# Routing in Opportunistic Networks

*Divya Alok Sharma*

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

January 2014

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

**2014/01/02**

# Abstract

Opportunistic networking is employed in scenarios where end-to-end communication paths between nodes cannot be assumed due to continuously changing network topology. In these networks, data is routed in a “store-carry-forward” fashion by exploiting the transient communication opportunities that arise when mobile nodes equipped with short-range wireless interfaces come within each other’s transmission range. Designing efficient routing schemes for opportunistic networks is challenging as they face a trade-off between delivery performance and resource consumption. In this thesis, we propose a novel opportunistic routing algorithm that addresses this trade-off by incorporating information about the node contact pattern in the network. Aggregating contact events into a weighted graph enables us to detect communities and learn about the neighbourhoods of nodes. This information is utilized by our algorithm to assess the usefulness of every forwarding opportunity that arises. Performance evaluation of the proposed algorithm shows that it is capable of achieving good delivery performance while consuming significantly fewer resources in comparison to the existing opportunistic routing algorithms.

# Sommaire

Le réseautage opportuniste est utilisé dans les scénarios où l'existence de routes de communication entre agents ne peut pas être présumé dû à l'altération constante de la topologie du réseau. Dans ces réseaux, l'information est acheminée grâce à une procédure de "sauvegarde-report" qui exploite les opportunités de communication transitoires qui ont lieu lorsque des agents équippés d'interface sans-fil à petite portée entrent dans leurs périmètres de portée respectifs. Établir des plans de communication efficaces dans le contexte de réseaux opportunistes est un défi puisque ceux-ci font face à un compromis entre performance de livraison et consommation de ressources. Dans cette thèse, nous proposons un algorithme de routage opportuniste original qui addresse ce compromis en incorporant l'information liée aux contacts entre noeuds du réseau. Assembler les évènements de contacts dans un graphe pondéré nous permet de détecter les communautés et d'apprendre le voisinage de chaque noeud. Cette information est utilisée par notre algorithme afin d'évaluer l'utilité de chaque opportunité de communication qui se présente. Une analyse de la performance de l'algorithme proposé montre qu'il est capable d'atteindre une bonne performance de livraison tout en consommant significativement moins de ressources lorsque comparé aux algorithmes de routage opportuniste actuels.

# Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Mark Coates, for his valuable guidance and support that helped me to successfully complete this thesis.

I am grateful to my parents and brother for their love and support throughout my studies abroad.

I would also like to thank my colleagues of the Computer Network Research Lab, especially Santosh, Rizwan, Arslan, Jay, Sean, Eric, Benjamin, Rodrigo, Milad, Niloufar, Yunpeng, Tao and Deniz, for their companionship and assistance.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AP | Access Point |
| CCDF | Complementary Cumulative Distribution Function |
| MAC | Media Access Control |
| MB | Megabyte |
| Mbps | Megabit per second |
| ONE | Opportunistic Network Environment |
| SD | Standard Deviation |

# Chapter 1

# Introduction

## 1.1 Motivation

Nowadays, most mobile devices are equipped with short-range wireless interfaces of Bluetooth and WiFi in addition to the cellular radio. In the absence of Internet connectivity, these short-range networking technologies can be exploited to enable transfer of messages[1] between devices that come within communication range of each other as a consequence of their mobility. Such mobile ad-hoc networks, which deliver messages by taking advantage of any transmission opportunities that arise, have been called opportunistic networks [1]. These networks have received significant attention from researchers in recent times as they can be deployed in a number of practical scenarios, such as non-intrusive wildlife tracking (e.g., ZebraNet [2] and SWIM [3]), providing data communication to remote and rural areas (e.g., DakNet [4]) and offloading mobile data traffic [5].

The main focus of research on opportunistic networks has been on the aspect of routing. In these networks, sending messages between nodes is challenging as the network topology changes with time due to node mobility. Hence, formation of end-to-end routes between source and destination of messages is not always possible. If the evolution of the network topology were deterministic, routes for messages could be selected so as to achieve optimal performance in terms of delay [6]. However, in most cases of interest, the movement pattern of nodes cannot be completely determined based on local information available to nodes. Messages have to be exchanged between nodes that come within communication range of

[1] In this thesis, the term "message" is used to denote a file that a node wishes to send to another node in the network.

each other in order to move them closer to their destinations. Thus, routing takes place in a store-carry-forward fashion [7]. In this approach, if the next hop is not immediately available to a node, it must buffer the message until it has the opportunity to forward it. A routing scheme is used to determine when to forward a message and to whom. Different factors such as delivery success rate, end-to-end delay, cost in terms of bandwidth, power and memory consumption have to be taken into account while designing it.

## 1.2 Thesis Problem Statement

Designing efficient routing algorithms for opportunistic networks is difficult as limited information is available about their topological evolution. Consequently, all opportunistic routing strategies face a trade-off between delivery performance (success ratio, end-to-end delay) and resource consumption (memory, bandwidth and power). An appropriate balance between them can be achieved by exploiting more information about the the expected topology of the network [6].

In this thesis, we have focussed our attention on opportunistic networks that consist of people carrying mobile devices. In these networks, the social interactions and daily routines of people are expected to heavily influence their movement pattern. The problem is to design an efficient opportunistic routing algorithm that utilizes information learned from the contact pattern of nodes to make forwarding decisions. In order to accomplish this task, the following sub-problems will have to be tackled: (i) determine a tractable way of representing the node encounter pattern in the network such that potentially useful information is not lost; (ii) use this representation to identify and extract important features that will enable our routing algorithm to distinguish nodes based on their forwarding capabilities. By including this additional information, our goal is to achieve a favourable trade-off between delivery performance and resource consumption.

## 1.3 Thesis Contribution and Organization

This thesis presents a novel opportunistic routing algorithm that relies on information obtained by processing node encounters within a network to make forwarding decisions. In order to accomplish this, we have described a technique for aggregating contact events between the nodes into a weighted graph. Neighbourhood and community information extracted from this graph are used by our algorithm to determine the suitability of nodes to act as relays for different messages. In contrast to the existing strategies, the use of a weighted graph enables our algorithm to determine the information exchange capacity between the nodes and their likelihood of encountering each other regularly. By differentiating between strong and weak ties in the network, our algorithm is able to avoid counterproductive forwarding of messages. This helps it to achieve good delivery performance while reducing its resource consumption. The organization of this thesis is summarized below.

Chapter 2 discusses the strengths and weaknesses of the existing routing algorithms available for use in opportunistic networks. At the end of this chapter, we list the different synthetic mobility models and real world datasets that are commonly used for assessing the performance of opportunistic routing algorithms.

Chapter 3 describes our procedure for building a weighted graph from the contact events in a network. Sample graphs are prepared by processing node encounters recorded in two real world datasets. A detailed discussion of their topological properties is presented towards the end of this chapter.

Chapter 4 presents the proposed opportunistic routing algorithm by providing a thorough description of its forwarding decisions and the rationale behind them.

Chapter 5 compares the performance of our algorithm with other well-known opportunistic routing schemes. The simulation scenarios have been built by using contact events from real world datasets. The evaluation metrics have been chosen carefully in order to capture both delivery performance and resource consumption of the algorithms under consideration.

Chapter 6 concludes the thesis by summarizing our work and discussing potential directions for future research.

# Chapter 2

# Literature Review

In this chapter, we briefly review the various routing algorithms that have been proposed for use in opportunistic networks. We begin by describing algorithms that do not distinguish nodes based on their forwarding capabilities and rely solely on replicating or coding the messages. These routing schemes are followed by algorithms that assess the usefulness of every forwarding opportunity in bringing a message closer to its destination. This is achieved by incorporating additional information about the network in the routing process. We end the chapter by describing the two main approaches that are adopted for evaluating the performance of opportunistic routing algorithms.

## 2.1 Replication based routing

In replication based routing, multiple copies of a message can be present in the network at any given time. For a message to be delivered successfully, at least one of the copies has to reach the destination before it expires. These algorithms do not make use of any information about the evolution of network topology. A node forwards messages stored in its buffer to all or a fraction of encountered nodes without assessing their capability to deliver them. The only parameters controlled by these algorithms are the level of replication and the nodes that are allowed to make new copies.

### 2.1.1 Epidemic Routing

Epidemic routing for partially connected ad-hoc networks was first proposed by Vahdat and Becker [8]. It floods the network with duplicate copies of messages. Whenever two nodes meet, they exchange summary vectors containing the list of messages stored in their respective buffers. After comparing the summary vectors, a node requests those messages that it does not already have. Thus, messages are replicated in the network until every node has one copy. If unlimited bandwidth and buffer space are available, this scheme would deliver the most messages with least delay. However, significantly high resource consumption makes it infeasible for most practical purposes.

Epidemic routing is oblivious to the variance in forwarding capability of a node to different destinations in the network. Its efficacy depends mainly on the amount of resources available.

### 2.1.2 Spray and Wait Routing

The spray and wait protocol [9] restricts resource consumption by limiting the number of copies of a message that can be made. This algorithm operates in two phases: the spray phase and the wait phase. During the spray phase, a predetermined number of copies (say $L$) of a message are spread in the network by following one of the two proposed mechanisms. In the simpler case, the source node forwards $L - 1$ copies of a message to the first $L - 1$ distinct nodes it encounters. The other mechanism sprays the copies in a binary fashion. Any node having $n > 1$ copies of a message forwards $\lfloor n/2 \rfloor$ copies to the first node it encounters which does not already have that message. This process continues until the node is left with only a single copy of the message after which it enters the wait phase. The wait phase, as its name suggests, involves direct transmission of a message to its destination. A node having a single copy of a message just carries it in the buffer until its destination is encountered.

As this routing algorithm is also oblivious to the dissimilar forwarding capabilities of nodes, level of message replication is the most important factor affecting its performance in terms of delivery success rate and latency. If it is set to a low value, fewer messages will be delivered with large end-to-end delay. On the other hand, if high replication is used, good delivery performance will be achieved at prohibitive levels of resource consumption. The optimal level of replication depends on the size of the network, expected delivery performance and the amount of resources available.

## 2.2 Coding based routing

An alternate approach to replication based routing involves transmission of coded packets rather than mere duplicates of original messages. These coded packets may be generated by either the source node or by intermediate relay nodes as a message traverses the network. The destination node can successfully decode the original message once it receives a sufficient number of encoded packets. This procedure achieves good delivery performance while limiting transmission overhead.

### 2.2.1 Erasure coding based approach

In erasure coding, an original message is converted into a large set of coded blocks by its source. A fraction of these coded blocks are required at the destination to decode the original message. If a replication factor of $r$ is employed, only $1/r$ fraction of the coded blocks are sufficient to retrieve a message.

The erasure coding based forwarding scheme proposed by Wang et al. [10] is quite similar to spray and wait routing in its operation. For a replication factor of $r$, a message of size $M$ is encoded into $\frac{M \times r}{b}$ blocks of size $b$ each. These coded blocks are then split equally among the first $k \times r$ nodes that the source encounters. Thus, each relay node carries $\frac{M}{k \times b}$ blocks having a cumulative size of $\frac{M}{k}$. The message can be decoded at its destination as soon as $k$ of the relay nodes deliver their coded blocks.

The main difference with spray and wait routing is the use of $k$ times more relay nodes for the same amount of message replication. As each relay carries blocks occupying only $1/k$ of the original message size, there is more uniform distribution of messages in the network. Moreover, the use of a greater number of relays increase the likelihood of delivering blocks with lower delay. However, erasure coding requires $k$ relays to succeed in comparison to just one in the case of replication based routing. Thus, the actual performance of this algorithm will depend on the mobility characteristics of nodes in the network.

Different erasure coding schemes such as Reed-Solomon codes [11] and Tornado codes [12] are available. They differ in their encoding/decoding times and decoding efficiency which determines the number of coded packets required to reconstruct the original message.

### 2.2.2 Network coding based approach

In erasure coding, only the source of a message was allowed to generate coded blocks by processing it. The rest of the nodes in the network merely forwarded these coded packets. Network coding, introduced by Ahlswede et al. [13], allows intermediate relay nodes to process the packets they receive and generate new coded packets from them. This helps to increase the network throughput. It was shown that a sender can communicate information to a set of receivers in the network at a rate equaling the broadcast capacity if network coding is allowed [13]. This is generally not possible if only replication or erasure coding is used. It was further shown that linear encoding functions were sufficient to achieve this capacity [14]. Thus, in linear network coding, an intermediate node forwards a linear combination of packets stored in its buffer. The coefficients of linear combination are taken from a finite field $F_q$ of size $q$. When a node receives a sufficient quantity of independent network coded packets, called innovative packets, it can decode the set of original messages.

Avalanche is a large scale file distribution system based on network coding [15]. Widmer et al. [16] have proposed a routing algorithm for delay-tolerant networks that uses network coding. Whenever a node receives an innovative packet, it broadcasts a predetermined number of linearly coded packets to its neighbours. This quantity is called the forwarding factor of this algorithm.

Selecting the identity and number of packets that will be linearly combined at any node is an important design decision for routing schemes based on network coding. Messages are grouped into generations in order to constrain the size of encoding and decoding matrices [16]. Only packets belonging to the same generation are linearly combined. The benefits of network coding decrease if a generation contains very few packets. Its optimal size depends on the characteristics of the network and the pattern of message creation.

## 2.3 Node encounter prediction based routing

One obvious drawback of replication and coding based routing is the indiscriminate forwarding of messages to all or a fraction of encountered nodes which results in high transmission overhead. In most cases, the movement pattern of nodes is not completely random. Consequently, encounters between them are not homogeneous across the network. Routing algorithms can assess the utility of forwarding opportunities by estimating the chance a node has of encountering specific members in the near future. Transmission overhead can be significantly reduced by carefully selecting relays for messages.

### 2.3.1 PRoPHET

PRoPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity) makes routing decisions by predicting the probability of future node encounters [17]. This is considered to be dependent on the number of past encounters and the time gap between them.

The likelihood that node $a$ can deliver a message to node $b$ is denoted quantitatively by a metric called delivery predictability $P_{(a,b)} \in [0,1]$. Each node maintains a delivery predictability vector containing entries for all other nodes in the network. Whenever two nodes, say $a$ and $b$, meet they exchange and update their delivery predictability vectors in the following manner:

- **Delivery predictability to the encountered node**: Node $a$ will update its metric for node $b$ as follows:

  $$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init} \tag{2.1}$$

  where $P_{init} \in (0,1]$ is an initialization constant. Similarly node $b$ will update its metric for node $a$.

