The Hong Kong Polytechnic University

Department of Computing

Mobile Social Networks:
System and Applications

Chu Hau Shing

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

Initial Submission: 16 Jan 2013

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_________________ (Signed)

CHU HAU SHING (Name of student)

Abstract

Mobile social networks (MSNs) enable the interactions between people through their mobile devices within a social networking environment. More and more MSN applications, services and systems are being proposed and developed, changing the way people communicate with each other and enabling ubiquitous collaboration among people. In this thesis, we make three contributions to the development of future MSNs. First, we present a social network framework based on an innovative concept called social vectors. Basically, a social vector can be used to define and quantify the social relationship between two social entities. In general, social vectors can also be used to facilitate the development of MSN system and applications. Second, based on social vectors and inspired by previous work, we present an opportunistic routing protocol called Hybrid Opportunistic Routing for Social Entities (HORSE). HORSE seeks to combine the advantages of the previous opportunistic routing protocols in order to forward social messages effectively and efficiently over MSNs. In particular, it is inspired by the "six degrees of separation" theory, which provides the basis for forwarding social messages through a chain of friends. Simulation results using real-world data traces are presented to evaluate the performance of HORSE as compared with other schemes. The results indicate that HORSE can achieve high delivery ratios while maintaining low overhead costs. Third, based on social vectors, we present two related applications. The first one is a Smart Shopping System, which identifies customers using RFID technology and finds similar customers using social vectors with the purpose of providing customized marketing messages for customers. The second one is a Dynamic Signage System with a social vector-based dynamic playlist mechanism, which enables the display of dynamic contents that matches with people interests. Simulation results are presented to show the effectiveness of this mechanism.

Acknowledgements

This thesis would not have been completed without the valuable advice and guidance from my supervisor, Dr. Henry Chan. I am therefore eager to give sincere gratitude to him.

I would also like to thank my co-supervisor, Dr. Eiko Yoneki of the University of Cambridge, for her invaluable guidance and constructive comments.

Besides, I would like to thank Xiping Hu and Prof. Victor Leung of The University of British Columbia and Prof. Jon Crowcroft of the University of Cambridge for research collaboration on mobile social networks.

Finally, I greatly thank Felix Hui for his help on developing our Smart Shopping System.

Table of Contents

List of Figures

List of Tables

1 Introduction

In recent years, there has been considerable research interest in social networks and social computing, such as establishing a framework for community detection [1], developing a social reputation model [2], designing recommendation algorithms [3], and protecting privacy for data sharing applications [4]. Currently, social networks are mostly web-based, allowing people to access the information of friends and friends of friends from the Internet through their computers and mobile terminals. This many-to-many connectivity enhances social relationships and enables a wide range of social applications to be used over the Internet. Recent technological advances have given smart phones the capability to generate, store, and share contents. Location data is also available by using a GPS receiver embedded in a mobile device. Such contents may be of interest to specific groups of people in a specific time or geographic area. With the advance and popularity of smartphones, mobile social networks (MSNs) can be formed to complement existing web-based social networks. For example, Facebook, Twitter, and Foursquare are very popular and widely used in both social networks and MSNs. Apart from using smartphones to get social content and services from social networks through the traditional communication network (i.e., Internet), opportunistic networks can also be formed with smartphones, allowing social messages to be forwarded in an ad-hoc manner using opportunistic channels between people.

Unlike traditional social networks that are centered on individual persons, future MSNs can take advantage of the additional capabilities of contemporary mobile devices such as smart phones. These capabilities, such as global position system (GPS) receiver, sensing modules (cameras, accelerometer, and gravity sensors, etc.), and multiple radios (second/third/fourth generation cellular, WiFi, Bluetooth, WiFi Direct, etc.), enable MSNs to enhance conventional social networks with additional features, such as location-awareness. Different from conventional social networks in which people interact over the Internet, the multiple radios in mobile devices enable future MSNs to also work over opportunistic networks, where each node can act as a host, a router, or a gateway, and connect with other nodes in an ad-hoc manner, without possessing or acquiring any knowledge about the network topology.

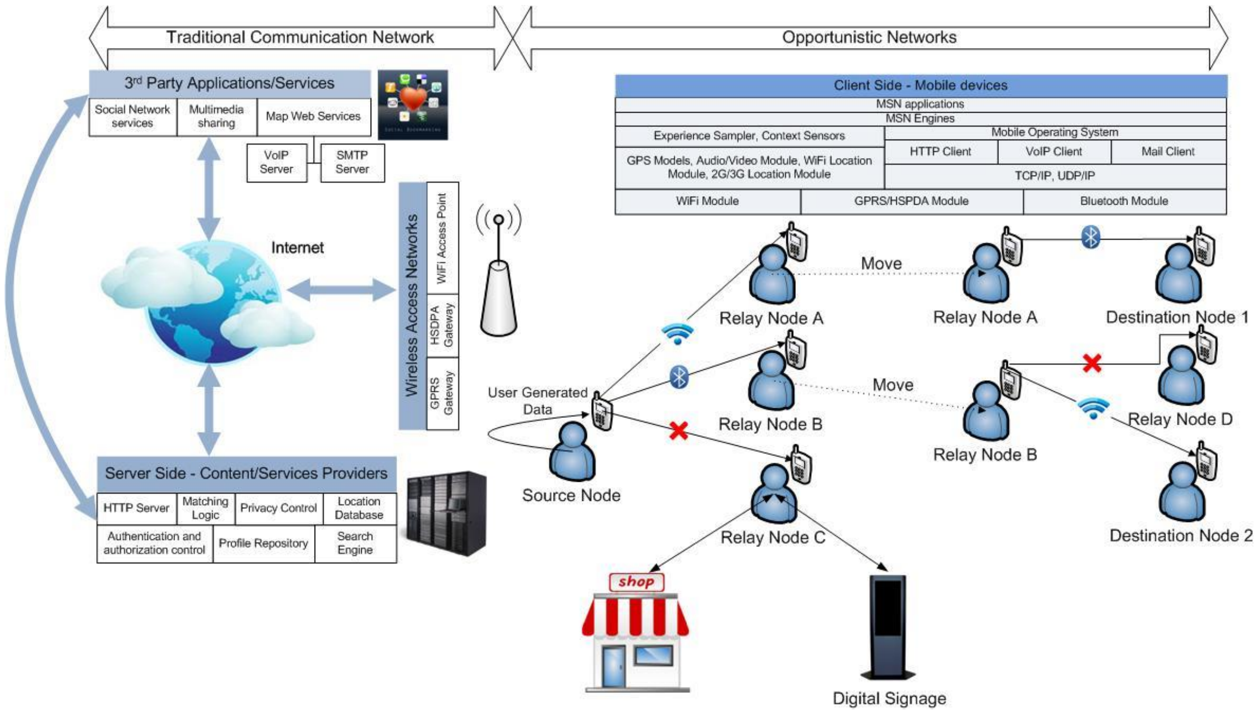


Fig. 1 The architecture of future MSNs

Future MSNs will likely be based on an integration of the traditional communication network and opportunistic network architectures, as shown in Fig. 1. Most widely used social network and MSN applications such as Facebook, Twitter and Foursquare rely on traditional communication network architecture solely. It is based on a client and server model and consists of four main components: server side (the content and service provider), the third party service provider, the client side (users' mobile devices and its applications) and the wireless access networks. The server side has several components, such as the Hypertext Transport Protocol server, databases, application logic, authentication and authorization control, and so on. It is responsible for delivering/updating contents and services to/from the client side via the wireless access networks. Some new MSN services are enabled by third-party service providers via their Application Program Interface (API), such as the multimedia sharing services [5] and map web services. The client side is distributed over different mobile devices such as smart phones. The MSN applications in client side can be classified into two types. One is purely web-based and the other one is a hybrid type in which MSN services are encapsulated as a mobile application running on mobile operating systems such as iOS and Android [6]. The MSN application in client side are responsible for communicating with the server side so that users can get interested social content and services via the wireless access networks. The main advantages of using the server side to provide most MSN services are that it can simplify service implementation, reduce the hardware requirements on client side, and provide centralized control and coordination efficiently. The disadvantages are in common to those in client-server architectures. For example, the servers need extremely high reliability in order to provide reliable MSN services to users. In some cases such as disaster situations, MSN services may even be unavailable due to the disruption of the wireless access networks. Besides, the traditional communication network cannot take advantage of the benefits arising from opportunistic contacts since it assumes a connected path from a source to a destination with a low propagation delay and packet loss rate. In contrast, opportunistic networks do not assume that a connected path exists between a source node and a destination node. Instead, opportunistic interactions between mobile devices take place when data is sent from a source node to one or more destination nodes via either direct communications between devices using Wi-Fi Direct or Bluetooth, or other mobile devices as relay nodes using a store, carry and forward approach. Since the exchange of data between nodes consumes resources such as energy and storage, only encountered nodes that have higher chances of delivering the data to the destination node(s) are selected as relay nodes. The estimation is based on contact history, mobility pattern, common interests, or social relationships between users. With the integration of the traditional communication and opportunistic network architectures, many new features and services can be supported in future MSNs. One example that make uses of both the traditional communication and opportunistic network architectures is that news, weather forecasts, traffic alerts, and social media are retrieved from the traditional communication network by any node initially and then shared to all others over opportunistic networks. Note that even the traditional communication network is unavailable due to some special cases such as disaster situations we mentioned before, the future MSN services can still be maintained by using the opportunistic networks. In addition, the interaction between users have to rely on traditional communication networks in conventional MSNs (e.g., when user A wants to send a message to user B. The message need to be passed to the server side in the traditional communication network first and then retrieved by the user B from the traditional communication network). In future MSNs, users can interact with each other directly without relying on the traditional communication network. For example, user-generated data such as messages, photos, and micro-blogs can be collected and shared over opportunistic networks solely. Moreover, the interaction between users and outside environment is also enabled by future MSNs. For example, shops or digital signage can provide customized marketing messages to users over the opportunistic networks.

