communicate with each other at a low cost. Also, CBF can be used by epidemiologists to understand the spreading pattern of epidemics, such as cholera and Flu which affects villages after villages in third world countries every year. Furthermore, the improved version of CBF, HCBF, has more versatile features and wider coverage of communication types, so it can be used in any PSNs' applications such as transferring of media files [51] and for studying organizational behavior in banking environments [60].

6.2 Limitations: Scope for Improvement

While constituting a significant contribution to the study of PSN routing, our research does have several short-comings that could be addressed in future work.

1. **Datasets:** This thesis analyzes results using three available datasets, strongly biased toward University lifestyles and relationships. Replicating these results with datasets from other populations or on synthetic data [36] could shed further light on performance and design trade offs. Also as this work is for limited resource environment, an appropriate dataset that could be used is MIT badge data set [60][1]. The MIT badge data set [60] includes information on low capacity sensors (smart badges) The practical data on a low capacity PSN can help evaluate the actual performance of CBF and HCBF as low capacity sensors is the area for which they were designed to function. Due to time constraints, this dataset could not be included in the current work.

2. **Communications Modeling:** In this work messages are assumed to be passed successfully if sent, neglecting channel noise. This is a common assumption in DTN work, as packet drops are to some extent implicitly encoded in the datasets. However, modeling communication channel noise and investigating the role of packet size as well as count would substantially extend this work.

3. **Community Definitions:** This work only investigates a single form of community generation and did not determine the impact of different community formation techniques or frequencies on the results. Looking at additional community formation techniques including other algorithms and even self-identification would shed light on CBF generalizability.

Despite these limitations, this work represents a significant contribution to the PSN literature.

6.3 Future Algorithm: Hierarchical Community Based Forwarding

Forwarding schemes like BUBBLE, CBF and HCBF have exploited emergent social structures to improve routing in social contexts. A scope that has more to offer is the way clustering is done prior to routing. Current PSNs have been treated as flat or 'almost-flat' network. There exists additional heterogeneity in human behaviour and by using that, hierarchical communities can be extracted from datasets on PSNs. One way to do this can be by merging frequently contacting sub-clusters into bigger clusters, so that the structure of the entire graph is not two level, but rather forms a tree-like structure. This section covers some initial thoughts on the way hierarchical clustering, which can be a potential scope of improving social-based routing performance, can be done. In this section, for ease of reference the word 'Cluster' refers to community of communities.

In the hierarchical routing, a community will consist of a connected graph having weakly connected clusters, which in turn are built on strongly connected individuals. Inside this communities the densely connected nodes will exist. In other word, the graph has to have minimum three tiered cluster structure.

[1]MIT Badge Dataset: http://realitycommons.media.mit.edu/badgedataset.html

Algorithm 4 Hierarchical Community-Based-Forwarding (Node carrier$_{LEVEL}$, Node [ ] en$_{LEVEL}$, Node enMaxLP$_{LEVEL}$, Node enMaxUI$_{LEVEL}$, Node enMaxNCF$_{LEVEL}$, Node enMaxCBC$_{LEVEL}$, node dest$_{LEVEL}$)

for i=1 to n **do**

  if $(en[i]_{LEVEL} == dest_{LEVEL})$ **then** //*Destination node encountered*

    $en[i].addMessageToBuf(message)$; **break** ;

  else if $((C(en[i])_{LEVEL+1} == C(dest)_{LEVEL+1})$ **and** $(C(carrier)_{LEVEL+1} \neq C(dest)_{LEVEL+1}$ **then** //*Member of destination group encountered*

    $en[i]_{LEVEL}.addMessageToBuf(message)$; **break** ;

  else if $(C(en[i])_{LEVEL+2} == C(dest)_{LEVEL+2})$ **then** //*Member of destination's higher community found*

    $en[i]_{LEVEL}.addMessageToBuf(message)$; **break** ;

  else if $(C(carrier)_{LEVEL+2} \neq C(dest)_{LEVEL+2})$ **and** $(C(en[i])_{LEVEL+2} \neq C(carrier)_{LEVEL+2})$ **and** $(BridgeNodeSet(en[i]))$ **then** //*inter-higher-community routing*