- **Delivery predictability to other nodes**: This is based on the property of transitivity. For any node $c \neq a$ or $b$, $a$ will update the value of $P_{(a,c)}$ using $P_{(b,c)}$ as follows:

  $$P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{2.2}$$

  where $\beta \in [0,1]$ denotes how large the contribution of transitivity should be. A similar update is performed at node $b$ as well.

Between two successive updates of a delivery predictability vector, the values contained in it are decreased in magnitude to highlight their aging. This is based on the intuition that if two nodes have not encountered each other for a long time, they are probably not good forwarders of messages to each other. As an example of this process, node $a$ will age its metric for node $b$ according to the following equation:

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^{k} \tag{2.3}$$

where $\gamma \in (0, 1)$ is the aging constant and $k$ is the number of time units that have elapsed since the last update of node $a$'s delivery predictability vector.

Once the values have been exchanged and updated, a message is forwarded from node $a$ to $b$ if the delivery predictability to its destination is higher at $b$. Reasonable values for parameters $P_{init}$, $\beta$, $\gamma$ and $k$ depend on the scenario under consideration.

Under certain undesirable conditions, it is possible for the delivery predictability metric to represent spurious information. As updates are performed at every node encounter, repeated reconnections between nodes due to unreliable connections will cause their metric for each other to increase too much. This phenomenon is known as the parking lot problem [18]. Similarly, a less aggressive aging policy may cause inappropriate amplification of indirectly obtained metrics in regions of high encounter frequency. In addition to these problems, fluctuations in delivery predictability values may cause messages to be forwarded more often than would otherwise be necessary, leading to longer paths and an increase in transmission overhead.

Other algorithms which make use of delivery prediction are Context-Aware Routing (CAR) protocol [19], Meets and Visits (MV) protocol [20], Shortest Expected Path Routing (SEPR) [21] and Minimal Estimated Expected Delay (MEED) [22].

## 2.4 Application of social network analysis techniques to routing

As the nodes in an opportunistic network are generally people carrying mobile devices, social interactions and daily routines are expected to guide their movement pattern. Routing algorithms can benefit by incorporating these factors into their decision process. Useful information about the strength of social ties between the nodes can be obtained by building a graph from the contact events between them and applying social network analysis techniques.

### 2.4.1 Identification of central nodes

Important nodes in the network can be identified on the basis of their centrality. Two of the most commonly used node centrality measures are degree and betweenness centrality. In an undirected graph with $|V|$ vertices and $|E|$ edges, these quantities are defined as follows:

**Degree centrality** of a vertex is the number of edges attached to it. It measures the size of a node's neighbourhood and can be calculated locally. The normalized degree centrality of node $v$ ($C_d(v)$) is given by

$$C_d(v) = \frac{degree(v)}{|V| - 1} \tag{2.4}$$

**Betweenness centrality** of a vertex is proportional to the number of times it lies on the shortest paths between other pairs of nodes in the graph [23]. Nodes with high betweenness centrality facilitate communication in the network. The betweenness centrality of node $v$ ($C_b(v)$) is given by

$$C_b(v) = \sum_{i \neq v \neq j \in V} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{2.5}$$

where $\sigma_{ij}$ is the total number of different shortest paths between nodes $i$ and $j$ and $\sigma_{ij}(v)$ is the number of these paths passing through $v$.

### 2.4.2 Community Detection and Modularity

Social interactions in the real world are organized around communities or groups such as families, friends and others. A graph constructed from these interactions is expected to have modular structure related to these communities. It will contain groups of nodes that have high concentration of edges among them suggesting that they encounter each other regularly for long periods of time. Routing algorithms can preferentially select relays for messages by making use of information on node affiliations. This has been shown to significantly improve forwarding performance over oblivious routing algorithms in opportunistic networks [24]. Several algorithms are available to divide a graph into clusters of nodes representing communities [25]. Most commonly used algorithms are based on optimization of modularity.

**Modularity** is a quality function which assesses the goodness of clusters obtained from a graph [26]. It is based on the idea that a random graph is not expected to have communities. Hence, it compares the density of edges in a subgraph to the density that would be expected in an equivalent random graph. Quantitatively, modularity is expressed as:

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(C_i, C_j) \tag{2.6}$$

where $A$ is the adjacency matrix of the graph, $m = \frac{1}{2} \sum_{ij} A_{ij}$ is the total number of edges and $P_{ij}$ represents the expected number of edges between nodes $i$ and $j$ in an equivalent random graph. The summation is done over every node pair. The $\delta$-function evaluates to 1 when nodes $i$ and $j$ belong to the same community, that is $C_i = C_j$, and is 0 otherwise. Positive values of modularity indicate that the communities identified are good. Its value is always less than or equal to 1 [27].

Community detection algorithms based on modularity optimization yield mutually disjoint clusters in which every node belongs to a single community. However, it is reasonable to assume that in many real world networks communities may overlap and have common members. The Clique Percolation Method by Palla et al. [28] is one method that is commonly used to detect overlapping communities in graphs.

Routing algorithms have been put forward that make use of centrality calculation and community detection to identify better relays for messages [29–31].

### 2.4.3 SimBet Routing

In networks where nodes form cliques, sending messages between different groups may be difficult if only metrics based on past encounters are used to make forwarding decisions. None of the nodes belonging to the group of the message source may have directly or indirectly encountered the destination. Consequently, algorithms such as PRoPHET (Section 2.3.1) are bound to perform poorly in these networks. It becomes important to identify bridging links between the tightly-knit groups so that messages can be forwarded between them. This is the central idea underpinning SimBet routing [29].

High betweenness centrality of a node indicates that it can facilitate communication between other nodes in the network. However, calculation of betweenness centrality using Equation 2.5 (socio-centric betweenness) requires knowledge of the entire graph topology. To avoid this complexity, SimBet routing requires nodes to calculate betweenness from their ego-networks (ego-centric betweenness). This network consists of the node (ego) and its neighbours (alters) along with all the unweighted and undirected links among them [29]. It has been shown empirically that locally calculated ego-centric betweenness correlates well with global socio-centric betweenness [32]. SimBet uses ego-centric betweenness to identify bridging nodes in the network.

Social networks also exhibit the property of transitivity which implies that nodes have higher probability of being acquainted if they share common friends. It forms the basis used by SimBet for determining similarity between any two nodes in the network. Mathematically, the social similarity $S(x, y)$ between nodes $x$ and $y$ is calculated as

$$S(x, y) = |N(x) \cap N(y)| \tag{2.7}$$

where $N(x)$ and $N(y)$ are sets representing the neighbourhoods of nodes $x$ and $y$ respectively. Nodes with higher similarity are considered to be better forwarders of messages meant for each other by SimBet.

Every node in the network maintains an adjacency matrix representing its ego-network. It is augmented by additional entries for nodes indirectly accessible to the ego node through its neighbours. These are used during similarity calculations. On every contact in the network, the two encountering nodes (say $n$ and $m$) update their respective adjacency matrices and recalculate their betweenness and similarity scores. For a message destined for node $d$, relative similarity and betweenness utility are calculated as follows:

$$SimUtil_n(d) = \frac{Sim_n(d)}{Sim_n(d) + Sim_m(d)} \tag{2.8}$$

$$BetUtil_n = \frac{Bet_n}{Bet_n + Bet_m} \tag{2.9}$$

where $SimUtil_n(d)$ is the relative similarity utility of node $n$ with respect to node $m$ for a message being sent to $d$. $Sim_n(d)$ and $Sim_m(d)$ denote the social similarity of nodes $n$ and $m$ respectively with destination $d$. Similarly, $BetUtil_n$ is the relative betweenness utility of node $n$ with respect to $m$. $Bet_n$ and $Bet_m$ represent the ego-centric betweenness of $n$ and $m$ respectively.

These are combined into SimBet utility of node $n$ for a message destined for node $d$ by the following formula:

$$SimBetUtil_n(d) = \alpha \cdot SimUtil_n(d) + \beta \cdot BetUtil_n \tag{2.10}$$

where $\alpha$ and $\beta$ are tunable parameters such that $\alpha + \beta = 1$. SimBet utility of node $m$ is also calculated similarly.

A message is forwarded to the node having higher SimBet utility for its destination. This algorithm is able to deliver messages even in situations where there is low connectivity between sending and receiving nodes. There is only a single copy of every message at any given time in the network as SimBet does not allow replication.

One potential weakness of SimBet routing is in its method of constructing unweighted ego-networks. In these networks, there is an edge between nodes that have met at least once in the past. This may lead to wrong estimation of betweenness and similarity of nodes as the pattern of node encounters varies over time.

### 2.4.4 Bubble Routing

This routing scheme proposed by Hui et al. [30] makes use of community affiliation and varying centrality of nodes to make forwarding decisions. Nodes are ranked, both globally (over the entire network) and locally (within their community), on the basis of their betweenness centrality. Global rank of a node determines its overall popularity and forwarding capability for any message in the network. Community affiliation of nodes is used to identify members of the community to which the destination of a message belongs. These are more likely to meet and deliver the message to it. The local rank of a node determines its popularity and forwarding ability for messages meant to be delivered within its community.

Routing in Bubble proceeds by replicating a message to more popular nodes in the network on the basis of global ranks. When a node belonging to the community of the destination is encountered, the message is forwarded to it irrespective of its global rank. Within the community, the message is replicated to members having higher local rank until it reaches its destination.

Community detection is performed by building an unweighted graph that represents the node encounters observed in a network. There is an edge between nodes whose total contact duration is greater than a predetermined threshold. The clique percolation method [28] is used to identify communities. Betweenness centrality is calculated numerically by replaying the events used for building the graph and simulating unrestricted flooding of messages. Centrality of a node is taken to be proportional to the number of times it lies on the shortest delay paths of delivered messages. The procedure for generating messages (choice of source-destination pair) is different for calculating global and local centrality.

A distributed version of Bubble has also been proposed in which every node estimates its community and centrality separately. A new member is added to the local community of a node as soon as its total contact duration with that node exceeds a familiarity threshold [33]. Centrality estimation is done by counting the number of unique encounters a node has within a time window [33]. The forwarding scheme of distributed Bubble is the same as that of centralized version.

Distributed Bubble faces the problem that local communities only grow in size as time progresses. There is no mechanism for reducing the metric (cumulative contact duration between nodes) used for determining if there should be an edge between nodes. A node continues to be in the local community of another node even if they have not met even once for a long period of time. This diminishes the benefit of using community affiliation to make forwarding decisions. It is also difficult to select an appropriate value for the familiarity threshold without conducting several trials. Replication of messages by this algorithm makes it resource intensive.

### 2.4.5 PeopleRank Routing

The SimBet and Bubble routing algorithms infer node centrality and community structure from graphs representing node encounters. The emergence of online social networks has made it easier to obtain information about the social interactions of a user. Routing algorithms can benefit by incorporating this additional information. It has been shown that forwarding messages through social neighbours can achieve better delivery performance in a small conference environment than random forwarding [34].

PeopleRank uses a graph representing user declared social relationships in conjunction with opportunistic contacts to identify central nodes in the network [31]. The centrality metric (PeopleRank) is inspired by the PageRank algorithm used to rank webpages on the Internet [35]. The PeopleRank of a node is high if it is connected to other important nodes in the social graph obtained from an online social network. It is assumed that only neighbours in the social graph have direct impact on the rank of a node. The PeopleRank of node $N_i$ ($PeR(N_i)$) is defined by the following formula:

$$PeR(N_i) = (1 - d) + d \sum_{N_j \in F(N_i)} \frac{PeR(N_j)}{|F(N_j)|} \tag{2.11}$$

where $N_1, N_2, \ldots, N_n$ represent nodes in the network, $F(N_i)$ is the neighbourhood of node $N_i$ in the social graph and $d \in (0, 1]$ is the damping factor. The second term in the summation on the right hand side equally divides a node's PeopleRank among all its neighbours. The damping factor denotes the importance given to social relationships in determining a node's centrality. It is close to one if the social graph represents a strong relation between nodes such as friendship. A lower value of $d$ is chosen when the social graph is loosely defined.

Whenever two socially connected nodes meet, they update their respective ranks according to Equation 2.11. Thus, frequently seen nodes update their centrality more often. A node will forward messages only to nodes with higher PeopleRank than it. It is important to note that this routing algorithm does not take into account the destination of a message while forwarding it. As a consequence, central nodes in the network will be overburdened as every message will be forwarded towards them irrespective of its destination.

## 2.5 Performance evaluation using real world datasets

There are two main approaches to evaluate the performance of an opportunistic routing algorithm. The first involves the use of a synthetic mobility model, such as random walk or random waypoint [36], to simulate a network of mobile nodes. However, these abstract and simplistic models are unable to capture several several important characteristics (such as dependence on daily routines and social groups) of real node mobility [37,38]. Hence, a more sensible approach is to use data recorded by experiments that were specially designed to study the movement pattern of people in the real world. These datasets can be divided into two categories based on whether they contain records of direct contact between nodes or events from which contacts can only be indirectly inferred. A brief description of the techniques used to collect them is presented below.

**Direct contact datasets** are obtained by making a group of participants carry Bluetooth enabled devices that scan their surroundings at regular intervals to detect nearby

| Name | Scan Interval (in seconds) | Devices | Duration | Year |
|---|---|---|---|---|
| University of Toronto | 16 | 23 | 11 weeks | 2004 |
| MIT Reality Mining | 300 | 94 | 9 months | 2004-05 |
| Cambridge | 120 | 12 | 7 days | 2005 |
| Infocom 2005 | 120 | 41 | 4 days | 2005 |
| Infocom 2006 | 120 | 78 | 5 days | 2006 |
| Sigcomm 2009 | 120 | 76 | 3 days | 2009 |

**Table 2.1** Direct contact datasets

devices. These datasets contain the start and finish times of every contact between each node pair in the network. Table 2.1 summarizes the direct contact datasets collected at University of Toronto [39], MIT [40], Cambridge [41] and some conferences (Infocom 2005 [41], Infocom 2006 [24], Sigcomm 2009 [42]).

**Access point (AP) based datasets** contain the times for which users are associated with different WiFi access points (APs) or GSM cell towers in the network. In order to infer contacts in these networks, it is assumed that devices connected to the same AP or cell tower are located close enough to be able communicate directly. Table 2.2 summarizes three of these datasets collected at Dartmouth [43, 44], UCSD [45] and MIT [40].

| Name | Wireless Technology | Devices | Duration | Year |
|---|---|---|---|---|
| Dartmouth | WiFi | >7000 | 3 years | 2001-04 |
| UCSD | WiFi | 275 | 11 weeks | 2002 |
| MIT Reality Mining | GSM | 94 | 9 months | 2004-05 |

**Table 2.2** AP based datasets

The contact pattern of nodes observed in these datasets has inspired several synthetic models such as the community based mobility model [38], the time-variant mobility model [46] and the working day movement model [47].