As mentioned above, the architecture of future MSN is based on the integration of the traditional communication network and opportunistic network. Fig. 2 compares opportunistic networks with traditional communication networks. In general, a network has two components: network nodes or simply nodes and hosts. Fixed networks like the Internet have both fixed nodes and hosts. Mobile networks like cellular networks have mobile hosts but fixed nodes. Opportunistic networks have fixed hosts and mobile nodes and even both mobile nodes and mobile hosts. Due to their dynamic and volatile nature, opportunistic networks operate under a completely new networking paradigm such that traditional routing protocols cannot be applied to them. In fact, they introduce new technical challenges and problems. This thesis seeks to contribute to this interesting area for supporting MSNs. Although traditional networking protocols cannot be applied in opportunistic networks, opportunistic social networks do have similarities with traditional networks, as shown in Table 1. (Note that we define opportunistic social networks as a subset of opportunistic networks, in which the estimation of whether nodes store/forward a data packet is based on the use of human social relationships.) In traditional communication networks, intra-domain and inter-domain routing protocols are required for routing packets within the same domain and across different domains, respectively. Similarly, in opportunistic social networks, intra-community and inter-community routing protocols are required to forward messages within the same community and across different communities. Furthermore, similar to traditional unicast and multicast services, data forwarding and data dissemination methods are required for forwarding data from one host to another host and from one host to a selective group of hosts, respectively. Here our focus is on data forwarding. Ongoing work is also being conducted on the dissemination of data.

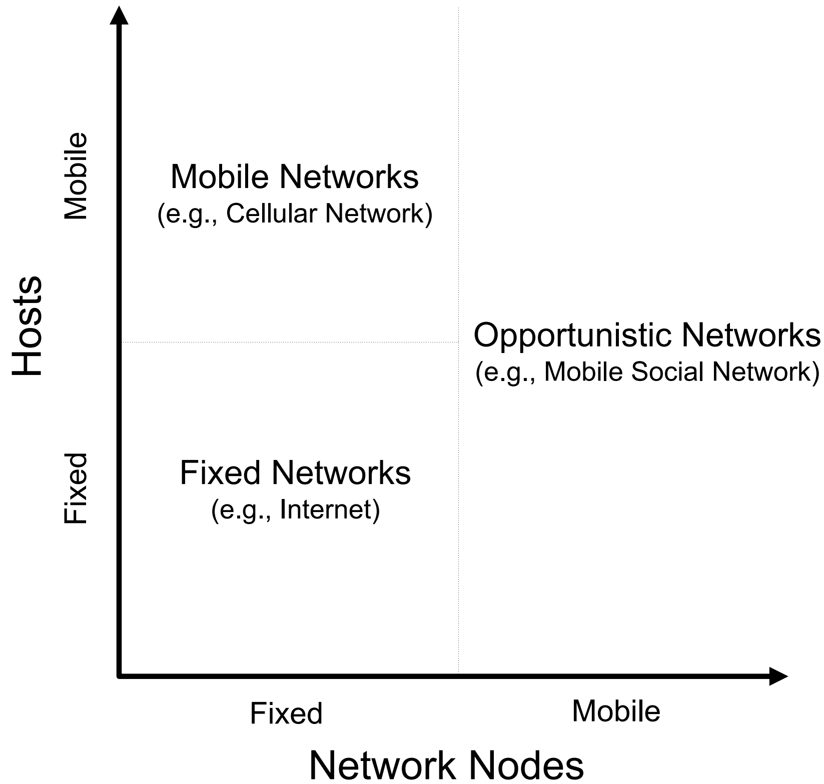


Fig. 2 Traditional communication networks and opportunistic networks

Table 1 Comparison between a traditional communication network and an opportunistic social network

Traditional communication network	**Opportunistic social network**
Intra-domain routing	Intra-community routing
Inter-domain routing	Inter-community routing
Unicast	Data forwarding
Multicast	Data dissemination

There are some fundamental questions in the research and development of future MSNs:

(1) How can we define and quantify the social relationships between people?

Going back to basics, we need a systematic and mathematical way to define social relationships. For example, when we say that X is a good friend of Y or Z is a better friend of X than Y, what do these statements mean? How can we quantity these relationships? Referring to Fig. 1, how can we determine the social relationships (i.e., friend, stranger, etc.) between mobile nodes. Note that these quantified social relationships are essential for the design and implementation of social network protocols and applications. To address the question, we define a social network framework based on a new concept called social vectors. Our proposed social vectors provide an effective and flexible framework to define social relationships in a systematic and quantitative manner. They even have the potential to open a new research framework (e.g., the development of social vector algebra or calculus).

(2) How can a message be forwarded effectively from a source node to a destination node over an opportunistic network?

Referring to Fig. 1, assume the source node want to send a message to the destination node 1. Which encountered nodes (i.e. relay nodes A, B and C) the source node should select to carry the message? As mentioned before, the exchange of data between nodes consumes resources such as energy and storage. Therefore, only encountered nodes that have higher chances of delivering the data to the destination node(s) are selected as relay nodes in data forwarding. Inspired by the previous work and based on social vectors, we propose an opportunistic routing protocol called Hybrid Opportunistic Routing for Social Entities (HORSE) for forwarding messages over opportunistic social networks. HORSE seeks to combine the advantages of previous opportunistic routing protocols. It is also inspired by the "six degrees of separation" theory, which provides the basis for forwarding social messages through a chain of friends (or a friend tree in our case). In particular, friends can facilitate intra-community data forwarding. Furthermore, strangers and highly sociable people can facilitate inter-domain data forwarding. Simulation results based on real-world data traces illustrate the advantages of HORSE over other opportunistic routing protocols.

(3) What MSN applications can be developed based on social vectors?

Besides using social vectors to investigate HORSE protocol, we present two social vector-based application systems for MSNs. The first one is a Smart Shopping System, which employs social vectors to identify similar customers. Products can then be recommended to customers based on purchase records. On the other hand, by using RFID technology, the location of customers can be identified effectively. Therefore, customized marketing messages (i.e., advertisements and e-coupons for the recommended products) can be delivered to customers' smart phones as soon as customers enter a shop or are near a shop.

The second one is a Dynamic Signage System, which employs a social vector-based dynamic playlist mechanism to generate dynamic contents. Traditional digital signage systems generally display static content without taking into consideration the interests of the consumers who are standing around the display. In future MSNs, next-generation digital signage systems will be more dynamic and intelligent by displaying content based on the interests. The social vector-based dynamic playlist mechanism enables digital signage system to show dynamic contents that highly matches people interests based on the information collected from the mobile phones of the users standing around the signage. Simulation results will be presented to show the effectiveness of this social vector-based mechanism.

The rest of this thesis is outlined as follows. Chapter 2 provides a literature review on related work. Chapter 3 presents the social network framework based mainly on social vectors. Chapter 4 presents the HORSE protocol. Chapter 5 presents the Smart Shopping System. Chapter 6 presents the Dynamic Signage System. Chapter 7 gives the conclusions and outlines future work.

2 Literature Review

2.1 Social Relationships in Social Network

There has been considerable research interest in social networks and social computing in recent years. For example, a framework is established for identifying online communities by using sentiment analysis and community detection in [1]. Its aim is to help companies design various marketing strategies or help governments understand people's political opinions better. In [2], a social reputation model is developed for assisting user to browse interested content. It first find the statistical correlation among users to determine various user interests and then make use of the inherent friend relationships to make reputation estimation and reliable social enhancements. In [3], a recommendation algorithms is designed by inferring category-specific social trust circles from social network data and rating data. Experiment results show that the recommendation accuracy can be increased by utilizing user's social trust information. On the other hand, research on MSNs is also flourishing. For example, influential people is identified through fixed-length random walks by using a lightweight and distributed protocol in [7]. Simulation results show that it can achieve a comparable performance with low overhead. In [8], human mobility trace such as inter-contact time and contact duration is generated by using a waypoint model. Simulation results show that this model is able to generate human mobility trace that is similar to the publicly available real traces. In [9], social network-based video sharing over a mobile social community is presented. It supports the exploration, sharing, and creation of video contents through MSNs. One of the important issues in social networks/MSNs is that of dealing with the social relationships between people. For example, how can we define and quantify social relationships between people or how can we classify people into different groups?

A social network is formed by a set of individuals and relationships among them [10]. It is usually represented as a graph by setting individuals as nodes and the relationships between them as edges. Therefore, many researchers have investigated social networks based on the techniques used in graph theory. In [11], the strength of social relationships between people is measured by a formula of user determined weights multiplied by the number of emails exchanged. In [12], an empirical analysis of social-network is performed, showing that the interactions between people can be inferred from the time stamp of emails. In [13], tightly connected communities are found by using hierarchical agglomerative clustering. In [14], the subgroup structure of a social network is identified based on an optimized modularity function. In [15], the structure of mobile phone networks and its persistent links are studied. It shows that people with high clustering and low degree are usually connected by the persistent links. In [16], a triangle approach is used to find communities of mobile users based on call detail records. In [17], social relationships are discovered based on the clique percolation method. In [18], the structure and tie strengths in mobile communication networks are studied. In [19], a spectral clustering method is employed for partitioning telephone call graph. In [20], a ghost edges approach which adds virtual edges to connect labeled and unlabeled nodes has been introduced. In [21], a WR-KMeans method that clusters instant messages on an extended vector space has been proposed. In [22], a nonparametric Bayesian approach is employed for discovering unsupervised group and predicting link in relational datasets.

The aforementioned approaches are mainly based on data retrieved from emails, World Wide Web and telephone calls in a centralized environment. Some approaches that deal with social relationships are based on proximity and location information of people in social networks. In [23], multiple regression quadratic assignment approach is employed to investigate social network data for finding behavioral characteristics based on proximity and mobile phone data. In [24], the behavior structure is identified by analyzing, predicting, and clustering multimodal data within the social network (from individuals and communities) using location data. In [25], a Gaussian model is employed to find patterns in proximity between users and determine the type of social relationship. In [26], data-mining techniques are applied on mining copresence events captured by Bluetooth devices to identify communities between individuals.

Besides, in [27], the social relationship between two people is simply classified to strangers, familiar strangers, community and friends based on contact duration and number of contacts. Moreover, three distributed community detection approaches, namely SIMPLE, k-CLIQUE, and MODULARITY, are proposed in [27]. They are able to detect static and dynamic communities based on different computation and resource requirements in a distributed environment. Another distributed community detection algorithm using neighborhood similarity metric is proposed in [28]. Note that some data routing and dissemination protocols such as [29][30] rely on distributed community detection to make data forwarding decision when nodes meet.