    $en[i]_{LEVEL}.addMessageToBuf(message)$; **break** ;

  end if

end for

// *Intra-community routing*

if $((C(carrier)_{LEVEL+1} == C(dest)_{LEVEL+1})$ **and** $(C(enMaxLP)_{LEVEL+1} == C(dest)_{LEVEL+1}))$ **then**

  if $(LP(carrier_{LEVEL}) < LP(enMaxLP_{LEVEL}))$ **and** $(UI(carrier_{LEVEL}) \leq UI(enMaxLP_{LEVEL}))$ **then**

    $enMaxLP_{LEVEL}.addMessageToBuf(message)$;

  end if

// *Inter-community routing*

else if $(C(carrier)_{LEVEL+2} == C(dest)_{LEVEL+2})$ **or** $(C(enMaxNCF)_{LEVEL+2} == C(carrier)_{LEVEL+2})$ **then**

  if $(C(enMaxNCF)_{LEVEL+1} == C(carrier)_{LEVEL+1})$

  and $(C(enMaxNCF)_{LEVEL+1} \neq C(dest)_{LEVEL+1})$

  and $(NCF_{[carrier_{LEVEL}][C(dest)_{LEVEL+1}]} < NCF_{[enMaxNCF_{LEVEL}][C(dest)_{LEVEL+1}]})$ **then**

      $enMaxNCF_{LEVEL}.addMessageToBuf(message)$;

  else if $(CBC_{[C(carrier)_{LEVEL+1}][C(dest)_{LEVEL+1}]} < CBC_{[C(enMaxCBC)_{LEVEL+1}][C(dest)_{LEVEL+1}]})$ **then**

      $enMaxCBC_{LEVEL}.addMessageToBuf(message)$;

  end if

//*Nodes belong to different Inter-higher-communities*

else if $(C(carrier)_{LEVEL+2} \neq C(dest)_{LEVEL+2})$ **and** $(C(en[i])_{LEVEL+2} \neq C(carrier)_{LEVEL+2})$ **then**

  if $(CBC_{[C(carrier)_{LEVEL+2}][C(dest)_{LEVEL+2}]} < CBC_{[C(enMaxCBC)_{LEVEL+2}][C(dest)_{LEVEL+2}]})$ **then**

    $enMaxCBC_{LEVEL}.addMessageToBuf(message)$;

  end if

end if

Such narrowing of focus on the hierarchical structure of the community will help to transfer messages to desired nodes in a shorter time. Adding this new advantage to those already obtained by HCBF can create an enhanced hierarchical routing algorithm for social contexts.

While considering the time frame of data collection, the structure of communities in the datasets available is dynamic. So in the tree like structure, the clusters higher up the hierarchy will be more stable than communities down the hierarchy. Communities in turn will be more stable than nodes as nodes happen to be the leaves in the hierarchical tree. In other words, as we will go down the tree of communities, we will encounter more instability. As DTN nodes' movements are dynamic, such a hierarchical tree has to be formed on every week's contact pattern. Knowing the lowest level at which leaf nodes are is important. This is because through out routing, carriers needs to be aware of the level encountered nodes to understand its authoritative power of its performance. In this suggested approach, From each level, comparison of hierarchy is done at least up to two levels from it.

Once the previous week's hierarchical clusters are known, then the most contributing nodes between clusters, located higher up the hierarchy, can be used to transfer inter-cluster messages. Once messages are inside clusters then existing schemes, like HCBF, can be used to forward messages to the destinations. If it is known that messages are generated between groups within the same cluster, then this hierarchical technique will help to keep the messages confined inside the cluster. In long run, this confinement will further help to channel messages and allow them to reach their destination using fewer number of transmission and delay. This is because by confining message inside clusters, nodes are protected from choosing derailing paths having deceptive local maximas, thus stopping message from going to unnecessary clusters. That is, it will keep messages localized within the stable cluster. Such a channeling technique and dependence on social nodes as bridges, is hoped to make Hierarchical community based forwarding a good solution for reducing delay, lowering transmission cost and increasing delivery ratio further. A proposed pseudo algorithm is provided in Algorithm 4.In this algorithm, 'X_{LEVEL} ' is showing the need of two input parameters. They are id of the node (X) and level of the tree it is positioned at (LEVEL). Here '$C(x)_L$' is a pseudo function that maps hierarchical community of node 'x' at level 'L' of the tree. The input parameters are the carrier node id and its level, destination node id and its level, the list of encountered node (encountered by the carrier node in time epoch t) having a size of 'n' along with their levels and nodes which have the maximum LP, maximum NCF, maximum CBC and maximum UI from the encountered list of nodes.