## 2.6 Summary

This chapter described the various categories of routing algorithms that have been proposed for use in opportunistic networks. All of them face a trade-off between delivery performance (success ratio, end-to-end delay) and resource consumption (memory, bandwidth and power). In general, the more information nodes have about the network, the more likely they are to make optimal forwarding decisions. Additional information used by routing algorithms includes past encounters between devices, their mobility pattern and social interactions between people carrying these devices. The last section listed the various synthetic models and real world datasets that are used for assessing the performance of opportunistic routing algorithms.

# Chapter 3

# Weighted Contact Graph and Community Detection

In many opportunistic networks, data about the strength of social ties between nodes is not directly available. However, the contact pattern of nodes is expected to reflect the underlying social structure as people are more likely to interact with friends, colleagues or family members than with strangers.

One technique to infer social ties in a network is to represent node encounters as an aggregated contact graph $G(V, \mathbf{A})$ and analyze them collectively. Vertices of this graph (V) correspond to the nodes in the network and the adjacency matrix ($\mathbf{A}$) encodes information about pairwise node encounters. This is the approach that is followed by the SimBet (see Section 2.4.3) and Bubble (see Section 2.4.4) routing algorithms. However, it may be argued that both of these algorithms lose a lot of useful information by considering only binary relations between the nodes. This implies that the elements in the corresponding adjacency matrix simply record the presence or absence of edges, that is:

$$A_{ij} = A_{ji} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

A major drawback of building such unweighted contact graphs is that equal significance is given to all the existing edges irrespective of the strength of ties they represent. This can harm the performance of a routing algorithm that depends on aggregate contact graphs to make forwarding decisions.

The underlying social structure of a network can be better analyzed by aggregating node encounters into a weighted contact graph. Every edge in this graph is assigned a weight which quantifies the strength of inferred social relationship between the two nodes joined by it. Elements in the adjacency matrix of this graph record the weights assigned to the corresponding edges:

$$A_{ij} = A_{ji} = \text{weight of edge between nodes } i \text{ and } j \tag{3.2}$$

This strategy allows one to retain both strong and weak ties in the network by appropriately assigning weights to them. Weak ties, representing sporadic encounters between nodes, have been shown to play an important role in achieving a favourable performance vs. bandwidth tradeoff in opportunistic networks [48].

This chapter describes our procedure for building a weighted contact graph from the contact events in a network and the conclusions obtained on analyzing it. We begin by presenting in Section 3.1 the properties of pairwise node encounters that are observed in Infocom 2006 (direct-contact) and Dartmouth (AP-based) datasets (see Section 2.5). Their characteristics are used to determine the contact metric on which edge weights will be based in the aggregate graph. Section 3.2 describes the topological properties of the weighted contact graphs derived from the two datasets under consideration. Section 3.3 presents the outcome of applying a community detection algorithm on them.

## 3.1 Characteristics of pairwise node encounters in real world datasets

Since every contact between two nodes in an opportunistic network provides a chance to exchange messages, contact duration and inter-contact time are the two most commonly analyzed metrics. Contact duration represents the time during which a pair of nodes are within communication range of each other. The time elapsed between two successive encounters between a pair of nodes is captured by the inter-contact time. The statistical properties of these metrics influence the general levels of throughput and latency that different categories of routing algorithms can achieve.

### 3.1.1 Infocom 2006 dataset

This dataset was collected during the IEEE Infocom conference held in Barcelona, Spain in April 2006 [24,49]. 78 Bluetooth enabled devices called iMotes, having a range of about 30 meters, were distributed to participants on April 23 between 7 p.m. and 9 p.m. Each iMote was programmed to scan its surroundings at intervals of 120 seconds and log contact data for all Bluetooth devices (including cell phones) it discovered. For every contact, the start and finish times were recorded along with the MAC address of the encountered device. The iMotes were collected from the participants on April 27.

Since we are only interested in studying the contact pattern between the iMotes, we disregard all encounters with external Bluetooth devices such as cell phones. Figure 3.1 shows how the number of node pairs in contact varies with time of day during the experiment. More encounters are observed in the daytime as people participate in various events at the conference. The level of activity during the night and early morning, on the other hand, is significantly lower. This plot clearly demonstrates the diurnal nature of contact pattern that should be expected in a network where node mobility is driven by human activities.

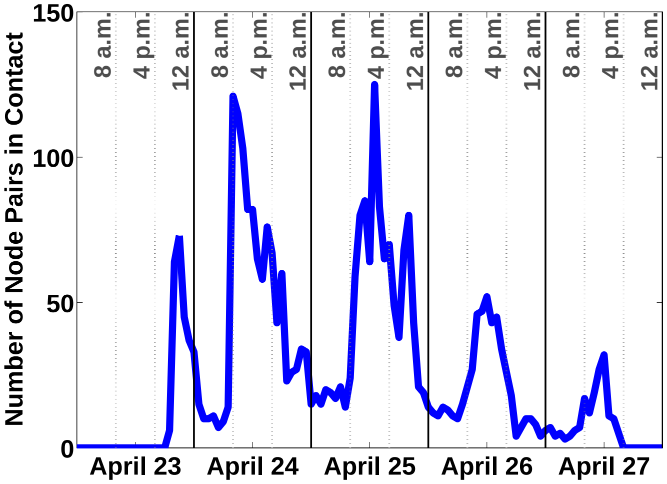
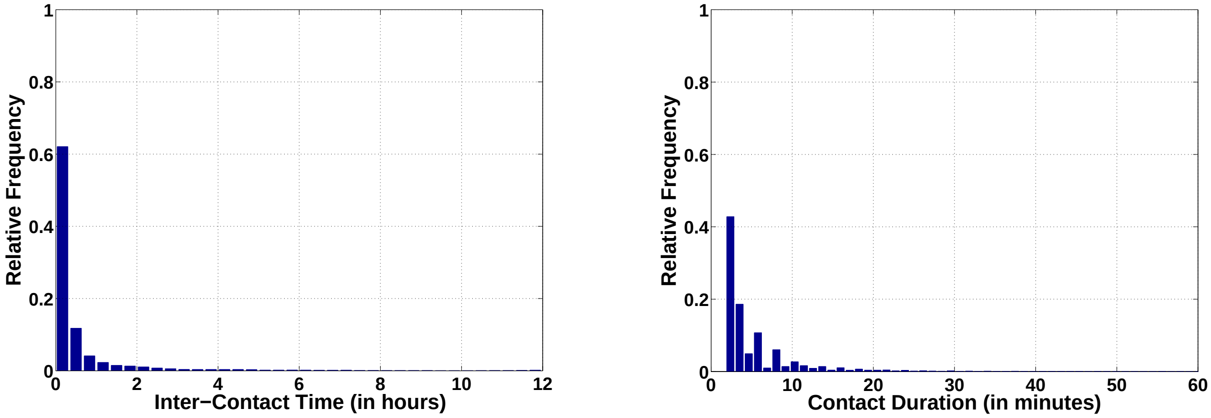


**Fig. 3.1** Number of iMote pairs within Bluetooth range of each other (sampled at intervals of 1 hour)

Empirical distributions of inter-contact time and contact duration are derived by using the start and finish times of encounters recorded by the iMotes. Since it is impractical to analyze these metrics for every possible iMote pair individually, we consider only their aggregate distributions. Figure 3.2 shows the histograms obtained from the samples of



(a) Histogram of inter-contact time samples

(b) Histogram of contact duration samples

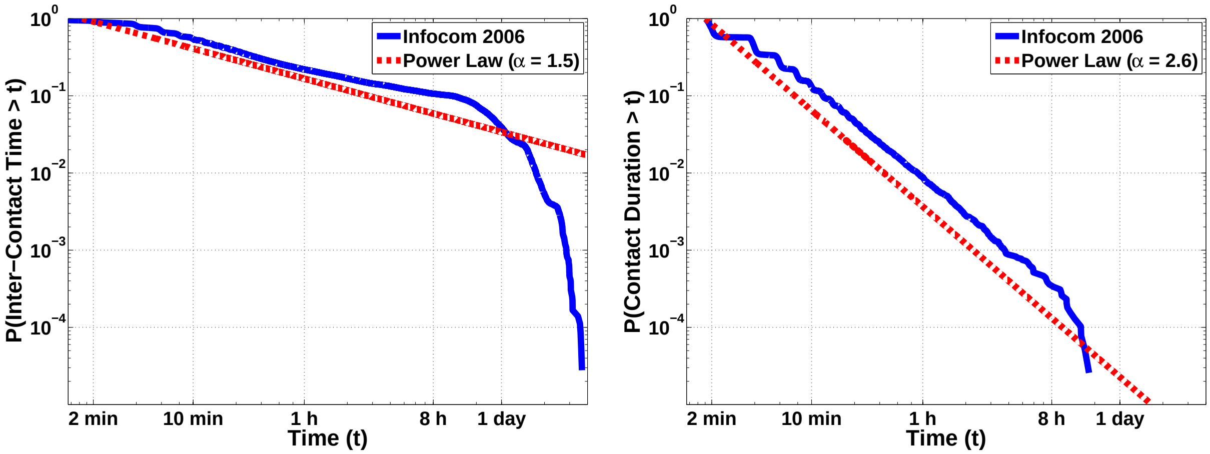
**Fig. 3.2** Distribution of the observed samples of inter-contact time and contact duration in Infocom 2006 dataset. The range of values depicted on the $x$-axis of both plots are partial.

inter-contact time and contact duration. A few summary statistics of these samples are presented in Table 3.1.

| | Mean (in minutes) | Median (in minutes) | Skewness | Kurtosis |
|---|---|---|---|---|
| Inter-contact time | 183.5 | 11.9 | 4.01 | 21.92 |
| Contact duration | 6.8 | 3.9 | 23.33 | 812.93 |

**Table 3.1** Summary statistics of the observed samples of inter-contact time and contact duration in Infocom 2006 dataset

The mean and median values of inter-contact time are 3.06 hours and 11.9 minutes respectively. In order to make inferences about the shape of its distribution, higher order moments of skewness and kurtosis have also been calculated [50]. Skewness measures the asymmetry of a distribution around its mean. Positive values indicate that the data is spread out more to the right of the mean than to the left. The opposite is true if skewness is negative. It is 0 for perfectly symmetric distributions such as the Gaussian. The fourth central moment of a distribution, called kurtosis, is used to measure the heaviness of its tail. Values greater than 3 indicate that a distribution is more outlier-prone than the Gaussian



(a) Aggregate CCDF of inter-contact time

(b) Aggregate CCDF of contact duration

**Fig. 3.3** Distribution of inter-contact time and contact duration obtained from pairwise encounters between 78 iMotes carried by participants in Infocom 2006 dataset

distribution and has a "fat" tail. The values of skewness and kurtosis for inter-contact time reveal that its distribution has a long and heavy right tail. Similar behaviour is exhibited by the distribution of contact duration which has a mean value of 6.8 minutes.

The log-log plots of the aggregate complementary cumulative distribution functions (CCDFs) of inter-contact time and contact duration are shown in Figure 3.3. The choice of logarithmic scale for the axes enables us to compare them with heavy-tailed power law[1] distributions. Figure 3.3(a) shows that the aggregate CCDF of inter-contact time approximately follows a straight line in the region from 2 minutes to 8 hours. Its slope is similar to that of the CCDF of power law with exponent 1.5. However, for higher values of inter-contact time, it decays much more rapidly. Thus, the distribution of inter-contact time has the characteristics of a power law over a finite interval. Analogous behaviour has been reported in several other datasets [37, 51]. Figure 3.3(b) shows that the slope of the aggregate CCDF of contact duration is comparable to that of the CCDF of power law with exponent 2.6 for almost its entire range of values. This explains the much higher value of kurtosis observed for contact duration in Table 3.1.

[1] The CCDF of a random variable $X \in [x_0, +\infty)$ having a power law distribution with exponent $\alpha$ is expressed as $P(X > x) = \left(\frac{x}{x_0}\right)^{-\alpha+1}$ for $x \geq x_0$ and $\alpha > 1$. Its plot on log-log scale is a straight line with slope $1 - \alpha$.

### 3.1.2 Dartmouth dataset

The Dartmouth dataset contains data recorded between April 11, 2001 and June 30, 2004 by WiFi APs installed in several buildings on the campus of Dartmouth college [43,52]. The APs were configured to transmit a message to a central server every time a wireless client associated or disassociated with them. Each message contained the AP name, the MAC address of the client and an indicator denoting the type of event. The server recorded every message along with its time of arrival. Several thousand users utilized the WiFi infrastructure on campus during the period for which data has been collected.

As it is infeasible to analyze the entire dataset, we concentrate only on the three week period from October 12, 2003 to November 1, 2003. This interval contains records from 537 APs. We assume that two devices are in contact when they are connected to the same AP at the same time. As the number of active devices changes daily, we consider only those nodes that were active on at least five days during each of the three weeks. This step resulted in the network containing 1811 nodes.