2.2 Data Routing and Dissemination

In addition to wide-area wireless networking interfaces such as 2G, 3G, and 4G, recent mobile devices are also equipped with short-range radio interfaces such as WiFi Direct and Bluetooth, which enable local peer-to-peer communications. WiFi Direct and Bluetooth can offer low-cost, fast, and always available local connectivity, which will enable future MSNs to operate even without an infrastructure. Currently, Wi-Fi Direct has peer-to-peer transfer speeds of up to 250Mbps with a maximum distance of 656 feet, while Bluetooth 4.0 has lower power consumption and operates with speeds up to 25Mbps over a distance of at least 200 feet.

There has been much research on data routing approaches in opportunistic networks, which store, carry and forward messages from a source node to a destination node via intermediate nodes based on the local peer-to-peer communications in a decentralized and distributed environment. The simplest approach is epidemic routing [31]. When a node meets another node, they exchange messages that the other node do not possess. The protocol obviously incurs the highest transmission and storage costs. However, in practice the number of messages exchanged during each contact between two mobile nodes is limited by the duration of the contact. To improve the message delivery success ratio, nodes coming into contact should only exchange those messages that have a higher probability of being delivered to their destinations when processed by the receiving nodes. Many different approaches exist to optimize epidemic routing by means of reducing the number of copies of messages sent over the network.

An optimized approach called Spray and Wait routing [32] simply limits the number of message copies entering the network. Another approach [33] makes use of the mobility predictability property of the users. When nodes carrying the same message meet, the node predicted to have the earliest encounter with the destination node keeps the message while the other nodes discard the message. A representative approach that is able to minimize the transmission and storage costs in data routing is PROPHET [34]. It uses the history of past encounters between nodes to predict whether or not an encountered node can deliver a message to a destination. For example, if node A wants to send a message to node C and node B has met node C previously, node A will forward the message to node B when they meet, with the expectation that node B will likely meet node C again.

Some advanced approaches exploit human social relationships to make forwarding decisions. SimBet routing [35] determines the centrality of a node and social similarity based on an ego network analysis. Messages are forwarded to nodes that have higher centrality so that the possibility of finding suitable relay nodes towards the destination can be increased. In [36], a social group is detected based on the history of contact. Two nodes form a group if the contact strength exceeds a threshold value. Groups can be merged into a bigger group if they have enough common members. Each node or group has a delivery probability for each node. A message is forwarded to the nodes or groups with a higher delivery probability. In Bubble Rap [29], three community detection algorithms, called SIMPLE, K-CLIQUE, and MODULARITY, are employed. A message is forwarded to a node that interacts with more nodes globally until reaching a node that is in the same community as the destination node. The message is then forwarded to nodes with higher centrality in the community until the destination node is found.

In addition to using contact history, mobility patterns, or social relationships to make forwarding decisions, some data routing approaches are based on the predefined interests or existing social relationships of users in online social networks. In [37], dynamic groups are formed among nodes in close proximity dynamically based on the declared interests of the users. Messages are sent via relay nodes in the same group. MobiClique in [38] makes use of the API of the online social network Facebook to retrieve the friend lists of users. Messages are sent via friends of the destination nodes.

The above-mentioned work/studies generally provide four key features and principles for an effective opportunistic routing protocol as shown in Table 2. Inspired by the previous work, HORSE protocol (see chapter 4) generally seeks to combine the advantages of the aforementioned protocols based on these key features/principles.

Table 2 Key features/principles for effective opportunistic routing

Key Feature/Principle	Example(s)
Maximizing the delivery opportunity through broadcasting	[31]
Limiting relayed messages in the network	[32]
Using the contact history to facilitate data forwarding	[34]
Using social relationship or social feature to facilitate data forwarding	[29][35][36]

Some research has also been conducted on data dissemination approaches that disseminate a type of message to as many interested nodes as possible. In [39], Data dissemination is based on the users' predefined interest groups. Group messages are flooded within that interest group. A cooperative user-centric approach was proposed in [40]. When nodes meet each other, they exchange summaries of stored data items and request all interested data items. In addition, they request non-interested data items with a rewards probability. The reward probability of a node that stores a non-interested data item is computed based on the number of encountered nodes that are interested in the data item in the last time window. Similarly, in [41], each node associates a utility function with each data item. The utility function is determined based on the number of encountered nodes that are interested and the number of times that the data item has already been disseminated. Data items are exchanged with the aim of maximizing the utility of data items among nodes. A socially aware data dissemination framework called ContentPlace was proposed in [42]. The utility function is the sum of a data item's access probability for its community divided by the size of the data item and the social weight of a user's association with a community. Whenever nodes meet, each node determines a set of data items that maximizes the local utility of its cache, fetches the data items that are wanted in the set from others, and discards data items that are not in the set from the local cache. Moreover, in [30], a socially aware overlay network is built between nodes that have a higher closeness centrality within their communities. Data are disseminated using publish and subscribe operations via these nodes.

2.3 Simulation of Human Mobility Patterns

Simulating human mobility plays an important role in evaluating the above data routing approaches in opportunistic networks. The simulators in [43, 44] can simulate human mobility based on real world traces or mobility patterns.

Real world traces of communications between Bluetooth devices have been captured by various projects [45-47], enabling useful data such as contact frequency, contact durations, and locations. These traces can help researchers design better data routing approaches by exploring the real world interactions between mobile devices. However, they are limited by the small size of the data set due to high costs of experimentation, and by the large Bluetooth scanning intervals due to energy constraints. For example, in the MIT Reality Mining project [45], the population size was limited to 100 students and the Bluetooth scanning interval was limited to every 5 minutes.

A mobility model is another option to model human mobility. Two of the most common mobility models are Random Walk and Random Waypoint. In Random Walk [48], node movements are based on random directions and speeds. With Random Waypoint [49], each node stays at a location for a period of time before moving to another location with a randomly chosen direction and speed. A map-based variant of the Random Waypoint model is the Random Map-based model [51]. Nodes move to randomly selected locations following defined roads on a specific map. In the Shortest Path Map-based model [50], nodes move to a randomly selected location on a map following the shortest path calculated by Dijkstra's shortest path algorithm. An advanced model is the Working Day Movement model [51]. It models three major activities of people during a work week: 1) Being at home, 2) Working in the office 3) Doing activities with friends after work. In addition, it takes into consideration communities and social relationships. When nodes are performing the same activity in the same place, groups are formed. Nodes in the same group use the same movement parameters (e.g., speed and pause time). Other advanced models such as CMM [52] and HCMM [53] are based on complex social network theory. In CMM, node movements are determined by the social relationships of nodes. HCMM extends CMM by considering spatial attraction in node movements. It assumes that people prefer to spend time in popular locations and select the next location a short distance away.

2.4 Features and Services

MSNs can provide many features and services. Here are some examples from the literature.

2.4.1 Messaging/File Transfer

Internet connectivity is sometimes expensive and slow. It may even not be available in underground areas, rural regions, and disaster areas. Future MSNs will enable users without Internet connectivity to access the Internet via the connectivity of peers that are willing to provide relaying services and send messages [54] or files [46] through other mobile devices.

2.4.2 Media Streaming

Cooperative media streaming services are proposed in [55] for future MSNs. All mobile devices send their location information to a centralized server via the Internet. The centralized server sends commands to mobile devices so that all of the mobile devices can connect together via ad-hoc connectivity such as WiFi Direct by moving to a specific location. Some of the mobile devices are further connected to a centralized server via the Internet. Media streaming services can then be shared among these mobile devices with the advantage of the high speed ad-hoc connectivity.

2.4.3 Content Dissemination

Mobile users can access a great deal of useful local contents such as news, weather forecast, traffic alerts, and social media via the Internet. The contents are often of interest to nearby users. Future MSNs enable users to get such contents from other mobile users without accessing the Internet. For example, in [30], contents are disseminated among mobile devices using a publish-and-subscribe model. In addition, micro-blogging services such as Twitter enable users to send short messages that are followed by a public audience as in conventional MSNs. Future MSNs will allow users to share micro-blogs directly over opportunistic networks [56]. The localized social structures in society may help to deliver micro-blogs to interested recipients in an effective manner.

It will also be possible to search for information locally in future MSNs. A query can initially be propagated to a mobile device in a specific geographic area via a centralized server, and then further propagated between neighboring nodes over opportunistic networks [57].

2.4.4 Neighbor Discovery

Neighbor discovery will be a vital service in future MSNs. Interactions between physically proximate people were facilitated in [58] by using Bluetooth discovery to find nearby devices and a centralized server to match the profiles of users. With this service, conference participants can find the right people to meet, large companies can facilitate internal collaboration between employees, and individuals can find people with common interests in various social environments.

Another neighbor discovery service was proposed in [59]. Mobile devices disseminate the results of local device scanning to alert each other to the presence of parties of interest in a larger area that is beyond local scanning range. Users may then send messages to others to arrange for meetings.

2.5 User Behavior and Resource Management

Since users of mobile devices have their own needs for resources local to the devices, such as bandwidth, processing power, and energy, contributions to MSNs or to the distributed mobile computing environment will inevitably lead to a decrease in resources available to the users. To solve this problem in MSNs, the resource management techniques used in distributed computing may be applied to MSNs, e.g., renting in advance resources offered by other mobile devices [60].

Human altruism has been investigated deeply in [61, 62], where it was shown that human cooperation relies on rewards or punishments. Similar reward-based and punishment-based approaches are also applicable in MSNs. The basic idea is to encourage nodes to store, carry and forward messages. In reward-based approaches [63-65], credits are paid by source nodes that send messages to others and given to intermediate nodes that carry and deliver messages. Another credit-based approach [66] uses the concept of message trading. A node can get a message from another if it can provide a message in return. A similar approach is used in PlanetLab [67], a globally distributed platform used for developing and evaluating network services. In punishment-based approaches [68-70], nodes detect selfish nodes and propagate announcements of the identity of the selfish nodes over the network. The announcements eventually result in the selfish nodes becoming unable to receive any of the messages that are sent to them. Most existing ad-hoc based MSN protocols are designed based on the assumption of altruistic cooperation among nodes. However, the exchange of messages between nodes consumes resources such as energy and storage. Some nodes may download interested data items but refuse to store and forward data items to others. Taking into consideration the behavior of nodes is therefore important in designing MSN systems. A routing approach that considers the willingness of nodes when selecting relay nodes was proposed in [71]. Based on the assumption that users are partially altruistic, [72] showed that if all users have an altruistic coefficient $ß > 0$, then the price of the anarchy of traffic routing is bounded by $1/ß$. The impacts of altruism on the throughput and delay of MSN systems are studied in [73], which shows that MSNs are robust to the distributions of altruism because of the existence of multiple paths.

