UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XXVciclo

Settore Scientifico Disciplinare ING-INF/03

Tesi di Dottorato

Using Multi-layer Social Networks for Opportunistic Routing

Socievole Annalisa

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XXV ciclo

Tesi di Dottorato

Using Multi-layer Social Networks for Opportunistic Routing

Socievole Annalisa

Annalisa Socievole

Coordinatore
Prof. Luigi Palopoli

Supervisori
Prof. Salvatore Marano

Ing. Floriano De Rango
Floriano De Rango

DEIS

DEIS- Dipartimento di Elettronica, Informatica e Sistemistica
Novembre 2012

Settore Scientifico Disciplinare: ING-INF/03 *to my beloved sister Eleonora*

and my husband Francesco with his sea waves

Acknowledgement

Thanks to my parents and grandparents for their love and support.

Thanks to my supervisors Floriano De Rango and Salvatore Marano for their suggestions throughout my studies.

Thanks to Jon Crowcroft and Eiko Yoneki for their suggestions during the time spent at Computer Laboratory in Cambridge.

Thanks to my coordinator Luigi Palopoli for his guidance throughout the PhD course.

Thanks to all my friends, in particular Elisabetta.

Thanks to my Yoga teacher Aravinda for getting me into shape physically and mentally.

Most importantly: thanks for reading my work.

Contents

List of Figures

List of Tables

1 Introduction

The diffusion of mobile devices carried by users, such as smartphones, has led to a growing interest in new network architectures without fixed infrastructure and exploiting peer-to-peer opportunistic connectivity. In a world where people are becoming increasingly reliant on mobile communication in several aspects of their life, being unable to communicate can negatively affect business and personal relationships. When there is no suitable network infrastructure, an alternative system is necessary. Delay Tolerant Networks (DTNs) [25] were developed to allow communication in scenarios where fixed infrastructure is not available and existing IP and GSM/UMTS network protocols are unsuitable. In such scenarios, where nodes often create sparse network topologies and the contacts between them are intermittent, DTNs use a *store-carry-forward* strategy to allow communication when a path through the network is not reliable (due to disconnections). A node receiving a packet from one of its contacts can buffer the message, carry it while moving, and forward it to the encountered nodes which are at least as useful (in terms of delivery) as itself. A network that routes packets using the store-carry-forward approach is also called *opportunistic network*, because nodes forward messages when the opportunity arises: during an encounter contact.

Opportunistic networks are a special case of DTNs in which there are frequent disconnections and unpredictable encounter patterns. Researchers have developed several routing protocols to deal with these scenarios. Social-based routing protocols, for example, are a class of opportunistic routing protocols exploiting social information. Studying the social relationship between individuals within the network, we can better understand the usefulness of encounters for forwarding.

Commonly, the social network information is extracted from encounters between Bluetooth-enabled devices. The ubiquity of smartphones permits to collect user co-presence information, which allows us to identify social ties grounded on real world interactions. However, the Internet added other social interaction techniques that are not based on physical meetings: email, chat and online social networks services such as Facebook, Twitter, MySpace, and

LinkedIn. Online social interactions may be as useful as co-presence data for improving opportunistic networks, if they provide us with insights into user behavior.

Most of the existing social-based protocols use social information extracted from real-world encounter networks. A protocol based on encounter history, however, takes time to build up a knowledge database from which to take routing decisions. An opportunistic routing protocol which extracts social information from multiple social networks, can be an alternative approach for deciding when to forward messages. While opportunistic contact information changes constantly and it takes time to identify strong social ties, online social network ties remain rather stable and can be used to augment available partial contact information.

1.1 Goals and approach

The aim of this thesis is to demonstrate that social information extracted from multiple social networks provide performance improvements to opportunistic routing. To do so, we perform extensive analysis before presenting our routing proposal.

First, we analyze the performance of different classes of existing opportunistic routing protocols in order to demonstrate that social-based algorithms are advantageous. Second, we compare user social behavior detected through mobile devices interactions to online behavior by using different social network analysis tools. This comparison demonstrates that online social information can be used to improve routing. Finally, we define a *multi-layer social network* model composed by several social networks and construct a new opportunistic routing approach which exploits multiple social network layers to perform routing decisions. The performance of this protocol are evaluated by carrying out tests via trace-driven simulation, with different representative scenarios and routing protocols.

1.2 Dissertation outline

The first part of Chapter 2 provides an overview of the related literature and the current state of the art in the area of DTNs and opportunistic networks. In the second part, the performance of some representative DTN routing schemes are compared through simulations and the impact of energy consumption on routing performance is discussed.

Chapter 3 analyzes the similarity in the graph structure between a social network detected through ZigBee encounters and and the Facebook network of a set of mobile users in order to understand if the online social network can provide useful information for opportunistic forwarding. First, social network models are described; second, the concepts of online and detected social networks are defined; lastly, online and detected social networks are compared using both a sociocentric and an egocentric approach for social network analysis.

Chapter 4 describes the analysis performed to investigate the similarity between multiple social networks layers by introducing a multi-layer social network model. The purpose of this analysis is to provide novel insights into the comparability of dynamic contact networks (detected social networks) and online social networks, and to better understand the social contact behavior of individuals and groups by considering an overall complex system where there are multiple social networks describing their social dynamics. In particular, this chapter focuses on a *joint diagonalization* technique used to produce static graphs from temporal graphs and on the analysis of node centrality, network motifs and detected communities on a multi-layer network.

Chapter 5 details how multi-layer social networks can be used for opportunistic forwarding by describing a proposal of opportunistic routing that exploits a multi-layer social network. This chapter demonstrates the benefits of the new protocol through extensive simulations by comparing the performance of the proposal to other existing routing schemes.

Finally, Chapter 6 concludes with a summary of the contributions of this thesis, as well as discussing the potential future research that can emerge from this work.

1.3 Publications

Book chapters

- F. De Rango and A. Socievole. Mobile Ad-Hoc Networks: Applications. Xin Wang, Chapter 11, *Meta-Heuristics Techniques and Swarm Intelligence in Mobile Ad Hoc Networks*. InTech - Open Access Publisher, pp. 245-264, 2011.

Conference papers

- A. Socievole, F. De Rango, and C. Coscarella. Routing approaches and performance evaluation in delay tolerant networks. In Wireless Telecommunications Symposium (WTS), 2011, pp. 1-6, April 2011.
- A. Socievole, F. De Rango, and C. Coscarella. Performance evaluation of distributed routing protocols over DTN stack for MANETs. In International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), June 2011. Poster paper.
- A. Socievole and S. Marano. Exploring user sociocentric and egocentric behaviors in online and detected social networks. In Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on, pp.140-147, April 2012.

- A. Socievole and F. De Rango. Evaluation of routing schemes in opportunistic networks considering energy consumption. In Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2012 International Symposium on, pp. 1-7, July 2012.
- A. Socievole and S. Marano. Evaluating the impact of energy consumption on routing performance in delay tolerant networks. In Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International, pp. 481-486, August 2012.

2

The development of opportunistic networks

In this chapter we discuss the development of opportunistic networks in scenarios where mobile nodes are sparse and the links between them are intermittent. In these networks there is no guarantee that a path between source and destination nodes exists at any time, rendering traditional routing protocols for mobile networks unsuitable to deliver messages between nodes.

First, we will describe Mobile Ad Hoc Networks (MANETs) and discuss why routing protocols for MANETs are unsuitable to deliver messages in these scenarios where connectivity is intermittent and high latency might be introduced. Second, we will discuss the development of Delay Tolerant Networking (DTN) and opportunistic networks designed to handle intermittent connectivity, high latency, long queuing delays and limited resources. Third, we will discuss DTN routing and compare the performance of some representative DTN routing schemes. Lastly, we will evaluate how the energy consumption impacts the routing performance and how the different forwarding algorithms for opportunistic networks influence the energy usage in the mobile devices.

2.1 MANETs

MANETs are wireless mobile networks where no infrastructure exists and the network topology may dynamically change in an unpredictable manner forming an arbitrary graph. In such networks, nodes may be asked to route packets without connecting to access points. An example of MANET topology is showed in Fig. 2.1.

MANETs were initially developed keeping in mind the military applications, such as battlefield where an infrastructure network is difficult to have. In such environments, ad hoc networks are able to self-organize and can be used where other technologies cannot be deployed. The capabilities of MANETs were suddenly used for other well-known applications such as:

- Collaborative computing - Some business environment could require collaborative work also outside offices. In such environments, it could be more

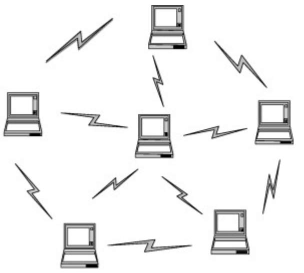


Fig. 2.1. A MANET topology for a set of laptops. Wireless connections between nodes are indicated with bolt symbols. There is no need of a dedicated router since every device routes packets.

important for people to have a possibility to cooperate and exchange information outside.

- Disaster recovery applications - After a natural disaster communications infrastructure is usually not available and there is the need to quickly restore communications. MANETs can be set up in hours instead of days or weeks required for a wired network.
- Personal Area Networking - A Personal Area Network (PAN) is short-range network where devices such as PDAs, laptops or digital cameras are usually associated with a given person. In such scenarios, the technology used to avoid the need of wires between devices is mainly Bluetooth.

It is clear that routing in MANETs is intrinsically different from routing in infrastructured networks. One of the major challenges in the design of a routing protocol for MANETs is to find a packet route quickly and efficiently considering that topology rapidly changes, routers have to be selected and a request has to be initiated. Moreover, the low resource availability in these networks requires efficient utilization and hence an optimal routing scheme.

MANETs routing protocols can be classified as being proactive or reactive. Proactive (or table-driven) protocols require that a node keep track of routes to all possible destinations so that a route is ready when needed. This implies a periodic update of routing tables and does not copy well, however, in highly dynamic environments. On the other hand, reactive (or on-demand) protocols require that a node only discover routes to destinations on demand. Reactive protocols often consume less bandwidth than proactive protocols, but the delay to discover a route to a destination can be significantly high. We can conclude that there is not a protocol suited for all possible environments, while some hybrid schemes have been proposed.

Destination-Sequenced Distance-Vector (DSDV) [59] (proactive), Dynamic Source Routing (DSR) [38] (reactive) and Ad Hoc on Demand Distance Vector Routing (AODV) [60] (reactive) are some of the most cited routing protocols for MANETs.

DSDV is a hop-by-hop distance vector routing protocol where each node periodically broadcasts routing updates and maintains a routing table with all the possible destinations of the particular network and the number of hops to each destination. Each route is labelled with a sequence number in order to have updated routes. There are two possible types of route updates: full dumps (all available routing information) that are sent infrequently during periods of occasional movements or small increment packets transmitted after a full dump. The mobile nodes also sent beacon messages in order to have updated information on neighbors.

In DSR protocol, mobile nodes maintain route caches containing the known source routes. The entries of the route cache are periodically updated in order to include new routes. There are two major phases: the *route discovery* and the *route maintenance* phase. The discovery phase starts when a node does not have an entry into the route cache to send a packet to a particular destination. It initiates a route discovery by broadcasting a route request containing the destination address, along with the source nodes address and a unique ID number. All the nodes receiving the route request check if they have an entry for that destination. If they do not have it, they add their own addresses to the route record of the request and forward the packet along their outgoing links. In order to limit the number of route requests propagated along the outgoing links, each node only propagates the route requests that have not yet been seen and the requests that do not contain the nodes address in the route record. When the route request reach a node having an unexpired route to the destination or the destination node itself is reached, a route reply is generated and forwarded along the route to the initiator of the request. Route maintenance is accomplished using acknowledgements and route error packets.

The AODV protocol is basically a combination of DSDV and DSR. Similar to DSR, it uses the on-demand mechanism of route discovery and route maintenance adding the use of hop-by-hop routing, sequence numbers, and periodic beacons used in DSDV. AODV is also able to minimize the number of broadcasts by creating routes on an on-demand basis without maintaining a complete list of the routes as in DSDV.

There are several other routing protocols for MANETs [85]:

- Optimized Link State Routing (OLSR) [16] - a proactive link state protocol where nodes floods routing table within the network and calculate the optimal forwarding locally.
- Location-Aided Routing (LAR) [41] - protocols that use GPS information to improve routing performances.

- EASE [31] - the history of encounters is used to improve routing performances.
- On Demand Multicast Routing Protocol (OMDRP) [43] - a reactive multicast protocol using mobility prediction.
- Dynamic MANET On-Demand (DYMO) [58] - a successor to AODV working both in proactive and in reactive mode.

All the MANET routing protocols mentioned above require that there exists an end-to-end path from the packet source node to the destination node. If nodes are characterized by high mobility and the deployment of nodes in the network is sparse, this condition is difficult to meet. If the path does not exist, packet transmission is delayed until a path becomes available. Moreover, if there is a large delay, transport protocols such as TCP does not work well, even is the end-to-end is still active. For a large delay, TCP congestion control mechanism assumes that a packet is lost. Another problem for TCP is that the window size might take long time to enlarge because of the time needed to wait for the ACK packets. These considerations let us conclude that a different strategy is needed when mobile networks are intermittently connected or when there is a large delay.

2.2 Opportunistic networks and Delay Tolerant Networks

Opportunistic networks are one of the most interesting evolutions of MANETs. In opportunistic networks, nodes in transmission range opportunistically cooperate during a contact to forward data towards a destination. In a context where there are frequent disconnections and high delays, thanks to this opportunistic behavior, mobile nodes are able to communicate with each other even if a route connecting them does not exist. Delay Tolerant Networks (DTNs) [25] also known as disruption-tolerant networks or intermittently connected networks can be considered a subset of opportunistic networks with few high delay links. Actually there is not in the literature a clear separation of concepts for opportunistic networks and DTNs and the two terms are often used interchangeably. We believe that opportunistic networks include DTNs and several concepts behind them come from studies on DTNs.

Delay tolerant networking architecture allows communication in scenarios where nodes are sparse and the contacts between them are intermittent, due to high node mobility. Examples of intermittently connected networks are:

- Inter-planet satellite networks - networks where the communication between satellites and ground nodes suffers from long delays and episodic connectivity.
- Sensor networks - networks where battery power is limited and sensors are scheduled to be wake/sleep periodically in order to send data to the sink node.

- Military ad hoc networks - networks where nodes may move randomly and are subject to being destroyed.

In the DTN domain the following assumptions of conventional IP networks are not valid:

- There exists and end-to-end path between the source and the destination node.
- The end-to-end packet drop is small.
- The maximum round trip time between nodes in the network is not high.

As a result the DTN routing protocols and data-delivery architecture differ from traditional TCP/IP networks. The RFC 4838 [14] describes the routing and the transport layer approaches used in DTNs. In these networks, a node receiving a packet from one of its contacts can buffer the message, carry it while moving, and forward it to the encountered nodes which are at least as useful (in terms of delivery) as itself. This strategy is also referred as *store-carry-forward*. The data unit used in DTNs can be a message, a packet, or *bundle*, which is defined as a number of messages to be delivered together. The DTN Bundle Protocol is used to communicate between regions with high delay links. This protocol sits between the transport and application layer of the traditional TCP/IP stack.

As we have discussed, DTNs solve the problem of disconnection and delay that MANETs are not able to cope with, using the store-carry-forward approach. DTNs, however, are designed for systems with a few high delay links. There are scenarios, in which the mobile nodes are subject to a high number of disconnections and delays. One example is a Pocket Switched Network (PSN) [33], where humans carry small personal computing devices capable of exchanging messages. The network is weighted using the encounters between the humans carrying the mobile devices during their daily lives. A network routing packets using the store-carry-forward approach in this scenario is an opportunistic network.

Opportunistic networks and their algorithms has led to the design of new applications that were not previously carried out on mobile devices:

- Mobile file sharing [50] [63]
- Crowdsourcing and messaging [13] [52]
- Opportunistic Computation [17] [55]
- Sensing [18] [21] and collaborative sensing [29] [74]
- Personal sensing for healthcare monitoring [44]
- Military surveillance [82] and human tracking [1]
- Animal tracking [39]
- Interaction with embedded AI in pervasive environments [32].

Where connectivity to existing infrastructure networks is not present, the message passing between the mobile devices could route a particular message to the destination. This concept is related to Stanley Milgrams famous small world experiment [75] where the letters sent between two unfamiliar individuals, one in Nebraska and the other in Massachusetts, were routed using six hops.

The task of the routing algorithm in an opportunistic network is to decide if a particular contact is appropriate for routing any of the outgoing messages. Studying the social relationship between individuals within the network, we can better understand the usefulness of encounters for forwarding. The Haggle architecture [73] was designed as a data-centric architecture for opportunistic networks taking advantage of brief encounters to route packets. Hui et al. showed that finding the correct groups of nodes to forward messages improve routing and node efficiency [34] [35] (we describe at length this social based routing strategy in section 2.4.5).

2.3 DTN routing

Routing in DTN or opportunistic networks is a difficult issue to deal with because the assumption of an existing end-to-end path between the source and the destination does not hold. Therefore, routing protocols have to deliver a message to the destination using the store-carry-forward strategy. Several routing protocols were proposed to handle frequent disconnections, opportunistic or predictable connections, high latency, long queuing delay and limited resources. In this section we classify these protocols and we briefly describe the most representative protocols belonging to each class.

Routing protocols for DTNs can be classified as follows:

- Flooding- or replication-based
- History- and encounter-based
- Probabilistic-based
- Social behavior-based
- Knowledge-based

Flooding- or *replication-based* protocols employ a simple routing strategy. The source node forwards a copy of the message to all its neighbors and the neighbors do the same with their neighbors (which do not have a copy of the message). Protocols belonging to this class vary according to the spreading mechanism used and the number of copies of the packet used to flood the network.*Epidemic routing* [76] was historically the first flooding based routing protocol. Each contact opportunity is used by the nodes to disseminate the data. When a node receives a packet, it is buffered and carried by the mobile node and all the packets kept by the node are forwarded to all the other encountered nodes. Clearly, a message spreads quickly throughout the network and this reduces the delivery delay. Moreover, considering that there are many copies of a message within the network, Epidemic routing increases the delivery ratio as well. Unfortunately, as the number of messages increases, Epidemic routing do not scale well due to its high resource requirement (i.e., storage and battery) and consequently, some complementary mechanisms are needed to overcome this problem. More details on Epidemic routing will be given in section 2.4.1. *Spray and Wait* protocol [71] is another well-known flooding-based routing protocol controlling the amount of copies of a packet to be spread in the network. Differently from Epidemic routing, the number of messages copies to be spread is fixed. The source node sprays L message copies to L distinct neighbors and then waits with the hope that one of these nodes will meet the destination. During the wait phase all L nodes storing a copy of the message perform *direct delivery*. Direct Delivery [70] is a single-copy routing technique where the message is forwarded by the current node only directly to the destination node. Spray and Wait obviously has an overhead lower than Epidemic routing and it can be adjusted to meet specific deadlines as well. For example, using a shorter *wait* phase, lower delays can be achieved. This controlled flooding-based protocol will be better described in section 2.4.2.

History- and *encounter-based* routing protocols are an alternative to flooding schemes considering history information on the past encounters in order increase the system's performance. A representative history-based routing protocol is PRoPHET (Probabilistic Routing Protocol using History if Encounters and Transitivity) [46]. Lindgren et al. assume that if a user visited a place several times in the past it is much likely that will visit this place again in the future. A *delivery predictability* indicating how likely it is that a node will be able to deliver a packet to a destination is calculated at each node and packets are forwarded accordingly. With MaxProp [12] routing, Burgess et. al also use history of the node's encounters to drive routing decisions. The *path likelihood* metric based on historic information is the metric used to decide whether to transmit (if time runs short) or delete a packet (if storage space is not enough). PRoPHET and MaxProp will be further described in sections 2.4.3 and 2.4.4 respectively.

Probabilistic-based routing protocols aim to reduce the cost of forwarding while retaining good performances by forwarding packets only to nodes that have high delivery probabilities. With the term "probabilistic-based" routing protocols we refer to the probabilistic ones that do not utilize history information. *(p-q)* routing [49] is an example of probabilistic scheme where a relay node encountering the source node accepts a copy of the message with probability q $(0 \leq p \leq 1)$ and encountering another potential relay, the latter accepts a copy with probability p $(0 \leq q \leq 1)$. The destination accepts the copy of the message with probability 1. In [77] and [78], Wang and Wu propose a *Delay-/Fault-Tolerant Mobile Sensor Network for Pervasive Information Gathering* (DFT-MSN). Their proposal define a delivery scheme based on the *delivery probability* to forward a message to the appropriate neighbor nodes and a queue management scheme deciding on whether to transmit or drop a stored message, based on the *message's importance*.

To improve routing performance further, many protocols use social network information. There exist several recent studies studies that investigate the impact of human mobility and the potential of social relations on the design of a routing protocol. *Social behavior-based* protocols usually extract social information from a contact network (also called *detected social network*). In [15], Chaintreau et al. analyze a large number of traces related to different human-mobility environments and find that their inter-contact time distribution is heavy-tailed. Consequently, routing algorithms for opportunistic networks have to be tested under different mobility models than the Random Way-Point. Bubble-Rap [34] is a social based routing protocol where nodes are ranked according to their centrality and their belonging to a particular community. This protocol allows nodes to "bubble up" a message to a node it has a higher rank within the same community, or is a member of a community that is closer to the destination. This protocol will be better analyzed in section 2.4.5. The SimbetTS [19] routing protocol is another example of social based protocol where a node forward a message to an encountered node according to three social metrics: *betweenness* (the number of shortest path on which a node lies), *similarity* (the number of ties that two nodes share), and *tie-strength* (the recency, duration and number of contacts between two nodes). In another work [81], Xu et al. also use centrality as metric for opportunistic forwarding. However, this approach overloads the most central nodes which have to perform a large percentage of the forwarding (63% of traffic by the 10 % of nodes). Boldrini et al. [7] use social information extracted from users' connections to users outside their home group to improve opportunistic routing and construct a middleware using history information on the past encounters to improve opportunistic services. In another work [48], Mashhadi et al. define a social-based routing scheme where messages are forwarded to nodes that are interested in the specific content of the message.

As we have seen, most protocols use social information extracted from detected social networks, however there exists the possibility to perform opportunistic routing using the information extracted from online social networks such as Facebook[1], Twitter[2], LinkedIn[3], etc. The protocols using this strategy will be described in Chapter 5.

Another class of routing protocols for DTNs, known as *knowledge-based* or *deterministic* routing protocols, assumes to have partial or full knowledge of either the network topology or the inter-contact times. In an early study [37], Jain et al. evaluate the performance of different routing algorithms varying their network knowledge, also known as *oracles*. The knowledge of the meeting times of DTN nodes or the storage space availability at each node are examples of oracles. The simplest algorithm, which has zero knowledge is called First Contact (FC). According to this scheme, a message is forwarded along an edge chosen randomly between all the current contacts. Other algorithms, such as the Earliest Delivery (ED) have oracles related to the contact times

[1] www.facebook.com
[2] www.twitter.com
[3] www.linkedin.com

and durations, while the most sophisticated one, called Linear Programming, has oracles regarding contacts, queuing and traffic. In [86], using topology information and group membership, Zhao et al. conclude that even with partial knowledge, multicast routing algorithms can perform efficiently. Demmer et al. [20] adjust Link State Routing for DTN environments such as connection-isolated villages in developing countries, assuming some knowledge regarding the approximate contact times and contact durations. Since the connectivity carrier to the Internet is a public transport vehicle, whose timetable can be known in advance, this knowledge is used to improve routing.

2.4 Performance comparison of DTN routing schemes

In this section we compare the performance of some of the DTN routing protocols described in section 2.3. This work was conducted with Prof. Floriano De Rango and Carmine Coscarella and published in [67] and [66]. The purpose of these first works on DTNs was to investigate the behavior of a set of routing protocols belonging to different classes in terms of number of transmitted messages, message delivery ratio, average latency, buffer occupancy and average number of hops. In section 2.3.2 we extend our analysis considering energy consumption as well. We first describe in a deep way some of the routing schemes we considered and the simulation environment, and subsequently we discuss the results of our performance comparison.