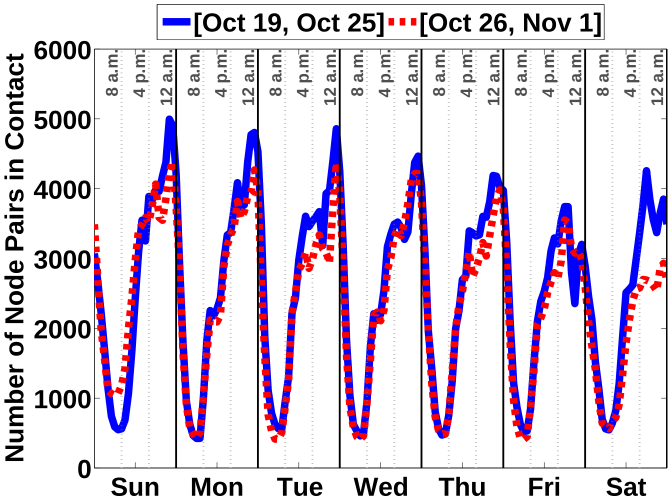
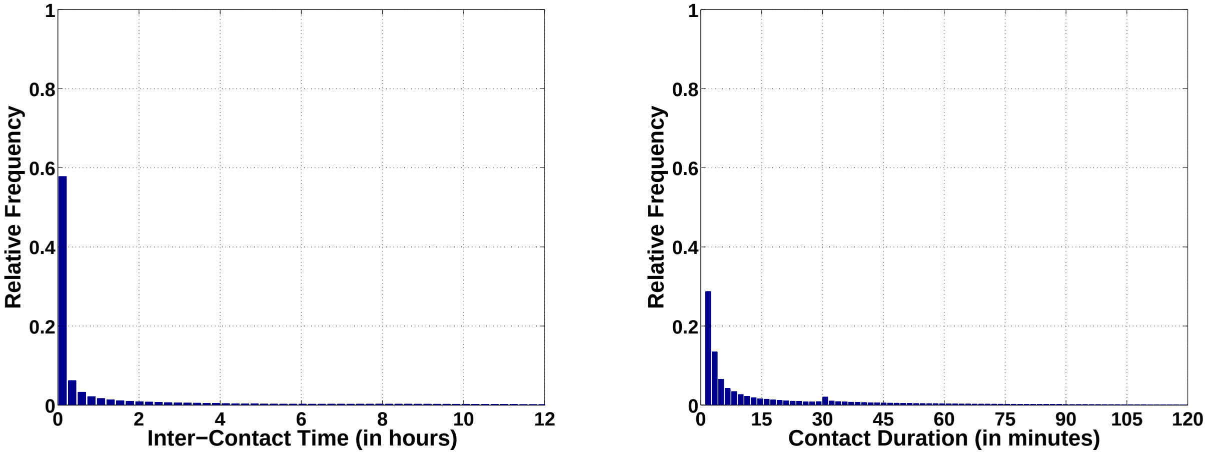


**Fig. 3.4** Number of user pairs connected to the same AP (sampled at intervals of 1 hour) during the two week period from October 19, 2003 to November 1, 2003

Figure 3.4 shows the number of node encounters observed in Dartmouth dataset at different instants of time during the two week period from October 19, 2003 to November 1, 2003. The diurnal and weekly nature of the contact pattern is clearly evident from



(a) Histogram of inter-contact time samples

(b) Histogram of contact duration samples

**Fig. 3.5** Distribution of the samples of inter-contact time and contact duration observed during the three week period from October 12, 2003 to November 1, 2003 in Dartmouth dataset. The range of values depicted on the $x$-axis of both plots are partial.

this plot. On most days, node encounters increase after 7 a.m. and attain their peak value around midnight. The interval from midnight to 7 a.m. witnesses a steady decline in activity within the network. As expected, the rise and fall of node encounters seem to closely follow the daily schedule of students living on campus.

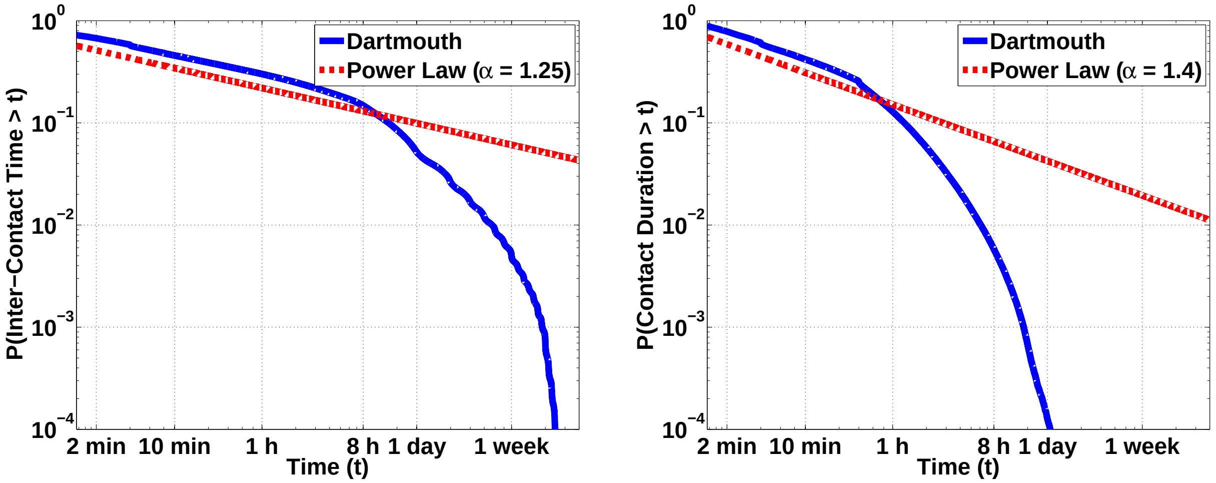
Statistical properties of pairwise node encounters are derived by processing the start and finish times of inferred contacts. In order to remove ephemeral contacts that occur when a device jumps back and forth between different APs in quick succession, we consider only those node encounters which last longer than 60 seconds.

| | Mean<br>(in minutes) | Median<br>(in minutes) | Skewness | Kurtosis |
|---|---|---|---|---|
| Inter-contact time | 359.2 | 6.9 | 8.73 | 102.40 |
| Contact duration | 31.2 | 6.3 | 7.65 | 106.42 |

**Table 3.2** Summary statistics of the samples of inter-contact time and contact duration observed during the three week period from October 12, 2003 to November 1, 2003 in Dartmouth dataset

Figure 3.5 shows the histograms obtained from the samples of inter-contact time and contact duration. The summary statistics of these samples are presented in Table 3.2. The values of skewness and kurtosis indicate that the two metrics have positively skewed heavy-tailed distributions.

Figure 3.6 shows the aggregate CCDF of inter-contact time and contact duration on log-log scale. Based on Figure 3.6(a), the distribution of inter-contact time can be compared to a power law with coefficient 1.25 for values less than 8 hours. For higher values, the CCDF decays much more rapidly. This behaviour is analogous to the observations made for the Infocom 2006 dataset. The aggregate CCDF of contact duration is shown in Figure 3.6(b). Its similarity with a power law distribution only holds for periods less than 1 hour.



(a) Aggregate CCDF of inter-contact time

(b) Aggregate CCDF of contact duartion

**Fig. 3.6** Distribution of inter-contact time and contact duration obtained from inferred pairwise encounters between 1811 users in Dartmouth dataset. The three week period from October 12, 2003 to November 1, 2003 has been considered

## 3.2 Construction of weighted contact graph

Based on the diurnal nature of the contact patterns observed in real world datasets, we have decided to use contact duration per day between two nodes to quantify the strength of tie between them. The amount of time for which they are located close to one another is expected to be related to the nature of social interaction between them. In addition,

| | Infocom 2006 | Dartmouth |
|---|---|---|
| Duration (days) | 3 | 14 |
| # Nodes | 78 | 1811 |
| Edge Density | 0.8884 | 0.0436 |
| # Connected Components | 1 | 9 |
| # Nodes (Largest Component) | 78 | 1803 |

**Table 3.3** Summary of contact graphs generated from Infocom 2006 and Dartmouth datasets. The metric of edge density denotes the fraction of edges with weight greater than zero.

it is also indicative of the amount of data that can be exchanged between the two nodes. Thus, the adjacency matrix ($\mathbf{A}$) of the weighted contact graph built from a dataset will have entries:

$$A_{ij} = A_{ji} = \frac{D_{ij}}{N} \tag{3.3}$$

where $D_{ij}$ is the cumulative contact duration between nodes $i$ and $j$ over the period of $N$ days that have been extracted from the dataset. We did not consider other contact metrics, such as number of encounters and inter-contact time, to derive edge weights as frequently dropped connections between devices can make them unreliable.

### 3.2.1 Structure of graphs obtained from the datasets

Using the above described procedure, we generate weighted contact graphs for both the datasets under consideration.

For the Infocom 2006 dataset, we have only used contact events occurring during the three day period from April 24 to April 26 to build the contact graph. As data is not available for the entire first and last day of this experiment, we do not include them in our analysis.

Based on the weekly nature of contact pattern observed in the Dartmouth dataset, we have used events from the first two weeks (October 12, 2003 - October 25, 2003) to build a weighted contact graph.

Table 3.3 summarizes the characteristics of the contact graphs generated from the datasets. The two graphs differ greatly in their edge density (fraction of edges with $A_{ij} > 0$).

The conference environment of Infocom 2006 dataset is small and dense in which every node is likely to meet most of the other nodes in the network. In comparison, the contact graph of Dartmouth dataset is quite sparse with a node encountering only 79 other nodes on average during the 14 day period under consideration. There are 8 isolated nodes with degree 0 that do not belong to the largest connected component of the contact graph representing the Dartmouth network.

### 3.2.2 Clustering coefficient and small-world effect

Transitivity or clustering is a typical property exhibited by graphs representing social ties in a network. In these graphs, the likelihood of two nodes being connected increases if they share a common neighbour. It is related to the commonly observed phenomenon that a friend of your friend is more likely to be your friend. The level of transitivity in a graph is indicated by the number of triangles present in its topology. Watts and Strogatz [53] gave the following definition for the local clustering coefficient of a vertex $v$ ($C_v$) in an unweighted graph:

$$C_v = \frac{\text{number of triangles connected to vertex } v}{\text{number of possible edges between its neighbours}} \tag{3.4}$$

If $d(v)$ denotes the number of neighbours (degree) of $v$, then the denominator in the above equation is equal to $\binom{d(v)}{2}$. For vertices with degree 0 or 1, $C_v$ is taken to be zero [54]. The clustering coefficient for the entire graph (C) is the average of the local clustering coefficients of all its vertices:

$$C = \frac{1}{|V|} \sum_{v \in V} C_v \tag{3.5}$$

From the above definition it is clear that $C \in [0, 1]$.

In order to calculate clustering coefficients for the two weighted contact graphs obtained from the datasets, we first construct their equivalent unweighted versions by pruning edges with weights less than a threshold value. This step retains stronger ties that represent regular node contacts in the network. The weight threshold is determined so that the resultant unweighted graph has a desired edge density. In order to present the general trend that is not distorted by truncation, we build three unweighted graphs with different edge densities for each dataset (0.04, 0.05 and 0.06 for Infocom 2006 and 0.01, 0.02 and 0.03 for Dartmouth).

| | Infocom 2006 | | | Dartmouth | | |
|---|---|---|---|---|---|---|
| | 4% | 5% | 6% | 1% | 2% | 3% |
| #Nodes (Largest Component) | 63 | 64 | 67 | 1634 | 1789 | 1799 |
| Clustering Coefficient | 0.35 | 0.39 | 0.37 | 0.74 | 0.53 | 0.45 |
| Avg. Shortest Path Length | 4.81 | 3.63 | 3.20 | 5.90 | 2.91 | 2.52 |

**Table 3.4** Clustering Coefficient and Average Shortest Path Length in the largest connected components of the truncated contact graphs of the two datasets. Lowest weight edges have been pruned in order to achieve the indicated edge densities.

We are also interested in calculating the average shortest path length ($l$), the formula for which is given by:

$$l = \binom{|V|}{2}^{-1} \sum_{i>j} d_{ij} \tag{3.6}$$

where $d_{ij}$ is the geodesic (shortest) distance between nodes $i$ and $j$ and $\binom{|V|}{2}$ is the total number of unordered node pairs in a graph with $|V|$ nodes. If a graph contains more than one connected component, the value of geodesic distance becomes ambiguous for nodes that lie in different components and do not have a connecting path. Hence, we concentrate only on the largest connected component of each unweighted graph for calculating the two metrics of interest (clustering coefficient and average shortest path length).

Table 3.4 presents the values of clustering coefficient and average shortest path length measured in each of the derived unweighted graphs. Significantly higher clustering is observed in these graphs in comparison to Erdős-Rényi (ER) random graphs with similar numbers of vertices and edges (expected clustering coefficient of an ER random graph is equal to its edge density) [55]. It indicates highly non-random behaviour of node encounters in the two scenarios under consideration. In addition, the low values of average shortest path lengths, especially in the graphs of Dartmouth dataset, point to the existence of the small-world phenomenon [56, 57] in these networks. It suggests the possibility that opportunistic routing algorithms can deliver messages in a few hops by incorporating information about the strength of mobility correlation between the nodes in the network.

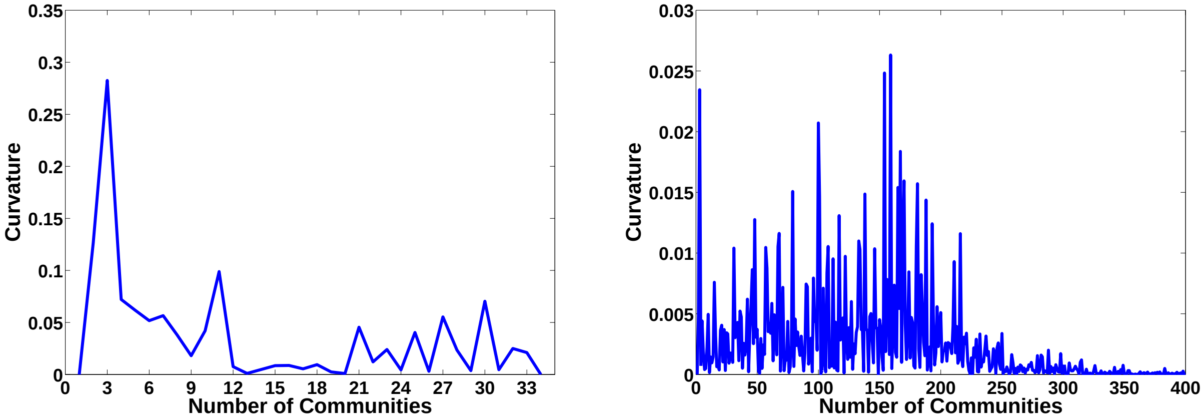
## 3.3 Decomposition of weighted contact graph into communities

Further information from the non-random topology of a weighted contact graph can be obtained by breaking it down into weakly linked groups of densely connected nodes. Each of these groups represents a community, the members of which are more likely to encounter each other regularly for long periods of time.

A number of techniques are available for identifying communities in a weighted graph [25]. However, the algorithms of interest to us are those that try to optimize either of the two most commonly used criteria for measuring the goodness of partitions obtained from a graph. These include the normalized cut criterion proposed by Shi and Malik [58] and the modularity function proposed by Newman and Girvan [26].