3 Social Network Framework

This chapter presents a general social network framework, which can be used for developing the HORSE protocol, the Smart Shopping System and the Dynamic Signage System in particular and supporting research on future MSNs in general. The major component of this framework is a new concept called social vectors.

3.1 Terminologies

First, we introduce or define the following terminologies for the social network framework:

Social Space – A social space is formed by social entities. A social relationship may exist between two social entities. This thesis puts focus on investigating an opportunistic routing protocol for routing messages over a social space using social relationships.

Social Entity – A social entity is the basic unit of a social space, which can be a person, a place, or an object. Social entities may interact with each other. Furthermore, messages can be sent across social entities.

Social Relationship – A social relationship may exist between two social entities. For example, a social relationship can be a “friend”, “colleague”, or “relative”. Chapter 3 makes use of social relationships between people to facilitate data forwarding. In the case of a social relationship between a shop and a person, the social relationship can be, for example, a “regular customer” or an “occasional visitor”. Chapter 4 makes use of social relationships between a shop and a person to identify similar customers.

Social Vector – A social vector $\overrightarrow{V_{X,Y}}$ quantifies a social relationship between two social entities X and Y by using social attributes. Details will be given in the next section.

Social Attribute – As mentioned, a social vector is comprised of social attributes, which may have continuous (e.g., contact duration), discrete (e.g., number of visits) or Boolean (e.g., whether the social entity has visited a certain place) values. Other values can also be defined.

Group – A group can be formed by similar social entities. Each group can be represented by a group social vector, which defines the main characteristics (or “centroid”) of the group.

Group Social Vector – A group social vector is the mean social vector among all the social vectors of the social entities belonging to the group. In other words, the mean social group vector can generally represent the group (i.e., its general characteristics).

3.2 Social Vector

An *m*-dimensional social vector $\overrightarrow{V_{X,Y}}$ = $[a_1, a_2, \ldots, a_m]$ (i.e., with m social attributes) can be defined to quantify the social relationship between two social entities X and Y in a social space. Besides, it has similar properties as traditional vectors.

Unit Social Vector

For the k^{th} social attribute, a unit vector $\overrightarrow{V_{X,Y}}(k)$ can be defined in the k^{th} dimension or direction to represent the value of attribute a_k.

Magnitude & Angle

The magnitude of $\overrightarrow{V_{X,Y}}$ is the strength of the social vector, which is defined by:

$$\left|\overrightarrow{V_{X,Y}}\right| = \sqrt{\sum_{i=1}^{m} \overrightarrow{V_{X,Y}}(i)^{\,2}} \tag{1}$$

The angle $\theta_{Y,Z}$ between two social vectors $\overrightarrow{V_{X,Y}}$ and $\overrightarrow{V_{X,Z}}$ is defined by:

$$\theta_{Y,Z} = \cos^{-1} \frac{\sum_{i=1}^{m} \overrightarrow{V_{X,Y}}(i) \times \overrightarrow{V_{X,Z}}(i)}{\left|\overrightarrow{V_{X,Y}}\right|\left|\overrightarrow{V_{X,Z}}\right|} \tag{2}$$

From the perspective of X, the social entities W and V are more similar than the social entities W and Z if

$$\theta_{W,V} < \theta_{W,Z} \tag{3}$$

and

$$\left|\overrightarrow{V_{X,W}} - \overrightarrow{V_{X,V}}\right| < \left|\overrightarrow{V_{X,W}} - \overrightarrow{V_{X,Z}}\right| \tag{4}$$

XNOR Operation

For Boolean attributes, the similarity between two social vectors can be found by using an XNOR operation. Given two social vectors $\overrightarrow{\mathrm{V_{X,Y}}}$ and $\overrightarrow{\mathrm{V_{X,Z}}}$ with *m* attributes:

$$\overrightarrow{\mathrm{V_{X,Y}}} \text{ XNOR } \overrightarrow{\mathrm{V_{X,Z}}} = [\overrightarrow{\mathrm{V_{X,Y}}}(1)\times \overrightarrow{\mathrm{V_{X,Z}}}(1), \overrightarrow{\mathrm{V_{X,Y}}}(2)\times \overrightarrow{\mathrm{V_{X,Z}}}(2),\ldots, \overrightarrow{\mathrm{V_{X,Y}}}(m)\times \overrightarrow{\mathrm{V_{X,Z}}}(m)] \tag{5}$$

Group Social Vector

A group of social vectors can be combined to form a group social vector. Suppose that there are *N* social vectors: $\overrightarrow{\mathrm{V_{X,E_1}}}$, $\overrightarrow{\mathrm{V_{X,E_2}}}$,…, $\overrightarrow{\mathrm{V_{X,E_{\mathit{N}}}}}$. If the corresponding social entities form a group, the group social vector will be $\overrightarrow{\mathrm{V_{X,G}}}$, which can be expressed as:

$$\overrightarrow{\mathrm{V_{X,G}}} = \frac{1}{N}[\sum_{i=1}^{N}\overrightarrow{\mathrm{V_{X,E_1}}}(1), \sum_{i=1}^{N}\overrightarrow{\mathrm{V_{X,E_1}}}(2), \ldots, \sum_{i=1}^{N}\overrightarrow{\mathrm{V_{X,E_1}}}(m)] \tag{6}$$

The vector values of this group social vector define the "centroid" of the group.

Weight Social Vector

In some cases, some attributes may be more important than others. A weight vector can therefore be introduced. Define W: $\{w_1, w_2,..., w_m\}$ as the weight social vectors. The weighted magnitude for the weighted social vector is defined by

$$\left|\overrightarrow{\mathrm{V_{X,Y}}}\right|_{\mathrm{w}} = \sqrt{\sum_{i=1}^{m} w_i \overrightarrow{\mathrm{V_{X,Y}}}(i)^{\,2}} \tag{7}$$

Similarly, the weighted angle $\theta_{\mathrm{Y,Z}}$ between two weighted social vectors is defined by

$$\theta_{\mathrm{Y,Z}} = \cos^{-1}\frac{\sum_{i=1}^{m} w_i \times \overrightarrow{\mathrm{V_{X,Y}}}(i) \times \overrightarrow{\mathrm{V_{X,Z}}}(i)}{\left|\overrightarrow{\mathrm{V_{X,Y}}}\right|_{\mathrm{w}}\left|\overrightarrow{\mathrm{V_{X,Z}}}\right|_{\mathrm{w}}} \tag{8}$$

The weighted XNOR result is

$$\overrightarrow{\mathrm{V_{X,Y}}} \text{ XNOR } \overrightarrow{\mathrm{V_{X,Z}}} = [w_1(\overrightarrow{\mathrm{V_{X,Y}}}(1) \times \overrightarrow{\mathrm{V_{X,Z}}}(1)), w_2(\overrightarrow{\mathrm{V_{X,Y}}}(2)\times \overrightarrow{\mathrm{V_{X,Z}}}(2)),\ldots,w_m(\overrightarrow{\mathrm{V_{X,Y}}}(m) \times \overrightarrow{\mathrm{V_{X,Z}}}(m))] \tag{9}$$

The weighed group social vector can also be defined as follows:

$$\overrightarrow{V_{\mathrm{X,G}}} = \frac{1}{N}\left[w_1 \sum_{i=1}^{N} \overrightarrow{\mathrm{V}_{\mathrm{X,E_i}}}(1), w_2 \sum_{i=1}^{N} \overrightarrow{\mathrm{V}_{\mathrm{X,E_i}}}(2), \ldots, w_m \sum_{i=1}^{N} \overrightarrow{\mathrm{V}_{\mathrm{X,E_i}}}(m)\right] \tag{10}$$

where $\sum_{i=0}^{m} w_i = 1$ and in general, $w_i = \frac{1}{m}$.

3.3 Application of Social Vectors in Previous Work

Social vectors can be applied in previous work and many other scenarios. As an example, we use social vectors to express the social relationships in [27]: Community, Familiar Stranger, Stranger, and Friend. In that case, a two-dimensional social vector is used with two attributes: a_1 = total number of contacts and a_2 = total contact duration. Define μ_1 and μ_2 as the average number of contacts and the mean contact duration, respectively. The four categories of social relationships can then be expressed as follows:

- Community - High number of contacts and longer contact duration:

$$\overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(1) > \mu_1 \text{ and } \overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(2) > \mu_2$$

- Familiar Stranger - High number of contacts and short contact duration:

$$\overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(1) > \mu_1 \text{ and } \overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(2) < \mu_2$$

- Stranger - Low number of contacts and short contact duration:

$$\overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(1) < \mu_1 \text{ and } \overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(2) < \mu_2$$

- Friend - Low number of contacts and longer contact duration:

$$\overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(1) < \mu_1 \text{ and } \overrightarrow{\mathrm{V}_{\mathrm{X,Y}}}(2) > \mu_2$$

3.4 Social Relationship between Two People

There are various ways to define “friends”. Social vectors provide an effective and flexible way to define friends and other social relationships, depending on different requirements. As an example, here we define friends using the angle and magnitude of a social vector. Again, we make use of the previously defined social vector $\overrightarrow{\mathrm{V_{X,Y}}}$ with two attributes: a_1 = total contact duration and a_2 = total number of contacts. Y is a friend of X if the following conditions are fulfilled:

$$\cos^{-1}\frac{\overrightarrow{\mathrm{V_{X,Y}}}\cdot\overrightarrow{\mathrm{U}_k}}{|\overrightarrow{\mathrm{V_{X,Y}}}||\overrightarrow{\mathrm{U}_k}|} > \alpha_k \; (k = 1, 2) \tag{10}$$

and

$$|\overrightarrow{\mathrm{V_{X,Y}}}| > \beta \tag{11}$$

These conditions are set to filter out people whose total contact duration and/or total number of contacts are too small to be classified as friends. Fig. 3 shows an example. T and Z are filtered (i.e., as non-friends). U is filtered too because the magnitude of the vector is too small. W, V, and Y are defined as friends of X. The one with the longest social vector (i.e., W) is X's best friend. Note that the above is just an example. Any other definitions can be used. As a further extension in later sections, we introduce a third Boolean attribute a_3, which indicates whether X has met Y recently (e.g., whether Y is among the recent λ persons met by X) (i.e., to identify recent friends). In other words, a_3 = 1 and a_3 = 0 mean that the person has recently and has not recently been met by X, respectively. It has been shown by previous research [14] that if X has met Y recently, there is a higher possibility that X will met Y again in the near future.