2.4.1 Epidemic routing

Epidemic routing [76] is a flooding based protocol where each contact opportunity is used by the nodes to forward the messages. When a node receives a packet, it is buffered and carried by the mobile node and all the packets kept by the node are forwarded to all the other encountered nodes (Fig. 2.2).The basic idea of epidemic routing is equivalent to the spread of an infection. When a node carrying a packet has a contact with a new node that does not have a copy of that packet, the carrier infects the new node forwarding a copy of the packet. Each node stores in a buffer the messages it has originated and that ones it is carrying on behalf of other nodes. A *summary vector* is kept by the nodes as an index of these messages, and when two nodes meet they exchange their summary vectors in order to understand which messages stored remotely have not been seen by the local node. Each message of the summary vector has a globally unique message ID which is used to establish if it has been previously seen. Fig. 2.3 shows a flow diagram of Epidemic forwarding scheme.

It is important to note that, in routing schemes adopting the store-carry-forward paradigm, many packets could remain in the network when a packet reaches its destination node. Those packets not only waste buffer resources but also continue to scatter packet copies. Since the nodes storage capacity

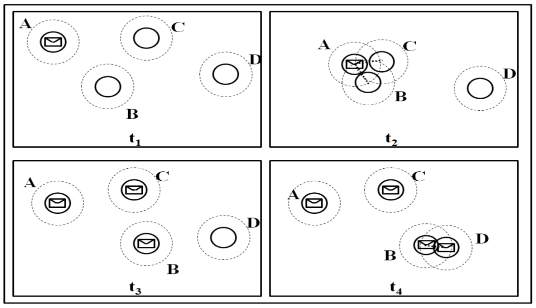


Fig. 2.2. Example of epidemic forwarding: a source node A forwards the packet to the encountered nodes B and C. Node B, encountering D, forwards the packet to the destination node D.

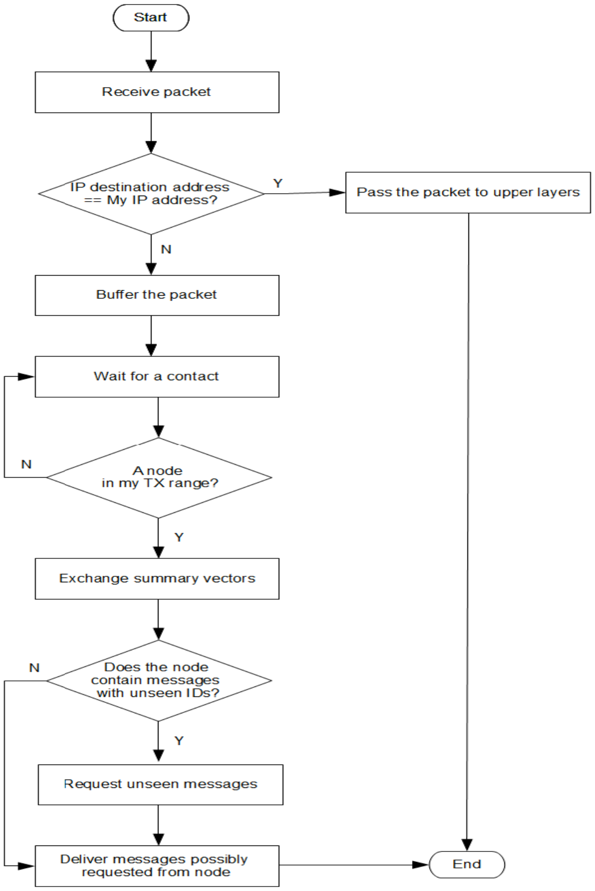

Start
Receive packet
IP destination address == My IP address?
Y
Pass the packet to upper layers
N
Buffer the packet
Wait for a contact
N
A node in my TX range?
Y
Exchange summary vectors
N
Does the node contain messages with unseen IDs?
Y
Request unseen messages
Deliver messages possibly requested from node
End

Fig. 2.3. Flow diagram of Epidemic forwarding scheme.

is limited,*recovery scheme* has to be combined with the forwarding scheme in order to delete unnecessary packets from the network. In [84] several recovery approaches are discussed. A timer-based approach deletes all packets from the network in a finite time interval. However, it is not simple to set the lifetime of packets because if the lifetime is too short, packets may not reach their destinations, while if it is too long, many unnecessary packets remain in the network for a long time. Another approach, referred as VACCINE, is based on explicit notification. When a packet reaches the destination node, this node generates the corresponding anti-packet. Moreover, when a node successfully delivers a packet to the destination node, it generates an anti-packet as well. These anti-packets are then flooded to the other nodes according the conventional epidemic scheme. When the anti-packet is received, the corresponding packet, if any, is deleted from the node. It is important to note that anti-packets also act as acknowledgements, so that source nodes can know the successful delivery of packets they transmitted. The recovery process terminates when anti-packets spread over all nodes. In the performance comparison described in section 2.4.7, we consider both classic Epidemic routing and Epidemic with the VACCINE scheme.

2.4.2 Spray and Wait

Spray and Wait protocol [71] is a different kind of epidemic routing which floods the network with a fixed number of copies of a packet. The source node sprays L message copies to L distinct neighbors and then waits hoping that one of these nodes will carry the message to the destination. If the destination node is not found during the spraying phase, each of the L nodes carrying a message copy will forward the message only to the destination node (*direct delivery*). It has been defined a binary version of Spray and Wait protocol. This version allows a source node A to start with L copies of the packet. Then, when any node L having $n > 1$ copies of the packet encounters another node B (with no copies), it forwards to B *L/2* copies. When node A has only one copy left, it forwards the packet only to the destination. In Fig. 2.4, the flow diagrams describing these protocols are showed.

2.4.3 PRoPHET

PRoPHET (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [46] is a routing scheme which estimates a node metric using the information on the number of meetings between nodes. When there is a contact between two nodes, they increase their link weight towards each other and towards the nodes met by the other node. In order to indicate how likely it is that a node will be able to deliver a packet to a destination, PRoPHET defines a metric called *delivery predictability.* When there is a contact between two nodes, they exchange their summary vectors containing the delivery predictability information. This information is used to choose which messages to

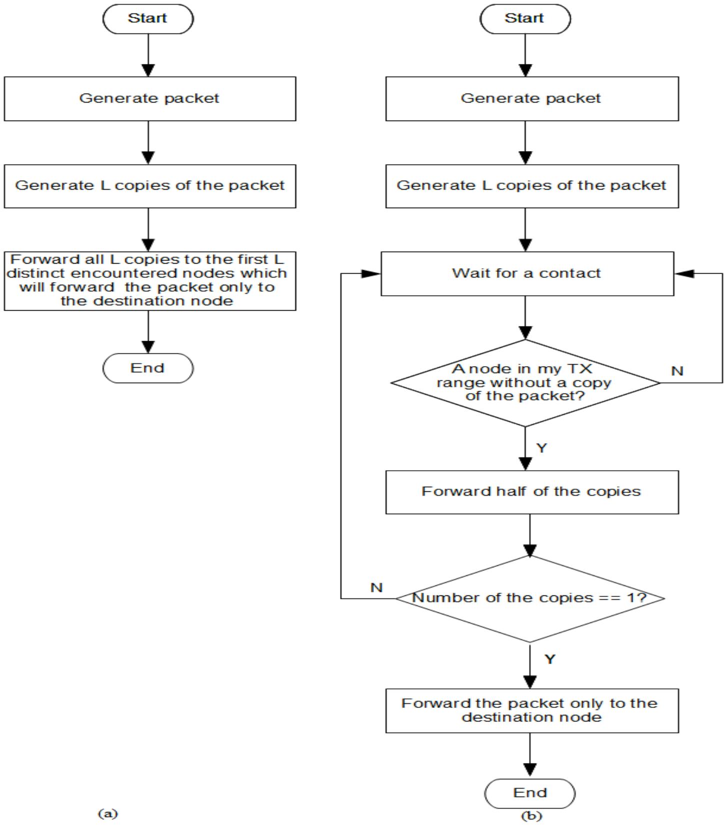

Start
Generate packet
Generate L copies of the packet
Forward all L copies to the first L distinct encountered nodes which will forward the packet only to the destination node
End
(a)
Start
Generate packet
Generate L copies of the packet
Wait for a contact
A node in my TX range without a copy of the packet?
N
Y
Forward half of the copies
Number of the copies == 1?
N
Y
Forward the packet only to the destination node
End
(b)

Fig. 2.4. Flow diagrams of Spray and Wait (a), and Binary Spray and Wait (b) message generation.

exchange and to update the internal predictability vector. In PRoPHET, a node forwards a packet to its encounter, only if this encounter's delivery predictability is higher. In Fig. 2.5, a flow diagram for this scheme is presented.

2.4.4 MaxProp

In a study similar to PRoPHET, nodes are weighted using historic information of the contacts between nodes. MaxProp [12] is a flooding-based DTN routing protocol, where each node keeps a particular vector listing estimations of meeting probabilities between nodes. In other words, this vector defines the likelihood a node has of encountering each of the other nodes of the network. During a contact between two nodes, each node increments the corresponding element of the vector and they exchange their node-meeting likelihood vectors. With this information they are able to compute the shortest paths to the

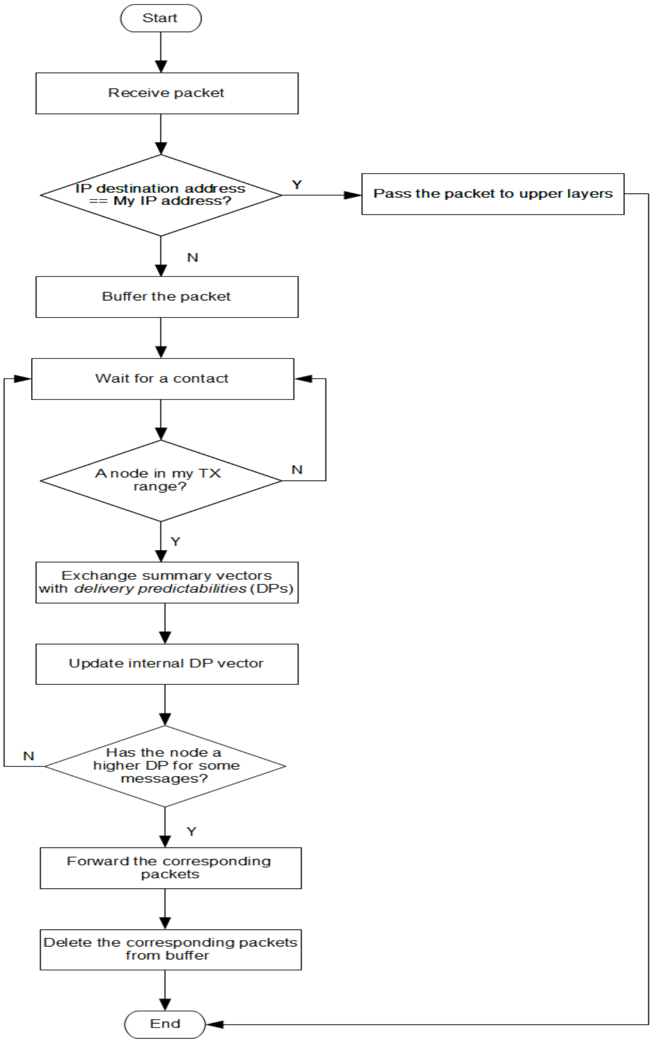

Start
Receive packet
IP destination address == My IP address?
Y
Pass the packet to upper layers
N
Buffer the packet
Wait for a contact
A node in my TX range?
N
Y
Exchange summary vectors with *delivery predictabilities* (DPs)
Update internal DP vector
Has the node a higher DP for some messages?
N
Y
Forward the corresponding packets
Delete the corresponding packets from buffer
End

Fig. 2.5. Flow diagram of PRoPHET forwarding scheme.

destinations desired. The messages are then ordered and sent considering the cost to a particular destination. In order to do that, MaxProp keeps an ordered queue for the messages, based on the destination of each message and ordered by the path likelihood to that destination. During a meeting between two nodes, all messages not held by the encounter are transferred. Moreover, in MaxProp nodes that successfully receive a message use acknowledgements to be injected into the network in order to instruct the other nodes to delete extra copies of a message from their buffers. In Fig. 2.6, the flow diagram of MaxProp forwarding scheme is presented.

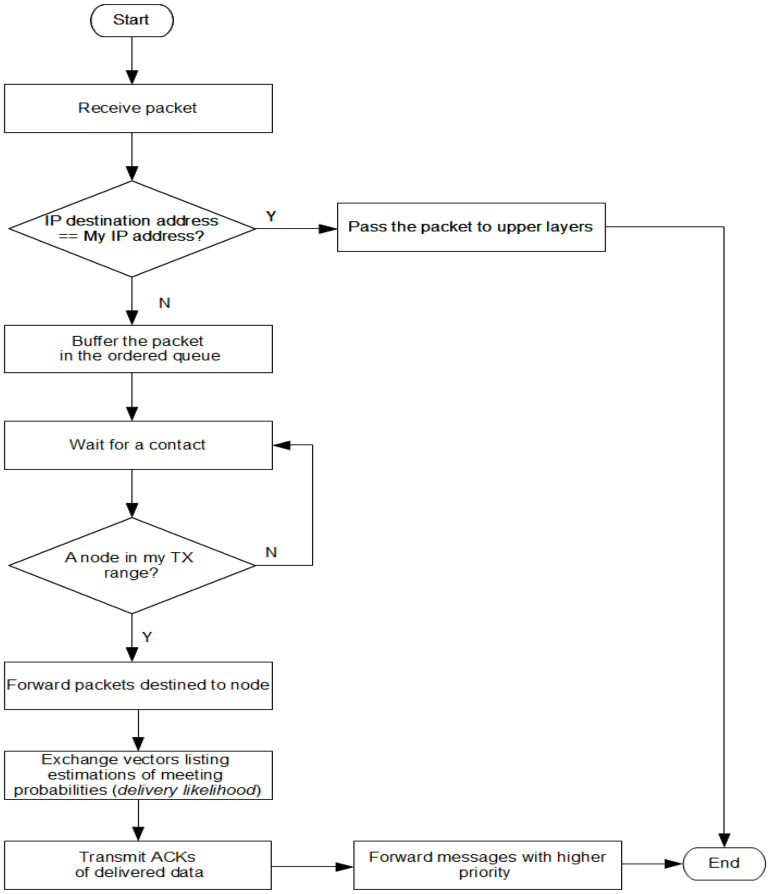

Start
Receive packet
IP destination address == My IP address?
Y
Pass the packet to upper layers
N
Buffer the packet in the ordered queue
Wait for a contact
A node in my TX range?
N
Y
Forward packets destined to node
Exchange vectors listing estimations of meeting probabilities (*delivery likelihood*)
Transmit ACKs of delivered data
Forward messages with higher priority
End

Fig. 2.6. Flow diagram of MaxProp forwarding scheme.

2.4.5 Bubble Rap

Bubble Rap [34] is among the most widely referenced social-based routing protocols for this class of mobile networks. Fig. 2.7 shows the flow diagram for this forwarding scheme. In Bubble Rap a node is characterized by two social metrics, namely *centrality* and *community*. A minimum of two centrality measures are associated to each node. One measure is based on the nodes global popularity (GP) in the whole network and the other measure is based on the local popularity (LP) within its community or communities. A message is forwarded to nodes with higher global popularity (centrality) until the carrier node meets a node with the same community label (CL) as the destination node. In this case, the message is forwarded to nodes with higher local popularity until successful delivery.

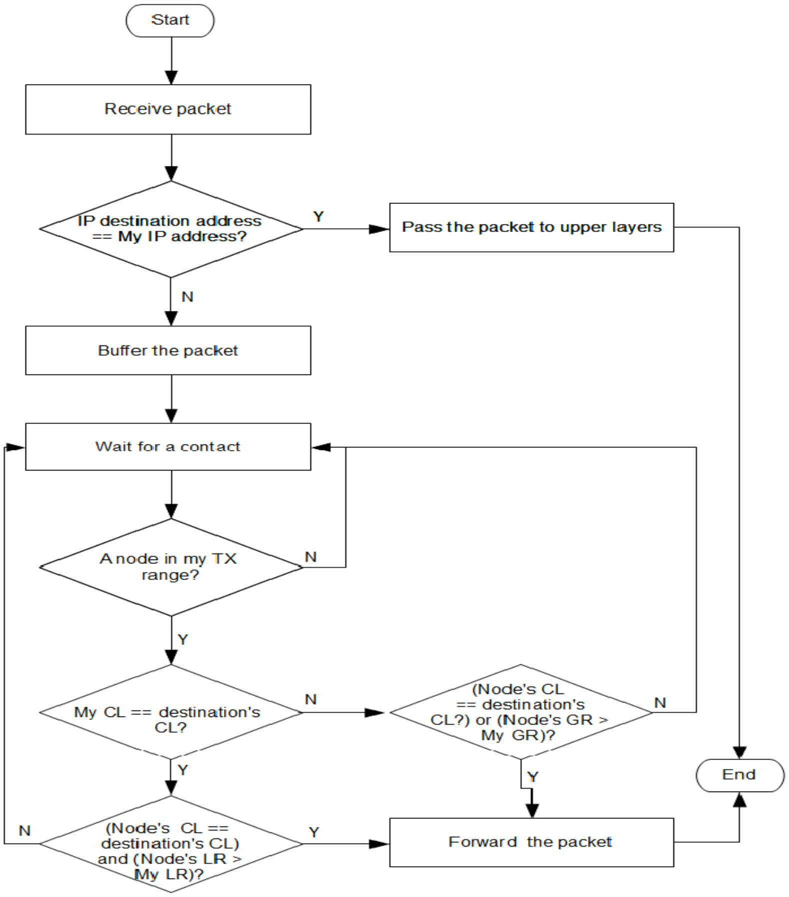

Start
Receive packet
IP destination address == My IP address?
Y
Pass the packet to upper layers
N
Buffer the packet
Wait for a contact
A node in my TX range?
N
Y
My CL == destination's CL?
N
(Node's CL == destination's CL?) or (Node's GR > My GR)?
N
Y
End
N
(Node's CL == destination's CL) and (Node's LR > My LR)?
Y
Forward the packet

Fig. 2.7. Flow diagram of Bubble Rap forwarding scheme.

2.4.6 Simulation environment

Our simulations are carried out on the Opportunistic Network Environment (ONE) simulator [40]. In this simulation environment each node is modeled with a radio interface, persistent storage, several movement models, several routing capabilities, a basic energy consumption module and application interactions. We compare the routing performance of First Contact, Direct Delivery, Epidemic Routing, Epidemic Routing with VACCINE recovery mechanism (we extend the simulator including this mechanism), Spray and Wait, PRoPHET and MaxProp.

The chosen mobility is the Random Waypoint [38], where the node movement is free of restrictions, both temporal and spatial, with a node speed selected between 0.5 m/s and 20 m/s. Nodes have a transmission range of 50 meters and move in an area of 2000m x 2000m. In the scenarios where the number of nodes is fixed, we have selected 50 nodes.

Message size varies from 5 to 15 kB, each message is generated at a random time with 3 second intervals [t, t+3s] and is sent to a random selected destination node. The TTL is set to 1800s.

The transmit speed of radio devices is 250 kBps and the buffer size is set to 20 MB. We run the simulations with Spray and Wait with 5 message copies for the scenarios with 50 nodes and choosing a number of copies equal to the 10% [72] of the number of nodes in the other scenarios. For the simulation of PRoPHET we choose $Pinit = 0.75$, $\beta = 0.25$ and $\gamma = 0.98$, as suggested in [46]. Finally, the simulation time is always 15000 seconds.

2.4.7 Performance comparison and results

We have focused on comparing the performance of the chosen DTN protocols with regard to the following metrics:

- *number of transmitted messages*
- *message delivery ratio*, i.e. how many of the transmitted messages the protocol is able to deliver to destination;
- *average latency*, i.e. how long time it takes a message to be delivered (even though applications using these protocols are relatively delay tolerant, it is still of interest to consider this performance metric);
- *buffer occupancy*, in order to analyze the resource utilization which is so crucial;
- *average number of hops* to reach the destination node.

Fig. 2.8 shows the number of transmitted messages as a function of the number of nodes. Direct Delivery sends the smallest amount of messages and this is not very surprising, since according to this scheme the sending node delivers a message only if it meets the destination. On the contrary, Epidemic routing generates the highest number of transmitted messages because of the unlimited replication of messages. Applying the VACCINE recovery mechanism,however, the number of messages sent by Epidemic routing is less than classic Epidemic routing and comparable to the number of messages sent by PRoPHET and MaxProp. It is important to underline that in PRoPHET and MaxProp, messages are only sent to better nodes, while epidemic routing sends all possible messages to the encountered nodes.

A single-copy routing protocol like First Contact decreases the number of transmitted messages, since a packet is delivered to the first encountered node trying to reach the final destination. Looking at Spray and Wait routing protocols, which use a fixed number of copies, the number of transmitted messages are reduced. Another thing that can be seen from the graph is that increasing the number of nodes PRoPHET increases the number of transmitted messages overcoming MaxProp and epidemic with VACCINE. This is due to the fact that the higher probability of two nodes meeting each other forces PRoPHET to more frequent computations and therefore the number of copies placed in the network is high.

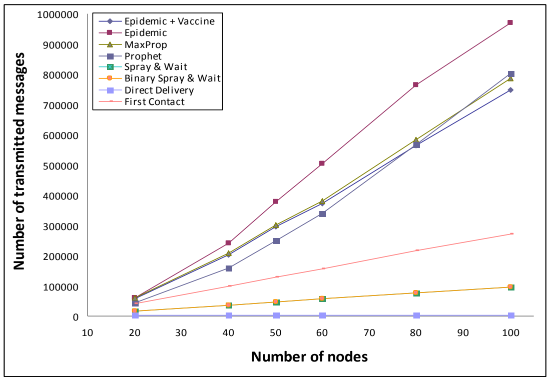


Fig. 2.8. Number of transmitted messages as a function of the number of nodes.

Fig. 2.9 the delivery ratio as a function of the number of nodes is showed. MaxProp and Epidemic routing protocols perform better and similarly because they make use of best path selection mechanisms and flooding respectively. Moreover, the VACCINE mechanism, combined with Epidemic routing outperforms the classic Epidemic routing because the deleted packets decrease the buffer occupancy allowing other transmissions. Looking at PRoPHET, based on local mobility information, it presents a better delivery ratio than Spray and Wait protocols but lower than MaxProp and epidemic protocols. Direct Delivery and First Contact are characterized by low delivery performance, since the first protocol delivers a message only if the encounter is the destination, while the second delivers a message to the first node encountered trying to reach the destination.

Fig. 2.10 shows the delivery ratio as a function of node speed. This performance metric increases with respect to speed for all the routing protocols and this behavior is closely related to the encounter process. The more the frequency of the encounters, the more the number of message exchange and hence delivery. The single-copy protocols as Direct Delivery and First Contact are characterized by the lowest delivery ratio. When the node speed is

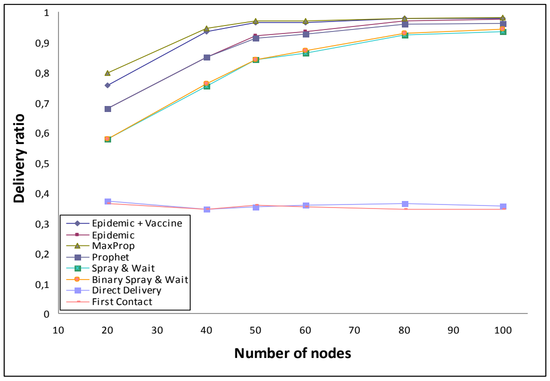


Fig. 2.9. Delivery ratio as a function of the number of nodes.

low, the best performance is reached by MaxProp and Epidemic routing with VACCINE, followed by classic Epidemic, PRoPHET, Spray and Wait protocols and finally by Direct Delivery and First Contact. When the node speed is high, the frequency of the encounters is so high that the difference between the performances of MaxProp, epidemic protocols, PRoPHET, and Spray and Wait protocols is not significant.