In graph theoretic terms, the cut associated with a particular division of a weighted graph into disjoint communities is simply the sum of the weights of edges running between them. Intuitively, low values of cut denote less similarity between the identified clusters. Hence, partitioning a graph by minimizing the normalized cut (a modification of cut) [58] identifies communities with strong internal connections that are weakly linked to each other by low weight edges. Several algorithms are available for this purpose [58–62]. However, most of them require the number of desired communities as input. This renders them unsuitable for use with real world networks where such information is not available a priori. The GANC (Greedy Agglomerative Normalized Cut) algorithm proposed by Tabatabaei et al. [62] tries to address this shortcoming by providing a mechanism for selecting the appropriate number of communities in a network. The plot of a metric called “curvature” against all the possible number of communities is expected to have a distinct peak at the optimal number of partitions identified by this algorithm. This behaviour is demonstrated in Figure 3.7(a) which shows the curvature plot obtained on applying GANC to the Zachary karate club network [63]. If there is no prior knowledge of the true number of communities, the location of the peak can be used to instruct GANC to divide the network into 3 clusters. However, the absence of any clear peak in the curvature plot shown in Figure 3.7(b) makes it difficult to reliably identify the optimal number of communities in the Dartmouth network. This requirement of providing the number of partitions is not encountered while using algorithms that optimize modularity.

Modularity is a quality function that assesses the goodness of clusters by measuring the extent to which the density of edges within them is higher than expected (see Equation



(a) Distinct peak observed in Zachary karate club network

(b) No distinct peak observed in the graph derived from Dartmouth dataset

**Fig. 3.7** Curvature plot used for identifying optimal number of partitions in GANC

2.6 in Section 2.4.2). It is based on the idea that a random graph is not expected to have a meaningful community structure [26]. Positive values of modularity indicate that the communities identified are good. According to this criterion, the optimal partitioning of a graph corresponds to the maximum value of modularity that can be achieved on it. As modularity optimization does not require any prior knowledge of the number of communities, this approach is appropriate for use with real world networks. Several algorithms are available that are applicable to large networks and often find partitionings with modularity values close to the maximum achievable modularity [64–69].

We have decided to use the modularity optimization based Louvain algorithm, proposed by Blondel et al. [69], to detect communities in the weighted contact graphs derived from the datasets. In addition to being extremely fast, it is one of the best algorithms in terms of consistently achieving high modularity partitions for a wide variety of network topologies.

### 3.3.1 Louvain algorithm

The Louvain algorithm is a greedy algorithm that maximizes modularity (see Equation 2.6 in Section 2.4.2) in order to extract the hierarchical community structure of a network [69].

The first step of this algorithm is initialized by assigning a different community to every node in the graph. The nodes are then processed sequentially in order to determine if any improvement in modularity can be achieved by moving a node to the community of one of

| | Infocom 2006 | | | | Dartmouth | | | |
|---|---|---|---|---|---|---|---|---|
| | L0 | L1 | L2 | L3 | L0 | L1 | L2 | L3 |
| # Communities | 78 | 19 | 8 | 7 | 1803 | 149 | 37 | 29 |
| Modularity | -0.015 | 0.250 | 0.286 | 0.288 | -0.001 | 0.801 | 0.890 | 0.892 |

**Table 3.5** Outcome of Louvain algorithm on the largest connected components of the weighted contact graphs derived from the two datasets. 'L$x$' denotes the hierarchy level '$x$' in the community structure. Modularity at every level is calculated with respect to the topology of the original graph.

its neighbours. The change which results in the highest gain in modularity is chosen. If no positive gain is possible, the node remains in its current community. This process is applied repeatedly in the same order until no further increase in modularity can be obtained by moving a single node. In the next step, a new weighted graph is constructed by merging the nodes belonging to a community into a *supernode*. The weight of an edge between any two *supernodes* is equal to the sum of the weights of the edges between the two corresponding communities in the previous graph. These two steps collectively denote one iteration of the algorithm. The *supernodes* identified at the end of every iteration yield a new level in the hierarchical community structure of the graph. The algorithm stops when no improvement in modularity is observed in an iteration.

### 3.3.2 Hierarchical community structure identified in the datasets

We applied the Louvain algorithm on the largest connected components of the weighted contact graphs prepared from the two datasets. The 8 isolated nodes in the contact graph of Dartmouth dataset have been excluded as they have no impact on the calculation of modularity.

Table 3.5 shows the number of communities and the corresponding value of modularity at every level in the hierarchical community structure uncovered by the algorithm. The bottommost level (L0) in both hierarchies corresponds to the initial partition in which there are as many communities as there are nodes. All the subsequent levels (L1, L2 and L3) present the statistics related to the communities identified at the end of every iteration. The algorithm stopped after the same number of iterations on both graphs.

The community partitions obtained in Infocom 2006 dataset have low modularity. This

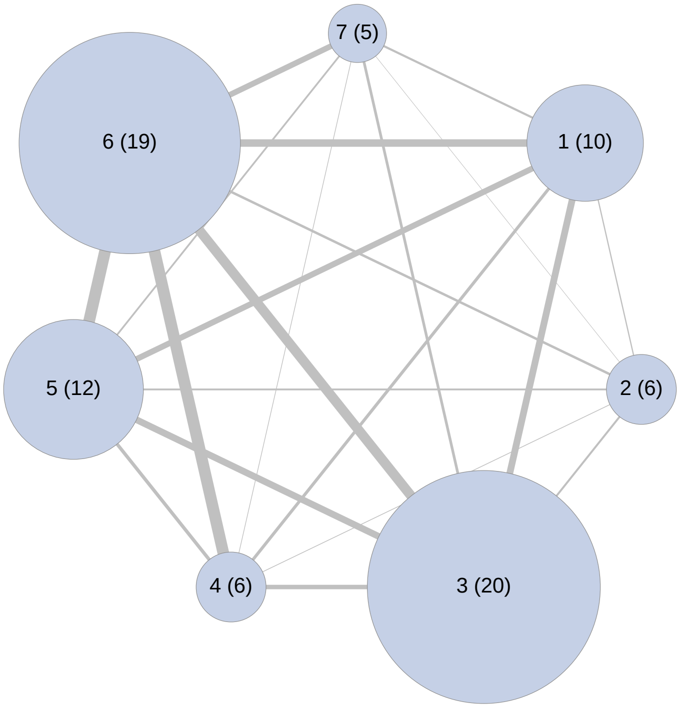


**Fig. 3.8** Graph representing the community structure identified by the Louvain algorithm in the weighted contact graph of Infocom 2006 dataset. In every node label, the community size is enclosed in parentheses.

outcome agrees with the experience that participants attending a conference tend to make new contacts by interacting with people other than their colleagues. The communities identified in Dartmouth network, on the other hand, are of much better quality as indicated by their significantly higher values of modularity.

On the basis of modularity, we choose the partitions corresponding to the topmost level (L3) in both hierarchies to represent the communities in their respective networks. Thus, the 78 nodes of Infocom 2006 dataset are divided into 7 communities that have 20, 19, 12, 10, 6, 6 and 5 members. The interconnection between these communities is depicted in the form of a graph in Figure 3.8. In this graph, the size of every node is proportional to the size of the community it represents and the thickness of every edge is proportional to the cumulative strength of the ties between the members of the two communities it connects.

In Dartmouth dataset, the 1803 nodes nodes lying in the largest connected component of its contact graph are divided into 29 communities. The remaining 8 nodes are assigned separate communities which takes the total count up to 37. Figure 3.9 depicts the interconnection between these communities.

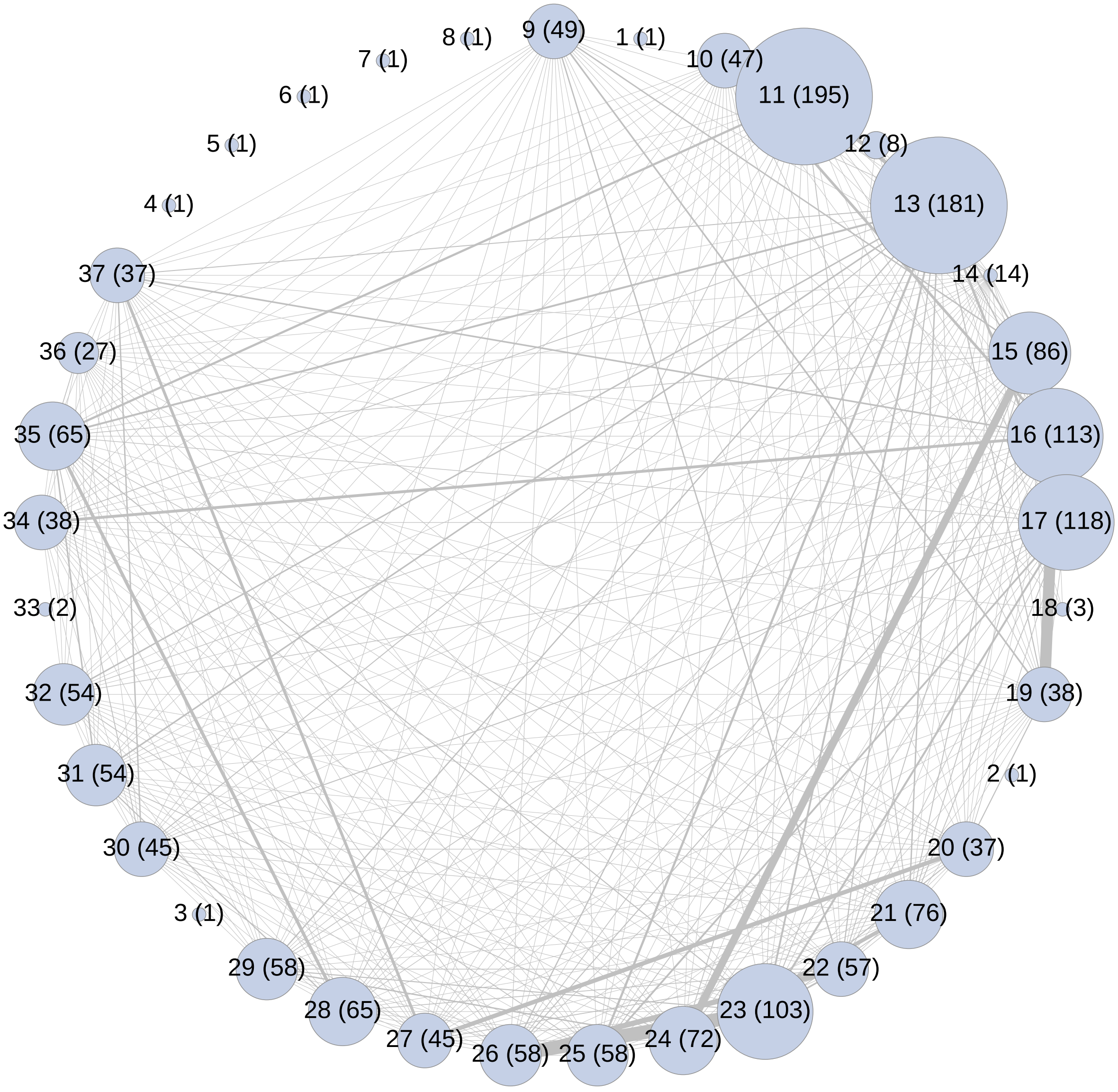


**Fig. 3.9** Graph representing the community structure identified by the Louvain algorithm in the weighted contact graph of Dartmouth dataset. In every node label, the community size is enclosed in parentheses.

A histogram of the size distribution of the 29 communities identified by the Louvain algorithm in the Dartmouth dataset is shown in Figure 3.10.

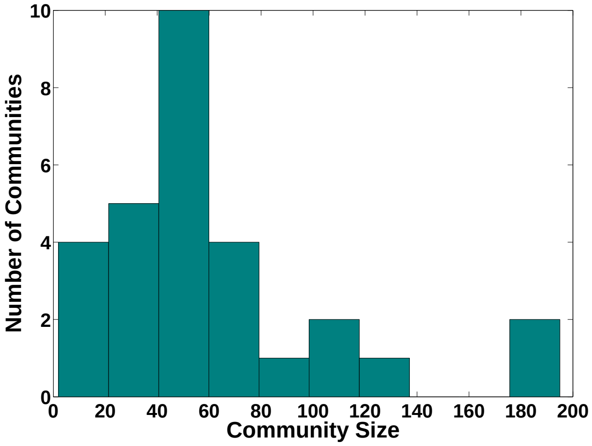


**Fig. 3.10** Size distribution of the 29 communities identified by the Louvain algorithm in the weighted contact graph of Dartmouth dataset

## 3.4 Summary

This chapter presented the analysis of node encounters recorded by two real world datasets. The diurnal nature of human activities was found to have a significant impact on the contact pattern of people in these datasets. They were further examined by constructing weighted contact graphs that captured the time spent by nodes in each other's vicinity. The highly modular structure exhibited by these graphs allowed us to divide them into small groups representing communities.

# Chapter 4

# Contact Graph based Routing Algorithm

An opportunistic routing algorithm can increase the efficiency of its delivery performance by assessing the usefulness of every forwarding opportunity. In order to accomplish this, it requires additional information about the node encounter pattern in the network. A weighted contact graph, built using the technique described in the previous chapter, is one possible source of information that a routing algorithm can use. Several of its characteristics, such as the varying strength of ties between the nodes, the high level of clustering and the presence of modular structure, clearly indicate that a node is suitable for delivering messages to only a fraction of destinations in the network. Incorporating information learned from a contact graph can help a routing algorithm to distinguish nodes based on their forwarding capabilities.

This chapter describes a novel opportunistic routing algorithm that utilizes information derived from a weighted contact graph to make forwarding decisions. As our algorithm strives to achieve good delivery performance while consuming less resources, it does not employ replication of messages. In Section 4.1, we present the factors that determine the selection of relays for messages in our scheme. Section 4.2 presents the details and pseudocode of our algorithm.

## 4.1 Information available to nodes

The weighted contact graph, obtained by following the steps described in Section 3.2, is used to provide every node in the network with three sets of information: the identity of all its neighbours, the strength of its ties with them and the list of members belonging to its community.

The first two sets represent the local neighbourhood information of a node. The inclusion of edge weights from the contact graph enables our algorithm to differentiate between strong and weak ties. This information is used to determine the ability of a node to deliver messages to destinations lying within its neighbourhood.

For destinations outside a node's neighbourhood, its usefulness as a relay is determined on the basis of its community affiliation or degree centrality. Community affiliation of a node simply denotes the community to which it belongs. As communities represent tightly-knit groups of nodes, a node belonging to the community of the destination is expected to have a better chance of encountering it or one of its neighbours. In cases where community information is inadequate for estimating the forwarding capability of a node, its degree centrality is considered. This measure is simply the number of neighbours connected to a node in the weighted contact graph (see Equation 2.4 in Section 2.4.1). It is treated as an indicator of the number of different nodes a relay is likely to encounter and can be easily calculated from the local information available at nodes.

Thus, the forwarding decisions of our algorithm are based entirely on information about the neighbourhood, community and degree centrality of nodes.