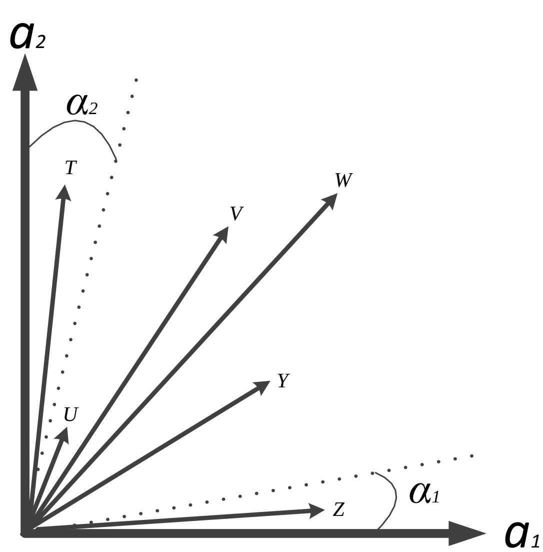


Fig. 3 Social vectors of a social entity with other people

3.5 Social Relationship between a Person and a Set of Places

Social vectors can also be used to define the social relationship between a person (e.g., A, B, C, D, E, and F) and a set of places (e.g., P_1 and P_2). Referring to Fig. 4, for purposes of illustration we define a simple social vector with two social attributes a_1 and a_2 as the number of visits to P_1 and P_2, respectively. The figure shows the social vectors for A, B, C, D, E, and F for a social space with P_1 and P_2. The following social relationships can be defined as follows: B is classified as a familiar visitor or a loyal customer of P_1 while E is classified as a familiar visitor or a loyal customer of P_2. C is a familiar visitor or a loyal customer of both P_1 and P_2. Since the number of visits is too low, A is classified as a new visitor to P_1 and F is classified as a new visitor to P_2. D is likely to be a new visitor to both P_1 and P_2. Based on the social vectors, the shops can, for example, conduct different marketing activities. For example, a shop can determine the similarity between two customers (e.g., for purposes of recommendation) and send electronic coupons to a certain group of customers (e.g., new visitors) when they enter the shop.

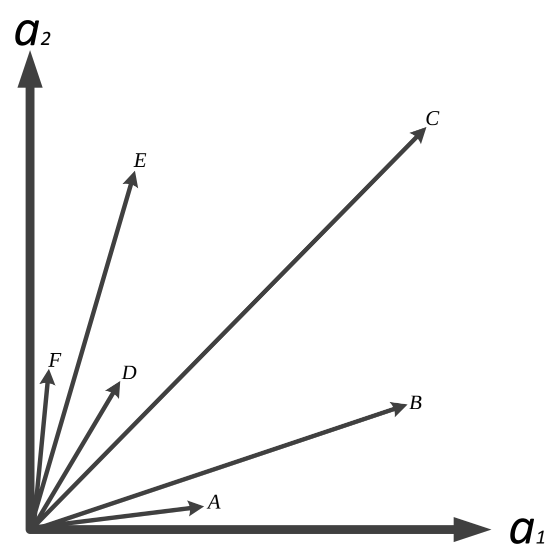


Fig. 4 Social vectors for a person and a set of places

3.6 Social Relationship between a Group of People

In reality, social relationships are not limited to one-to-one relationships such as “friend”, “colleague”, or “customer”. Social relationships such as “community” are often formed by a group of people. In this section, we investigate social vectors to define many-to-many social relationships between a group of people. People will be assigned to different communities or crowd sourcing teams. Note that communities and crowd sourcing teams are two opposite concepts. In essence, similar social entities form a community while dissimilar social entities form a crowd sourcing team. In this section, people are assigned into the required number of groups (i.e., either communities or crowd sourcing teams) by using social vectors and intelligent computing methods. In particular, intelligent computing methods based on Genetic Algorithm and K Means Clustering will be employed. Simulation results are presented to compare the performance of the methods we employed.

3.6.1 Community

Community [27] [74] is a group of similar social entities that has similar characteristics such as geographic location, culture and social relationship. With the advent of the Internet, people can communicate virtually and share common interests regardless of the physical location. People under a same community are probably doing similar things on the Internet. For example, they tend to visit a same website or communicate with a same group of people. Assume people that visited a website are under a same community. If we need to classify M people into N community, we first define a set of websites: $\{W_1, W_2, \ldots, W_N\}$. A social attribute a_i indicates whether a person P visited W_i. If P read W_i, a_i = 1. Otherwise a_i = 0. The social vector between a person P and a set of websites W is defined by

$$\overrightarrow{\mathrm{V}_{\mathrm{W,P}}} = [a_1, a_2, \ldots, a_k]$$

where *k* is any integer.

If there is a community C_i formed by a group of people P_j, where j = 1, 2,…, L_i and $\sum_{i=0}^{N} L_i = M$, the social vector between a community and a set of book is defined by

$$\overrightarrow{\mathrm{V}_{\mathrm{C_1,W}}} = \frac{1}{L}[\sum_{j=1}^{L} \overrightarrow{\mathrm{V}_{\mathrm{P}_j,\mathrm{W}}}(1), \sum_{j=1}^{L} \overrightarrow{\mathrm{V}_{\mathrm{P}_j,\mathrm{W}}}(2), \ldots, \sum_{j=1}^{L} \overrightarrow{\mathrm{V}_{\mathrm{P}_j,\mathrm{W}}}(L)] \tag{12}$$

In this example, people are well assigned to different communities if $\left|\overrightarrow{\mathrm{V}_{\mathrm{C_1,W}}}\right|$ is as small as possible for all i.

3.6.2 Crowd Sourcing Team

Crowd sourcing [75][76] is a novel approach that divides a large task into many small tasks. The small tasks are then distributed to people that are familiar to handle. Obviously, a good crowd sourcing team should be formed by people that have different knowledge or strengths. In other words, a good crowd sourcing team is formed by dissimilar social entities.

If we assume there are *M* people and *N* tasks, the easiest way to construct *N* crowd sourcing team is to assign these *M* people to *N* tasks randomly. However, some people may have knowledge in a particular field such as programming. If the people who have similar knowledge are assigned to a same task rather than distributed to *N* tasks, all tasks may not be executed in an effective manner. Therefore, people who have different knowledge or strengths should be assigned to various different crowd sourcing teams. Below is an example of how to use social vector to assign people to different crowd sourcing teams effectively.

Similar to assigning people into different communities, we first define a set of knowledge K:{ $K_1, K_2, \ldots, K_m$}. A social attribute a_i indicates whether a person P has knowledge K_i. In other words, if P has K_i, $a_i = 1$. Otherwise $a_i = 0$. The social vector between a person P and a set of knowledge K is defined by

$$\overrightarrow{\mathrm{V}_{\mathrm{P,K}}} = [a_1, a_2, \ldots, a_m]$$

where *m* is any integer.

Assume there is a crowd sourcing team T_i formed by a group of people P_j, where $j = 1,2,\ldots, L_i$, and $\sum_{i=0}^{N} L_i = M$.The social vector between a crowd sourcing team and a set of knowledge K is defined by

$$\overrightarrow{V_{T_i,K}} = \frac{1}{L}\left[\sum_{j=1}^{L}\overrightarrow{V_{P_j,K}}(1), \sum_{j=1}^{L}\overrightarrow{V_{P_j,K}}(2)\ , \ldots, \sum_{j=1}^{L}\overrightarrow{V_{P_j,K}}(L)\right] \tag{13}$$

In this example, good crowd sourcing teams are found if $\left|\overrightarrow{V_{T_i,K}}\right|$ is as large as possible for all *i*.

3.6.3 GA-Based Clustering

In this section, we present the GA-based clustering algorithm for assigning people to different communities and crowd sourcing teams effectively. Inspired by the evolution of living organisms, genetic algorithms are intelligent computing techniques for finding optimized solutions. Basically, practical solutions are expressed as “chromosomes”, which can be mixed to generate a new chromosome through a crossover process. Sometimes, a mutation process can also be employed to introduce small changes in chromosomes after the crossover operation. After many generations of crossover and mutation operations, a close-to-optimal solution is expressed to be obtained.

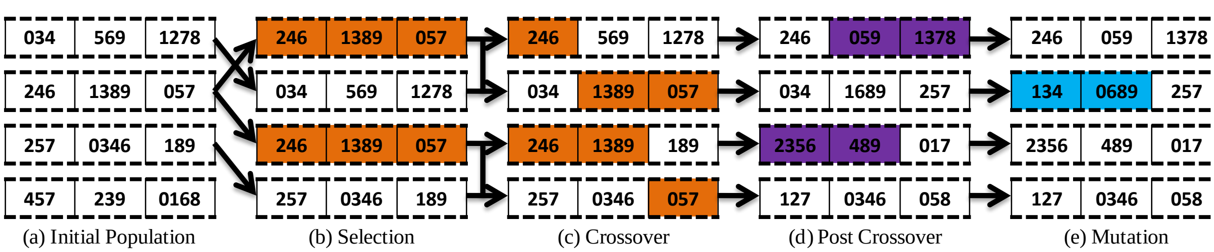


Fig. 5 GA-Based clustering

Fig. 5 gives an example. Assume there are ten people. Each of them has an identity number {0,1,…,9}. Initially, random solutions are generated to assign the people into groups. Every three groups function like a chromosome. Chromosomes are selected based on their fitness values. The fitness value is defined by the sum of distance between each group member's social vector and the centroid of the group social vector. In the selection process, α chromosomes are selected through a random process based on their fitness value. For assigning people to communities effectively, chromosomes with a lower fitness have a higher probability of being selected. On the other hand, for assigning people to different crowd sourcing teams effectively, chromosomes with a higher fitness have a higher probability of being selected. Note that a chromosome may be selected more than once. After selecting $\frac{\alpha}{2}$ pairs of chromosomes, each of them is mixed in the crossover process. Assume that two parent chromosomes, A and B have been selected. In the crossover process, β% of A is mixed with (1- β%) of B to produce C and (1- β%) of B is mixed with β% of B to produce D. For the post crossover process, some identity numbers may need to be added or removed to ensure the identity numbers of all people are in each chromosome and without duplication. The best two chromosomes among A, B, C and D can survive. As for the mutation process, any identity number may be replaced by another with a pre-defined mutation probability. The above steps are repeated 1000 times (i.e., 1000 generations).