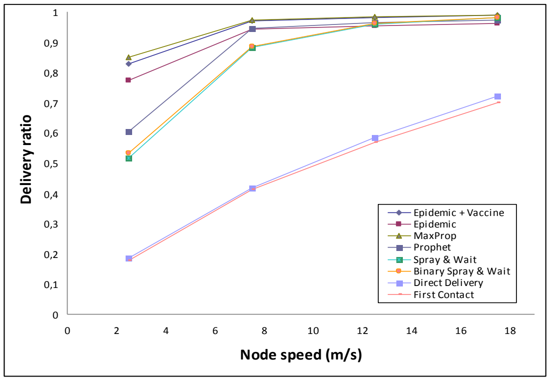


Fig. 2.10. Delivery ratio as a function of node speed.

The average latency as a function of the number of nodes is showed in Fig. 2.11. First Contact and Direct Delivery take about 800 seconds to deliver a message, both with a small number of nodes and with a large number of nodes. The motivation is the same as the previous one: the two protocols deliver a message to a single node, so the low delivery ratio leads to a higher latency. Spray and Wait protocols behave similarly between 20 and 40 nodes simulations, but when the number of nodes increases the binary version of

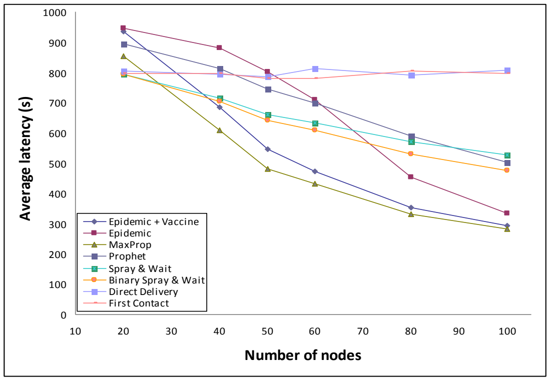


Fig. 2.11. Average latency as a function of the number of nodes.

the protocol delivers a message in a smaller amount of time. It is important to note that the two versions of epidemic protocols start with a high average latency for 20 nodes when the number of nodes increases there is a significant difference between them. The mechanism of anti-packets operates in a good way if many nodes try to communicate with each other, increasing the level of propagation of anti-packets. Consequently, the version without VACCINE offers the highest average latency as a result of the large distribution of messages within the network. Looking at PRoPHET, this protocol does not offer a fulfilling performance: predicting good forwarding nodes in this scenario is difficult because of the randomness in the mobility of nodes, leading to a higher latency. Nevertheless, MaxProp is always the best protocol: the higher delivery ratio leads to a lower average latency.

Looking at the average latency as a function of node speed (Fig. 2.12), it can be seen that the more the node speed increases, the more the nodes have the possibility to encounter different nodes. Hence, the delivery ratio increases and the average latency decreases. At low speeds, First Contact and Direct Delivery protocols are characterized by the lowest average latency, since the contacts between nodes are more sporadic; protocols causing a high average latency are classic Epidemic and PRoPHET, since they send messages to all the encountered nodes and during this process it is not sure that one of these nodes is the final destination (especially for Epidemic protocol) or that the delivery predictability of the destination may enable the transmission of the message (in the case of PRoPHET). At higher speeds, MaxProp, epidemic protocols, PRoPHET and Spray and Wait protocols outperform Direct Delivery and First Contact. Classic epidemic protocol and PRoPHET, according to their epidemic nature, have the highest average latency, followed by Spray and Wait protocols. A lower average latency is guaranteed by Epidemic with VACCINE and MaxProp. The reason is that the mechanism based on anti-packets provided by VACCINE and the buffering process provided by MaxProp allow a better management of the buffer space leading to a lower average latency.

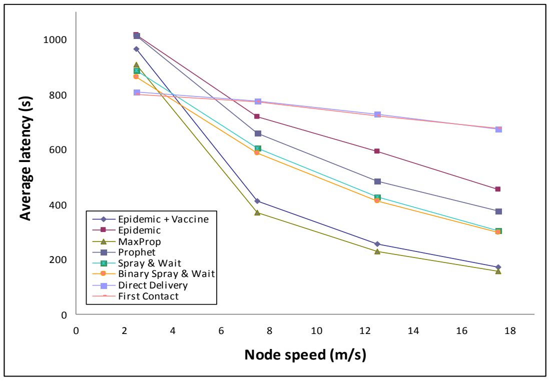


Fig. 2.12. Average latency as a function of node speed.

Fig. 2.13 and Fig. 2.14 show the buffer occupancy percentage as a function of the number of nodes and node speed respectively. In Fig. 2.13, it is intuitive to see that the classic Epidemic protocol shows the higher buffer occupancy percentage and this is due to the flooding process and the absence of a recovery mechanism deleting unnecessary packets from the network. In PRoPHET, when two nodes meet, a message is transferred to the other node only if the delivery predictability of the destination of the message is higher at the other node, so a message may have to stay in the buffer for a longer period. Analyzing the behavior of MaxProp and epidemic with VACCINE, it can be seen that it is similar. The mechanism of assigning priorities to buffered packets and the recovery mechanism guarantee a lower and acceptable percentage of buffer occupancy. In the case of Spray and Wait protocols, where the number of message copies is fixed, the percentage of buffer occupancy further decreases. Similarly, Direct Delivery and First Contact, using only one copy of the message, provide a lower percentage.

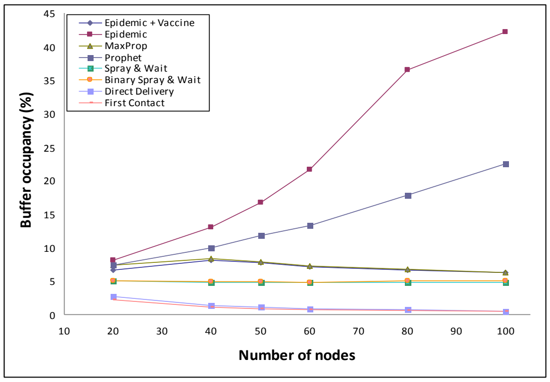


Fig. 2.13. Percentage of buffer occupancy as a function of the number of nodes.

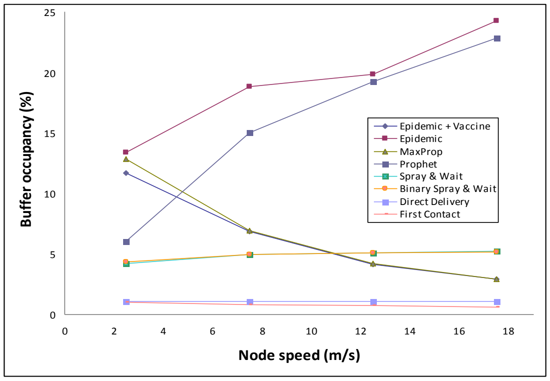


Fig. 2.14. Percentage of buffer occupancy as a function of node speed.

In Fig. 2.14 the buffer occupancy increases with respect to speed for protocols that rely on flooding or a probabilistic forwarding, such as the classic Epidemic protocol and PRoPHET respectively. For protocols that rely on path selection and a better buffer management, such as MaxProp, and on the use of anti-packets, such as Epidemic with VACCINE, the buffer occupancy decreases with respect to speed. This is intuitive, since the recovery mechanism allows a node to refuse or delete a message from its buffer in case of successful delivery to the destination. For protocols using a fixed number of copies of the message, such as Spray and Wait protocols, Direct Delivery and First Contact, the percentage of buffer occupancy can be considered constant.

The behavior of the different routing approaches can be further explained looking at Fig. 2.15 where the average hop count as a function of the number of nodes is showed.

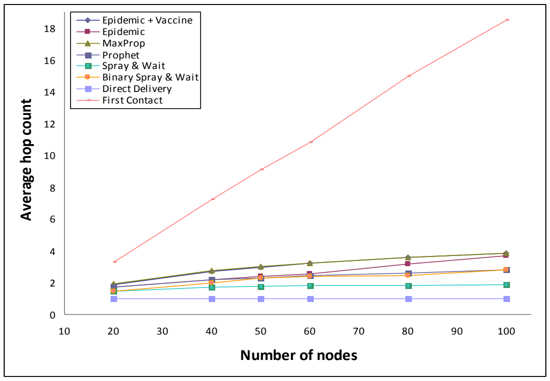


Fig. 2.15. Average hop count as a function of the number of nodes.

Direct Delivery maintains a hop count of 1 in any case, since it delivers a message only to the final destination. In the case of First Contact, increasing the number of nodes increases the average number of hops, since a node delivers a message to the first encountered node. Similar is the behavior of the other protocols where the average number of hops slightly increases with the increase in the number of hops.

2.4.8 Discussion

Considering a set of DTN routing protocols composed by Direct Delivery, First Contact, Epidemic routing (both without a recovery mechanism and with VACCINE), Source Spray and Wait, Binary Spray and Wait, PRoPHET and MaxProp, we compared their routing performance in different scenarios in terms of delivery ratio, average latency, buffer occupancy and average hop count. Our simulations show that MaxProp and epidemic routing with VACCINE give better performance than the other protocols. Moreover, MaxProp and Epidemic with VACCINE perform very similarly considering delivery ratio, average latency, the percentage of buffer occupancy and average hop count. The use of best path selection mechanisms with an appropriate buffer management, as in the case of MaxProp, and flooding combined with the VACCINE recovery scheme, as in the case of epidemic routing, succeed in the goal of providing communication between nodes in a intermittently connected network with different forwarding strategies, and better performance than other examined protocols.

2.5 Impact of energy consumption on routing performance

After analyzing many DTN routing protocols in the literature, we observed that authors compare the performance of their routing algorithm in order to show the improvement of their proposal with regard to others in the literature, focusing on performance metrics like delivery ratio and delivery latency. A DTN system should attempt to achieve high delivery ratio and low delivery delay, but it is very difficult to achieve both targets, considering that the system is constrained with regard to energy consumption and storage space. As a result, we believe that the choice of a DTN routing protocol should be based on the desired performance outcome, taking also into account system constraints. Starting from this consideration, we extended our performance comparison of a set of representative DTN routing protocols (Epidemic routing, Spray and Wait, MaxProp, PRoPHET and Bubble Rap) considering energy consumption, in order to study if their behavior vary when the energy consumption constraint is considered. This work was conducted with Prof. Floriano De Rango and Prof. Salvatore Marano and published in [65] and [69].

2.5.1 Simulation environment

In order to test the different protocols considering energy consumption, we carried several simulations using again the Opportunistic Network Environment (ONE) simulator. This simulator has been especially designed to test routing protocols for DTN networks including Epidemic routing, Spray and Wait, PRoPHET and MaxProp. We extended the simulator implementing the social based protocol Bubble Rap.

2.5.1.1 Energy consumption model

In ONE, each node is modeled with a battery characterized by a limited energy budget. Energy is subtracted from the budget every time a node transmits a packet or scans the area looking for other nodes. If a node has not enough energy level is not allowed to perform scanning or forwarding.

For our simulations we consider each node as a smartphone with a 1200 mAh battery, using Bluetooth to connect to the other nodes and the energy consumption for Bluetooth as derived in [4]. In this last work, authors measured energy depletion for a Nokia N95 smartphone (loss per second considering scanning and energy consumption per send), using the Nokia Energy Profiler v1.2.

2.5.1.2 Mobility model

For this performance evaluation considering energy consumption, we choose a more realistic mobility model: the Working Day Movement (WDM) [22] with the Helsinki Map available in ONE. Although the Random Waypoint mobility model is popular to use in evaluations of mobile ad hoc protocols, real users are not likely to move around randomly. If a node has visited a location several times before, it is likely that it will visit that location again. WDM model increases the reality of human node mobility by modeling some of the major activities performed by humans during a working week: working at the office, going out with friends in the evening and sleeping at home. Nodes can also move alone or in groups by walking, driving or riding a bus. In [22], it has been shown that the inter-contact time and contact time distributions generated by the WDM model follow closely the ones found in real-world traces.

2.5.1.3 Parameter settings

The main parameters used in our simulations are showed in Table 4.1. Table 4.2 lists the parameters for the Working Day Movement model. In Table 2.3 , the parameters for the routing protocols are specified. In particular, we choose for PRoPHET the values suggested in [46], while for Bubble Rap we use K-Clique as community detection algorithm and CWindow as centrality algorithm [36]. As in [22], the Helsinki map is divided into 4 main districts, with 3 additional overlapping districts simulating movements between the city center and the other districts, and one district covering the whole map. This last district has 4 buses, while the other districts have 2 buses. Moreover, each district has 10 offices (100 m x 100 m) and 4 meeting spots, and nodes are randomly distributed over these districts.

Table 2.1. Main Parameter Settings

Parameter	*Value*
Simulation area	7800 m x 8500 m
Transmission range	10 m
Message size	[5k,15k]
Transmit speed	250 kBps
Buffer size	20 MB
Simulation time	43200 s

Table 2.2. WDM Parameter Settings

Parameter	*Value*
Pedestrian speed	[0.5, 1.5] m/s
Pedestrian pause time	[0, 120] s
Car speed	[2.7, 13.9] m/s
Car pause time	[0, 120] s
Bus speed	[7, 10] m/s
Bus pause time	[10, 130] s
Working day length	28800 s
Pause time inside office	[10, 10000] s

Table 2.3. Routing Protocols Parameter Settings

Parameter	*Value*
Spray and Wait L	6
PRoPHET Pinit	0.75
PRoPHET β	0.25
PRoPHET γ	0.98
Bubble Rap K (K-Clique)	5
Bubble Rap time to wait before recalculating centrality values	600 s
Bubble Rap centrality time window	6 h
Bubble Rap # of time intervals to average node's centrality	5

2.5.2 Performance comparison and results

The focus of this analysis is to study the routing performance of Epidemic, Spray and Wait, PRoPHET, MaxProp and Bubble Rap protocols when energy consumption is taken into account. Our performance comparison starts with the *overhead cost* as a function of the number of nodes. This metric is calculated as the number of packets transmitted across the air divided by the number of unique packets created. When a node transmits a packet, its available energy is decremented. Consequently we believe that determining the ratio of the packets transmitted across the air to the number of unique packets created is important. Fig. 2.16 indicates that MaxProp protocol has the lowest overhead cost. Epidemic routing, Spray and Wait and PRoPHET behave very similarly, showing higher overhead costs than MaxProp. Even if MaxProp is flooding-based in nature, it orders packets by destination costs and it is able to transmit and drop packets in that order. Moreover, it allows the use of acknowledgements after a successful reception in order to instruct the other nodes to delete extra copies of the packet from their buffers. PRoPHET, which is forwarding-based in nature, do not blindly forward packets to every encounter and has an overhead cost slightly lower than the flooding-based Epidemic routing and Spray and Wait protocols. It is interesting to note that the social-based routing protocol Bubble Rap shows the highest overhead cost but comparable to flooding-based schemes. In this case, the higher number of packets transmitted across the air is due to the community structures and contact opportunities of the particular simulation scenario we considered.

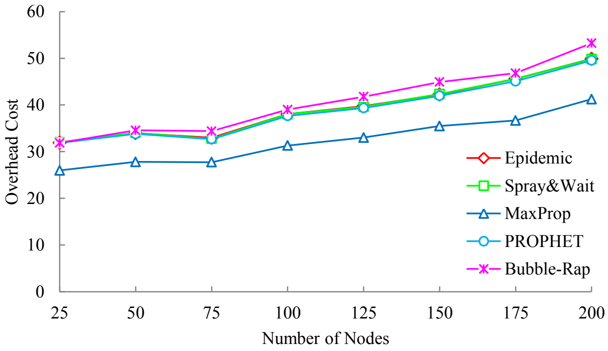


Fig. 2.16. Overhead cost as a function of the number of nodes.

The *delivery ratio* as a function of the number of nodes is showed in Fig. 2.17. This metric is calculated as the number of delivered packets divided by the number of unique packets created. The results of our simulations show the close relationship between the overhead cost and this performance metric. In the scenario with 25 nodes, the delivery cost is quite similar for all the protocols and is about 0.45. As the number of nodes increases, Bubble Rap shows the highest delivery ratio, while Epidemic routing, Spray and Wait, and PRoPHET have a lower and similar delivery ratio. As expected, MaxProp performs the worst, showing the lowest delivery ratio.

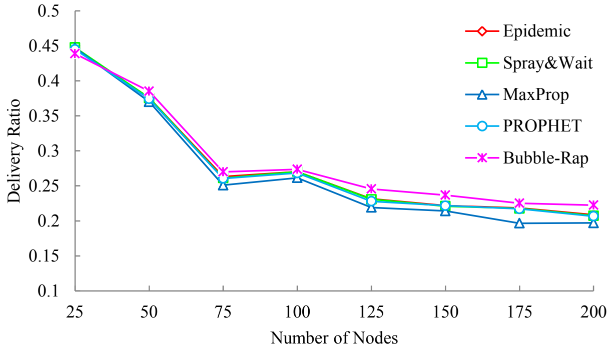


Fig. 2.17. Delivery ratio as a function of the number of nodes.

In Fig. 2.18, the *average latency* (the average time it takes a packet to be delivered) as a function of the number of nodes is presented. This metric measures the time it takes a message to be delivered. Even though applications using these protocols are relatively delay tolerant, it is still of interest to consider this performance metric when energy consumption is taken into account. Our results highlight that Bubble Rap is characterized by the highest average latency, while the other protocols show a similar and lower delay. These results confirm that in opportunistic networks is very difficult to achieve both high delivery ratio and low delivery latency considering energy consumption. Based on the above, we assert that the choice of a routing protocol for opportunistic network has to be based on the desired performance outcome, taking into account system constraints.

Figs. 2.19, 2.20, and 2.21 show the CDF of *hop counts* for 25, 50 and 100 nodes, respectively. We compute the hop count distribution for the deliveries in order to show the hop distance between sources and destinations. In Fig. 2.19, Epidemic routing delivers the better performance with 30% of hop counts within less than 2 hops. It also reveals that Bubble Rap delivers the worst performance with 26% of hop counts within less than 2 hops. The other protocols perform very similarly and better than Bubble Rap. As the number of nodes increases, all the protocols give similar results and perform worse than the scenario with 25 nodes. Figs. 2.20 and 2.21illustrate that there are 26% and 19% of packets delivered within 2 hops, respectively.

Fig. 2.22 exhibits the *average energy consumption* as a function of the number of nodes. This metric is calculated as the average percentage of the

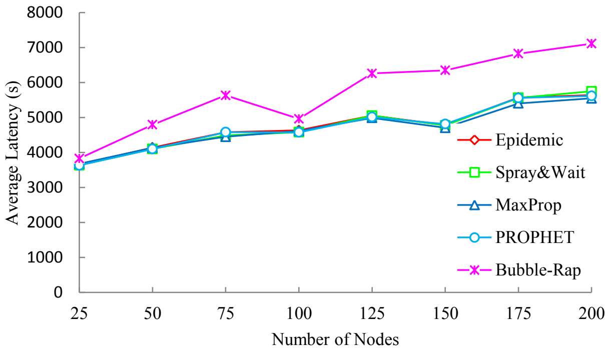


Fig. 2.18. Average latency as a function of the number of nodes.

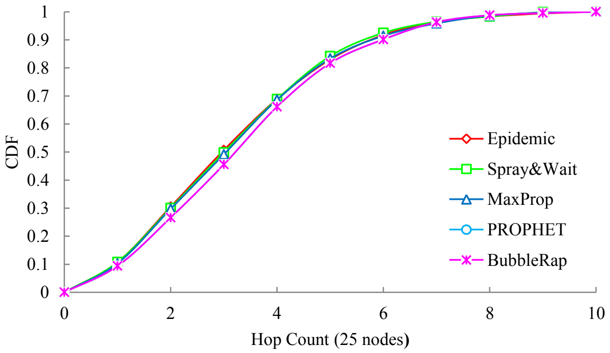


Fig. 2.19. CDF of hop counts for 25 nodes.

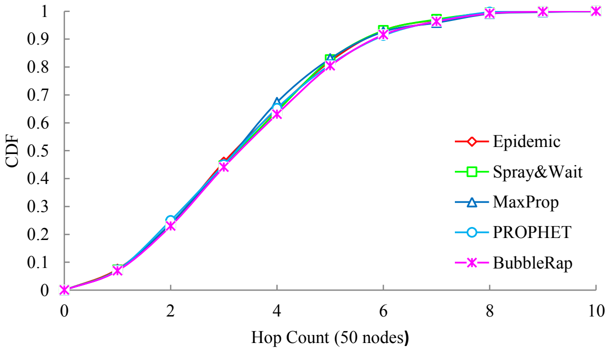


Fig. 2.20. CDF of hop counts for 50 nodes.

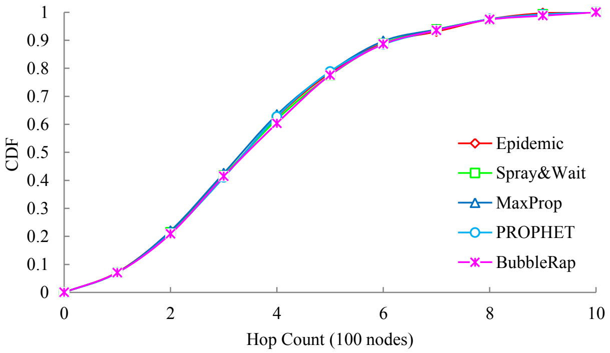


Fig. 2.21. CDF of hop counts for 100 nodes.

energy spent in transmission and scanning. We can see that MaxProp is characterized by the best performance, with an energy consumption between 49% and 50%. Epidemic routing, Spray and Wait, and PRoPHET perform similarly with an energy consumption which can be considered constant as the number of nodes increases and is about 51.5%. Looking at Bubble Rap, this protocol shows the worst behavior, with the highest energy consumption values. These results highlight that the performance differences between the routing protocols analyzed in this work are strictly related to the transmission overhead of redundant packet copies.

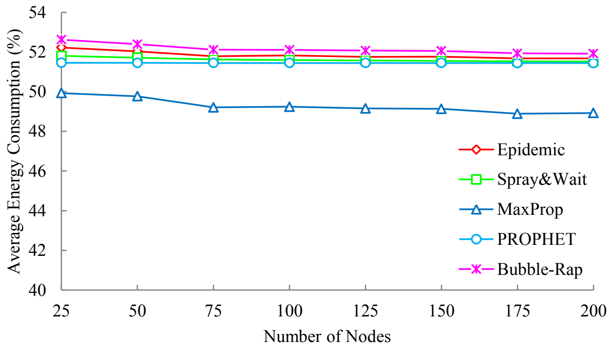


Fig. 2.22. Average energy consumption as a function of the number of nodes.

Fig. 2.23 presents the *average residual energy* as a function of the number of nodes. We analyze this metric in order to compare the performance of the different routing protocols, evaluating the average percentage of the residual energy of a node at the end of the simulation. These results confirm that

MaxProp performs better than the other protocols, with a residual energy which is about 0.55%, followed by PRoPHET, Epidemic routing, Spray and Wait, and Bubble Rap which maintain a residual energy between 0.13% and 0.4%

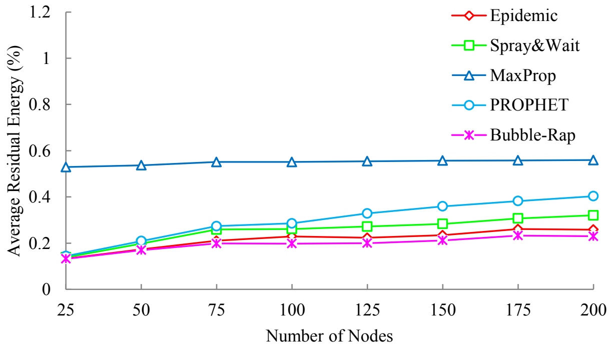


Fig. 2.23. Average residual energy as a function of the number of nodes.

2.5.3 Discussion

Analyzing the performance of Epidemic routing, Spray and Wait, PRoPHET, MaxProp and Bubble-Rap protocols when energy consumption is considered, we found that Bubble Rap performs better than the other routing protocols with regard to delivery ratio, while considering the delivery delay and the energy consumption it performs worse than the others, mainly due to its higher transmission overhead.