## 4.2 Algorithm

Contact Graph based routing is a replication-free algorithm in which the suitability of a node for carrying a message is determined by considering three factors in the following order: (i) strength of its tie to the destination; (ii) its community affiliation; and (iii) its degree centrality.

Preference is given to relays having a stronger connection to the destination of a message. The neighbourhood information available at nodes is used for this purpose. Community affiliation is considered only when the destination of a message is not present in the neighbourhoods of both encountering nodes. Thus, direct ties are given more importance than simply the presence of a destination within a community. This is motivated by the high values of clustering coefficient observed in the contact graphs prepared in the previous chapter (see Section 3.2.2). The property of transitivity (or clustering) suggests that even a weakly linked neighbour of a node is capable of delivering messages to it by passing them on to its other more strongly linked neighbours. On the other hand, a community that overlaps partially with the neighbourhood of a node, often contains a few members that are only indirectly linked to it through many intermediate nodes.

The degree centrality of encountering nodes is used only when both neighbourhood and community information are insufficient to differentiate their forwarding capability. This situation can arise in two scenarios: (i) when the destination is not present in the neighbourhood and community of either node; and (ii) when the destination is not present in the neighbourhood of either node but is present in both of their communities. In these two cases, our algorithm prefers the node with higher degree centrality as it is expected to encounter a greater number of nodes in the network.

The pseudo-code presented on the next page describes the forwarding decisions made at a typical node, say $A$, when it encounters another node, say $B$, in the network.

A message is forwarded to node $B$ if it has a stronger tie to its destination. However, if the destination lies neither in the neighbourhood of $A$ nor that of $B$, a few additional conditions are checked. If the destination lies exclusively in the community of node $B$, the message is forwarded to it irrespective of its degree. As mentioned previously, degree centrality is used only in scenarios where neighbourhood and community information are insufficient.

There are several possible ways of assigning priority to messages in the outgoing list of a node in Contact Graph based routing. In our implementation, the two encountering nodes, $A$ and $B$, first exchange all immediately deliverable messages. These are messages that have $A$ or $B$ as their destination. Next, the two nodes try to exchange any message for which the forwarding decision was made by comparing the strength of their ties to its destination. This is followed by messages for which the decision was made by using the community affiliation and degree centrality of nodes $A$ and $B$. The priority order becomes important when the contact duration is not long enough to transfer all messages in the outgoing list.

**Algorithm 1 Contact Graph based routing algorithm**: Forwarding decisions at node $A$ upon encountering node $B$

```
$C_A \leftarrow$ Community of node $A$
$C_B \leftarrow$ Community of node $B$
$N_A \leftarrow$ Neighbourhood of node $A$
$N_B \leftarrow$ Neighbourhood of node $B$
for all messages $m \in buffer(A)$ do
    $D \leftarrow$ Destination of $m$
    $w_{AD} \leftarrow$ Strength of tie between nodes $A$ and $D$
    $w_{BD} \leftarrow$ Strength of tie between nodes $B$ and $D$
    if $D == B$ then                                   ▷ Node $B$ is the destination
        $outgoingList(B).add(m)$
    else if $(w_{BD} > w_{AD})$ then                   ▷ Node $B$ has stronger tie to destination
        $outgoingList(B).add(m)$
    else if $(w_{AD} == 0) \wedge (w_{BD} == 0)$ then
                                                       ▷ Destination not in either neighbourhood
        if $D \in C_B \wedge D \notin C_A$ then         ▷ Destination in node $B$'s community only
            $outgoingList(B).add(m)$
        else if $D \in (C_A \cap C_B) \vee D \notin (C_A \cup C_B)$ then
                                                       ▷ Destination in both or neither communities
            if $|N_B| > |N_B|$ then                    ▷ Node $B$ has higher degree centrality
                $outgoingList(B).add(m)$
            end if
        end if
    end if
end for
```

## 4.3 Summary

This chapter described the Contact Graph based routing algorithm. It is a replication-free algorithm that uses neighbourhood, community and degree information derived from a weighted contact graph to make forwarding decisions. It gives preference to relays having a stronger tie to the destination of a message. Community affiliation and degree centrality are considered when neighbourhood information is insufficient to make a forwarding decision. We have also provided a priority order for transferring messages between the nodes while using Contact Graph based routing.

# Chapter 5

# Performance Evaluation

We have conducted a comprehensive assessment of Contact Graph based routing by comparing its performance with other well-known opportunistic forwarding schemes. In order to simulate networks with realistic node contact pattern, we have used real world datasets to create the simulation scenarios. As opportunistic routing algorithms face a trade-off between delivery performance and resource consumption, the evaluation metrics have been carefully selected to capture their transmission efficiency, end-to-end delay and storage cost in addition to delivery success rate.

We begin by providing, in Section 5.1, a brief overview of the simulation environment we used. Section 5.2 describes the simulation setups and performance metrics that are used for comparing the routing algorithms. In Sections 5.3 and 5.4 we provide the results obtained from each of the simulation scenarios and discuss their implications.

## 5.1 Simulator

The simulations have been performed using the Opportunistic Network Environment (ONE) simulator (version 1.4.1) created by Keränen et al. [70]. It is written in Java 1.6 and can be easily extended to include additional features by creating appropriately derived *subclasses*. The main services provided by this simulator are:

- **Mobility modelling**: The simulator is capable of generating node movement using both synthetic mobility models and events derived from real world datasets.

- **Routing simulation**: The simulator provides a framework for defining the routing algorithms used by nodes. Opportunistic communication is enabled by providing a mechanism for detecting nodes that are within each other's transmission range.

- **Report generation**: The report modules provided with the simulator are capable of gathering data from events related to node contacts and message transfer. The output files written by them are used for performance analysis.

The ONE simulator does not implement the details of the physical layer such as the effect of distance and interference on data transmission. However, the speed at which nodes transfer messages can be configured.

The versatility of this simulator and the ability to take input from external datasets make it suitable for use in our work.

## 5.2 Simulation setup

The performance of Contact Graph based routing has been compared with four other routing algorithms in two different simulation scenarios.

### 5.2.1 Routing algorithms

The four additional algorithms can be divided into two categories based on whether or not they employ replication of messages.

- **Replication-based algorithms**

  1. Epidemic routing (see Section 2.1.1)
  2. Distributed Bubble (see Section 2.4.4)

- **Replication-free algorithms**

  3. PRoPHET (see Section 2.3.1)
  4. SimBet (see Section 2.4.3)

The parameters of PRoPHET are set to values prescribed in [17]. For SimBet, we use the parameter values described in [29]. K-Clique community detection (K = 3) [28] is used in Distributed Bubble algorithm with the value of familiarity threshold chosen such that the best possible delivery performance is achieved in the scenario under consideration.

### 5.2.2 Simulation scenarios

Two separate simulation scenarios have been created by using direct contact events from the Infocom 2006 dataset (see Section 3.1.1) and inferred contact events from the Dartmouth dataset (see Section 3.1.2).

#### Infocom scenario

Contact events between the iMotes of Infocom 2006 dataset are extracted for the period from April 24, 2006 to April 26, 2006 and provided as input to the ONE simulator. This creates a simulation scenario that spans 3 days and contains 78 mobile nodes.

In order to initialize the Contact Graph based routing algorithm in this scenario, we have used the weighted contact graph prepared from the Infocom 2006 dataset in Section 3.2.

#### Dartmouth scenario

From the Dartmouth dataset, inferred contact events between the 1811 nodes identified in Section 3.1.2 are obtained for the period from October 27, 2003 to October 31, 2003. These are provided as input to the ONE simulator in order to create a simulation scenario that spans 5 days.

In this scenario, neighbourhood, degree and community information learned from past encounters between the 1811 nodes are used to initialize the Contact Graph based routing algorithm. For this purpose, the weighted contact graph prepared from data that spans the 14 day period from October 12, 2003 to October 25, 2003 is used (see Section 3.2). Past encounters could not be used in the Infocom scenario as data is available only for 3 complete days.

### 5.2.3 Simulation parameters

The Infocom and Dartmouth scenarios share a few common simulation parameters. In both of them, the nodes are assumed to be equipped with a Bluetooth interface having a transfer rate of 2.1 Mbps. A built-in functionality provided by the ONE simulator is used to generate messages of size 4 MB at intervals of 30 seconds throughout the duration of both simulations. The source and destination pair for the messages are selected uniformly at random from all the nodes in the network.

In both scenarios, no restrictions have been imposed on the buffer capacity of nodes and the time-to-live (TTL) of messages. This enables us to compare the delay performance and memory consumption of routing algorithms.

### 5.2.4 Performance metrics

The performance of routing algorithms is compared by using the following metrics:

- **Delivery Ratio**: The fraction of all generated messages that a routing algorithm successfully delivers before the simulation ends.
- **Overhead Ratio**: The transmission efficiency of a routing algorithm as calculated by the following equation:

$$\text{Overhead ratio} = \frac{\text{Number of Messages Relayed}}{\text{Number of Delivered Messages}} - 1 \tag{5.1}$$

- **Average and Median Hop Count for Delivered Messages**: The average and median number of hops messages traverse before they reach their destination.
- **Average and Median Delivery Latency (in hours)**: The average and median delay experienced by messages in reaching their destination.
- **Average Buffer Occupancy (in number of messages)**: The number of undelivered messages (including duplicates) present in the buffer of a node on average over the entire duration of a simulation.
- **Maximum Buffer Occupancy (in number of messages)**: The highest buffer occupancy recorded in the network at some point in a simulation.

## 5.3 Results from Infocom scenario

This section presents the results obtained from evaluating every routing algorithm under consideration 10 times on the Infocom 2006 network. In each of the 10 simulations, 8640 messages were generated randomly according to the policy described in Section 5.2.3. The familiarity threshold for Distributed Bubble was configured to 1 hour as it gave the best performance in terms of delivery ratio.

Figure 5.1 depicts the average number of messages successfully delivered by the routing algorithms as the simulation progresses. As expected, the delivery pattern of messages follows the diurnal nature of user activity in the network (see Figure 3.1 in Section 3.1.1). Every algorithm delivers most of its messages in the interval from 8 a.m. to 8 p.m. on all three days of the simulation.

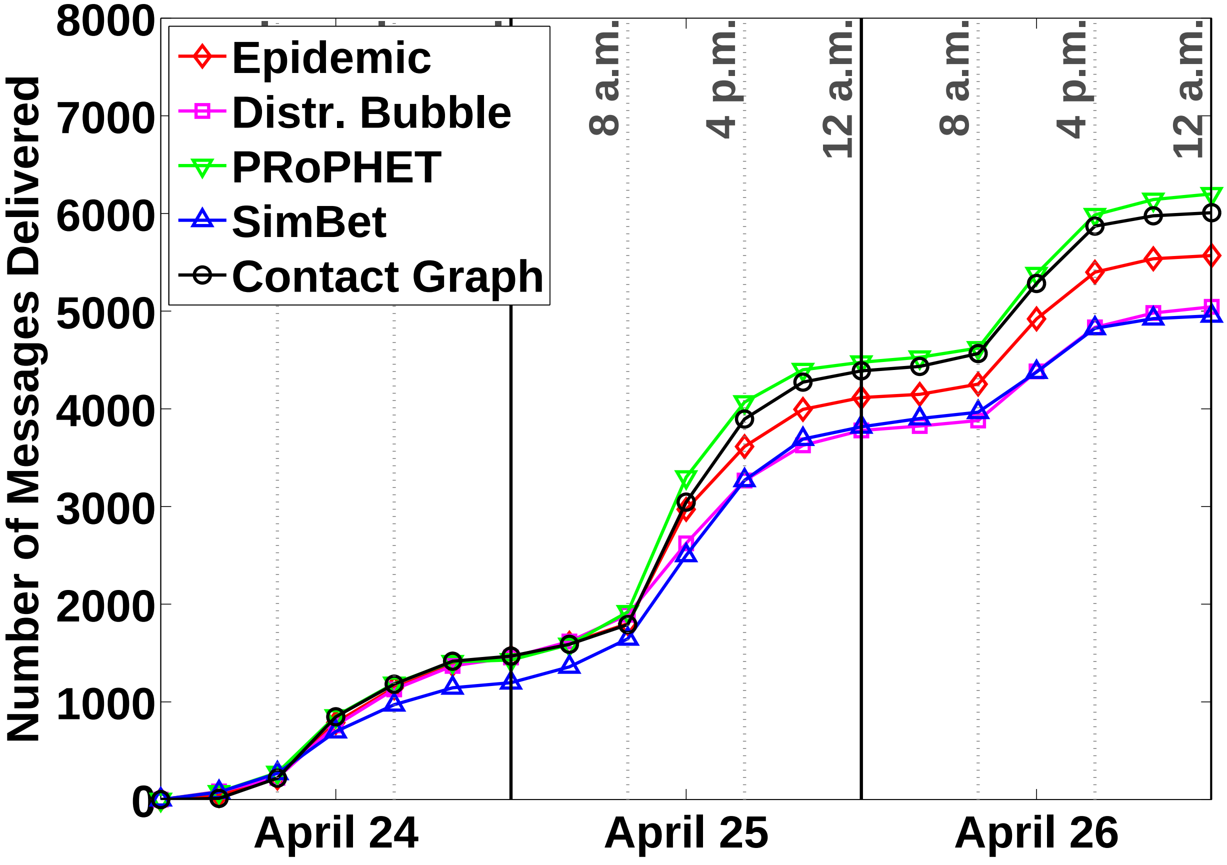


**Fig. 5.1** Number of messages delivered over time in the Infocom scenario (average over 10 runs)

### 5.3.1 Delivery Ratio

| | Delivered Messages | Delivery Ratio | |
|---|---|---|---|
| | Mean | Mean | SD |
| Epidemic | 5570 | 0.64 | 0.003 |
| Distributed Bubble | 5045 | 0.58 | 0.004 |
| PRoPHET | 6200 | 0.72 | 0.006 |
| SimBet | 4952 | 0.57 | 0.005 |
| Contact Graph | 6008 | 0.70 | 0.004 |

**Table 5.1** Aggregate delivery performance of routing algorithms over 10 simulation runs on the Infocom 2006 network

Table 5.1 shows the mean and standard deviation of the delivery ratio achieved by the routing algorithms when the simulation ends. PRoPHET achieves the highest delivery ratio by delivering about 6200 messages on average across all the simulations. It is closely followed by Contact Graph based routing that delivers about 6008 messages on average.

Epidemic routing and Distributed Bubble deliver fewer messages than PRoPHET and Contact Graph, suggesting that there are limited benefits of replicating messages in this network. As a node acquires more and more messages, there is increased contention between them when forwarding opportunities arise. Only those messages that are replicated sufficiently have a good chance of reaching their destination. Hence, unequal consumption of limited resources by different messages results in inferior delivery performance of these algorithms.