3.6.4 GMin and GMax

In this section, we present the GMin and GMax algorithms for assigning people to different communities and crowd sourcing teams effectively. Basically, GMin and GMax are developed based on the K Means clustering algorithm. The K Means clustering algorithm is an evolutionary algorithm which clusters *n* observations into *k* groups and then assigns each observation to clusters according to the distance between the observation and the cluster's mean. The process is repeated until all observations in clusters are unchanged. For GMin and GMax, the number of observations assigned into a group is limited. When the number of observations in a group exceeds the pre-defined limit, GMin and GMax will rearrange one of the observations in a group to another. GMin is designed to assign people to different communities effectively while GMax is designed to assign people to crowd sourcing teams effectively.

Fig. 6 gives an example. We randomly assign people into three communities or crowd sourcing teams so that each community or crowd sourcing team has roughly the same number of people. For each person, we calculate the distance between his/her social vector and the centroid of the social vector of his/her community or crowd sourcing team. For GMin, we assign the person into the closest community if the social vector is not closest to his/her own community. For maintaining the desired number of people in each community, a person who is farther in his/her community is assigned to the second closest community if the number of people in the original crowd sourcing team exceeds the predefined limit. By contrast, for GMax, we assign the person into the farthest crowd sourcing team if the social vector is not farthest to his/her own crowd sourcing team. Similarly, for maintaining the desired number of people in each crowd sourcing team, a person who is closest in his/her crowd sourcing team is assigned to the second farthest crowd sourcing team if the number of people in the original crowd sourcing team exceeds the predefined limit. Above process is repeated until no people need to be assigned from one community or crowd sourcing team to another. In other words, the process ends when all groups become stable. Since the choice of initial people in communities or crowd sourcing teams can greatly affect the final result, the best result of multiple trials of different initial people in communities or crowd sourcing teams will be adopted.

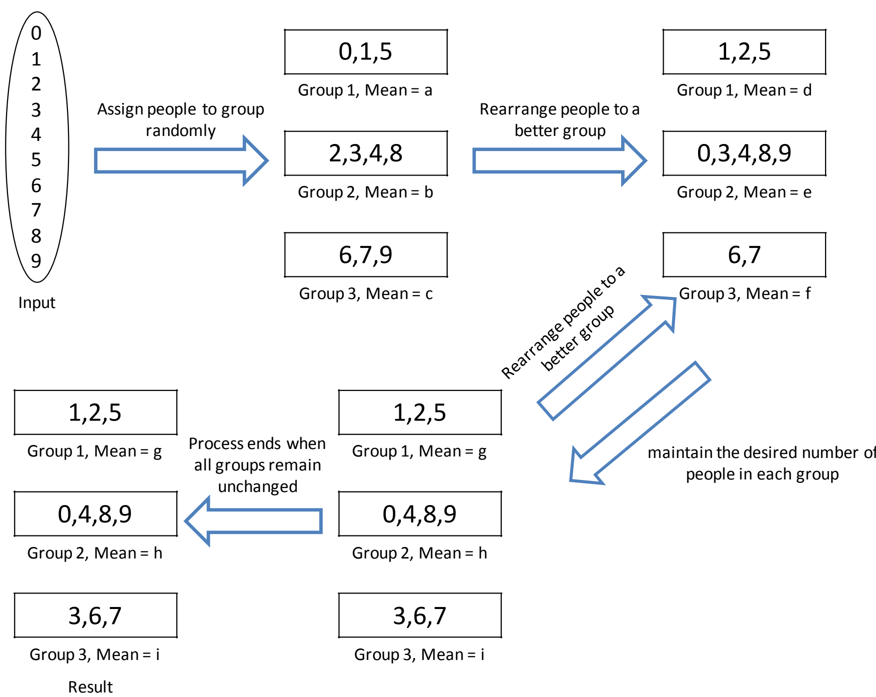


Fig. 6 GMin and GMax

3.6.5 Simulation Results and Discussion

We have evaluated the performance of Genetic Algorithm (GA), GMin and GMax for finding communities and crowd sourcing teams based on a Java simulation program. Assume that there are 15 people. Each of them has *i* social attributes indicating whether a person P read B_i or has knowledge K_i. Each social attribute has value between 0.0 and 1.0, generated by a random number generator. We assign these people to *K* communities or crowd sourcing teams. For GA, the simulation process ends after 1000 generations. As for GMin and GMax, the simulation process is repeated until no people need to be assigned from one community or crowd sourcing team to another. Simulation results are shown below.

We first compare the effectiveness of GA and GMin under different required numbers of communities. Fig. 7 shows the comparison of GA and GMin. Shorter distance between people and the centroids of communities implies that people is closer to the community. It can be seen that when the number of communities increases, the average distance between people and the centroids of communities decreases. It is because when the number of communities increases, the probability that people can be arranged into a closer community is also increased. GA clearly outperforms GMin because it can arrange people to a closer community in an intelligent way. With GA, the average distance between people and the centroids of the communities is less than 0.6 when the required number of communities is 5. GMin performs worse than GA. With GMin, the average distance between people and the centroids of communities is higher than 0.6 when the required number of communities is 5.

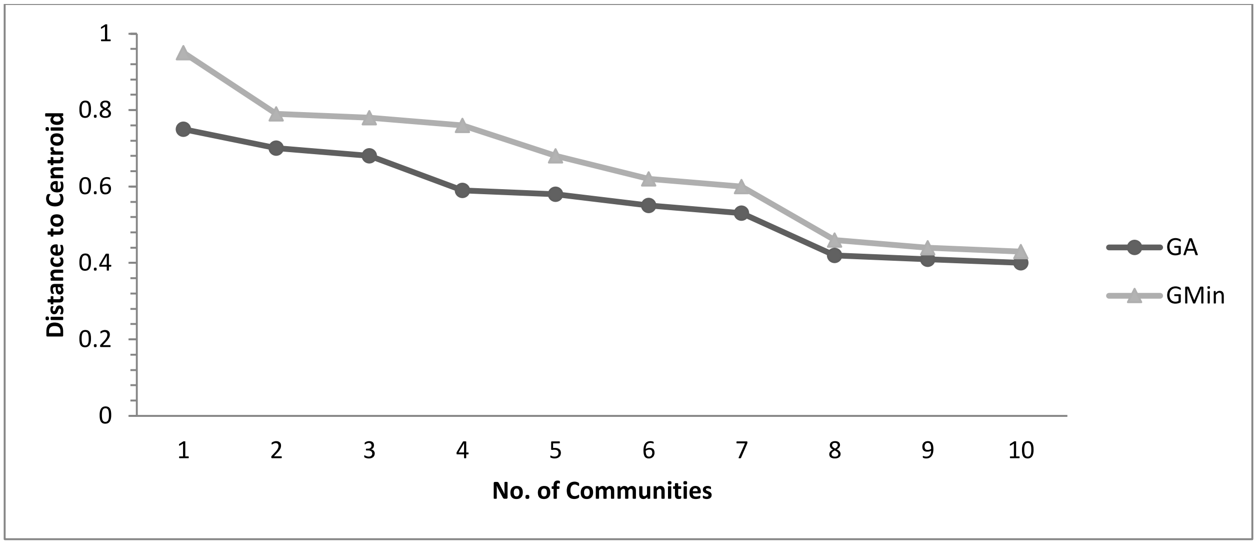


Fig. 7 Comparison of GA and GMin (different no. of communities)

Then we compare the effectiveness of GA and GMax under different required numbers of crowd sourcing teams. Fig. 8 shows the comparison of GA and GMax. Longer distance between the people and the centroids of crowd sourcing teams implies that every person in the crowd sourcing teams has knowledge in a wider range. It can be seen that the average distance between people and the centroids of the crowd sourcing teams decreases when the number of crowd sourcing teams increases. It is because when the number of crowd sourcing teams increases, the probability that people can be arranged into a farthest crowd sourcing team is decreased. Besides, GA achieves better performance because it can arrange people to crowd sourcing teams more intelligently. With GA, the average distance between people and the centroids of crowdsourcing teams is about 0.9 when the required number of communities is 5. GMax performs worse than GA. With GMax, the average distance between people and the centroids of crowdsourcing teams is shorter than 0.8 when the required number of communities is 5.