Considering delivery ratio, Max Prop performs clearly worse than the other protocols, but the energy consumption is the lowest.

When we compare Epidemic routing, Spray and Wait, MaxProp and PRoPHET protocols in terms of average latency, we find similar performances. They show a delivery latency smaller than Bubble-Rap, wasting a similar amount of energy.

Considering hop counts, Bubble Rap shows the lowest percentage of packet deliveries with 2 hops, while the behavior of the other routing protocols can be considered very similar.

2.6 Summary

In this chapter we described the development of Delay Tolerant Networks and opportunistic networks and their usefulness for challenged communication environments. We have noted the following points:

- We have motivated the importance of Delay Tolerant Architecture and opportunistic networks through example applications.
- We have motivated the need for a Delay Tolerant Architecture and opportunistic network research instead of using existing solutions for Mobile Ad Hoc Networks.
- We have seen that the main challenge for DTN and opportunistic routing is to decide which encounter nodes use for forwarding.
- We have analyzed a set of representative routing protocols for opportunistic networks and discussed their differences in terms of routing performance metrics.
- We have seen that in opportunistic networks is very difficult to achieve both high delivery ratio and low delivery latency when energy consumption is taken into account.

3

The usefulness of social networks and opportunistic networks

We have seen in Chapter 2 that the task of an opportunistic routing protocol is to decide if the encounter taking place between two humans carrying mobile devices is appropriate for routing any of their outgoing messages. Even if there are different classes of routing algorithms, in this thesis we focus on social-based algorithms and on the usefulness of the social connections between individuals for forwarding. In an opportunistic network the social connection is an encounter that takes place because individuals are co-located. However, opportunistic encounter patterns are not the only kind of relationship between individuals. Traditional online social networks (e.g. Facebook, Twitter, LinkedIn, etc.) represent other kind of social connections.

Both opportunistic networks and online social networks can be represented by a graph of edges and vertices. In this chapter, we investigate the similarity in the graph structure between these social networks in order to understand if the online social network can provide useful information for opportunistic forwarding. First, we will describe social network models; second, we will define the concepts of online and detected social networks; lastly, we will compare online and detected social networks using both a sociocentric and an egocentric approach for social network analysis.

3.1 Social networks

In general, a social network is a theoretical contract useful to study the relationships between individuals, groups, organizations, or even entire societies. The term *social network* is used to describe the social structure determined by the interactions between entities of the the network. The links (edge/ties) between these entities can represent different kind of social connections; co-location, friendship or business ties are good examples. In this thesis we are interested in social networks whose ties represent the following kind of relationships:

- co-location (extracted from Bluetooth, Zigbee or Wi-Fi interactions)
- online interactions (extracted from Facebook, or similar online social networks)
- common interests (extracted from online information)

To understand the structure of the social graphs, it is interesting to understand how social networks can be modeled. The simplest way to generate a network is to create random connections between nodes. The Erdős-Rényi model [23] can be used to create random graphs where every possible edge is created with the same constant probability. An example of Erdős-Rényi is showed in Fig. 3.1. This model, however does not capture the features of a social network because the distribution of the links does not follow a power-law. The model by Barabási and Albert [2], on the contrary, generates scale-free networks and is believed to be more similar to human social networks. Scale-free networks have power-law (scale-free) degree distributions and are widely observed in the Internet, in the world wide web, in citation networks, e-mail networks, in opportunistic contact traces and online social networks. The Barabási-Albert model incorporates two important general concepts that makes networks generated by this model more similar to real networks: *growth* and *preferential attachment*. Growth means that the number of nodes in the network increases over time, while preferential attachment means that the more connected a node is, the more likely it is to receive new links.

A good property of scale-free networks is their tolerance to random link failures, where up to 5% of nodes can fail before the communication capability decreases. However, these networks are susceptible to hub failures due to targeted attacks, where a loss of 5% of nodes can double the network diameter. An example of this model is shown in Fig. 3.2.

3.2 Online and detected social networks

The diffusion of mobile devices carried by users, such as smartphones, has led to a growing interest in new network architectures exploiting the social information. Commonly, the social network information is extracted from encounters between Bluetooth-enabled devices. The ubiquity of smartphones, in fact, permits to collect user co-presence information, which allows us to identify social ties grounded on real world interactions. We refer to the resulting co-presence network as the *detected social network* (DSN). However, the Internet added other social interaction techniques not based on physical meetings: email, chat and online social networks services such as Facebook, Twitter, MySpace[1], Orkut[2] and LinkedIn. We refer to the network describing online interactions as the *online social network* (OSN). Even though both face to face interactions and online social networks help people to construct and

[1] www.myspace.com

[2] www.orkut.com

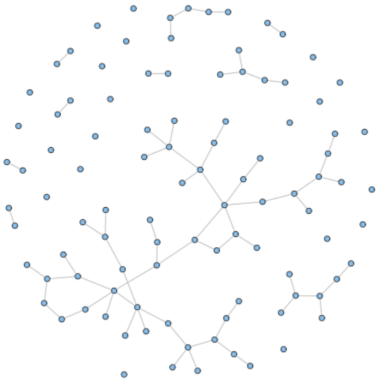


Fig. 3.1. Example of graph created with the Erdős-Rényi random graph model.

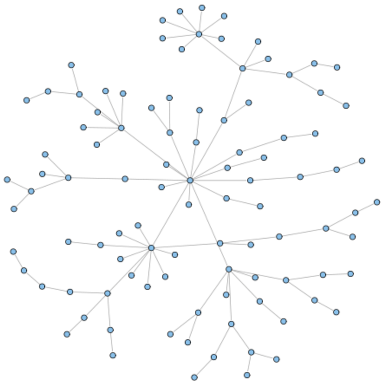


Fig. 3.2. Example of graph created using Barabási-Albert scale-free network model.

maintain social ties, it is not clear how individuals position themselves in the context of both online and real life social networks.

A considerable body of work analyzes the properties of online social networks and networks of physical encounters, but there exists little work that directly compares the online social network and the detected social network for the same set of users. The first work analyzing the differences between online and detected social networks [54] showed that individuals generally spend more time with their friends, therefore concluding that the two networks are not different. However, their experiment was performed at a three-day conference. As such, these results cannot be broadened to more than a contained event, where it is expected that subjects spend more time with their friends.

Another research [5] explored the use of detected social networks for routing in ubiquitous computing environments in comparison with detected social networks. This work focused on revealing the structure and role similarity and dissimilarity between online and detected social networks and showed that the two networks are not similar.

A more recent work has tackled the fusion of online and detected social network [42]. Studying the equivalence, the micro-correlation and the value in terms of acquaintances and navigating through the social ties of the two networks, the work showed that individuals involvement in each network vary considerably. However, the networks considered in this work are very sparse, and therefore have limited social information. Another work [57] considered the fusion of online and detected social networks, showing that the online and the detected social networks represent two different classes of social engagement that complement each other.

3.3 Exploring sociocentric and egocentric behaviors in online and detected social networks

In social network theory, there exist two distinct approaches to network analysis, deriving from two distinct historical traditions. The *sociocentric* network approach comes from sociology and involves the quantification of relationships between people within a defined group. On the other hand, the *egocentric* network approach comes from anthropology and this form of social network analysis is almost always about people rather than about groups. Egocentric networks, in fact, are defined as networks of a single actor together with the actors they are directly connected to, that is, their neighbors.

Both approaches capture important aspects of peoples social ties. In this In this section we analyze the similarities between online and detected social networks focusing on the egocentric and the sociocentric behaviors of users. This analysis was conducted with Prof. Salvatore Marano and published in [68]. Our work captures meaningful social similarities and differences between online and detected social networks using several structural measures, providing more exhaustive analysis than the previous studies related to online and detected social networks.

3.3.1 Data and methodology

We consider a real world trace a real world trace dataset, named SASSY [6], which contains ZigBee co-presence data and Facebook friend-list data. We used GEPHI [3] and UCINET [10] tools to perform social network analysis and compute egocentric and sociocentric measures.

SASSY is a dataset of sensor mote encounter records and corresponding social network data of a group of participants at University of St. Andrews. 25 individuals carried IEEE 802.15.4 T-mote sensor nodes for 3 months in order to collect co-location data. The ZigBee devices were able to detect each other within a radius of 10m and were programmed to broadcast a beacon every 6.67 seconds. At the beginning of SASSY experiment, participants declared their Facebook friends. Many participants knew each other: the mean friend-list size (i.e. the number of Facebook friends also participating in the experiment) was 9.8 with a standard deviation of 5.0.

We converted the SASSY data to two distinct social network graphs, where the number of vertices in each graph is N. We refer to the resulting opportunistic contact network as the detected social network (DSN), and to the Facebook related topology as the online social network (OSN). Fig. 3.3 and Fig. 3.4 show the DSN graph and OSN graph, respectively. We used the Yifan-Hu layout available in GEPHI to better visualize these graphs. In Table 3.1, the basic structural properties of the DSN and the OSN graphs are summarized.

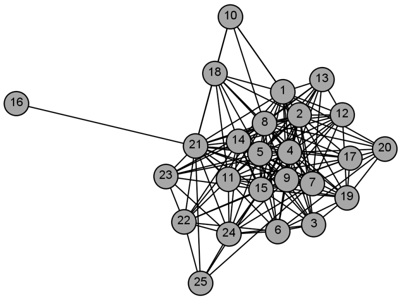


Fig. 3.3. The DSN graph for SASSY dataset.

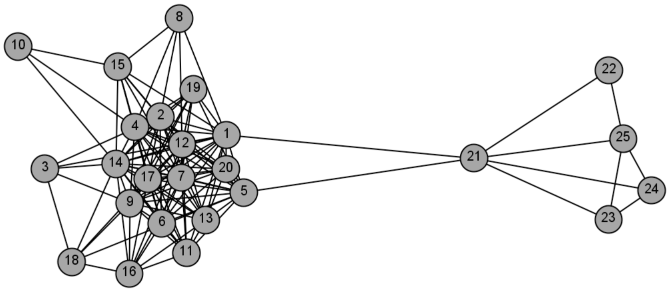


Fig. 3.4. The OSN graph for SASSY dataset.

Table 3.1. Structural properties of SASSY DSN and OSN graphs.

Property	*DSN*	*DSN*
Number of vertices	25	25
Number of edges	155	127
Average clustering coefficient	0.712	0.806
Graph density	0.517	0.423
Graph components	1	1
Average path length	1.503	1.853

3.3.2 Sociocentric analysis

The sociocentric method for social network analysis involves the quantification of interactions among a socially well-defined group of people and focuses on identifying global structural patterns. In particular, the method examines sets of relationships among actors that are regarded for analytical purposes as bounded social collectives. In this section, we assess the similarities and the differences between the DSN and the OSN in terms of sociocentric centrality measures and community structures.

3.3.2.1 Betweenness centrality

Betweenness centrality [28], which is also called sociocentric betweenness centrality, measures the influence a node has over the flow of information between every pair of nodes in the network graph under the assumption that information flows over the shortest path between them. Formally, the betweenness centrality of a node i is defined as

$$C_{betwenness}(i) = \sum_{i \neq j \neq k} \frac{g_{jk}(i)}{g_{jk}} \tag{3.1}$$

where g_{jk} is is the number of geodesic paths from j to k, and $g_{jk}(i)$ is the number of shortest paths from j to k which traverse i. Fig. 3.5 shows the empirical cumulative distribution functions (ECDFs) of sociocentric betweenness for the DSN and the OSN. The median betweenness is higher in the DSN (3.18 compared to 1.43 for the OSN). This indicates that nodes have more control over information in the DSN. Fig. 3.6 shows the correlation of sociocentric betweenness in the DSN and the OSN. We observe a relatively high correlation of this global metric, with a Pearson coefficient $\rho = 0.6901$, suggesting that users sociocentric betweenness is similar in both DSN and OSN, and that in general a user makes the same amount of relative effort to control information in each network.

3.3.2.2 Closeness centrality

Closeness centrality is a measure of the average shortest path of each node to each other node. With this measure, it is possible to identify the nodes which could reach others quickly. A main limitation of closeness is the lack of applicability to networks with disconnected components: two nodes belonging to different components do not have a finite distance between them. Thus, closeness is generally related to nodes within the largest component of a network. There are different definitions of closeness centrality in the literature [64] [79] [45]. We compute closeness centrality, for each node i, as

$$C_{closeness}(i) = \frac{1}{N-1} \sum_{j=1}^{N} d_{ij} \tag{3.2}$$

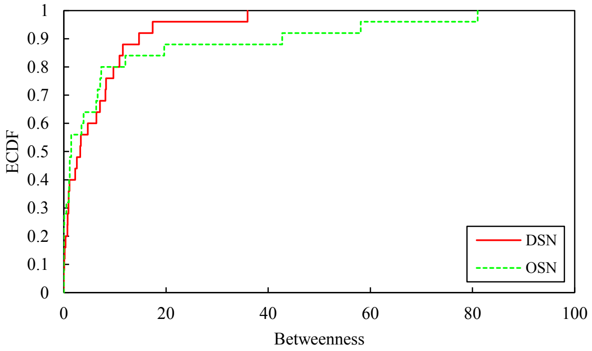


Fig. 3.5. DSN and OSN betweenness distributions.

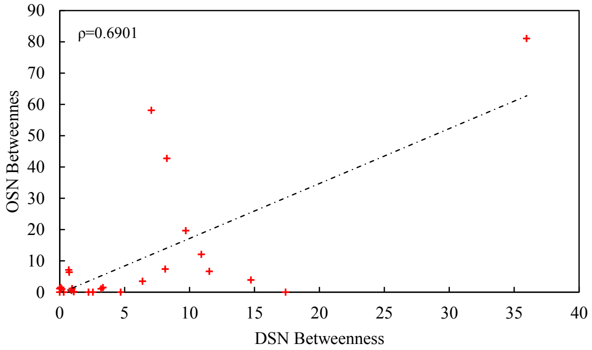


Fig. 3.6. Correlation between OSN and DSN betwenness centrality values.

where d_{ij} is the largest geodesic path from i to j. The results in Fig. 3.7 show that nodes in the DSN on average have a lower closeness (median of 1.54) than the OSN (median of 1.70). This result indicates that nodes in the DSN have a lower total distance to all the other nodes. The correlation between DSN closeness centrality and OSN closeness centrality is showed in Fig. 3.8. The low Pearson coefficient ($\rho = 0.4068$) suggests that there is not a significant correlation among the two closeness centrality metrics.

3.3.2.3 Eigenvector centrality

Eigenvector centrality [8] is a centrality measure defined in a circular manner. The centrality of a node is proportional to the sum of the centrality values of all its neighboring nodes. In other words, an important person can be characterized by its links to other important people. This measure is calculated using the adjacency matrix A of the undirected graph to find central nodes

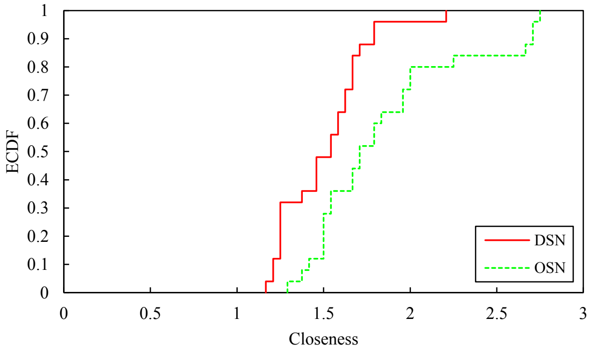


Fig. 3.7. DSN and OSN closeness distributions.

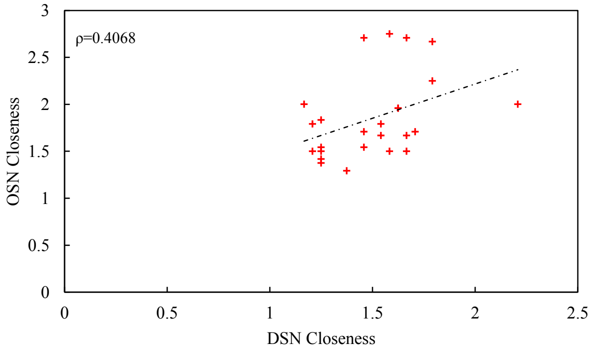


Fig. 3.8. Correlation between OSN and DSN closeness centrality values.

in the network. The eigenvector centrality for a node i is proportional to the sum of the eigenvector centrality values of its neighbor nodes. It is defined as

$$C_{eigenvector}(i) = \frac{1}{\lambda}\sum_{j=1}^{N} A_{ij}C_{eigenvector}(j) \tag{3.3}$$

where λ is the largest eigenvalue to assure the centrality is non-negative. Thus, is the i_{th} component of the eigenvector associated with the largest eigenvalue λ of the network. The results obtained for the DSN and the OSN eigenvector centrality distributions are presented in Fig. 3.9. Analyzing the median eigenvector centrality values, DSN shows a value of 0.66, while OSN is characterized by a higher median value of 0.73. These results indicate that nodes in the DSN on average are more likely to connect to nodes which are more central. As in the case of closeness centrality measure, the DSN and OSN eigenvector centrality values show a low correlation, with a Pearson coefficient of $\rho = 0.4007$ (Fig. 3.10).

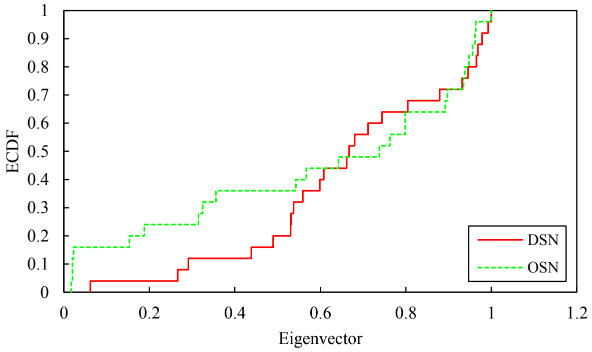


Fig. 3.9. DSN and OSN eigenvector centrality distributions.

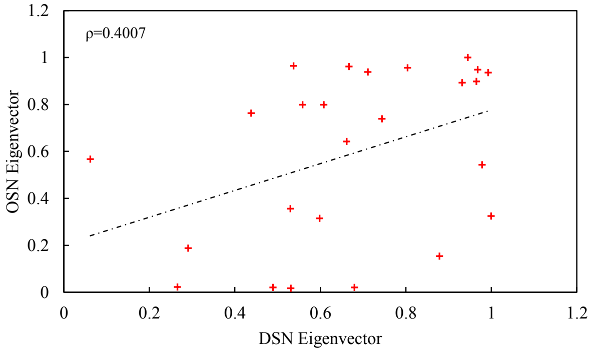


Fig. 3.10. Correlation between OSN and DSN eigenvector centrality values.

3.3.2.4 Bonacich power

Bonacich [9] proposed that power was a function of the connections of people in one's neighborhood. He defined two types of power. The first one states that the more connections the people in your neighborhood have, the more powerful you are. The second one considers nodes having neighbors with fewer connections more powerful. For a node i, it is defined as

$$C_{Bonacich,\beta}(i) = \sum_{j=1}^{N}(\alpha + \beta C_{Bonacich,\beta}(j))A_{ij} \qquad (3.4)$$

neighbors are likely to be dependent on ego, making ego more powerful. Negative values of the attenuation factor (between zero and negative one) compute power based on this idea. In our analysis, we investigate the Bonacich power based on dependency and we set $\beta = -0.5$. Fig. 3.11 shows the differences between the DSN and the OSN Bonacich power distributions: nodes in the OSN on average are more powerful (median of 2.20 compared to 2.17 for the DSN). In other words, nodes in the OSN are connected to weak neighbors which make them more powerful. Moreover, in Fig. 3.12 we observe an extremely very weak negative correlation of Bonacich power ($\rho = -0.2697$), suggesting that the Bluetooth network and Facebook network are completely different with regard to Bonacich power values.

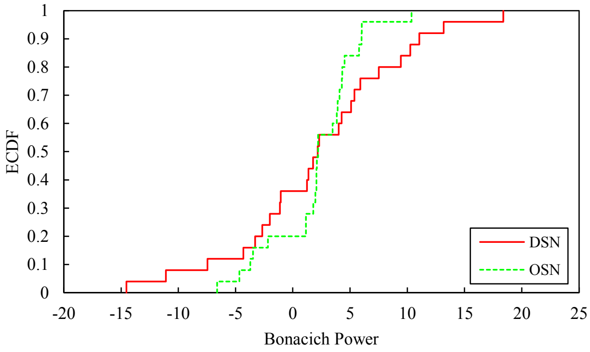


Fig. 3.11. DSN and OSN Bonacich power distributions.

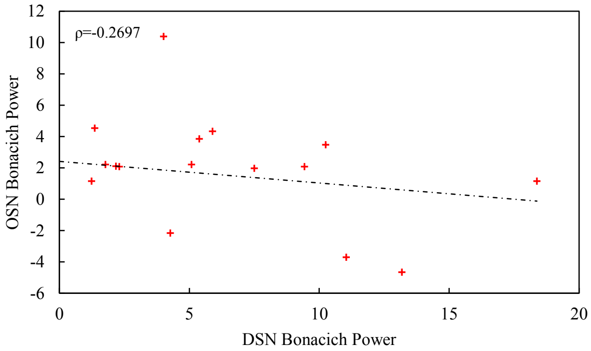


Fig. 3.12. Correlation between OSN and DSN Bonacich power values.

3.3.2.5 Modularity

Modularity [56] is a type of community detection approach measuring the chance of seeing a node in the network versus it its occurrence of being completely random. Formally, it can be defined as the sum of the random chance

$A_{ij} - \frac{C_{degree}(i)C_{degree}(j)}{2m}$ (where A_{ij} is an element of the adjacency matrix and $m = \frac{1}{2}\sum_i C_{degree}(i)$ the total edges in the network) over all pairs of nodes i,j falling in the same group, where s_i equals 1 if the two nodes fall in the same group and -1 otherwise:

$$Q = \frac{1}{4m}\sum_{ij}(A_{ij} - \frac{C_{degree}(i)C_{degree}(j)}{2m}s_i s_j). \tag{3.5}$$

Fig. 3.13 and Fig. 3.14 depicts the communities detected in the DSN and the OSN, respectively. The nodes are colored with their respective communities found by modularity algorithm. We observe three communities both in DSN and OSN. In the DSN the community sizes are 6, 9 and 10. The OSN shows communities of similar sizes (5, 8, and 12 nodes). Nodes 21, 22, 23, 24, 25 belong to the same community in both networks. On the contrary, the other nodes group differently in DSN and OSN. These results suggest that the overall DSN and OSN community structures are different, with regard to modularity community detection algorithm.

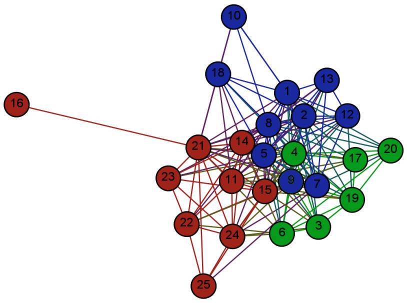


Fig. 3.13. Modularity community detection for DSN.

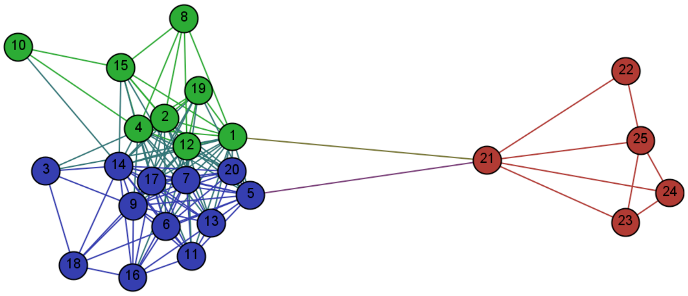


Fig. 3.14. Modularity community detection for OSN.

3.3.3 Egocentric analysis

After analyzing user behavior in the DSN and the OSN looking at sociocentric measures, we take a closer look at individuals behaviors at their local circumstances. Egocentric networks are defined as networks of a single actor together with the actors they are directly connected to. Describing and quantifying the variation across individuals in the way they are embedded in local social structures is the goal of the egocentric analysis. In this section, we compare the DSN and the OSN in terms of egocentric centrality measures.