SimBet routing, which does not employ replication, has the lowest delivery ratio. It does not perform well as it calculates betweenness and similarity from unweighted ego-graphs that only consider whether two nodes have met at least once in the past. This simple method of constructing ego-graphs harms its delivery performance in a scenario where nodes encounter each other quite often.

### 5.3.2 Overhead Ratio

An important factor determining the efficiency of a forwarding scheme is the number of transmissions required by it in order to achieve a particular level of delivery performance. Low overhead ratio is desirable as it reduces the energy consumed by nodes.

| | # Transmissions | Overhead Ratio | |
|---|---|---|---|
| | Mean | Mean | SD |
| Epidemic | 189410 | 33.0 | 0.17 |
| Distributed Bubble | 142080 | 27.2 | 0.26 |
| PRoPHET | 99934 | 15.1 | 0.14 |
| SimBet | 27807 | 4.6 | 0.05 |
| Contact Graph | 27585 | 3.6 | 0.02 |

**Table 5.2** Delivery overhead ratio achieved by routing algorithms over 10 simulation runs on the Infocom 2006 network

Table 5.2 shows that the routing algorithms under consideration differ greatly in their transmission efficiency. Epidemic and Distributed Bubble algorithms have significantly higher overhead ratio in comparison to the other three. This behaviour is expected as replication based forwarding schemes require extra transmissions to create new copies of messages.

PRoPHET has the highest overhead ratio among the replication-free routing algorithms. Its chief cause is the fluctuating delivery predictability values at nodes that leads to messages being forwarded more often than would otherwise be necessary. As the excess transmissions do not translate to a significant increase in delivered messages, PRoPHET has about 4.2 times larger overhead in comparison to Contact Graph based routing which avoids unnecessary transmissions. On the other hand, the average number of transmissions in SimBet routing is low as the relative importance of nodes changes slowly during the simulation.

### 5.3.3 Hop Count

| | Avg. Hop Count | | Med. Hop Count | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 2.2 | 0.02 | 2 | 0 |
| Distributed Bubble | 2.5 | 0.02 | 2 | 0 |
| PRoPHET | 13.3 | 0.12 | 8 | 0 |
| SimBet | 3.9 | 0.02 | 3 | 0 |
| Contact Graph | 3.9 | 0.01 | 4 | 0 |

**Table 5.3** Average and median number of hops traversed by messages before they are delivered by the routing algorithms in the Infocom scenario. Mean and standard deviation are obtained from 10 sample points generated by different simulation runs.

It is evident from Table 5.3 that all routing algorithms except PRoPHET are able to deliver their messages over short paths. The main factor contributing to this outcome is the small and dense nature of this network. Most of the nodes encounter each other relatively often. Significantly longer delivery paths in PRoPHET are caused by the same factors that led to its high overhead ratio.

### 5.3.4 Delivery Latency

In order to assess the effectiveness of a routing algorithm, it is important to consider its delivery ratio together with the delay experienced by messages in reaching their destination. Ideally, opportunistic forwarding schemes must strive to achieve good delivery performance with low latency.

Table 5.4 presents the average and median latency (in hours) with which routing algorithms deliver their messages in Infocom scenario. It can be seen that PRoPHET achieves the lowest average and median delays of 8.4 hours and 5.2 hours respectively. When Contact Graph based routing and SimBet are used, the average delay increases by about 37 minutes and 82 minutes respectively.

| | Avg. Delay (hrs) | | Med. Delay (hrs) | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 10.4 | 0.17 | 8.0 | 0.23 |
| Distributed Bubble | 11.0 | 0.30 | 8.4 | 0.38 |
| PRoPHET | 8.4 | 0.08 | 5.2 | 0.11 |
| SimBet | 9.8 | 0.16 | 6.2 | 0.07 |
| Contact Graph | 9.0 | 0.14 | 6.1 | 0.14 |

**Table 5.4** Average and median delay (in hours) experienced by delivered messages in Infocom 2006 network across 10 simulation runs

It is also evident that Epidemic and Distributed Bubble algorithms suffer greater delays in delivering their messages. In both these algorithms, as mentioned earlier, messages that have been created early on are replicated more by the nodes and have a greater chance of reaching their destination. This bias towards messages that have spent more time in the network leads to greater average and median delays in these algorithms.

### 5.3.5 Buffer Occupancy

In order to estimate the storage consumption of a forwarding scheme, the number of messages present in the buffer of every node is sampled at intervals of one hour during a simulation and the average is calculated. This metric is shown as the average buffer occupancy (in messages) in Table 5.5. Another interesting quantity derived from the hourly samples is the maximum buffer occupancy that was observed at some node in the network during a simulation. This quantity, which is also presented in Table 5.5, enables us to evaluate the maximum burden of storage that a node can experience when using a particular opportunistic routing algorithm.

As expected, replication based Epidemic and Distributed Bubble algorithms have significantly higher average and maximum buffer occupancy in comparison to the other three. The average buffer occupancy of replication-free algorithms is dependent on the number of messages delivered by them. Hence, PRoPHET has the lowest average occupancy as it delivers the most messages. It is closely followed by Contact Graph based routing and SimBet which deliver fewer messages. In terms of maximum buffer occupancy, Contact Graph

| | Avg. Occup. (msgs) | | Max. Occup. (msgs) | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 629 | 6.2 | 2144 | 85.3 |
| Distributed Bubble | 357 | 8.8 | 2040 | 147.8 |
| PRoPHET | 16 | 0.2 | 299 | 24.3 |
| SimBet | 23 | 0.3 | 509 | 29.7 |
| Contact Graph | 17 | 0.2 | 127 | 7.8 |

**Table 5.5** Average and maximum buffer occupancy (in messages) of nodes in the Infocom 2006 network while implementing different forwarding schemes

based routing has the best performance. This implies that it reduces the probability that messages are accumulated at certain nodes.

The low buffer occupancy (both average and maximum) of Contact Graph based routing suggests that its delivery performance will be least affected by any decrease in the buffer capacity of nodes.

## 5.4 Results from Dartmouth scenario

The performance metrics in this section were also obtained by carrying out 10 simulations per routing algorithm. Every simulation spanned a period of 5 days during which 14400 messages were generated randomly according to the policy described in Section 5.2.3. The familiarity threshold for Distributed Bubble was set to 5.5 hours in this scenario.

The delivery performance of routing algorithms is depicted in Figure 5.2. On all five days of the simulation, most of the messages are delivered in the interval from 8 a.m. to midnight. This is consistent with the aggregate contact pattern of nodes in the network (see Figure 3.4 in Section 3.1.2).

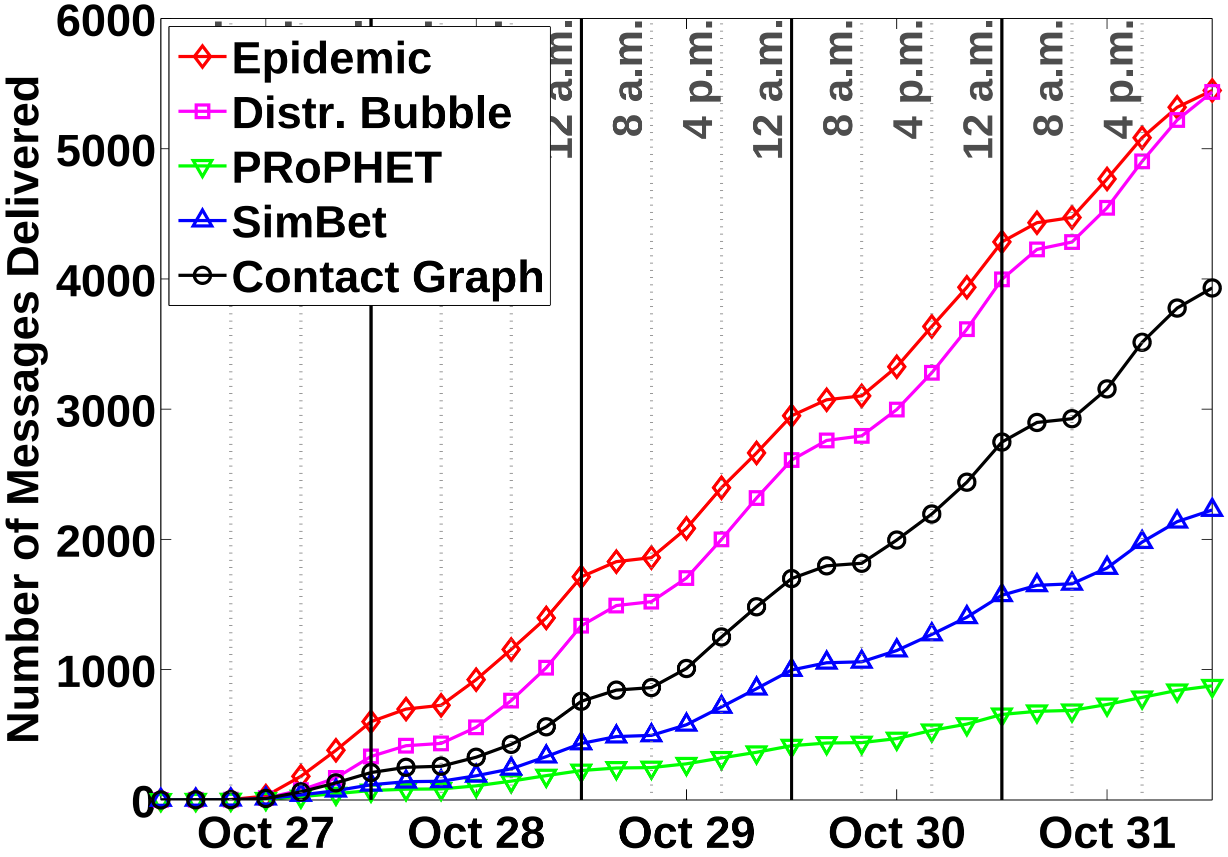


**Fig. 5.2** Number of messages delivered over time in the Dartmouth scenario (average over 10 runs)

### 5.4.1 Delivery Ratio

It is clear from Figure 5.2 that routing algorithms differ significantly in the number of messages delivered by them. As this network is much larger than the Infocom 2006 network, replication based Epidemic and Distributed Bubble algorithms outperform the others. The delivery ratios achieved by all the algorithms are shown in Table 5.6.

Among the replication-free algorithms, Contact Graph based routing easily surpasses the other two. It delivers about 1.8 and 4.5 times more messages than SimBet and PRoPHET respectively.

PRoPHET has by far the lowest delivery ratio in the Dartmouth scenario. As a typical node encounters only a very small fraction of other nodes in the network, the delivery predictability values are not propagated effectively. This impedes the selection of suitable

| | Delivered Messages | Delivery Ratio | |
|---|---|---|---|
| | Mean | Mean | SD |
| Epidemic | 5447 | 0.38 | 0.003 |
| Distributed Bubble | 5435 | 0.38 | 0.005 |
| PRoPHET | 876 | 0.06 | 0.002 |
| SimBet | 2226 | 0.15 | 0.004 |
| Contact Graph | 3931 | 0.27 | 0.002 |

**Table 5.6** Delivery ratio of routing algorithms over 10 simulation runs on the Dartmouth network

relays for messages by PRoPHET and degrades its delivery performance. On the other hand, the sparseness of the Dartmouth network is favourable to the use of unweighted ego-graphs by SimBet and improves its delivery performance. Consequently, it no longer has the lowest delivery ratio as was the case in the Infocom scenario.

### 5.4.2 Overhead Ratio

Table 5.7 presents the transmission efficiency of routing algorithms in terms of the overhead ratio achieved by them.

| | # Transmissions | Overhead Ratio | |
|---|---|---|---|
| | Mean | Mean | SD |
| Epidemic | 6888716 | 1263.8 | 9.06 |
| Distributed Bubble | 4080875 | 749.9 | 7.44 |
| PRoPHET | 389882 | 444.5 | 13.83 |
| SimBet | 77140 | 33.7 | 0.73 |
| Contact Graph | 66320 | 15.9 | 0.11 |

**Table 5.7** Delivery overhead ratio achieved by routing algorithms over 10 simulation runs on the Dartmouth network

As the simulations last five days on a large network, Epidemic and Distributed Bubble algorithms have much higher average number of transmissions in comparison to the other three. However, the significantly better delivery performance of these algorithms means that the discrepancies in the overhead ratios are not as large. Note that the use of node centrality and community affiliation by Distributed Bubble enables it to deliver almost the same number of messages as Epidemic routing while having about 41% lower overhead.

Contact Graph based routing has the lowest overhead ratio in this simulation scenario. It is able to deliver 77% more messages than SimBet while having 14% fewer transmissions. As was the case in the Infocom scenario, PRoPHET has the highest number of transmissions and overhead ratio among the replication-free algorithms.

### 5.4.3 Hop Count

| | Avg. Hop Count | | Med. Hop Count | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 8.9 | 0.06 | 8.2 | 0.42 |
| Distributed Bubble | 6.5 | 0.05 | 6.0 | 0 |
| PRoPHET | 25.6 | 0.78 | 17.3 | 0.67 |
| SimBet | 7.3 | 0.10 | 7.0 | 0 |
| Contact Graph | 7.1 | 0.05 | 7.0 | 0 |

**Table 5.8** Average and median number of hops traversed by messages before they are delivered by the routing algorithms in the Dartmouth scenario

Table 5.8 shows the average and median lengths of the paths through which routing algorithms deliver their messages in the Dartmouth network. Significantly longer paths are observed in the case of PRoPHET as the delivery predictability values are not propagated effectively. All the other algorithms are able to find shorter paths.

### 5.4.4 Delivery Latency

In the Dartmouth simulation scenario, there is a very wide variation in delivery performance. Because latency is calculated only for the messages that are successfully delivered, it is only reasonable to compare the latencies of algorithms that deliver a similar number of messages.

| | Avg. Delay (hrs) | | Med. Delay (hrs) | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 39.1 | 0.18 | 37.2 | 0.30 |
| Distributed Bubble | 42.4 | 0.28 | 40.4 | 0.36 |
| PRoPHET | 29.5 | 0.67 | 21.5 | 0.78 |
| SimBet | 38.0 | 0.69 | 33.5 | 0.80 |
| Contact Graph | 37.2 | 0.59 | 33.0 | 0.66 |

**Table 5.9** Average and median delay (in hours) experienced by delivered messages in the Dartmouth network across 10 simulation runs

The values presented in Table 5.9 show that Epidemic routing, due to its higher replication of messages, is able to achieve lower delays in comparison to Distributed Bubble. Contact Graph based routing achieves similar median latency to SimBet while delivering considerably more messages.