Before pursuing in the direction of hierarchical forwarding in social contexts, a large enough dataset needs to be collected in order to explore all the benefits hierarchical clustering has to offer to routing. Datasets need to be large enough so that there are enough number of nodes to form hierarchies among communities or for detecting heterogeneity among social behaviour of nodes, so that on this discrepancies the clustering tree can be built. The datasets used in this thesis are too small for such a set of 'communities of communities' to be formed.

For initial exploration purpose, sample dataset that can be helpful is the MIT Reality mining dataset [22]. The data set, MIT Reality mining dataset, has been chosen as a sample as it is three times larger than any of the current data sets. A summary of details on the dataset is included in Table 6.1.

Table 6.1: New Dataset Overview

	Dataset	
	Reality[22]	**Badge[60]**
Active Participants	97	22
Period(days)	246	20
Setting	Grads & Ugrads	Bank employee
Total clusters	Not known yet	Not known yet
Device	Nokia 6600 smart phone	Sociometric badges

References

[1] A. Telcs A. Korn, A. Schubert. Lobby Index in Networks. *In Physica A: Statistical Mechanics and its Applications*, 388(11):2221–2226, June 2009.

[2] S. Ahmed and S. S. Kanhere. Cluster-based Forwarding in Delay Tolerant Public Transport Networks. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 625–634, Washington, DC, October 2007.

[3] L.A.N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of Small-world networks. *In Proceedings of the National Academy of Science USA*, 97, 2000.

[4] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine. Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking*, pages 81–91, San Francisco, CA, September 2008.

[5] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. Exploiting Self-Reported Social Networks for Routing in Ubiquitous Computing Environments. In *Proceedings of International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 484–489, Avignon, France, October 2008.

[6] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic Clustering of Data. *In Physical Review Letters*, 76(18):3251–3254, April 1996.

[7] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding of Communities in Large Networks. *In Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), October 2008.

[8] C. Boldrini, M. Conti, I. Iacopini, and A. Passarella. Hibop: a History Based Routing Protocol for Opportunistic Networks. In *Proceedings of the 8th IEEE World of Wireless, Mobile and Multimedia Networks*, pages 1–12, Espoo, Finland, June 2007.

[9] C. Boldrini, M. Conti, and A. Passarella. Impact of Social Mobility on Routing Protocols for Opportunistic Networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6, Helsinki, Finland, June 2007.

[10] C. Boldrini, M. Conti, and A. Passarella. Contentplace: Social-Aware Data Dissemination in Opportunistic Networks. In *Proceedings of the 11th international symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 203–210, Vancouver, British Columbia, Canada, October 2008.

[11] C. Boldrini, M. Conti, and A. Passarella. Modelling Data Dissemination in Opportunistic Networks. In *Proceedings of the 3rd ACM Workshop on Challenged Networks*, pages 89–96, San Francisco, CA, September 2008.

[12] E. Bulut and B. K. Szymanski. Friendship Based Routing in Delay Tolerant Mobile Social Networks. In *Proceedings of IEEE Conference on Global Communication*, pages 1–5, Miami, Florida, December 2010.

[13] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. In *Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1 –13, Barcelona, Spain, April 2006.

[14] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot. The Diameter of Opportunistic Mobile Networks. In *Proceedings of the 3rd International Conference on emerging Networking EXperiments and Technologies*, pages 12:1–12:12, Copenhagen, Denmark, December 2007.

[15] Min Chen, Victor Leung, Shiwen Mao, Taekyoung Kwon, and Ming Li. Energy-efficient Itinerary Planning for Mobile Agents in Wireless Sensor Networks. In *Proceedings of the 2009 IEEE International Conference on Communications*, pages 5026–5030, Dresden, Germany, June 2009.