3.3.3.1 Degree centrality

Degree centrality counts how many connections a node has and can be considered the most basic of all centrality measures. It is defined, for a node i, as

$$C_{degree}(i) = \sum_{j=1}^{N} a_{ij} \tag{3.6}$$

where $a_{ij} = 1$ if nodes i and j are connected by an edge, $a_{ij} = 0$.Fig. 3.15 presents the DSN and OSN degree centrality distributions. The median betweenness is higher in the DSN (12 compared to 11 for the OSN). This indicates that nodes have more contacts in the DSN. Fig. 3.16 shows the correlation of degree in the DSN and the OSN. We observe a low correlation of this ego metric, with a Pearson coefficient $\rho = 0.4049$, suggesting that users local connection to their social network vary in OSN and DSN, and that in general they do not make the same amount of relative effort to establish links in each network.

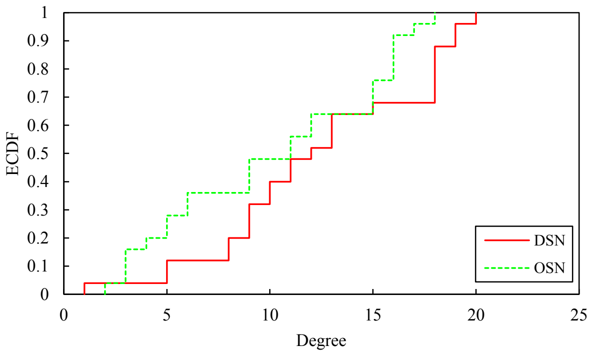


Fig. 3.15. DSN and OSN degree centrality distributions.

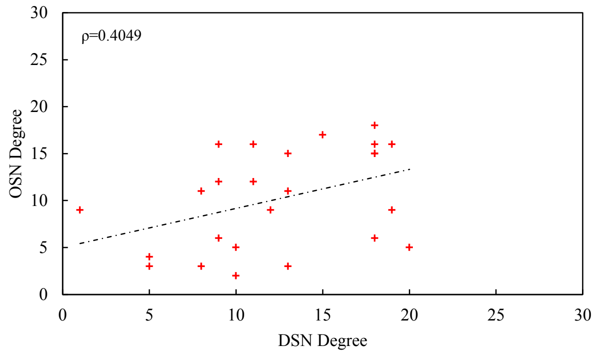


Fig. 3.16. Correlation between OSN and DSN degree centrality values.

3.3.3.2 Ego betweenness centrality

Ego betweenness is calculated using just the one-hop adjacency matrix of a node, as opposed to the global adjacency matrix used for sociocentric betweenness. The ego betweenness centrality metric can be calculated efficiently in a distributed way since only local information is required at each node. It has been shown that ego betweenness centrality values have a strong correlation to sociocentric betweenness values for most networks [47] [24]. Our results confirm this strong correlation. We found a Pearson coefficient of 0.9954 between the sociocentric betweenness and the ego betweenness measured on the DSN, and a Pearson coefficient of 0.7084 between the sociocentric betweenness and the ego betweenness measured on OSN. Fig. 3.17 depicts the OSN and DSN ego betweenness centrality distributions. We see that on average nodes in the DSN have a higher betweenness (median of 3.30), while nodes in OSN have a lower betweenness (median 1.25), as in the case of sociocentric betweenness. Fig. 3.18 shows the correlation of ego betweenness in the DSN and the OSN. Differently from sociocentric betweenness, we observe a low correlation of this ego metric, with a Pearson coefficient $\rho = 0.4076$.

3.3.3.3 Brokerage

Gould and Fernandez [30] explored the roles that ego plays in connecting groups. In Fig. 3.19 are depicted the five types of brokerage roles.

Brokerage roles are defined in terms of group membership as follows:

- coordinator: the broker mediates contact between two individuals from his own group;
- gatekeeper: the broker mediates an incoming contact from an out-group member to an in-group member;
- representative: the broker mediates an outgoing contact from an in-group member to an out-group member;

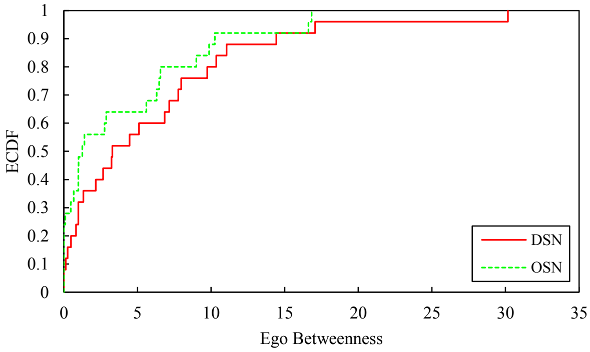


Fig. 3.17. DSN and OSN ego betweenness centrality distributions.

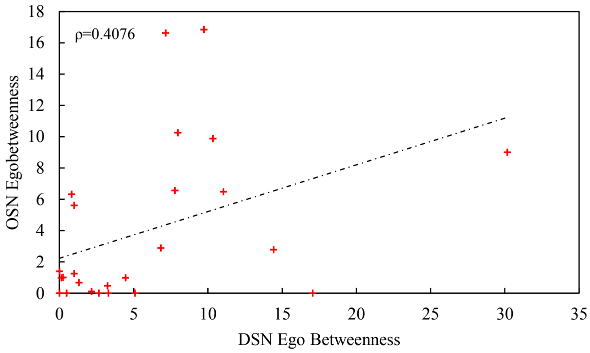


Fig. 3.18. Correlation between DSN and OSN ego betweenness centrality values.

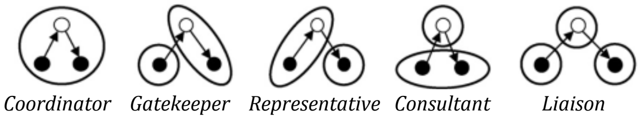


Fig. 3.19. Graphic representation of the five types of brokerage roles; the white nodes are the brokers, ellipses correspond to community boundaries.

- consultant: the broker mediates contact between two individuals from a different group;
- liaison: the broker mediates contact between two individuals from different groups, neither of which is the group to which he belongs.

The brokerage score for a given node with respect to a given role is the number of ordered pairs having the appropriate group membership(s) brokered by said node. We computed these scores for each node in DSN and OSN, grouping nodes with respect to modularity community detection algorithm. Fig. 3.20 shows the correlation between DSN and OSN brokerage scores. Considering the different roles (we do not take into account representative score because is not different from the gatekeeper score in undirected graphs), we observe a negative very weak correlation of coordinator score (ρ=-0.2319), a poor correlation of gatekeeper and consultant scores ($\rho = 0.3603$ and $\rho = 0.3603$, respectively) and a very weak correlation of liaison score ($\rho = 0.0492$). We can further observe that there are many gatekeeper brokers with also an high score both in DSN and OSN, while there are few liaison brokers, which are strategic for the information flow between different communities. In Fig. 3.21 the total brokerage score for each node in the DSN and the OSN is presented. This total score correspond to the total frequency of each role type within the network structure. We see that nodes in the DSN on average have a higher total brokerage score.

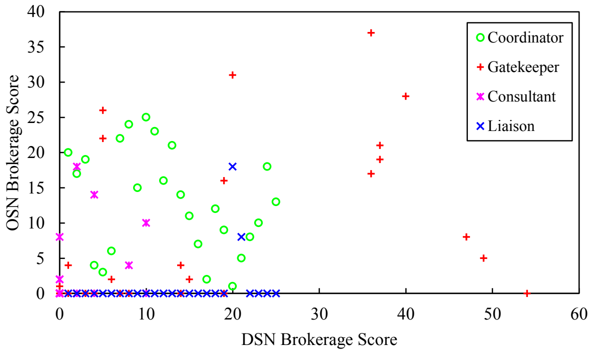


Fig. 3.20. Correlation between DSN and OSN brokerage scores.

3.4 Discussion

In this chapter we analyzed the structural properties of online and detected social networks, for a particular set of users. Specifically, we explored and compared the sociocentric and the egocentric behaviors of nodes, highlighting the structural similarities between the two types of networks and the differences in how individuals take part in co-presence network and Facebook network. Performing a sociocentric network analysis, we observed a relatively high correlation of betweenness centrality. On the contrary, the other users centrality measures in the online social network and the detected social network vary considerably. Moreover, we showed that the community structures of the two networks are different. The egocentric analysis further confirmed that the online and the detected social networks have different structural characteristics.

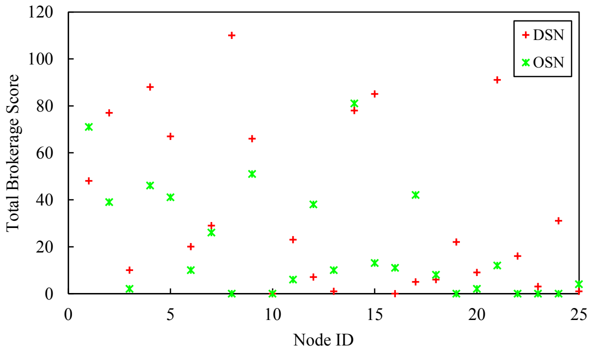


Fig. 3.21. Total brokerage scores in the DSN and the OSN.

We believe that our results, although limited to a single dataset, are representative of other similar experimental environments, but further generalization can be made only analyzing other similar datasets.

The relevant aspect of our work is the analysis of the contribution of central nodes within the online and the detected social networks. We feel that applications such as friend recommendation or routing schemes for opportunistic networks can benefit from this study providing a more complete understanding of user sociocentric and egocentric behaviors in real and virtual social networks. %

h

4

Multi-layer social networks

In Chapter 3 we have analyzed the structural similarities between online and detected social networks for a particular dataset and we have showed how individuals take part in co-presence network and Facebook network. In this chapter, we analyze more than two types of social networks for a particular set of users in order to investigate if the similarities between multiple social networks layers can be exploited for opportunistic forwarding. The aim of this study is to understand how much different online social networks reflect the user behavior in the detected social network. First, we will describe multiple social networks through a multi-layer social network model. Second, we will describe the dataset we analyzed and the multi-layer social network representing this dataset. Finally, we will compare the different social network layers focusing on node centrality, network motifs and detected communities.

4.1 Multi-layer social network model

Detected social network and online social network are representative of two different social contexts. If we extend the number of social contexts and consider multiple social networks for a particular set of users, we obtain a pillar, representing a single user connected to other users on several autonomous layers. Two users might be connected by many layers at the same time - e.g. two users may be connected in the detected social network, in the Facebook network, in LinkedIn and Twitter - while other users may be connected on just one layer - e.g. like co-workers connected only through LinkedIn or friends only through Facebook. The result is a complex system where there are several social network layers and where users exploit different kind of connections. The study of the whole system instead of each single network is useful to understand the overall role and position of users.

Our definition of a multi-layer social network model is based on simple undirected weighted networks. Weights can be used to represent the strength of the relationship.

Definition 1: a social network layer is an undirected weighted graph $G < V, E >$, where V is a set of vertices and E is a set of edges.

Definition 2: a multi-layer social network is a tuple $MLSN = (L_1, L_2, ..., L_n)$ where $L_i = G_i < V, E_i >$, $i \in 1, ..., n$ are social network layers.

In Fig. 4.1 an example of multi-layer social network with three users (without specifying edge weights) is showed.

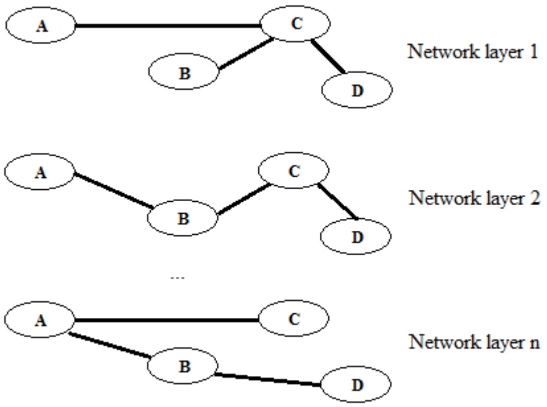


Fig. 4.1. A multi-layer social network.

The study of a multi-layer social network instead of each single social network is useful to understand the overall role and position of users. By comparing centrality measures computed on the multi-layer network, the frequency of network motifs or the communities, for example, it is possible to understand how much the single networks are complementary to each other or have a similar social function. In the following section, we study a particular multi-layer social network extracted from an experiment performed during a scientific conference and we present the results of the multi-layer network analysis.

4.2 Multi-layer social network in a conference environment

In this section, an example of multi-layer social network is described. In particular, we analyze both the patterns of contacts at a scientific conference and online social networking patterns for a particular set of users. The purpose of our analysis is to provide novel insights into the comparability of dynamic contact networks (detected social network) and online social networks, and to better understand the social contact behavior of individuals and groups by considering an overall complex system where there are multiple social networks describing their social dynamics. Such knowledge can feed into the design of better opportunistic schemes for supporting networking at conferences and at similar events.

In Chapter 3, we performed a similar analysis by taking into account only two layers of social networks: detected and online social network. In that case, we considered two static networks, the detected social graph where an edge between two nodes existed if there was at least one ZigBee contact between them, and the online social graph where an edge between two nodes existed if they were Facebook friends. In this section, we continue to consider a static graph for each network layer, but we use a different technique to produce a static graph describing a temporal network such as the detected social network. In particular, we apply a *Joint Diagonalisation* technique [26] to the dynamic contact network in order to decompose the behavior in time of the network and produce average static graphs for each times. These static graphs are representative of the most frequent propagation paths in the contact network. Then, we use these static graphs along with the online social network graphs to build the multi-layer social network.

In the following sections, the features of the dataset we considered and the Joint Diagonalisation technique will be described.

4.2.1 Lapland dataset

Lapland is a dataset collected during the ExtremeCom09 workshop in Padjelanta National Park (Sweden) [83]. The dataset contains Bluetooth scans of 17 conference attendees over a period of 4 days, their Facebook friendlists and their scientific interests. We use the participants' Facebook social network information to generate a *Facebook network* social graph, where a link between two nodes exists if they are friends, and the participants' scientific interests to generate an *Interest network* social graph where a link between two nodes i and j measures the similarity Sim_{ij} between them. This similarity measure is determined by examining each of k interests on the two nodes and counting the number of interests they have in common:

$$Sim_{ij} = \sum_{k=1}^{N} s_k(i,j) \tag{4.1}$$

$$s_k(i,j) = \begin{cases} 1 \; if \quad k_i = k_j \\ 0 \quad\; otherwise \end{cases} \tag{4.2}$$

where N is the number of scientific topics. It is important to underline that a link between two nodes is present if the nodes shares at least one scientific topic ($Sim_{ij} \neq 0$). Facebook network and Interest network are showed in Fig. 4.2 and Fig. 4.3, respectively.

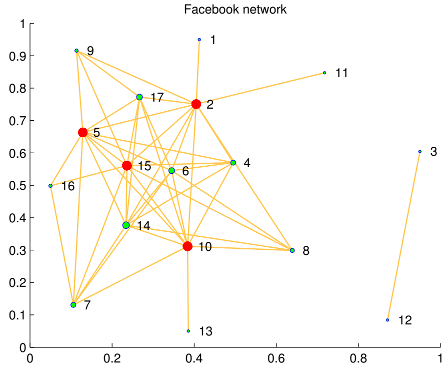


Fig. 4.2. Facebook network graph.

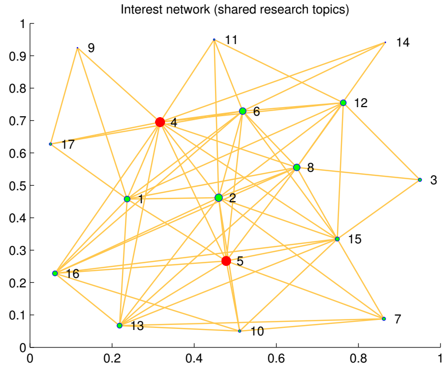


Fig. 4.3. Interest network graph.

The detected social network static graphs will be generated from Bluetooth contact data, using Joint Diagonalization technique. Before describing this technique and applying it to contact network, we show the statistical properties of the considered scenario. In Fig. 4.4 and Fig. 4.5 the contacts duration and the number of contacts distributions are presented. 52% of contact durations last more than one hour, and 4% last more than 3 hours. By looking at the number of contacts, we can see that 50% of number of contacts is greater than 26, and 15% is greater than 50.

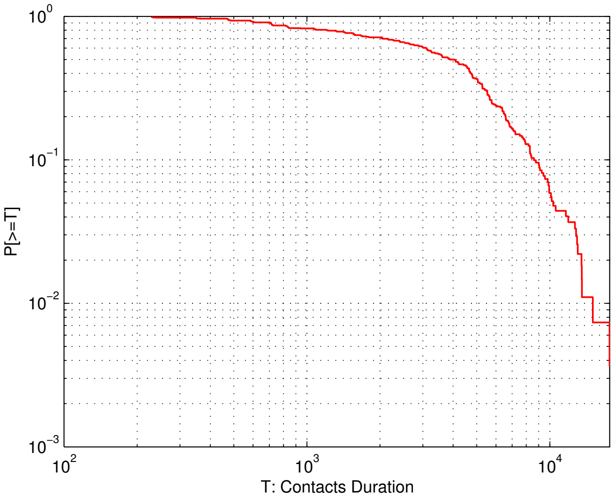


Fig. 4.4. Contacts duration distribution.

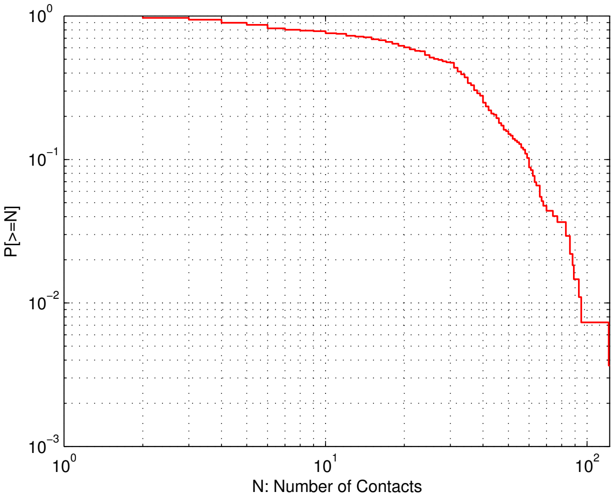


Fig. 4.5. Number of contacts distribution.

The correlation between contacts duration and number of contacts is showed in Fig. 4.6. Here the contacts duration is positively correlated to the number of contacts with a correlation coefficient of 0.9918. Most of the nodes do not meet regularly (low number of contacts) and have short contacts durations. Few nodes meet regularly (high number of contacts) and spend quite a lot of time together (high contacts duration).

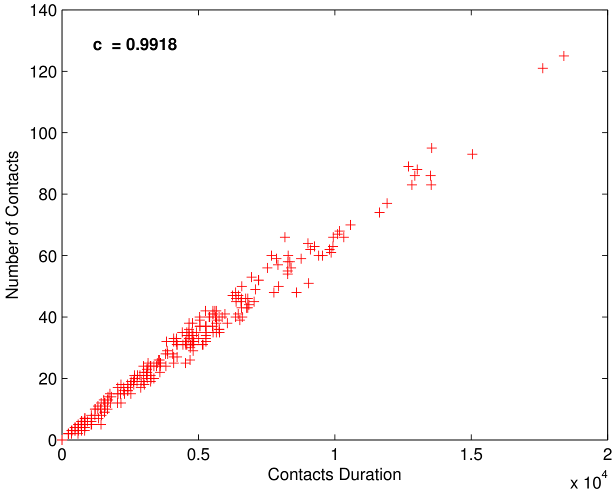


Fig. 4.6. Contacts duration versus number of contacts.

4.2.2 Joint Diagonalisation for dynamic network analysis

Joint Diagonalisation (JD) [26] is a technique used to track the changes in eigenspace (i.e. eigenvectors and eigenvalues) of a system. JD has been used successfully in different areas to track the evolution of systems via their eigenvectors and the application to the social network analysis is quite recent. In real-world contact networks, which are temporal networks, a single corresponding static graph is difficult to define. For this reason, JD can be used to decompose the behavior, in times, of contact network in order to create average static graphs for each time. Each of these static graphs, called *mode*, is a representation of the most common propagation paths corresponding to a particular time interval. This technique can be viewed as a mixture between a dynamic and static graph approach to social network analysis.

Given M samples of a network, $A_1 ... A_M$, JD produces an average eigenspace of the network. This technique seeks an orthogonal matrix such that:

$$A_i = UC_iU^T \ \forall i \tag{4.3}$$

If U correspond to the eigenvectors of A_i then C_i is diagonal, however no matrix U exists where all C_i are diagonal (except for the case in which all A_i are equal). JD seeks average eigenvectors $\bar{U}$ where the off-diagonal elements of C_i are minimized:

$$\bar{U} = \underset{U}{argmin}\ off_2(\textstyle\sum_{j=1}^{M} C_i) \tag{4.4}$$

where off_2 is the sum of the off diagonal elements squared, called the *deviation* of A_i from $\bar{A}$, δ_i:

$$\delta_i = off_2(C_i) = \sum_{k \neq j} |C_i^{k,j}|^2 \tag{4.5}$$

where $C_i^{k,j}$ is the k_{th} row and j_{th} column of C_i. Given the average eigenstructure of the the sample matrices, an average sampling matrix may be constructed from the eigenvector decomposition as:

$$\bar{A} = U\bar{C}U^T \tag{4.6}$$

where $\bar{A}$ is a matrix in which each entry is the average weight of the link as observed by the samples in the network and $\bar{C}$ is the average of diagonals of A_i projected onto $\bar{U}$.

Considering Lapland bluetooth contact data, we generated as samples 10000 spanning trees starting from a random node with the messages starting at random times (uniformly distributed). Then these trees were combined using JD in order to create $\bar{A}$. The average graph, $\bar{A}$, is represented in Fig. 4.7. Here, each link is a weighted link representing the proportion of trees using that connection and the size of a node is proportional to the sum of weights incident on that node. As it can be seen, there are some nodes bigger than others (e.g., node 12 and 14).

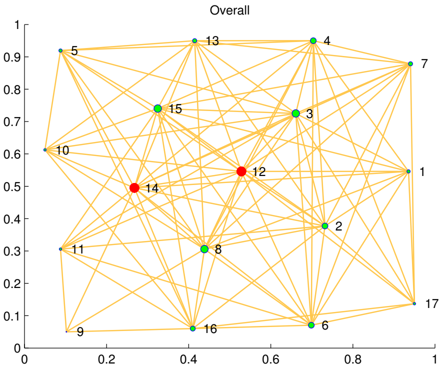


Fig. 4.7. Overall graph.

The average behavior of contact network is interesting because from a dynamic network we extracted an average static graph. However, if we examine the distribution of deviations, δ_i, from the average (Fig. 4.8), a more interesting behavior may be noticed. As it can be seen the distribution is multi-modal. A Gaussian mixture model was used to extract the two modes.

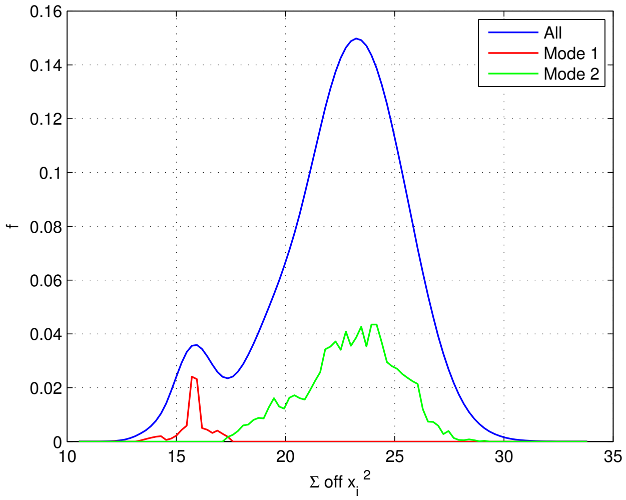


Fig. 4.8. Distribution of δ_i.