The low latency values of PRoPHET are not meaningful because its delivery performance is so poor.

### 5.4.5 Buffer Occupancy

Table 5.10 summarizes the average and maximum buffer occupancy observed at nodes in the Dartmouth network when using different routing algorithms.

As expected, Epidemic and Distributed Bubble algorithms have much higher storage consumption in comparison to the other three. However, lower replication of messages by Distributed Bubble results in it having 55% less average buffer occupancy than Epidemic

| | Avg. Occup. (msgs) | | Max. Occup. (msgs) | |
|---|---|---|---|---|
| | Mean | SD | Mean | SD |
| Epidemic | 1553 | 2.95 | 6585 | 87.7 |
| Distributed Bubble | 694 | 13.33 | 5896 | 113.8 |
| PRoPHET | 4 | 0.01 | 1087 | 24.9 |
| SimBet | 4 | 0.01 | 414 | 13.2 |
| Contact Graph | 3 | 0.01 | 232 | 9.6 |

**Table 5.10** Average and maximum buffer occupancy (in messages) of nodes in the Dartmouth network while implementing different forwarding schemes

routing.

As was the case in the Infocom scenario, the average buffer occupancy of replication-free algorithms is dependent on the number of messages they deliver. Thus, Contact Graph based routing achieves the best performance in this regard. Moreover, its low maximum occupancy implies that the algorithm reduces the concentrated accumulation of messages at individual nodes in the network in comparison to PRoPHET and SimBet.

## 5.5 Summary

This chapter presented an evaluation of the Contact Graph based routing algorithm by comparing its performance with existing opportunistic forwarding schemes. The simulation scenarios were created from two real world datasets that have distinct characteristics.

In the smaller Infocom scenario, Contact Graph based routing achieved an impressive delivery performance. It was able to deliver more messages than replication based forwarding schemes while consuming much fewer resources. It had the lowest overhead ratio among all the routing algorithms that were evaluated.

However, in the larger Dartmouth network, Contact Graph based routing was unable to match the delivery ratio achieved by replication based forwarding schemes. It delivered approximately 28% fewer messages than Epidemic and Distributed Bubble algorithms but achieved a dramatically lower overhead ratio and buffer occupancy. The other two replication-free algorithms had much poorer results in comparison.

The performance of the Contact Graph based routing algorithm shows that information learned from an appropriately constructed weighted contact graph can improve delivery performance while lowering other costs. Further improvement in the performance of this routing algorithm can be attained by incorporating a replication strategy that achieves the desired balance between delivery and overhead.

# Chapter 6

# Conclusion and Future Work

## 6.1 Summary and Discussion

In this thesis, we have proposed a novel opportunistic routing algorithm that makes forwarding decisions based on information obtained by processing node encounters within a network. By aggregating contact events into a weighted contact graph, we were able to learn about the neighbourhoods and community affiliation of nodes. This information was utilized by the proposed Contact Graph based routing algorithm to select suitable relays for messages. The simulations we performed show that it is capable of achieving good delivery success ratio while limiting the network resources consumed.

The literature review chapter described how the opportunistic routing algorithms have evolved to include additional information about the network in order to improve their delivery performance. The earliest routing schemes that relied on replication or coding of messages did not assess the forwarding capabilities of nodes. The algorithms that were introduced later tried to incorporate additional information about the network in order to identify important nodes. This approach can be used to achieve a favourable trade-off between delivery performance and resource consumption in opportunistic networks.

In Chapter 3 we argued that useful information for opportunistic routing can be obtained by constructing weighted contact graphs that capture the time spent by nodes in each other's vicinity. Significant deviation from randomness was observed in the topological properties of the sample graphs that were built using the technique we described. Their modular structure enabled us to divide them into communities of densely connected nodes.

In Chapter 4, the properties exhibited by the graphs presented in the previous chapter were used to design the Contact Graph based routing algorithm. We have provided a detailed discussion of the factors that determine the selection of relays in our algorithm and the reasons for assigning different importance to them.

Chapter 5 presented an evaluation of the Contact Graph based routing algorithm by comparing its performance with four other opportunistic forwarding schemes. We have provided details of the simulation environment that was used. Real world datasets were utilized to create two simulation scenarios that have distinct characteristics. The performance of our routing algorithm shows that information obtained from a weighted contact graph can be utilized to improve delivery performance while lowering other costs, such as overhead ratio and buffer occupancy.

## 6.2 Future Work

From the simulation results on the Dartmouth network, we observe that further improvement in the delivery performance of Contact Graph based routing can be achieved by incorporating a suitable replication strategy. We expect that by allowing limited duplication of messages, this algorithm can attain higher delivery ratio with slightly increased levels of transmission overhead and buffer occupancy.

Another possible direction in which the current work can be extended is to define a distributed version of the proposed routing algorithm. In this thesis, we have not addressed the mechanism for gathering the information necessary to build a weighted contact graph. In order to implement a distributed version of our algorithm, nodes will have to devise a strategy to obtain the information about their neighbourhood and adopt a decentralized algorithm to detect the members belonging to their community.

# References

[1] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *IEEE Commun. Mag.*, vol. 44, no. 11, pp. 134–141, Nov. 2006.

[2] 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,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, pp. 96–107, Oct. 2002.

[3] T. Small and Z. J. Haas, “The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way),” in *Proc. ACM Int. Symp. Mobile ad hoc Networking & Comput.*, Annapolis, MD, Jun. 2003, pp. 233–244.

[4] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: rethinking connectivity in developing nations,” *IEEE Computer*, vol. 37, no. 1, pp. 78–83, Jan. 2004.

[5] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation,” *IEEE Trans. Mobile Computing*, vol. 11, pp. 821–834, May 2012.

[6] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 145–158, Oct. 2004.

[7] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proc. Conf. Applicat., Technologies, Architectures, and Protocols Comput. Commun.*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[8] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” Duke Univ., Durham, NC, Tech. Rep., 2000.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. ACM SIGCOMM Workshop Delay-tolerant Netw.*, Philadelphia, PA, Aug. 2005, pp. 252–259.

[10] Y. Wang, S. Jain, M. Martonosi, and K. Fall, “Erasure-coding based routing for opportunistic networks,” in *Proc. ACM SIGCOMM Workshop Delay-tolerant Netw.*, Philadelphia, PA, Aug. 2005, pp. 229–236.

[11] I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[12] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proc. Annu. ACM Symp. Theory of Comput.*, El Paso, TX, May 1997, pp. 150–159.

[13] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[14] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 371–381, Feb. 2003.

[15] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” in *Proc. Annu. Joint Conf. IEEE Comput. and Commun. Soc.*, Miami, FL, Mar. 2005, pp. 2235–2245.

[16] J. Widmer and J.-Y. Le Boudec, “Network coding for efficient communication in extreme networks,” in *Proc. ACM SIGCOMM Workshop Delay-tolerant Netw.*, Philadelphia, PA, Aug. 2005, pp. 284–291.

[17] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[18] S. Grasic, E. Davies, A. Lindgren, and A. Doria, “The evolution of a dtn routing protocol - prophetv2,” in *Proc. ACM Workshop Challenged Netw.*, Las Vegas, NV, Sep. 2011, pp. 27–30.

[19] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive routing for intermittently connected mobile ad hoc networks,” in *Proc. IEEE Int. Symp. World of Wireless Mobile and Multimedia Netw.*, Taormina, Italy, Jun. 2005, pp. 183–189.

[20] B. Burns, O. Brock, and B. Levine, “Mv routing and capacity building in disruption tolerant networks,” in *Proc. Annu. Joint Conf. IEEE Comput. and Commun. Soc.*, Miami, FL, Mar. 2005, pp. 398–408.

[21] K. Tan, Q. Zhang, and W. Zhu, “Shortest path routing in partially connected ad hoc networks,” in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, Dec. 2003, pp. 1038–1042.

[22] E. Jones, L. Li, J. Schmidtke, and P. Ward, “Practical routing in delay-tolerant networks,” *IEEE Trans. Mobile Comput.*, vol. 6, pp. 943–959, Aug. 2007.

[23] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[24] P. Hui and J. Crowcroft, “How small labels create big improvements,” in *Proc. Annu. IEEE Int. Conf. Pervasive Comput. and Commun. Workshops*, White Plains, NY, Mar. 2007, pp. 65–70.

[25] S. Fortunato, “Community detection in graphs,” *Physics Rep.*, vol. 486, no. 3 - 5, pp. 75 – 174, Feb. 2010.

[26] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[27] M. E. J. Newman, “Analysis of weighted networks,” *Phys. Rev. E*, vol. 70, no. 5, p. 056131, Nov. 2004.

[28] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[29] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proc. ACM Int. Symp. Mobile ad hoc Netw. and Comput.*, Montreal, QC, Sep. 2007, pp. 32–40.

[30] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: social-based forwarding in delay tolerant networks,” in *Proc. ACM Int. Symp. Mobile ad hoc Netw. and Comput.*, Hong Kong, Hong Kong, May 2008, pp. 241–250.

[31] A. Mtibaa, M. May, C. Diot, and M. Ammar, “Peoplerank: Social opportunistic forwarding,” in *Proc. Conf. Inform. Commun.*, San Diego, CA, Mar. 2010, pp. 111–115.

[32] P. V. Marsden, “Egocentric and sociocentric measures of network centrality,” *Social Netw.*, vol. 24, no. 4, pp. 407 – 422, Oct. 2002.

[33] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *Proc. ACM/IEEE Int. Workshop Mobility in the evolving internet architecture*, Kyoto, Japan, Aug. 2007, pp. 7:1–7:8.

[34] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A.-K. Pietilainen, and C. Diot, “Are you moved by your social network application?” in *Proc. Workshop Online Social Netw.*, Seattle, WA, Aug. 2008, pp. 67–72.

[35] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Comput. Netw. and ISDN Syst.*, vol. 30, no. 1-7, pp. 107 – 117, Apr. 1998.

[36] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” *The Kluwer Int. Series Eng. and Comput. Sci.*, vol. 353, pp. 153–181, 1996.

[37] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Trans. Mobile Comput.*, vol. 6, pp. 606–620, Jun. 2007.

[38] M. Musolesi and C. Mascolo, “A community based mobility model for ad hoc network research,” in *Proc. Int. Workshop Multi-hop ad hoc Netw.: from theory to reality*, Florence, Italy, May 2006, pp. 31–38.

[39] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, “User mobility for opportunistic ad-hoc networking,” in *Proc. IEEE Workshop Mobile Comput. Syst. and Applicat.*, Low Wood, UK, Dec. 2004, pp. 41–50.

[40] N. Eagle and A. (Sandy) Pentland, “Reality mining: sensing complex social systems,” *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, Mar. 2006.

[41] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in *Proc. ACM SIGCOMM Workshop Delay-tolerant Netw.*, Philadelphia, PA, Aug. 2005, pp. 244–251.

[42] A.-K. Pietilänen and C. Diot, “Dissemination in opportunistic social networks: the role of temporal communities,” in *Proc. ACM Int. Symp. Mobile ad hoc Netw. and Comput.*, Jun. 2012, pp. 165–174.

[43] T. Henderson, D. Kotz, and I. Abyzov, “The changing usage of a mature campus-wide wireless network,” in *Proc. Annu. Int. Conf. Mobile Comput. and Netw.*, Philadelphia, PA, Sep. 2004, pp. 187–201.

[44] D. Kotz and K. Essien, “Analysis of a campus-wide wireless network,” *Wireless Netw.*, vol. 11, no. 1-2, pp. 115–133, Jan. 2005.

[45] M. McNett and G. M. Voelker, “Access and mobility of wireless pda users,” *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 2, pp. 40–55, Apr. 2005.

[46] W. jen Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, “Modeling time-variant user mobility in wireless mobile networks,” in *Proc. IEEE Int. Conf. Comput. Commun.*, Anchorage, AK, May 2007, pp. 758–766.

[47] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *Proc. ACM SIGMOBILE Workshop Mobility Models*, Hong Kong, Hong Kong, May 2008, pp. 33–40.

[48] S. Ioannidis and A. Chaintreau, "On the strength of weak ties in mobile social networks," in *Proc. ACM EuroSys Workshop Social Netw. Syst.*, Nuremberg, Germany, Mar. 2009, pp. 19–25.

[49] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[50] W. Feller, *An introduction to probability theory and its applications.* New York, NY: Wiley, 1968.

[51] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in *Proc. Annu. ACM Int. Conf. Mobile Comput. and Netw.*, Montreal, QC, Sep. 2007, pp. 183–194.

[52] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD data set dartmouth/campus (v. 2009-09-09)," Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus, Sep. 2009.

[53] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Apr. 1998.

[54] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[55] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci.*, vol. 5, pp. 17–61, 1960.

[56] S. Milgram, "The small world problem," *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

[57] J. Travers and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, no. 4, pp. 425–443, Dec. 1969.

[58] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 888–905, Aug. 2000.

[59] I. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1944–1957, Nov. 2007.

[60] M. Meila and J. Shi, “A random walks view of spectral segmentation,” in *Proc. Int. Workshop Artificial Intell. Stat.*, Jan. 2001.

[61] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Proc. Advances Neural Inform. Process. Syst.*, Dec. 2001, pp. 849–856.

[62] S. S. Tabatabaei, M. Coates, and M. Rabbat, “GANC: Greedy agglomerative normalized cut for graph clustering,” *Pattern Recognition*, vol. 45, no. 2, pp. 831 – 843, Feb. 2012.

[63] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *J. Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[64] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Phys. Rev. E*, vol. 69, no. 6, p. 066133, Jun. 2004.

[65] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E*, vol. 70, no. 6, p. 066111, Dec. 2004.

[66] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Phys. Rev. E*, vol. 74, no. 3, p. 036104, Sep. 2006.

[67] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, “Modularity from fluctuations in random graphs and complex networks,” *Phys. Rev. E*, vol. 70, no. 2, p. 025101, Aug. 2004.

[68] J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization,” *Phys. Rev. E*, vol. 72, no. 2, p. 027104, Aug. 2005.

[69] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *J. Stat. Mech.: Theory and Experiment*, vol. 2008, no. 10, p. P10008, Oct. 2008.

[70] A. Keränen, J. Ott, and T. Kärkkäinen, “The one simulator for dtn protocol evaluation,” in *Proc. Int. Conf. Simulation Tools and Techniques*, Rome, Italy, Mar. 2009, pp. 55:1–55:10.