[16] A. Clauset, M. E. J. Newman, and C. Moore. Finding Community Structure in very Large Networks. *In Physical Review E*, 70(6), December 2004.

[17] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, January 1999.

[18] E. M. Daly and M. Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 32–40, Montreal, Canada, September 2007.

[19] H. Dang and H. Wu. Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Networks. *In IEEE Transactions on Wireless Communication*, 9(6):1874–1881, June 2010.

[20] L. Danon, A. Daz-guilera, and J. Duch. Comparing Community Structure Identification. *In Journal of Statistical Mechanics: Theory and Experiment*, page 09008, 2005.

[21] A. Doria, M. Uden, and P. Pandey, D. Providing Connectivity to the Saami Nomadic Community. In *Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Innovation*, pages 20–28, Bangalore, India, December 2002.

[22] N. Eagle and A. Pentland. Reality Mining: Sensing Complex Social Systems. *In Personal Ubiquitous Computing*, 10(4):255–268, March 2006.

[23] F. Fabbri and R. Verdone. A Sociability-Based Routing Scheme for Delay-Tolerant Networks. *In EURASIP Journal on Wireless Communications and Networking*, 2011:1:1–1:13, 2011.

[24] K. Fall. A Delay-tolerant Network Architecture for Challenged Internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pages 27–34, Karlsruhe, Germany, August 2003.

[25] M. Fiedler. Algebraic Connectivity of Graphs. *In Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[26] S. Fortunato. Community Detection in Graphs. *In Physics Reports*, 486(3-5):75 – 174, 2010.

[27] S. Fortunato and M. Barthélemy. Resolution Limit in Community Detection. *In Proceedings of the National Academy of Sciences*, 104(1):36–41, January 2007.

[28] L.C. Freeman. A Set of Measures of Centrality Based on Betweenness. *In Sociometry*, 40(1):35–41, March 1977.

[29] G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms and Applications*. Society For Industrial And Applied Mathematics, Philadelphia, PA, 2007.

[30] M. Girvan and M. E. J. Newman. Community Structure in Social and Biological Networks. *In Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[31] J. C. Gower and G. J. S. Ross. Minimum Spanning Trees and Single Linkage Cluster Analysis. *In Journal of the Royal Statistical Society*, 18, 1969.

[32] Z. J. Haas and T. Small. A New Networking Model for Biological Applications of Ad hoc Sensor Networks. *In IEEE/ACM Transactions on Networking*, 14(1):27–40, February 2006.

[33] M. S. Hashemian, K. G. Stanley, D. L. Knowles, J. Calver, and N. Osgood. Human Network Data Collection in the Wild: the Epidemiological Utility of Micro-contact and Location Data. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 255–264, Miami, FL, January 2012.

[34] M. S. Hashemian, K. G. Stanley, and N. Osgood. Flunet: Automated Tacking of Contacts during Flu Season. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 348–353, Avignon, France, May 2010.

[35] M.S. Hashemian and K. G. Stanley. Effective Utilization of Place as a Resource in Pocket Switched Networks. In *Proceeding of IEEE 36th Conference on Local Computer Networks*, pages 247 –250, Bonn, Germany, October 2011.

[36] T. Hossmann, T. Spyropoulos, and F. Legendre. Putting Contacts into Context: Mobility Modeling beyond Inter-Contact Times. In *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 18:1–18:11, Paris, France, May 2011.

[37] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket Switched Networks and Human Mobility in Conference Environments. In *ACM SIGCOMM Workshop on Delay-tolerant Networking*, pages 244–251, Philadelphia, PA, August 2005.

[38] P. Hui and J. Crowcroft. How Small Labels Create Big Improvements. In *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 65 –70, March 2007.

[39] P. Hui, J. Crowcroft, and E. Yoneki. Bubble Rap: Social-Based forwarding in Delay Tolerant Networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 241–250, Hong Kong, China, May 2008.

[40] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, 2011.