Fig. 4.9 shows the distribution of the sample start times. As it can be seen the contact network has different modes of operation at different times. Mode 1 covers part of the data with a low frequency, while mode 2 is the predominant one, being the first mode to occur and covering all the times. The topology of mode 1 is shown in Fig. 4.10. This mode shows a highly structured network. Mode 2, on the contrary, is less well defined (Fig. 4.11) and more similar to the overall mode.

4.2.3 Lapland multi-layer social network

JD technique allowed us to extract two static social graphs, namely Mode 1 and Mode 2, from a dynamic detected social network which are representative of the most common propagation paths. Moreover, we defined two types of online social graphs: Facebook network and Interest network. We consider this last network as an online social network because we extracted the scientific topics (interests) from participants' scientific papers available online.

Once we have a static graph for each social network level, we are able to define a multi-layer social network for Lapland dataset as follows:

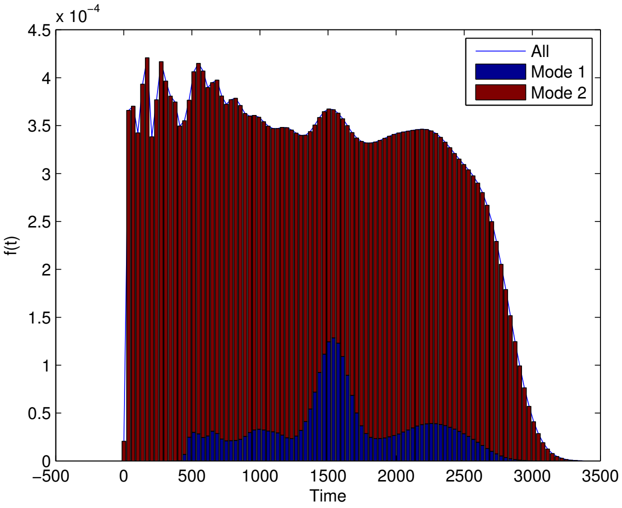


Fig. 4.9. Distribution of times by mode.

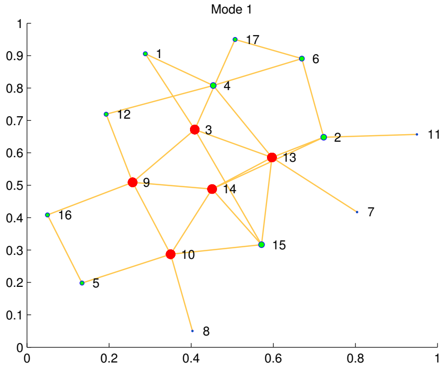


Fig. 4.10. Mode 1 graph.

- Layer 1: Mode 1
- Layer 2: Mode 2
- Layer 3: Facebook network
- Layer 4: Interest network

The features of this multi-layer social network will be analyzed in the following section.

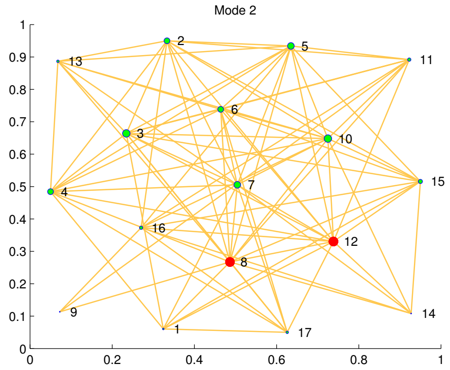


Fig. 4.11. Mode 2 graph.

4.3 Lapland multi-layer social network analysis

We now analyze Lapland multi-layer social using different social network analysis techniques in order to understand the social behavior of individuals by considering an overall complex system composed by multiple social networks. As previously said, such knowledge can feed into the design of better opportunistic routing schemes for supporting networking at conferences and at similar events. We structured our analysis into three sections: the first one, deals with the most recurrent sub-graphs or network motifs that repeat themselves among the various network layers, the second one, deals with node centrality measures in a multi-layer network and the third one, deals with community detection and groups dynamics at different layers.

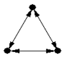
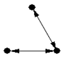
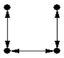
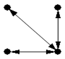
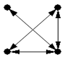
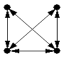
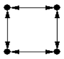
4.3.1 Network motifs analysis

Network motifs are repeated network structures that constitute meaningful building blocks of a more complex network [51]. The analysis of network motifs across all Lapland social network layers reveals which particular interactions are most common. By examining the networks motifs that repeat themselves among the various network layers, we are able to capture similar interaction behaviors between network layers. We use network motifs as a further method for comparing multiple social network layers.

The set of candidate motifs was selected from all subgraphs made of three and four nodes, as showed in the first column of Table 4.1. We considered seven different types of network motifs. The importance of motifs is evaluated by calculating the frequency of each considered motif.

In Table 4.1, for each network motif the frequency with which it occurred in the Mode 1 network (F_{M1}), in the Mode 2 network (F_{M2}), in the Facebook network (F_{FB}) and in the Interest network (F_{Int}) is showed. The most

Table 4.1. Network motifs frequencies at different network layers.

Motif type		F_{M1}	F_{M2}	F_{FB}	F_{Int}
m1		6.15%	43.64%	39.01%	38%
m2		93.85%	56.36%	60.99%	62%
m3		61.27%	13.81%	12.57%	19.78%
m4		20.81%	7.18%	13.91%	9.80%
m5		12.72%	32.91%	36.95%	38.40%
m6		1.73%	27.95%	23.62%	21.13%
m7		3.47%	4.01%	1.52%	1.08%

frequent motif is the subgraph of size 3 *m2*, where the frequency for Mode 1, Mode 2, Facebook network and Interest network are respectively 93.85%, 56.36%, 60.99% and 62%. This result is not surprising, since this structural pattern is quite frequent in social network. Another observation that can be made from Table 4.1 is that, for each network motif, except for motif *m7* which has low frequency and hence it is less significant, the frequency with which it occurs in Mode 2 network, Facebook network and Interest network is quite similar. These results further confirm that Mode 1 network differs from the other network layers. Moreover, if we compare only Mode 2 network, Facebook network and Interest network, we find that both for motif *m1* and motif *m2*, which are more frequent than the other motifs, the most similar networks are Facebook and Interest networks. As it can be seen, the difference between F_{FB} and F_{Int} is 1.01%, both for motif *m1* and *m2*. By looking at network motifs of size 4, for the most frequent motif *m5*, the difference between F_{FB} and F_{Int} is 1.45%.

4.3.2 Node centrality analysis

Another interesting study that can be made on a multi-layer network in order to compare the structure of each layer, is the node centrality analysis. The aim of this analysis is to understand how a particular centrality measures varies for a given node in each network. Similarly to the centrality analysis performed in Chapter 3 on Sassy dataset, we analyze degree, ego-betweenness, closeness and eigenvector centrality among the various social network layers. We computed for each node its degree, ego betweenness, closeness and eigenvector centrality at each network layer (*M1*, *M2*, *FB*, *Int*). In Table 4.2, the euclidean distance between the values of a particular centrality measure computed among two different layers is showed. $D_{M1,M2}$ measures, for example, the euclidean distance between a centrality measure in Mode1 network and the same centrality measure in Mode 2 network. As it can be seen, the most similar networks are Facebook network and Interest network, having the lowest distance for each centrality measure. This is an interesting result because we found that node centrality values on Interest network predict how much central a node will be on contact network (Mode 2). This result suggests that we could use centrality values measured on Interest network to drive routing decisions.

Table 4.2. Distance of node degree, ego betweenness, closeness and eigenvector centrality values.

	$D_{M1,M2}$	$D_{M1,FB}$	$D_{M2,FB}$	$D_{M1,Int}$	$D_{M2,Int}$	$D_{FB,Int}$
Degree	32.710	18.165	26.305	24.939	17.888	20.049
Ego betweenness	25.332	25.858	28.823	23.871	22.445	26.258
Closeness	139.219	301.634	251.793	108.298	64.549	285.413
Eigenvector	0.633	0.681	0.599	0.684	0.370	0.596

In Fig. 4.12, the degree distribution for each network layer is showed. Mode 2 network has the highest degree values with a median of 11 (Table 4.2), followed by Interest network with a median of 9, Facebook network with a median of 6 and Mode 1 network with a median of 3. By looking at Fig. 4.13, Mode 2 network shows again the highest values with a median of 3.115, followed by Mode 1 network with a median of 2, Interest network with a median of 1.85 and Facebook network with a median of 0.25. In the case of ego betweenness Mode 1 network and Mode 2 network are more similar. If we consider closeness distribution (Fig. 4.14), we find that Mode 2 network presents the highest centrality values with a median of 76.190, followed by Interest network with a median of 69.565, Facebook network with a median of 57 and Mode 1 network with a median of 42.105. Finally, as it can be seen from eigenvector centrality distribution (Fig. 4.15), Interest network shows for the first time the highest median value (0.272), followed by Mode 2 network (0.262), Facebook network (0.232) and Mode 1 network (0.141).

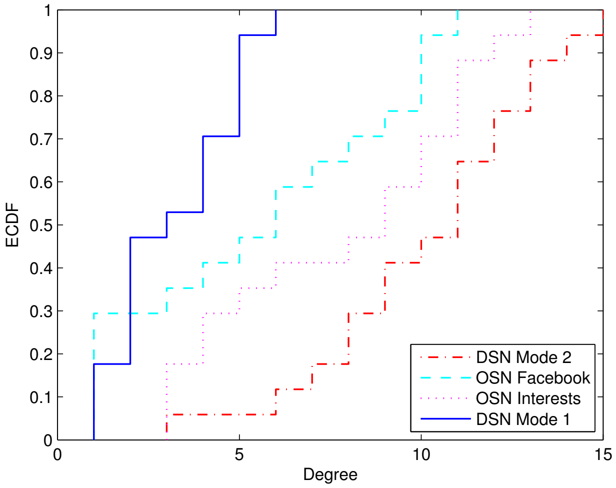


Fig. 4.12. Degree distribution for different social networks layers (Lapland dataset).

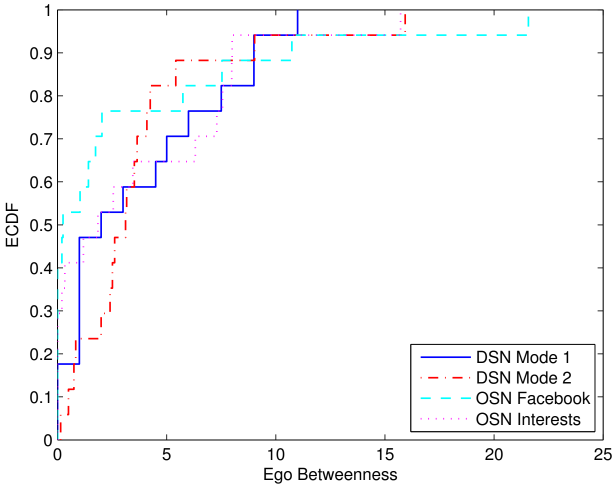


Fig. 4.13. Ego betweenness distribution for different social networks layers (Lapland dataset).

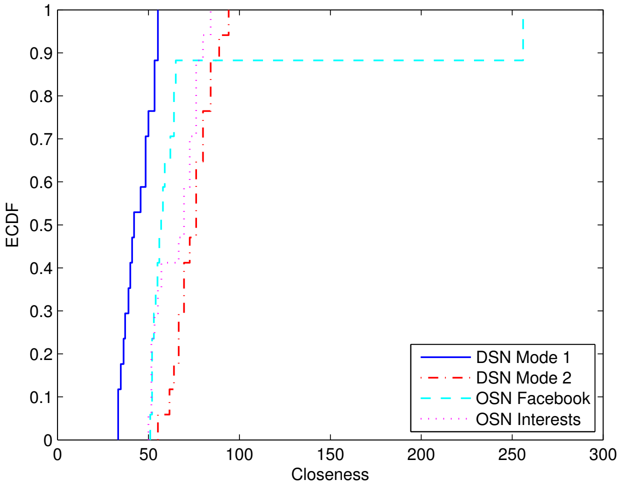


Fig. 4.14. Closeness distribution for different social networks layers (Lapland dataset).

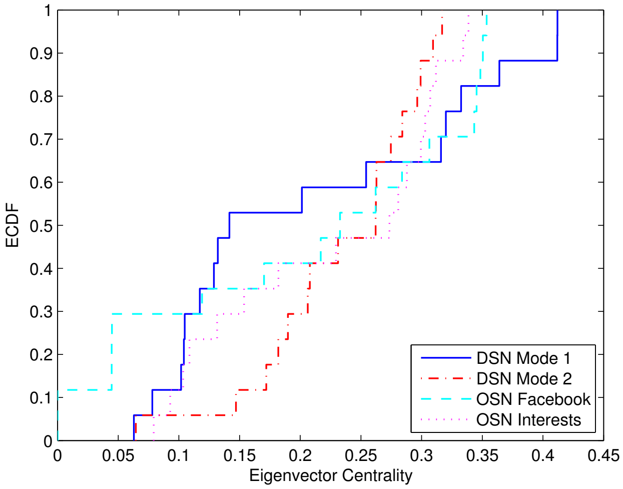


Fig. 4.15. Eigenvector centrality distribution for different social networks layers (Lapland dataset).

Table 4.3. Median values for different centrality measures distributions.

	Degree	*Ego betweenness*	*Closeness*	*Eigevector*
Mode 1	3	2	42.105	0.141
Mode 2	11	3.116	76.190	0.262
Facebook	6	0.25	57	0.232
Interest	9	1.85	69.565	0.273

4.3.3 Multi-layer community detection analysis

In this section we focus on groups by analyzing the communities at each network layer. We use Fiedler Clustering algorithm [27] as community detection method and then we compute similarity between communities at different layers. The eigenvector for the nonzero smallest eigenvalue of a Laplacian matrix is called Fiedler vector and can be used for decomposing graphs into structural components. The following summarizes the communities detected by this method for each social network layer:

- Mode 1 network:
  - C1=[1,3,17,12,2,11,4,7,13,6]
  - C2=[5,8,10,9,16,12,14,15]
- Mode 2 network :
  - C1=[8,9,1,11,2,16,5,10,13,3,6,4,12,7,17]
  - C2=[14,15]
- Facebook network:
  - C1=[2,11,1]
  - C2=[4,14,6,8,5,9,15,17,7,16,10,13]
- Interest network:
  - C1=[1,4,17,9,5,16,13,7,10,2,11,3,12,15]
  - C2=[6,8,14]

In Fig. 4.16, Fig. 4.17, Fig. 4.18 and Fig. 4.19, the dendrograms representing the hierarchical clustering for each network layers are showed.

Since we are interested in measuring the similarity between the communities detected in Mode 2 network, Facebook network and Interest network, we consider the biggest community for each of these social network layers and then we compute the Jaccard index as similarity measure. Given two sets A and B, Jaccard index is measured as follows:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.7}$$

The computed similarity values are the following:

- $Jaccard(C1_{M2}, C2_{FB}) = 0.558$
- $Jaccard(C1_{M2}, C1_{Int}) = 0.812$

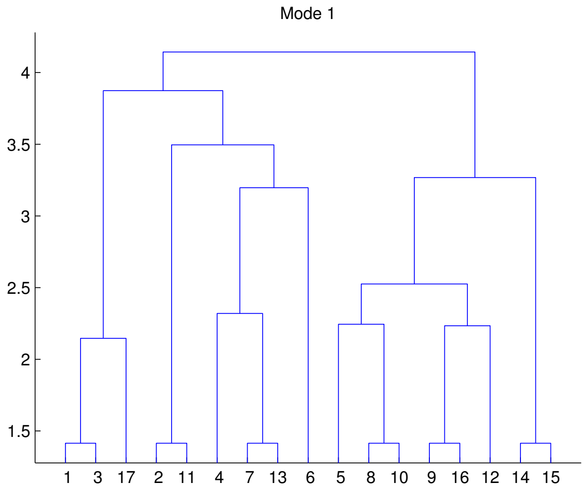


Fig. 4.16. Communities based on Fiedler clustering (Mode 1 network).

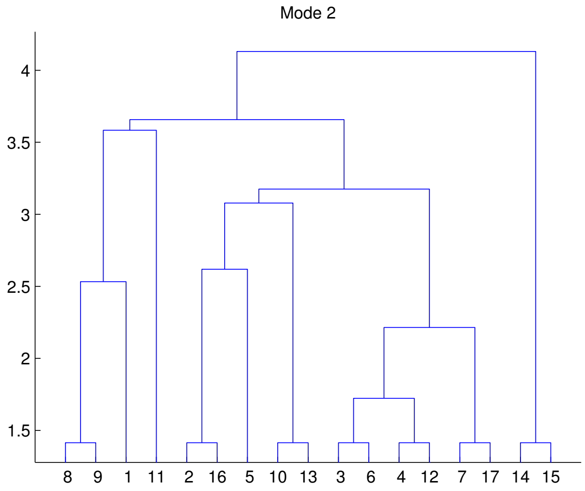


Fig. 4.17. Communities based on Fiedler clustering (Mode 2 network).

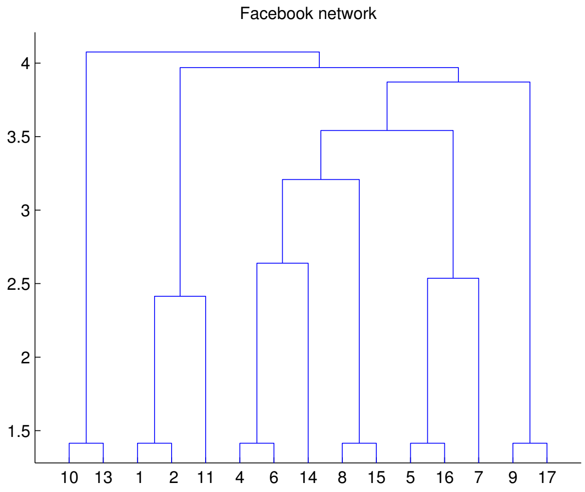


Fig. 4.18. Communities based on Fiedler clustering (Facebook network).

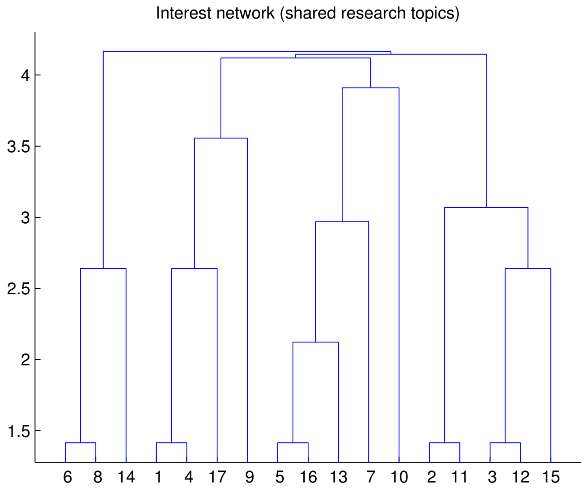


Fig. 4.19. Communities based on Fiedler clustering (Interest network).

- $Jaccard(C2_{FB}, C1_{Int}) = 0.529$

As it can be observed, Mode 2 network and Interest network are more similar, with a Jaccard index of 0.812, while the similarity indexes computed between Mode 2 network and Facebook network, and between Facebook network and Interest network are comparable.

4.4 Discussion

In this chapter we introduced the concept of multi-layer social network and analyzed the structural properties of this complex network. Specifically, we considered a particular dataset, namely Lapland, and defined four different social network layers. Using a Joint Diagonalisation technique, we extracted from the dynamic contact network two modes of operation, namely Mode 1 network and Mode 2 network , which are two static graphs representing the most common propagation paths in the detected social network. Adding to Mode 1 network and Mode 2 network, Facebook network and Interest network layers, we defined a multi-layer model for Lapland dataset. Then, we explored the network motifs occurring at each layer, discovering that Mode 2 network, Facebook network and Interest network are similar.

Analyzing node centrality in the multi-layer network we found again that nodes behave similarly in Mode 2 network, Facebook network and Interest network showing similar centrality values at each network layer. Finally, we measured the similarity between communities and we found the highest Jaccard index for Mode 2 network and Interest network.

The relevant aspect of the analysis of this multi-layer social network is that we found similarities between different levels of online and detected social networks which could be exploited for opportunistic routing. In the following chapter we describe our proposal of an opportunistic routing scheme using multi-layer social networks to improve routing performances.

5

Multi-layer social networks for opportunistic routing

Opportunistic networks may constitute an important part of future mobile networks and understanding how efficiently route information within these networks is an important research challenge. Recent work has demonstrated the importance of social network information for routing in mobile-computing environments, focusing on encounter histories or both on encounter histories and online social network information. Many of these works determine forwarding paths using detected communities or centrality measures extracted from contact network. By examining the social network of the nodes encountered by a particular node, it may be possible to optimize routing by forwarding messages to nodes which are encountered more often. Communities and centrality measures computed on contact network, however, may miss important aspects. For instance, during the bootstrapping phase of the network, the detected communities or centralities may produce sub-optimal forwarding paths because the detected social network may omit important ties. A user may have strong social ties to another user that he does not encounter frequently and the detected social network, which considers a low number of encounters between these users, may produce sub-optimal routing. In such situations, online social networks could identify and predict strong ties.

This chapter explores the use of multiple social networks for opportunistic routing. First, we will discuss how current research addresses the problem of opportunistic routing using both detected social network and online social network information. Second, we introduce a new opportunistic routing protocol, ML-SOR (Multi-Layer Social Network based Opportunistic Routing), which uses information from a multi-layer social network to drive routing decisions. Third, we will evaluate the performance of our proposal in different simulation scenarios and compare it to other existing routing schemes, demonstrating that the use of multi-layer social network information improves opportunistic routing.

5.1 Detected and online social networks for opportunistic routing

In this section we describe some protocols which drive routing decision using both detected social network and online social network information or online social network only. Considering our multi-layer social network model, these protocols exploit at most two social network levels to perform routing: contact network and online social network.

5.1.1 MobiClique

In [62], Pietilăinen et al. design and implement a novel mobile social networking middleware, named MobiClique, that uses Facebook network to bootstrap an opportunistic network. Authors designed MobiClique as a way to leverage virtual and physical worlds so that users can move between them in a way that enhances both.

MobiClique bootstraps the network using the existing Facebook user social profiles consisting of a unique user identifier, the friendlist and a list of groups (or networks) consisting of users sharing some common interest. During an opportunistic encounter, if the two user profiles are friend or share some interest, the users are alerted and can choose to exchange messages. Each MobiClique node executes a periodic loop that consists of three phases:

- (1) *neighbor discovery* (using Bluetooth or WiFi) - Bluetooth device discovery or broadcast beacons on a well-known WiFi SSID are used to perform neighbor discovery.
- (2) *user identification* - During the first encounter devices exchange their full social profile. During subsequent contacts the full profiles are exchanged only if the profile has changed since the last encounter.
- (3) *data exchange* - Messages are forwarded according to two rules: (i) unicast messages are sent either if the destination is met directly or forwarded through friends of the destination, and (ii) group messages are flooded within the corresponding interest group so that each member of the group will participate to the forwarding until everybody has received the data.

5.1.2 PeopleRank

Mtibaa et al. [53] make use of online social network information to compute node rankings. Their protocol is similar to the PageRank algorithm [11] used by Google search engine to measure the relative importance of a Web page within a set of pages. PeopleRank gives higher weight to nodes if they are socially connected to other important nodes of the network. In a completely distributed fashion, PeopleRank identifies the most popular nodes (in a social context) to forward the message to, given that popular nodes are more likely to meet other nodes in the network.

PeopleRank considers the online social network (called *social graph*) to compute node rankings. In the social graph, a social relationship between two nodes is defined either if nodes are declared friends, or if they share interests. When two neighbor nodes in the social graph meet, they exchange two pieces of information: their current PeopleRank values and the number of social graph neighbors they have. Then, the two neighbors update their PeopleRank values. Implicitly, the algorithm exploits contact networks since the PeopleRank value is updated every time the nodes meet.