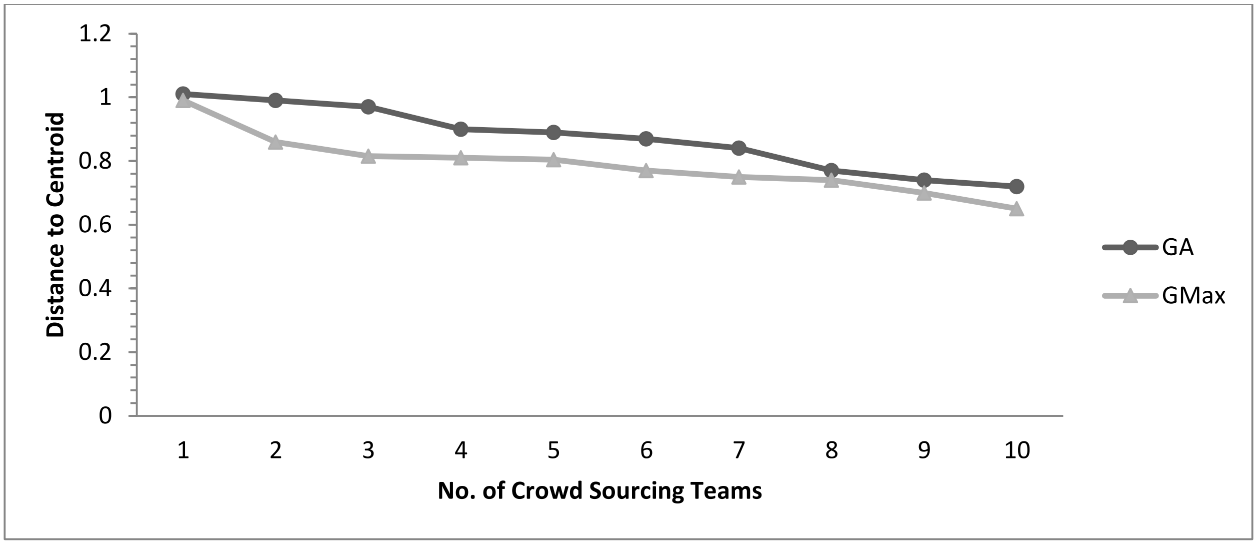


Fig. 8 Comparison of GA and GMax (different no. of communities)

Next, we compare the effectiveness of GA and GMin under different numbers of social attributes. Fig. 9 shows the comparison of GA and GMin. Shorter distance between people and the centroids of communities implies that people is closer to their communities. It can be seen that when the number of social attributes increases, the average distance between people and the centroids of communities increases. In the simulation, both GA and GMin have similar performance. GA can assign people to different communities in an intelligent way while GMin can assign people into different communities in a systematic way. With them, the average distance between people and community centroids is less than 0.9 when there are 20 social attributes.

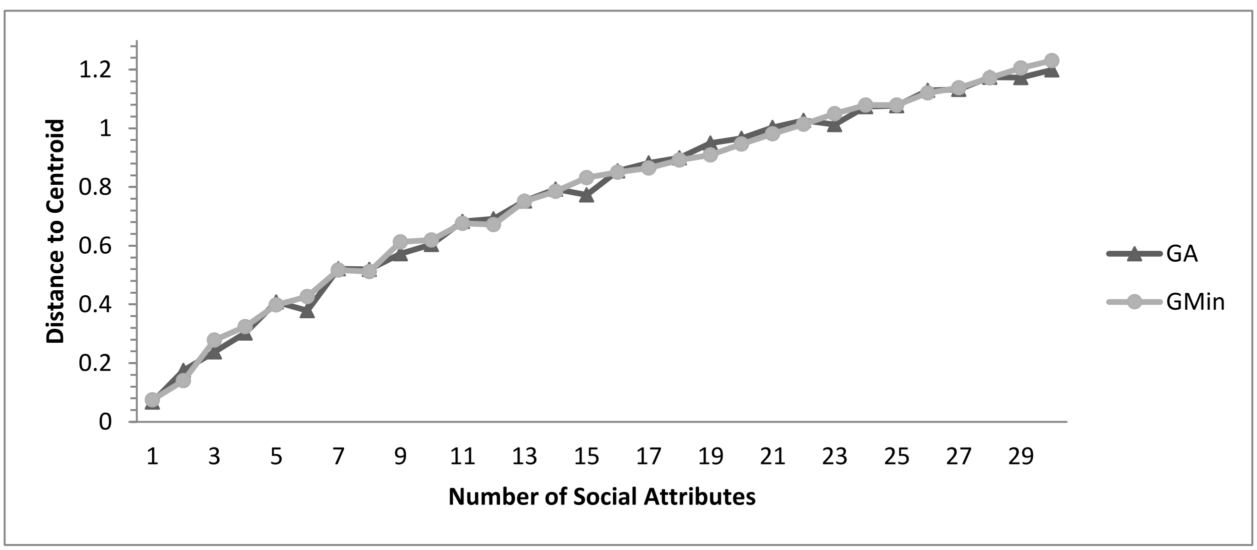


Fig. 9 Comparison of GA and GMin (different no. of social attributes)

Last, but not least, we compare the effectiveness of GA and GMax under different number of social attributes. Fig. 10 shows the comparison of GA and GMax. Longer distance between the people and the centroids of crowd sourcing teams implies that every person in the crowd sourcing team has different knowledge. It can be seen that the average distance between people and the centroids of the crowd sourcing teams increases when the number of social attributes increases. Besides, GA achieves better performance because it can arrange people to different crowd sourcing teams more intelligently. With GA, the average distance between people and community centroids is longer than 1.2 when the number of social attributes is 20. GMax performs worse than GA. With GMax, the average distance between people and the centroids of crowd sourcing teams is about 1.1 when the number of social attributes is 20.

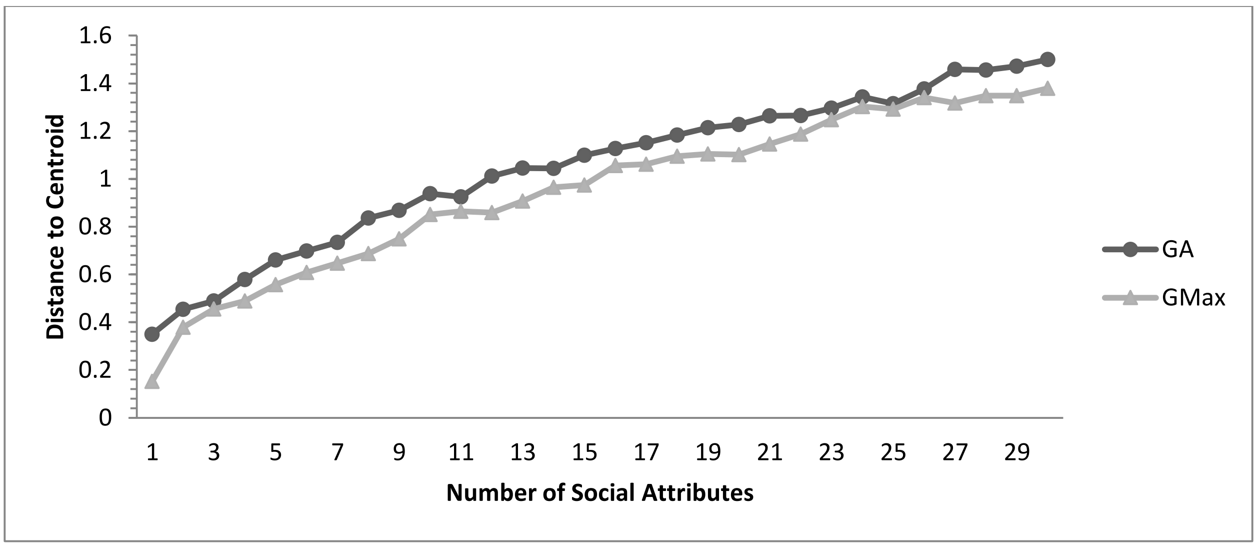


Fig. 10 Comparison of GA and GMax (different no. of social attributes)

4 Hybrid Opportunistic Routing for Social Entities (HORSE)

In this chapter, we present an opportunistic routing protocol called Hybrid Opportunistic Routing for Social Entities (HORSE)[1]. With the help of social vectors defined in Chapter 2, HORSE seeks to combine the advantages of the previous opportunistic routing protocols in order to forward social messages effectively and efficiently over future MSNs. In particular, it is inspired by the "six degrees of separation" theory, which provides the basis for forwarding social messages through a chain of friends.

4.1 Basic Concept

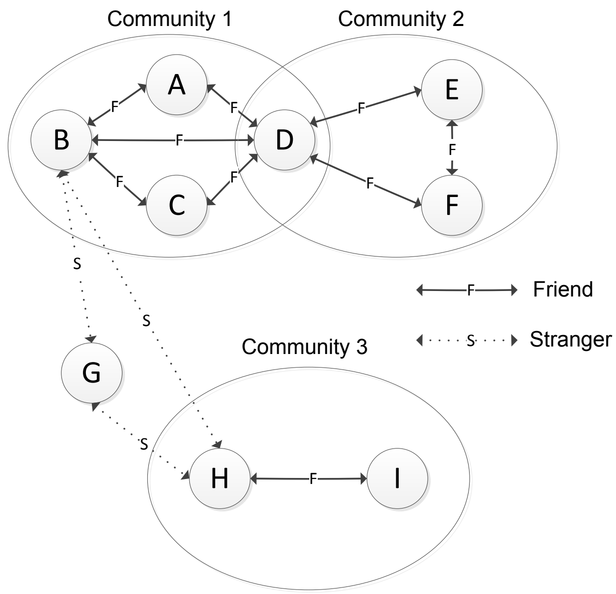


Fig. 11 Basic concept of HORSE

Fig. 11 shows the basic concept or big picture of HORSE with an example of three communities. In a community, each person should have a friend relationship (i.e., determined by a social vector) with one or more person(s). For example, A and B are friends in Community 1. Note that some people may belong to multiple communities. For example, D belongs to Community 1 and Community 2 in this example. These people can function as "inter-community routers". Besides

[1] There is a protocol called Hybrid Opportunistic Routing in the literature [82]. That protocol works quite differently and is designed for another purpose (i.e., not for mobile social networks).

friends, people also meet strangers. If a stranger is highly sociable (i.e., meeting many people), he/she may also serve as an "inter-community router", relaying messages across communities. For example, G is a stranger to both Community 1 and Community 3 but he/she may relay messages between the two communities. Furthermore, when a person meets a stranger who knows completely different people (i.e., an exceptional stranger), it may be desirable to pass the message to that person because that stranger may relay messages to other communities. In summary, HORSE relies on hybrid channels (i.e., friends, highly sociable strangers, and exceptional strangers) to relay messages by means of opportunistic routing. In essence, these people function like "virtual routers". For each message, a list of virtual routers is maintained. These channels or virtual routers are explained in further detail below.

4.2 Friends

In HORSE, messages are primarily relayed by friends. This is inspired by the well-known theory of "six degrees of separation", which means that any two individuals in the world could be connected by five other individuals (i.e., friends of friends). We call this chain a "friend chain". Therefore, if X wants to forward a message to Y and X meets Z who has Y in the friend chain, Z can help X relay the message to Y. The friend relationship is determined based on social vectors, discussed in the last section. To identify friend chains, each person keeps a friend tree with a breadth b and a depth d. In essence, based on social vectors, each person identifies b best friends (i.e., as determined by the b longest qualified social vectors).