[41] S. Ioannidis, A. Chaintreau, and L. Massoulie. Optimal and Scalable Distribution of Content Updates over a Mobile Social Network. In *Proceedings of the 28th IEEE Conference on Computer Communications*, pages 1422–1430, Rio de Janeiro, Brazil, April 2009.

[42] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 145–158, Portland, OR, August 2004.

[43] S. C. Johnson. Hierarchical Clustering Schemes. *In Psychometrika*, 2:241–254, 1967.

[44] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical Routing in Delay-Tolerant Networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, pages 237–243, Philadelphia, PA, August 2005.

[45] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 96–107, San Jose, CA, October 2002.

[46] S.K.A. Khan, R.J. Mondragon, and L.N. Tokarchuk. Lobby Influence: Opportunistic Forwarding Algorithm Based on Human Social Relationship Patterns. In *Workshop Proceedings of the 10th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 211–216, Lugano, Switzerland, March 2012.

[47] C. Konstantopoulos, A. Mpitziopoulos, Damianos Gavalas, and G. Pantziou. Effective Determination of Mobile Agent Itineraries for Data Aggregation on Sensor Networks. *In IEEE Transactions on Knowledge and Data Engineering*, 22(12):1679–1693, 2010.

[48] F. Kopos. *Biological Networks (Complex Systems and Interdisciplinary Science)*. World Scientific Publishing Company, 1st edition, December 2007.

[49] R. Lambiotte, J. C. Delvenne, and M. Barahona. Laplacian Dynamics and Multiscale Modular Structure in Networks, October 2009.

[50] A. Lindgren, A.Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. *In SIGMOBILE Mobile Computing and Communication Review*, 7(3):19–20, 2004.

[51] L. McNamara, C. Mascolo, and L. Capra. Media Sharing Based on Colocation Prediction in Urban Transport. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pages 58–69, San Francisco, CA, September 2008.

[52] A. G. Miklas, K. K. Gollu, K. K.W. Chan, S. Saroiu, K. P. Gummadi, and E. Lara. Exploiting Social Interactions in Mobile Systems. In *Proceedings of 9th International Conference on Ubiquitous Computing*, volume 4717, pages 409–428. Springer, Seoul, South Korea, 2007.

[53] A. Mtibaa, M. May., C. Diot, and M. Ammar. Peoplerank: Social Opportunistic Forwarding. In *Proccedings of 29th IEEE International Conference on Computer Communications*, pages 1 –5, San Diego, CA, March 2010.

[54] M. Musolesi and C. Mascolo. A Community Based Mobility Model for Ad Hoc Network Research. In *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, pages 31–38, Florence, Italy, May 2006.

[55] M. Musolesi and C. Mascolo. CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks. *In IEEE Transactions on Mobile Computing*, 8(2):246–260, 2009.

[56] M. E. J. Newman. Mixing Patterns in Networks. *In PRE*, 67(2):026126, February 2003.

[57] M. E. J. Newman. The Structure and Function of Complex Networks. *In SIAM REVIEW*, 45:167–256, 2003.

[58] M. E. J. Newman. Modularity and Community Structure in Networks. *In Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006.

[59] M. E. J. Newman. *Networks: An Introduction.* Oxford University Press, 2010.

[60] D. Olguín, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland. Sensible Organizations: Technology and Methodology for Automatically Measuring Organizational Behavior. *In IEEE Transactions on Systems, Man, and Cybernetics*, 39(1):43–55, February 2009.

[61] E.C.R.D. Oliveira and C. V. N.D. Albuquerque. NECTAR: a DTN Routing Protocol Based on Neighborhood Contact History. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, pages 40–46, Honolulu, HI, March 2009.

[62] G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying Social Group Evolution. *In Nature*, 446(7136):664–667, April 2007.

[63] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking Connectivity in Developing Nations. *In Computer*, 37(1):78–83, 2004.

[64] A.K. Pietilänen and C. Diot. Dissemination in Opportunistic Social Networks: the Role of Temporal Communities. In *Proceedings of the 13th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 165–174, Hilton Head, SC, June 2012.

[65] A. Pothen, H. D. Simon, and K.P. Liou. Partitioning Sparse Matrices with Eigenvectors of Graphs. *In SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, May 1990.