5.1.3 Social Role Routing (SRR)

In [4], Bigwood and Henderson present an opportunistic routing protocol, called Social Role Routing (SRR), that uses online social network information to bootstrap the opportunistic network and applies the social network analysis technique of role analysis to select nodes to act as message relays. SRR employs the social science technique of *regular equivalence* [80] to compute nodes' roles. This technique partitions nodes into classes, where all nodes in a class are connected to the same classes of nodes. Considering these roles, messages are forwarded only to intermediate nodes that are in the same role, or in a role adjacent to the destination's role.

Before the network starts up, each node stores a copy of a *Role Connectivity Graph* (RCG), which has been precomputed using the online social network of the participating nodes, allowing them to compute the geodesic distance between roles. Each node is characterized by a unique identifier (ID) and stores the identifier of the role to which it belongs (RoleID). When two nodes meet, they exchange their ID, RoleID and identifiers of each of the messages they are carrying. If a node does not have in its buffer a particular message carried by the encountered node, the node will check the geodesic distance of the encountered node's role from the destination node's role. If this distance is less than or equal to 1, it forwards a copy of the message to the encountered node.

It has been showed that this scheme is particularly advantageous during the network network startup, where bootstrapping the network using roles provides an advantage over having to create an encounter history. Moreover, it reduces message duplication prolonging battery lifetime.

5.2 ML-SOR: Multi-Layer Social network based Opportunistic Routing

In the last section we described some opportunistic routing schemes using detected and online social network information to drive routing decisions. Considering the multi-layer social model described in Chapter 4, these routing protocols exploit at most two social network levels: contact network and online social network.

The results of our multi-layer social network analysis showed that there are online social network layers similar to contact network. Our idea is to exploit more stable social information provided by several online social network layers to augment available partial contact information. In other words, we use a multi-layer social network to provide efficient data routing in opportunistic networks.

In this section we describe ML-SOR, our multi-layer social network based opportunistic routing proposal. Simulating real mobility traces and their social interactions, we will show how a multi-layer social network can be used to improve opportunistic routing.

5.2.1 ML-SOR social metric

ML-SOR is based on a social metric which exploits information extracted from different social network layers. This social metric is calculated using a combination of three measures:

- *centrality*
- *tie strength*
- *tie predictor*

Centrality in graph theory and network analysis quantifies the structural importance of a vertex within the graph (for example, how important a person is within a social network); typically, a central node has a stronger capability of connecting other network members. As previously seen, there are several ways to measure centrality. ML-SOR social metric computes node centrality using a long-term cumulative estimate of degree centrality, named $C_{CDegree}$. We choose degree as centrality measure because it is simply to be computed and requires only local knowledge of the network. Degree centrality counts how many connections a node has and it is defined, for a node i, as

$$C_{degree}(i) = \sum_{j=1}^{N} a_{ij} \tag{5.1}$$

where $a_{ij} = 1$ if nodes i and j are connected by an edge, $a_{ij} = 0$. We fix a time slot (e.g. 6 hours), so that each node calculates the number of unique nodes seen throughout this time interval. $C_{CDegree}(i)$ will be calculated as the node's average degree over a set of time slots including the most recent time slot and all the previous ones.

ML-SOR computes $C_{CDegree}(i)$ at contact network (detected social network) layer. This choice accounts for the dynamic evolution of this network layer over time.

Considering that centrality is measured using the contact history and does not account for the future links availability, we include into ML-SOR social metric a *tie strength* indicator which identifies links that have a higher probability to be activated. Social ties on online social networking websites, such as Facebook, Twitter[1] or LinkedIn, are more stable and hence stronger than contact network ties. Consequently, they are a good measure of whether a tie will be activated. ML-SOR calculates tie strength between a node i and a node j at online social network layer l as:

$$TS(i,j,l) = \begin{cases} 1 \; if \; i \; and \; j \; are \; connected \; at \; layer \; l \\ 0 \qquad\qquad\qquad otherwise \end{cases} \tag{5.2}$$

The total tie strength between two nodes will be an aggregation of the indicators measured at each online social network layer:

$$TS_{TOT}(i,j) = \sum_{l=1}^{L} TS(i,j,l) \tag{5.3}$$

where L is the total number of considered online social networking websites.

ML-SOR social metric takes into account a third measure useful to predict future collaborations between two nodes. A *tie predictor* is computed on an interest network layer, where a link between two nodes exists if they have in common at least one interest. Examining common neighbors of a pair of nodes i and j at interest network layer, we can predict a future interaction between i and j. If i and j have one or more common neighbors, the probability of future collaboration increases. ML-SOR compute the tie predictor $TP(i,j)$ of a possible future collaboration between i and j as a common neighbor measure based on Jaccard index:

$$TP(i,j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} \tag{5.4}$$

where $N(i)$ are the number of neighbors of node i and $N(j)$ are the number of neighbors of node j.

For each measure, ML-SOR computes the score of node i for delivering a message to node d compared to node j as follows:

$$CScore(i,j) = \frac{C_{CDegree}(i)}{C_{CDegree}(i) + C_{CDegree}(j)} \tag{5.5}$$

$$TSScore(i,j,d) = \frac{TS_{TOT}(i,d)}{TS_{TOT}(i,d) + TS_{TOT}(j,d)} \tag{5.6}$$

$$TPScore(i,j,d) = \frac{TP(i,d)}{TP(i,d) + TP(j,d)} \tag{5.7}$$

The ML-SOR social metric *MLScore* considers all scores of equal importance and is given by the sum of the contributing score values:

[1] Here we consider a tie between a user A and a user B, if A follows B and vice versa

$$MLScore(i,j,d) = CScore(i,j) + TSScore(i,j,d) + TPScore(i,j,d) \quad (5.8)$$

The ML-SOR social metric $MLScore$ captures the overall value a node has when compared to an encountered node across all measures (centrality, tie strength, tie predictor).

5.2.2 ML-SOR scheme

The forwarding process in ML-SOR is based on the comparison of the $MLScore$ social metric. When two nodes meet they exchange their centrality values, one or more lists of online social contacts (one list for each online social networking website) and a list of contacts with common interests. Each node then examines the messages it is carrying and computes the $MLScore$ of each message destination. Messages are then forwarded to the nodes with higher $MLScore$ for the message destination node. We can see the pseudo code for ML-SOR in Algorithm 1.

Algorithm 1 ML-SOR forwarding algorithm

```
function encounterNode(N):
    exchangeCentralityValues()
    exchangeContactLists()
    for all message in message_buffer do
    myMLScore ← computeMLScore()
    encounterMLScore ← computePeerMLScore()
    if encounterMLScore ≥ myMLScore OR N=destination then
    forwardMessage(message,N)
```

5.3 Performance evaluation

To evaluate ML-SOR, we perform trace-driven simulations using Lapland dataset described in Section 4.2.1 and Sigcomm2009 dataset [61], which contains data collected by the opportunistic mobile social application MobiClique. The application was used by 76 participants during SIGCOMM 2009 conference in Barcelona, Spain. The dataset includes traces of Bluetooth device proximity and the social profiles (Facebook friends and interests) of the participants. For both dataset we consider a multi-layer social network composed by the following layers:

- Bluetooth contact network
- Facebook network
- Interest network

5.3.1 Simulation environment

Our simulations are carried out on the Opportunistic Network Environment (ONE) simulator [40], considering five routing protocols: epidemic routing, PRoPHET, Bubble Rap, H-Bubble Rap and ML-SOR. Epidemic routing, PRoPHET and Bubble Rap were described in Section 2.4. We implemented H-Bubble Rap as an hybrid version of Bubble Rap where local centrality and global centrality metrics are respectively replaced with an $MLScore$ metric computed with a local $C_{CDegree}$ and a $MLScore$ metric computed with global $C_{CDegree}$. For PRoPHET and Bubble Rap we set the same parameters of Table 2.3. For ML-SOR and H-Bubble Rap the time to wait before recalculating centrality values is set to 600 s, the centrality time slot is set to 6 hours and the number of time intervals to average node centrality is set to 5. Finally, the total simulation time for Lapland dataset is set to 399812 s, while for Sigcomm dataset is set to 320593 s.

To compare the routing protocols, we analyze the following commonly used metrics:

- Delivery ratio: the ratio of the number of delivered messages to the number of all messages.
- Overhead cost: the number of packets transmitted across the air divided by the number of unique packets created.
- Average latency: the average time it takes a packet to be delivered.
- Average hop count: the average number of hops a message requires to reach destination.

5.3.2 Results

In this section we present the results of the trace-driven simulations performed to evaluate the routing performance of our proposed opportunist routing scheme. First, we discuss the scenario in which message TTL varies, and then the scenario in which we simulate different inter-message creation intervals.

5.3.2.1 Different TTLs scenario

For this scenario we set the inter-message creation interval to 30 minutes. This means that nodes generate one message every 30 minutes. Delivery ratio and overhead cost for Lapland Dataset are showed respectively in Fig. 5.1 and Fig. 5.2. Epidemic routing, with its unlimited flooding strategy, outperforms all the other protocols with the highest delivery ratio. However, the cost is also very high and in this case an opportunistic protocol with a similar performance at lower cost would be the right choice in order to reduce energy consumption.

Bubble Rap and H-Bubble Rap perform almost as well both in terms of delivery ratio and overhead cost. The delivery ratio is the lowest and clearly, the overhead cost is low. We can see that the overhead cost of H-Bubble Rap is slightly lower because it uses the multi-layer social metric.

PRoPHET clearly outperforms ML-SOR in terms of delivery ratio, but when TTL increases the performance of ML-SOR improves, with a comparable performance for TTL of 2 days and a better performance for 4 days. In terms of overhead cost, however, PRoPHET costs more than ML-SOR, especially for TTLs higher than 12 hours. These results demonstrate that the use of multi-layer social network information in ML-SOR reduces significantly the overhead cost, with a delivery ratio comparable to Epidemic routing.

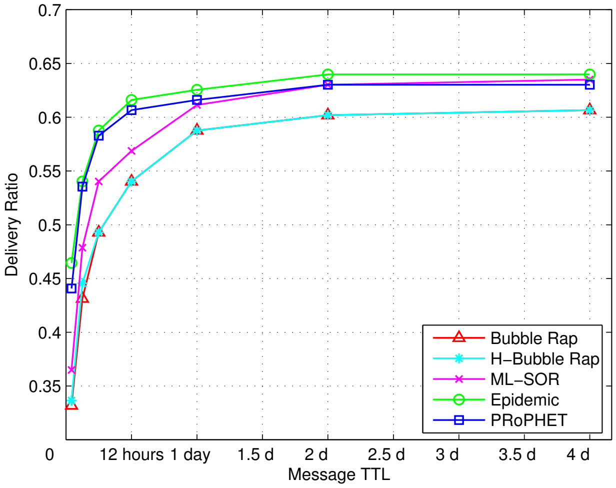


Fig. 5.1. Delivery ratio as a function of message TTL (Lapland dataset).

Fig. 5.3 shows the average latency for Lapland dataset. For low TTLs (1 hour and 3 hours) all protocols show a similar average latency. When TTL increases, Epidemic routing and PROPHET are able to deliver messages faster than the other protocols, as expected.

Bubble Rap shows the worst performance with the highest average latency for each TTL value. As we can see, the hybrid version of Bubble Rap, H-Bubble Rap, performs slightly better than classic Bubble Rap, while our ML-SOR outperforms these two social-based protocols.

We can see from Fig. 5.4 that Epidemic routing has the highest average hop count with a value that is around 2.8. PRoPHET shows a lower hop count, with an average value of 2.5, while social-based routing protocols as Bubble Rap, H-Bubble Rap and ML-SOR have lower values. These results confirm that the forwarding strategies of social-based schemes are able to reach the destination within less hops by exploiting the social behavior of nodes.

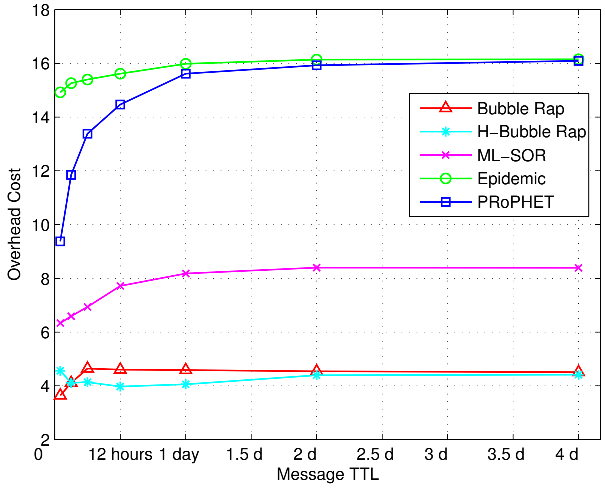


Fig. 5.2. Overhead cost as a function of message TTL (Lapland dataset).

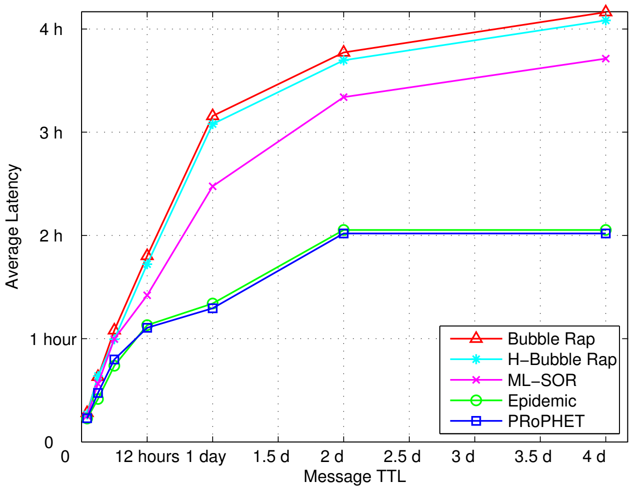


Fig. 5.3. Average latency as a function of message TTL (Lapland dataset).

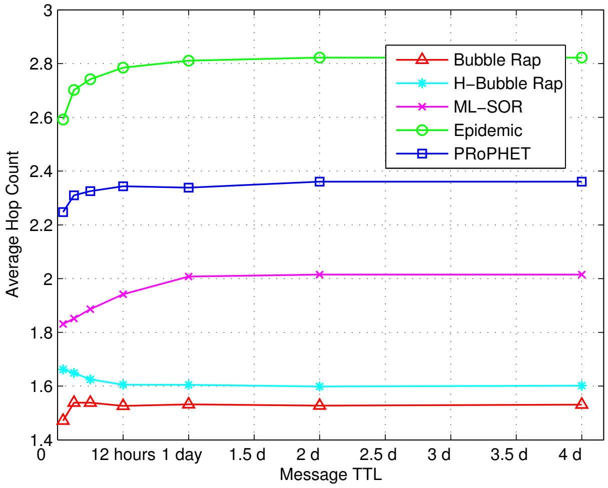


Fig. 5.4. Average hop count as a function of message TTL (Lapland dataset).

Now we evaluate the same routing strategies on Sigcomm data, which is a conference scenario as Lapland but with a higher number of nodes (76). Here the maximum TTL is set to 1 day because this dataset covers a lower number of hours than Lapland. In Fig. 5.5, we can see the delivery ratio for the different protocols. For this dataset, the overall delivery ratio is higher, with values that achieve more than 95% of message delivery. Since the number of nodes is higher, the possibilities of forwarding are higher too.

As expected, Epidemic routing is still characterized by the highest delivery ratio. Even if, for TTLs set to 12 hours and 1 day, PRoPHET performs as Epidemic routing. Differently from Lapland dataset, as TTL increases, the difference between the various routing schemes is not constant. For a TTL of 2 minutes, Epidemic routing achieves around 78% of the delivery ratio, H-Bubble Rap and ML-SOR 75%, Bubble Rap 67% and PRoPHET 60%. For a TTL of 10 minutes, all delivery ratio values grow except for Epidemic routing and ML-SOR which maintains the same previous values. As TTL increases, H-Bubble Rap and ML-SOR show the same performance, with a delivery ratio which is slightly lower than Epidemic routing and higher than Bubble Rap. This means that the ML-SOR social metric improves performance of Bubble Rap, both in the case of ML-SOR which does not consider communities to drive routing decisions and of H-Bubble Rap which is community-based. Moreover, for a TTL of 1 day, PRoPHET, H-BubbleRap and ML-SOR achieve 97% of delivery ratio as Epidemic routing.

In terms of cost, in Fig. 5.6 we can see that Epidemic routing costs much more than the other protocols. For low TTLs (2 minutes, 10 minutes, 1 hour), PRoPHETs has a cost lower than Epidemic, ML-SOR and H-Bubble

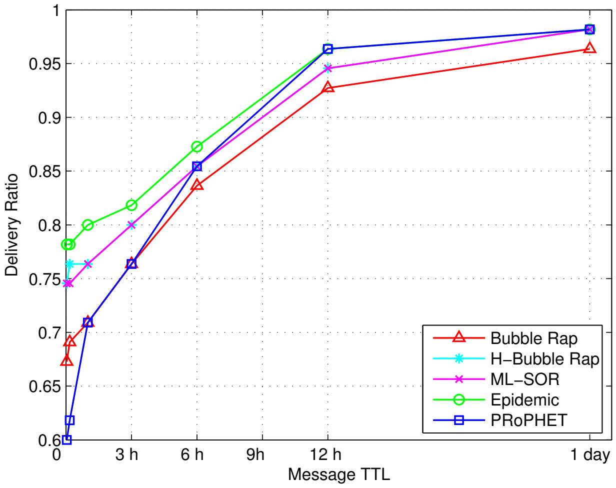


Fig. 5.5. Delivery ratio as a function of message TTL (Sigcomm dataset).

Rap. When TTL increases, PRoPHET costs more than the three social-based schemes. Clearly, Bubble Rap has the lower overhead cost, considering that has low delivery ratio. However, for TTLs set to 10 minutes or 1 hour, Bubble Rap performs as PRoPHET, but with a lower overhead cost.

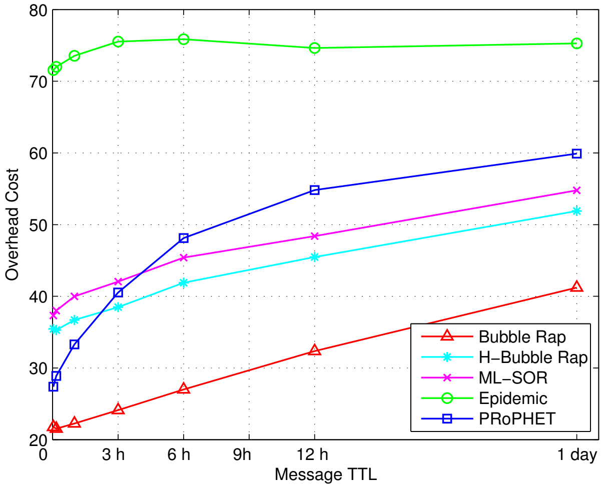


Fig. 5.6. Overhead cost as a function of message TTL (Sigcomm dataset).

As we can see from Fig. 5.7, Epidemic routing clearly performs the best, while PRoPHET is characterized by the worst performance. Social-based schemes perform better than PRoPHET. Here, the multi-layer social metric of ML-SOR and H-Bubble Rap works quite well producing a latency lower than classic Bubble Rap.

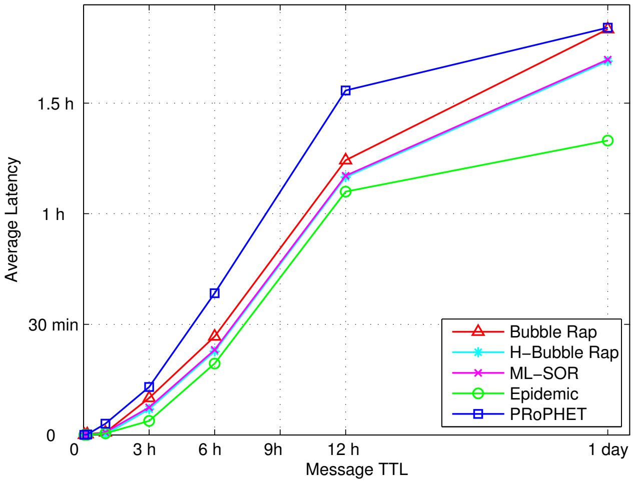


Fig. 5.7. Average latency as a function of message TTL (Sigcomm dataset).

In Fig. 5.8 the average hop count is showed. Differently from Lapland dataset, social-based strategies here shows average hop counts higher than Epidemic routing and PRoPHET. The interesting result is that ML-SOR is able to deliver the same number of messages of H-BubbleRap within less hops.

5.3.2.2 Different inter-message creation intervals scenario

Here we evaluate the performance of the protocols considering a scenario where TTL is fixed and the inter-message creation interval varies. For Lapland we choose 1 hour, while for Sigcomm dataset we set a TTL of 6 hours. Each message can be created every 1 hour, 3 hours, 6 hours, 12 hours, and 1 day. We start showing the results for Lapland dataset. In Fig. 5.9 and Fig. 5.10 are showed respectively the delivery ratio and the overhead cost. We can see that Epidemic routing and PRoPHET perform almost as well both in terms of delivery ratio, but if we consider overhead cost we can see that PRoPHET has a lower cost.

The two versions of Bubble Rap perform as well in terms of both delivery ratio and overhead cost (except for the case when inter-message creation inter-

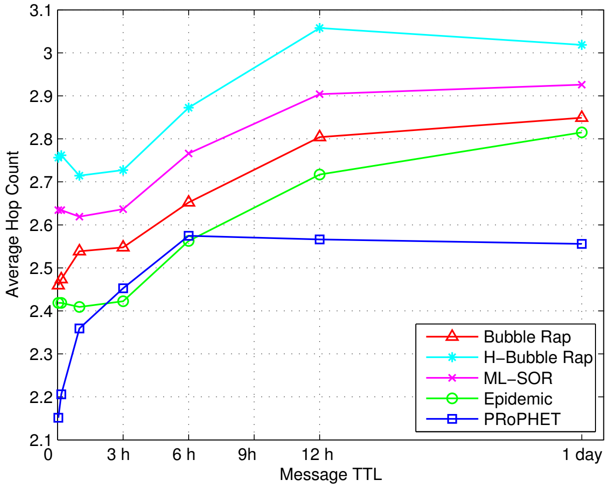


Fig. 5.8. Average hop count as a function of message TTL (Sigcomm dataset).

val is set to 1 day where H-Bubble Rap outperforms Bubble Rap). Moreover, if compared to the other protocols, H-Bubble Rap shows the lowest overhead cost.

ML-SOR has a delivery ratio higher than H-Bubble Rap for inter-message creation intervals set to 1 hour and 1 day.

Fig. 5.11 shows the results for the average latency. This routing metric is highly variable for Bubble Rap and H-Bubble Rap, which perform almost as well both in terms of delivery ratio and overhead cost. ML-SOR outperforms significantly the other two social based schemes when the interval is set to 6 hours and 12 hours. Moreover, it outperforms PRoPHET when the interval is set to 6 hours, 12 hours and 1 day.

The average hop count is showed Fig. 5.12. As we can see, the difference between the protocols can vary considerably for different inter-message creation intervals. As expected, Epidemic routing delivers messages using the highest number of hops, except for the case where nodes generate one message per day. In this case, Bubble Rap shows a higher average hop count.

In Fig. 5.13 and Fig. 5.10 are showed respectively the delivery ratio and the overhead cost. As expected, Epidemic routing shows the highest delivery ratio. However, the cost is also very high. Here ML-SOR shows quite good performance, with respect to H-Bubble Rap, having an overhead cost which can be considered similar.

As we can see from Fig. 5.15, PRoPHET shows the highest average latency, while the performance of the other protocols can be considered similar.

The average hop count is showed Fig. 5.16. As we can see, PRoPHET shows the lowest average hop count, while H-Bubble Rap and ML-SOR perform

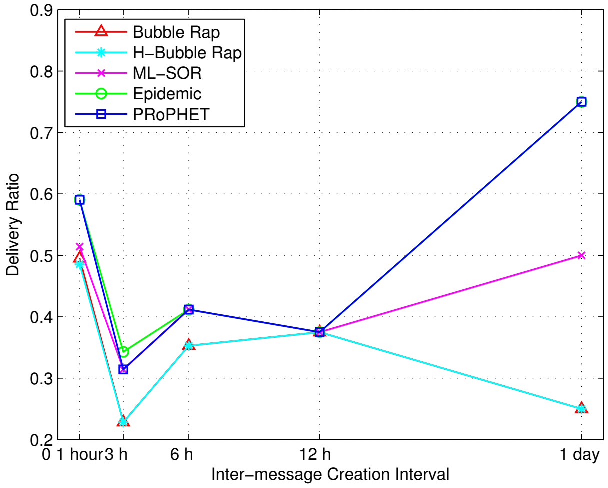


Fig. 5.9. Delivery ratio as a function of inter-message creation interval (Lapland dataset).

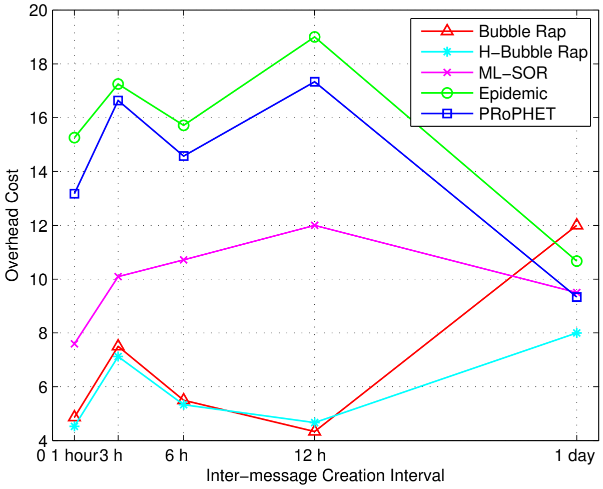


Fig. 5.10. Overhead cost as a function of inter-message creation interval (Lapland dataset).

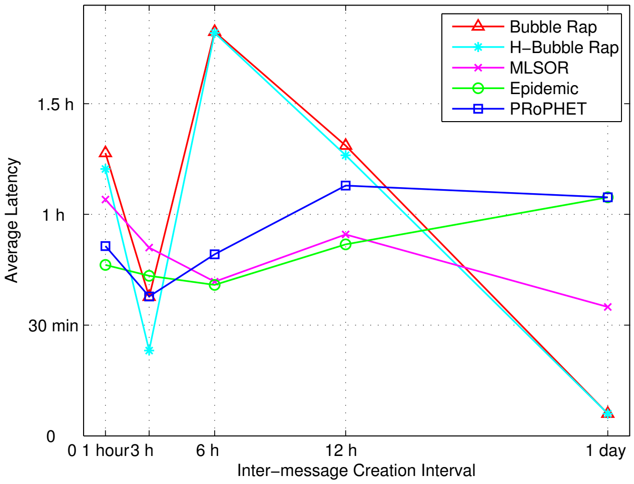


Fig. 5.11. Average latency as a function of inter-message creation interval (Lapland dataset).

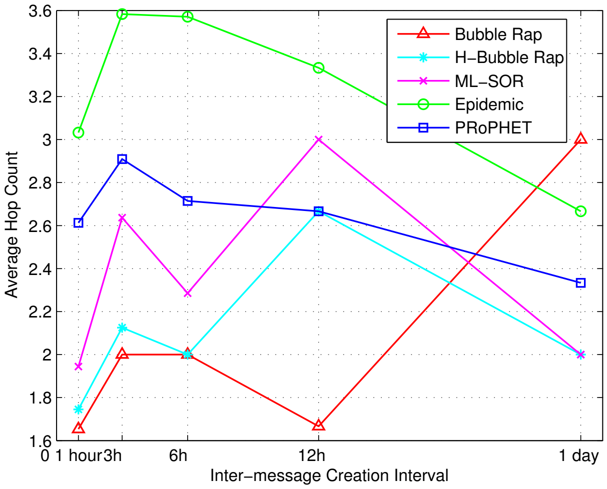


Fig. 5.12. Average hop count as a function of inter-message creation interval (Lapland dataset).

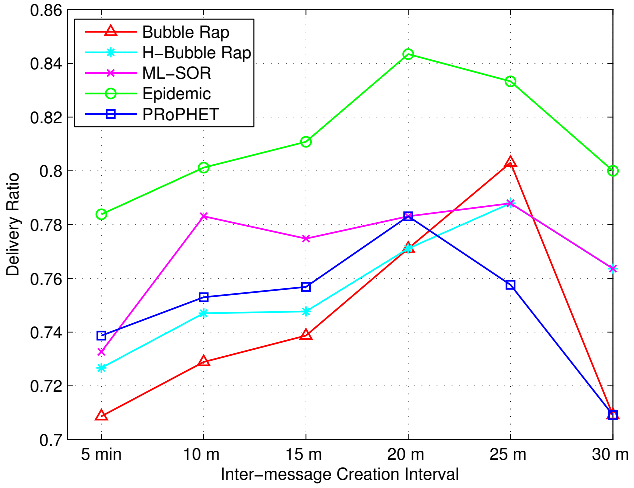


Fig. 5.13. Delivery ratio as a function of inter-message creation interval (Sigcomm dataset).

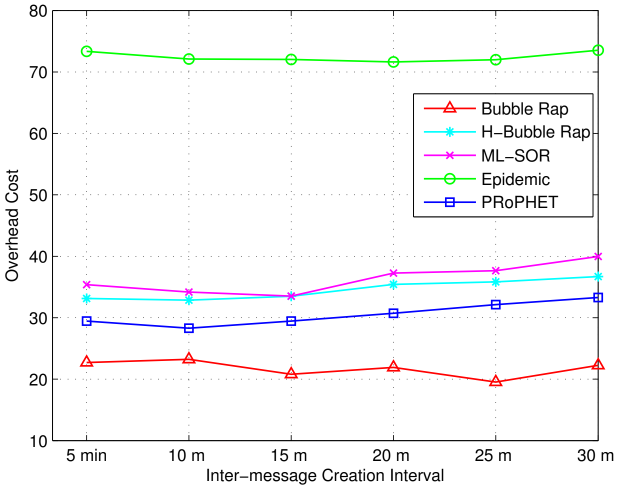


Fig. 5.14. Overhead cost as a function of inter-message creation interval (Sigcomm dataset).

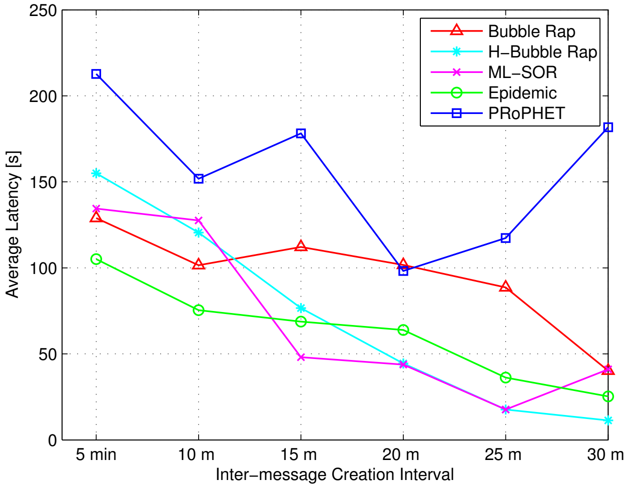


Fig. 5.15. Average latency as a function of inter-message creation interval (Sigcomm dataset).

similarly showing a higher average hop count than the the other protocols. Also the behavior of Bubble Rap and Epidemic can be considered similar.

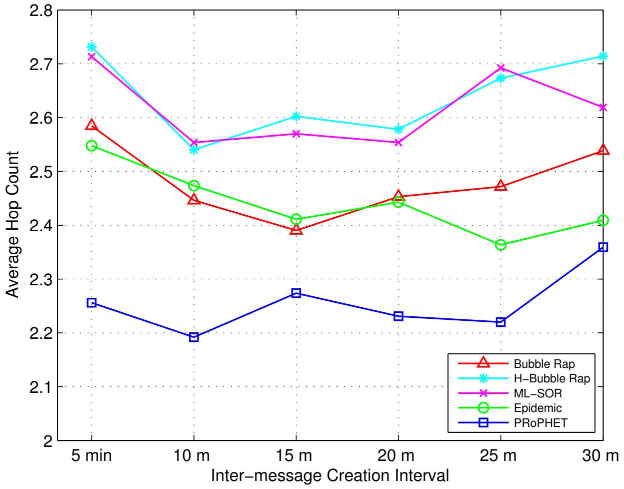


Fig. 5.16. Average hop count as a function of inter-message creation interval (Sigcomm dataset).

5.4 Discussion

In this chapter we have presented a novel opportunistic routing protocol, ML-SOR, that uses a multi-layer social network to select nodes to act as message relays. This allows the protocol to route efficiently messages within the network, exploiting not only the social information extracted from the detected social network layer but also social information extracted from several types of social network layers. We demonstrated that more stable social information provided by several social network layers is able to augment available partial contact information improving message forwarding.

We compared our protocol to Epidemic routing, PRoPHET, Bubble Rap and an hybrid version of Bubble Rap, called H-Bubble Rap, which adopts the same metric of ML-SOR and is community-based, and found that ML-SOR shows good performance in most scenarios. We have seen that maximizing delivery cost is not necessarily an indication that a routing protocol performs better than a protocol that does not. ML-SOR performs well despite having a lower delivery ratio than Epidemic routing, for example, as it has a lower overhead cost produced by its better forwarding strategy.

We evaluated the performance of the protocols using two real-world traces, each with different number of nodes and connectivity patterns. We found that there is not a protocol which consistently performs better than the other protocols across all traces. The question if there is a trace indicative an opportunistic network in general is yet an open question.

6

Conclusions

Opportunistic routing exploits the interactions between mobile devices to exchange data. Such interactions may arise because of social behaviour; the study of social networks can therefore be useful for routing in opportunistic networks.

In Chapter 2 we started describing the development of Delay Tolerant Networks and opportunistic networks and their usefulness for challenged communication environments. In particular, we motivated the importance of Delay Tolerant Architecture and opportunistic networks through example applications and the need for opportunistic network research instead of using existing solutions for Mobile Ad Hoc Networks.

We have seen that the main challenge for DTN and opportunistic routing is to decide which encounter nodes use for forwarding. Consequently, we analyzed a set of representative routing protocols for opportunistic networks and discussed their differences in terms of several routing performance metrics. From this analysis, we demonstrated that social-based forwarding performs well in opportunistic networks and that is very difficult to achieve both high delivery ratio and low delivery latency when energy consumption is taken into account.

In Chapter 3 we analyzed the structural properties of online and detected social networks, for a particular set of users. Specifically, we explored and compared the sociocentric and the egocentric behaviors of nodes, highlighting the structural similarities between the two types of networks and the differences in how individuals take part in co-presence network and Facebook network. Performing a sociocentric network analysis, we observed a relatively high correlation of betweenness centrality. On the contrary, the other centrality measures in the online social network and the detected social network vary considerably for the dataset considered. The relevant aspect of our analysis is the study of the contribution of central nodes within the online and the detected social networks. We feel that applications such as friend recommendation or routing schemes for opportunistic networks can benefit from this study providing a more complete understanding of user sociocentric and egocentric behaviors in real and virtual social networks.

In Chapter 4, we introduced the concept of multi-layer social network and analyzed the structural properties of this complex network. Specifically, we considered a particular dataset, namely Lapland, and defined four different social network layers. Using a Joint Diagonalisation technique, we extracted from the dynamic contact network two modes of operation, namely Mode 1 network and Mode 2 network, which are two static graphs representing the most common propagation paths in the detected social network. Adding to Mode 1 network and Mode 2 network, Facebook network and Interest network layers, we defined a multi-layer model for Lapland dataset. Then, we explored the network motifs occurring at each layer, discovering that Mode 2 network, Facebook network and Interest network are similar.

Analyzing node centrality in the multi-layer network we found again that nodes behave similarly in Mode 2 network, Facebook network and Interest network showing similar centrality values at each network layer. Finally, we measured the similarity between communities and we found the highest Jaccard index for Mode 2 network and Interest network.

The relevant aspect of the analysis of this multi-layer social network is that we found similarities between different levels of online and detected social networks which could be exploited for opportunistic routing.

In Chapter 5 we presented a novel opportunistic routing protocol, ML-SOR, that uses a multi-layer social network to select nodes to act as message relays. This allows the protocol to route efficiently messages within the network, exploiting not only the social information extracted from the detected social network layer but also social information extracted from several types of social network layers. We demonstrated that more stable social information provided by several social network layers is able to augment available partial contact information improving message forwarding.

We compared our protocol to Epidemic routing, PRoPHET, Bubble Rap and an hybrid version of Bubble Rap, called H-Bubble Rap, which adopts the same metric of ML-SOR and is community-based, and found that ML-SOR shows good performance in most scenarios. We have seen that maximizing delivery cost is not necessarily an indication that a routing protocol performs better than a protocol that does not. ML-SOR performs well despite having a lower delivery ratio than Epidemic routing, for example, as it has a lower overhead cost produced by its better forwarding strategy.

We finally evaluated the performance of the protocols using two real-world traces, each with different number of nodes and connectivity patterns. We found that there is not a protocol which consistently performs better than the other protocols across all traces. The question if there is a trace indicative an opportunistic network in general is yet an open question.

Our plans for future work include the analysis of other datasets with different connectivity pattern and social network layers such as Twitter or LinkedIn. We could also refine our multi-layer social network metric in order to exploit different types of tie strength indicators. Finally, we also wish to change the weights of the multi-layer social network metric. Currently we equally weight centrality, tie strength indicator and tie predictor. We think that it will be interesting to assign different weights to each of the three metrics used to compose the multi-layer social metric.

References

1. Xueli An, Jing Wang, R. Venkatesha Prasad, and I. G. M. M. Niemegeers. OPT: online person tracking system for context-awareness in wireless personal network. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, REALMAN '06, pages 47–54, New York, NY, USA, 2006. ACM.
2. Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, pages 286(5439):509–512, October 1999.
3. Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.
4. G. Bigwood and T. Henderson. Bootstrapping opportunistic networks using social roles. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, pages 1 –6, june 2011.
5. G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. Exploiting self-reported social networks for routing in ubiquitous computing environments. In *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, pages 484 –489, oct. 2008.
6. Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti. CRAWDAD trace set st_andrews/sassy/mobile (v. 2011-06-03). http://crawdad.cs.dartmouth.edu/st_andrews/sassy/mobile, june 2011.
7. Chiara Boldrini, Marco Conti, and Andrea Passarella. Social-based autonomic routing in opportunistic networks.
8. P. BONACICH. Factoring and weighting approaches to status scores and clique identification, 1972.
9. Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.
10. SP Borgatti, MG Everett, and LC Freeman. Ucinet for windows: Software for social network analysis. 2002.
11. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.
12. J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –11, april 2006.

13. Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, and Ronald A. Peterson. People-centric urban sensing. In *Proceedings of the 2nd annual international workshop on Wireless internet*, WICON '06, New York, NY, USA, 2006. ACM.
14. V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, R. Scott, and K. Fall. Delay-tolerant networking architecture. http://tools.ietf.org/html/rfc4838, April 2007.
15. Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *Mobile Computing, IEEE Transactions on*, 6(6):606 –620, june 2007.
16. T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). http://tools.ietf.org/html/rfc3626, October 2003.
17. M. Conti, S. Giordano, M. May, and A. Passarella. From opportunistic networks to opportunistic computing. *Communications Magazine, IEEE*, 48(9):126 –139, sept. 2010.
18. Dana Cuff, Mark Hansen, and Jerry Kang. Urban sensing: out of the woods. *Commun. ACM*, 51(3):24–33, 2008.
19. E.M. Daly and M. Haahr. Social network analysis for information flow in disconnected delay-tolerant manets. *Mobile Computing, IEEE Transactions on*, 8(5):606 –621, may 2009.
20. Michael Demmer and Kevin Fall. Dtlsr: delay tolerant routing for developing regions. In *Proceedings of the 2007 workshop on Networked systems for developing regions*, NSDR '07, pages 5:1–5:6, New York, NY, USA, 2007. ACM.
21. Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Gahng S. Ahn, and Andrew T. Campbell. MetroSense project: people-centric sensing at scale. World-Sensor-Web at SenSys, October 2006.
22. Frans Ekman, Ari Keränen, Jouni Karvo, and Jörg Ott. Working day movement model. In *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*, MobilityModels '08, pages 33–40, New York, NY, USA, 2008. ACM.
23. P. ERDOS. On the evolution of random graphs, 1960.
24. Martin Everett and Stephen P. Borgatti. Ego network betweenness. *Social Networks*, 27(1):31 – 38, 2005.
25. Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.
26. Damien Fay, Jérôme Kunegis, and Eiko Yoneki. On joint diagonalisation for dynamic network analysis. 10 2011.
27. M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. J.*, 25(100):619–633, 1975.
28. L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35-41, 1977.
29. Saurabh Ganeriwal, Laura K. Balzano, and Mani B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, 4(3):15:1–15:37, June 2008.
30. R. V. Gould and Fernandez R.M. Structures of mediation: a formal approach to brokerage in transaction networks. *Sociological Methodology (19)*, page 91, 1989.

31. Matthias Grossglauser and Martin Vetterli. Locating mobile nodes with EASE: learning efficient routes from encounter histories alone. *IEEE/ACM Trans. Netw.*, 14(3):457–469, June 2006.
32. H. Hagras. Embedding computational intelligence in pervasive spaces. *Pervasive Computing, IEEE*, 6(3):85 –89, july-sept. 2007.
33. Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.
34. Pan Hui and Jon Crowcroft. Bubble Rap: Forwarding in small world DTNs in ever decreasing circles. Technical Report UCAM-CL-TR-684, University of Cambridge, Computer Laboratory, May 2007.
35. Pan Hui and Jon Crowcroft. How small labels create big improvements. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 65 –70, march 2007.
36. Pan Hui, Eiko Yoneki, Shu Yan Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, MobiArch '07, pages 7:1–7:8, New York, NY, USA, 2007. ACM.
37. Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.
38. David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks, 1996.
39. Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *SIGARCH Comput. Archit. News*, 30(5):96–107, October 2002.
40. Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Simutools '09, pages 55:1–55:10, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
41. YoungBae Ko and Nitin H. Vaidya. Locationaided routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.
42. Vassilis Kostakos and Jayant Venkatanathan. Making friends in life and online: Equivalence, micro-correlation and value in spatial and transpatial social networks. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 587–594, Washington, DC, USA, 2010. IEEE Computer Society.
43. Sung-Ju Lee, William Su, and Mario Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *Mobile Networks and Applications*, 7(6):441–453, 2002.
44. Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities, 2006.
45. N. Lin. *Foundations of Social Research*. New York, 1976.
46. Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks, 2004.

47. Peter V Marsden. Egocentric and sociocentric measures of network centrality. *Social Networks*, 24(4):407 – 422, 2002.
48. A.J. Mashhadi, S. Ben Mokhtar, and L. Capra. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, pages 1 –6, june 2009.
49. T. Matsuda and T. Takine. (p,q)-Epidemic routing for sparsely populated mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):783 –793, june 2008.
50. Liam McNamara, Cecilia Mascolo, and Licia Capra. Media sharing based on colocation prediction in urban transport. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, MobiCom '08, pages 58–69, New York, NY, USA, 2008. ACM.
51. R. Milo, S. Shen-Orr, S., S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Quat. Res*, 52:143, 1999.
52. Mehul Motani, Vikram Srinivasan, and Pavan S. Nuggehalli. PeopleNet: engineering a wireless virtual social network. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, pages 243–257, New York, NY, USA, 2005. ACM.
53. A. Mtibaa, M. May, C. Diot, and M. Ammar. Peoplerank: Social opportunistic forwarding. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –5, march 2010.
54. Abderrahmen Mtibaa, Augustin Chaintreau, Jason LeBrun, Earl Oliver, Anna-Kaisa Pietilainen, and Christophe Diot. Are you moved by your social network application? In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 67–72, New York, NY, USA, 2008. ACM.
55. Derek G. Murray, Eiko Yoneki, Jon Crowcroft, and Steven Hand. The case for crowd computing. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 39–44, New York, NY, USA, 2010. ACM.
56. MEJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.
57. S.J. Pan, D.J. Boston, and C. Borcea. Analysis of fusing online and co-presence social networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 496 –501, march 2011.
58. Charles Perkins and Ian Chakeres. Dynamic MANET on-demand (DYMO) routing. Technical report draft-ietf-manet-dymo-21.txt, IETF Secretariat, Fremont, CA, USA, July 2010.
59. Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244, New York, NY, USA, 1994. ACM.
60. Charles E. Perkins and Elizabeth M. Royer. Ad hoc networking. chapter The Ad Hoc on-demand distance-vector protocol, pages 173–219. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
61. Anna-Kaisa Pietilainen and Christophe Diot. CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15). Downloaded from http://crawdad.cs.dartmouth.edu/thlab/sigcomm2009, July 2012.

62. Anna-Kaisa Pietiläinen, Earl Oliver, Jason LeBrun, George Varghese, and Christophe Diot. Mobiclique: middleware for mobile social networking. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 49–54, New York, NY, USA, 2009. ACM.
63. Calicrates Policroniades, Pablo Vidales, Martin Roth, and Daniel Kreienbühl. Data management in human networks. In *Proceedings of the second ACM workshop on Challenged networks*, CHANTS '07, pages 83–90, New York, NY, USA, 2007. ACM.
64. Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.
65. A. Socievole and F. De Rango. Evaluation of routing schemes in opportunistic networks considering energy consumption. In *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2012 International Symposium on*, pages 1 –7, july 2012.
66. A. Socievole, F. De Rango, and C. Coscarella. Performance evaluation of distributed routing protocols over DTN stack for MANETs. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2011*, june 2011.
67. A. Socievole, F. De Rango, and C. Coscarella. Routing approaches and performance evaluation in delay tolerant networks. In *Wireless Telecommunications Symposium (WTS), 2011*, pages 1 –6, april 2011.
68. A. Socievole and S. Marano. Exploring user sociocentric and egocentric behaviors in online and detected social networks. In *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on*, pages 140 –147, april 2012.
69. Annalisa Socievole and Salvatore Marano. Evaluating the impact of energy consumption on routing performance in delay tolerant networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 481 –486, aug. 2012.
70. T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 235 – 244, oct. 2004.
71. Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 252–259, New York, NY, USA, 2005. ACM.
72. Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, pages 79 –85, march 2007.
73. Jing Su, James Scott, Pan Hui, Jon Crowcroft, Eyal de Lara, Christophe Diot, Ashvin Goel, Meng How Lim, and Eben Upton. Haggle: Seamless networking for mobile applications, 2007.
74. Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 214–226, New York, NY, USA, 2004. ACM.

75. Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):pp. 425–443, 1969.
76. Amin Vahdat and David Becker. Epidemic routing for Partially-Connected ad hoc networks. Technical report, Duke University, April 2000.
77. Y. Wang and H. Wu. DFT-MSN: The delay/fault-tolerant mobile sensor network for pervasive information gathering. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –12, april 2006.
78. Yu Wang and Hongyi Wu. Delay/fault-tolerant mobile sensor network (DFT-MSN): A new paradigm for pervasive information gathering. *Mobile Computing, IEEE Transactions on*, 6(9):1021 –1034, sept. 2007.
79. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Cambridge Univ. Press, 1994.
80. Douglas R. White and Karl P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193 – 234, 1983.
81. Kuang Xu, V.O.K. Li, and Jaewoo Chung. Exploring centrality for message forwarding in opportunistic networks. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1 –6, april 2010.
82. Ting Yan, Tian He, and John A. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 51–62, New York, NY, USA, 2003. ACM.
83. Eiko Yoneki and Fehmi Ben Abdesslem. Finding a data blackhole in bluetooth scanning. ExtremeCom, 2009.
84. Xiaolan Zhang, Giovanni Neglia, Jim Kurose, and Don Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867 – 2891, 2007.
85. Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys Tutorials, IEEE*, 8(1):24 –37, quarter 2006.
86. Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. Multicasting in delay tolerant networks: semantic models and routing algorithms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 268–275, New York, NY, USA, 2005. ACM.