[66] V. Pulimi, T. Paul, K.G. Stanley, and D. Eager. Near-optimal Routing for Contour Detection in Wireless Sensor Networks. In *Proceedings of the 37th Annual IEEE Conference on Local Computer Networks*, pages 462–469, Washington, DC, USA, October 2012.

[67] R. Ramanathan, R. Hansen, P. Basu, R. R. Hain, and R. Krishnan. Prioritized Epidemic Eouting for Opportunistic Networks. In *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, pages 62–66, San Juan, Puerto Rico, June 2007.

[68] K. Rasul, S.A. Chowdhury, D Makaroff, and K.G. Stanley. Community-Based Forwarding for Low-Capacity Pocket Switched Networks. *Submitted to Ad Hoc Networks*, 2013.

[69] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the Levy-Walk Nature of Human Mobility. In *Proceedings of the 27th Conference on Computer Communications*, pages 924–932, Phoenix,AZ, April 2008.

[70] Chen S., Zhang J., and Gao Q. Social-based Hierarchical Routing in Delay-tolerant Networks. *In Journal of Information and Computational Science*, 9(16), 2012.

[71] Islam M. S. and A. Gronlund. Digital Bangladesh: A Change We Can Believe In? In *Proceedings of the 2nd International Conference on Electronic Government and the Information Systems Perspective*, pages 107–121, Toulouse, France, August 2011.

[72] C. V. Samaras, I. Komnios, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou, and N. Peccia. Extending Internet into Space - ESA DTN Testbed Implementation and Evaluation. In *Proceeding of the 1st International ICST Conference on Mobile Lightweight Wireless Systems*, volume 13, pages 397–404. Springer, 2009.

[73] N. Sastry, K. Sollins, and J. Crowcroft. Delivery Properties of Human Social Networks. In *Procedings of IEEE INFOCOM Miniconference*, pages 2586–2590, Rio de Janiero, Brazil, April 2009.

[74] S.E. Schaeffer. Graph Clustering. *In Computer Science Review*, 1(1):27–64, August 2007.

[75] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, pages 252–259, Philadelphia, PA, August 2005.

[76] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Performance Analysis of Mobility-Assisted Routing. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 49–60, Florence, Italy, May 2006.

[77] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case. *In IEEE/ACM Transactions on Networking*, 16(1):63 –76, February 2008.

[78] V. Srinivasan, M. Motani, and W. T. Ooi. Analysis and Implications of Student Contact Patterns Derived from Campus Schedules. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pages 86–97, Los Angeles, CA, September 2006.

[79] J. P. G. Sterbenz, R. Krishnan, R. R. Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao. Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions. In *Proceedings of the 1st ACM Workshop on Wireless Security*, pages 31–40, Atlanta, GA, December 2002.

[80] J. Su, A. Chin, A. Popivanova, A. Goel, and E. D. Lara. User Mobility for Opportunistic Ad-hoc Networking. In *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications*, pages 41–50, Low Wood, Lake Windermere,UK, December 2004.

[81] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Community Structure in Online Collegiate Social Networks. *In SIAM Review 53*, October 2010.

[82] A. Vahdat and D. Becker. Epidemic Routing for Partially-connected Ad hoc Networks. Technical Report CS-2000-06, Duke University, April 2000.

[83] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[84] J. Widmer and J.Y. L. Boudec. Network Coding for Efficient Communication in Extreme Networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 284–291, Philadelphia, PA, August 2005.

[85] J. Yang and J. Leskovec. Patterns of Temporal Variation in Online Media. In *Proceedings of the 4th International Conference on Web Search and Web Data Mining*, pages 177–186, Hong Kong, China, February 2011.

[86] E. Yoneki, P. Hui, and J. Crowcroft. Visualizing Community Detection in Opportunistic Networks. In *Proceedings of the 2nd ACM Workshop on Challenged Networks*, pages 93–96, Montreal, Quebec, Canada, September 2007.

[87] Y. Zhang and J. Zhao. Social Network Analysis on Data Diffusion in Delay Tolerant Networks. In *Proceedings of the 10th ACM International Symposium on Mobile Ad hoc Networking and Computing*, pages 345–346, New Orleans, LA, May 2009.