When a person meets other people, the information on friends can be exchanged so that a friend tree can be built. We assume that there is a warming up period for building up friend trees and the friend trees are updated continuously. To facilitate implementation, each person only keeps a friend tree up to d levels (i.e., a depth of d). Note that if b is very large and d is 5, a person may keep the whole population of the world according to the theory of "six degrees of separation". For a friend tree with a breadth b and a depth d, there are f people, where

$$f = \sum_{i=1}^{d} b^i = \frac{b(1 - b^d)}{1 - b} \tag{14}$$

Pseudo code of building a friend tree

```
/*called whenever disconnected from a social entity x*/
updateSocialVector(x) {
    x.updateNoOfContacts()
    x.updateContactDuration()
    x.updateRecentContactsList()
    a1 = x.getNoOfContacts() / maxNoOfContacts
    a2 = x.getContactDuration() / maxContactDuration
    a3 = 1 if(recentContactsList.contains(x))
    a3 = 0 otherwise
    x.setSocialVector (a1,a2,a3)
    if(x.getSocialVector().getAngle(1) > α1 and
       x.getSocialVector().getAngle(2) > α2 )
       friendList.add(x)
    }

buildFriendTree(){
    sort friendList based on its social vector length
    for first b social entities x in friendList
        rootNode.addChild(x.copyFriendTree(d - 1))
}
```

Fig. 12 Pseudo code of building a friend tree

Table 3 Definition of key variables/functions for Fig. 12

Variable/Function	Definition
x	A social entity
noOfContacts	Cumulated number of contacts with a social entity
contactDuration	Cumulated contact duration with a social entity
maxNoOfContacts	Maximum number of contacts among all encountered social entities
maxContactDuration	Maximum contact duration among all encountered social entities
recentContactsList	List of recent λ contacted social entities
friendList	List of friends
rootNode	Root node of the friend tree
updateS()/setS()/getS()	Update/Set/Get the value of S
copyFriendTree(*d*)	Copy the friend tree up to level *d*

After passing a message to Y, Z and Z's friends (i.e., based on Z's friend tree) are included as virtual routers for the message. Fig. 12 gives the pseudo code of building a friend tree. Table 3 shows the definition of its key variables and functions.

4.3 Highly Sociable Strangers

We may not only rely on friends to relay messages for a recipient because the recipient may belong to another community that has no social relationships with the sender and his/her friends. In general, friends are "intra-community routers" and occasionally "inter-community routers". Strangers can also serve as "inter-community routers". In particular, it is desirable to pass a message to a highly sociable stranger in the hope that the message will be spread more quickly because that people is likely to meet many more people than an average person. In HORSE, each person keeps track of the average number of contact persons per week, which indicates how sociable that person is. Suppose that for a message M in X's buffer, the overall average number of contact persons per week of the virtual routers for the message M is g. If X meets Y, who has a higher average number of contact persons per week by a factor of ψ (i.e., greater than $\psi \times g$), the message M will be passed to Y for delivery. Furthermore, once a message is passed to Y, Y and his/her friends are included as virtual routers for the corresponding message.

4.4 Exceptional Strangers

To further enhance the delivery probability while maintaining a low overhead cost, HORSE also makes use of exceptional strangers to reach out to unknown or remote communities. Suppose that X meets Y, and that for a message M none of the virtual routers for M is Y or Y's friends. In other words, Y knows people whom X does not know at all. In this case, X will use Y to deliver the message with probability p. Again, once the message is passed to Y, Y and Y's friends will be included as the virtual routers for the corresponding message M. In general, p should be set to a low value to maintain a low overhead cost. Note that this feature can also be disabled by setting $p = 0$, if required.

4.5 HORSE Protocol

Putting together the above delivery mechanisms, the following summarizes the HORSE protocol for a person or social entity X (i.e., focusing on the operation at X). Based on the aforementioned social vectors, X identifies *b* best friends and maintains a friend tree with a breadth *b* and a depth *d*. Note that the best friends and the friend tree can be updated continuously based on ongoing contact information. X stores the messages to be forwarded in a buffer of finite size *B*.

On a regular basis, X identifies the people within his/her area or range of communication. Note that people typically come and go dynamically and communications are carried out in an opportunistic manner (i.e., the following operations may be terminated suddenly). X randomly chooses one of the available people within the communication range for communication, as follows. Assume that Y is chosen. X and Y first exchange a summary of messages that they have. If X finds that Y is the recipient of a message in the buffer, X will obviously immediately deliver the message to Y. Then, X and Y exchange their friend trees. If X finds that the recipient of a message in the buffer is in Y's friend tree, X will deliver the message to Y. After that, for each remaining message in the buffer, if X finds that Y is more sociable than all of the virtual routers for the message as discussed above, it will send the message to Y for forwarding to other people.

Finally, for each remaining message in the buffer, if none of the virtual routers is Y or Y's friends, X will forward the message to Y with probability p. Whenever a message is sent to Y for delivery to the destined recipient Z, the virtual routers for the message will be updated (i.e., updated with Y and Y's friends). Besides, if a message is already forwarded to *b* Z's friends, it will not be forwarded to other people except Z. Note that after a message is sent to Z, it should be deleted from the corresponding buffer. Once a connection is set up between two people, the connection will be terminated if they have sent all required messages or they are no longer in contact (i.e., outside the communication range). Consistent with the arrangement used by other protocols (i.e., for fair comparison), if a buffer is full, an incoming message will be dropped automatically. Alternatively, old messages can be overwritten by new messages if the buffer is full. Fig. 13 gives the pseudo code of HORSE. Table 4 shows the definition of its key variables and functions.

Pseudo code of HORSE

```
        /*send message m to x*/
        /* and update virtual router & friend count*/
        for a message m in outgoingMessages
            isReceived = m.forwardTo(x)
            if(isReceived)
                updateVirtualRouter(x)
                if(m.isFirstPriority())
                    m.friendCount++
```

Fig. 13 Pseudo code of HORSE

Table 4 Definition of key variables/functions for Fig. 13

Variable/Function	Definition
outgoingMessages	Messages to be sent to x
forwardTo(x)	Forward the message to x
friendCount	Number of friend nodes passed
isMoreSociable(x)	Whether x is more sociable
isVirtualRouter(x)	Whether x is a virtual router
updateVirtualRouter(x)	Update x and his/her friend tree as virtual routers
put(m, #)	Put message m into the outgoing messages queue with priority #

In summary, Fig. 14 summarizes the possible data forwarding paths of HORSE for a message through the virtual routers. S and D are respectively the sender and receiver of the message. F is a social entity in the friend chain of the destination. H is a highly sociable social entity. N is an exceptional stranger.

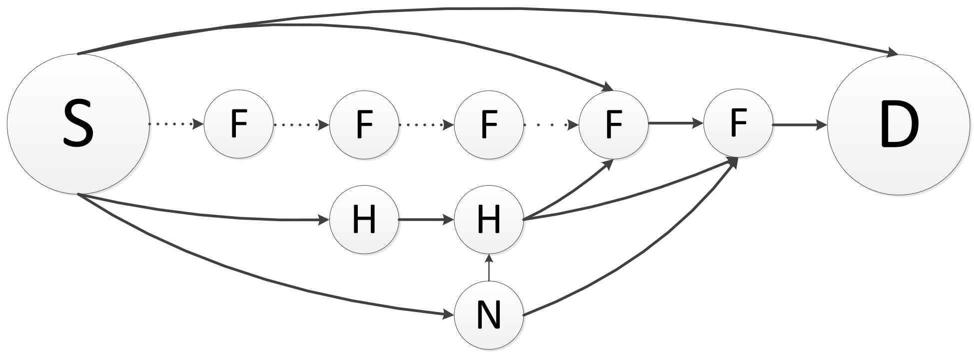


Fig. 14 Data forwarding with HORSE using virtual routers

4.6 Simulation Results and Discussion

We have evaluated the performance of the proposed HORSE protocol based on simulations using the Opportunistic Networking Environment (ONE) simulator [77], simulating all network protocol layers except the physical layer. Simulation results using real-world data traces are employed to evaluate the performance of HORSE in comparison with other schemes. Real-world data traces of mobile devices (e.g., contact frequency, contact duration and location) have been recorded by Bluetooth devices in various projects [46] [78] [79]. These data traces provide real-life data for testing and evaluating opportunistic routing protocols based on real interactions among mobile devices. In this section, we use the real-world data traces from the MIT Reality Mining project [78], where 100 Nokia 6600 smart phones were distributed to students and faculty members at MIT for a period of 9 months. These phones were pre-installed with a program to record contact information using the Bluetooth device discovery protocol every 5 minutes.

The ONE simulator offers a set of tools to simulate complex mobility scenarios in the real world. It allows user to use real-world data traces or mobility traces generated by other simulators to perform human mobility simulation. It also has some standard performance matrix for the evaluation of different routing protocols. One of its main drawbacks is that its message generation does not take group relationships and context information into consideration.

In the simulation environment, we used a friend tree with b = 3 and d = 2. Besides, as an example, we set $\alpha_1 = \alpha_2 = 10$, ψ = 0.7, p = 0.001 and λ = 10. 200 messages are to be delivered from a source node to a destination node. The size of the messages varies uniformly from 1KByte to 500KByte. Each node has a Bluetooth networking interface with a transmission speed of 250Kbps and a buffer size of B = 40MByte. After a message is sent to the destination (i.e., intended recipient), it is deleted automatically. An incoming message will be dropped if the buffer is full. Once a connection between two nodes is set, it will only be terminated if all required messages have been exchanged or if they are out of contact (i.e., outside the Bluetooth communication range).

The performance of the routing protocols is evaluated based on five common performance indicators: delivery ratio, delivery cost, overhead, hop count, and number of dropped messages. The delivery ratio is defined as the number of messages delivered to destinations out of the total number of messages created by senders. The delivery cost is defined as the total number of messages relayed out of the total number of messages created by senders. The overhead is defined as the total number of extra messages generated out of the total number of messages delivered successfully to the destinations. The hop count is the average number of hops that a message is sent until reaching the destination. The number of dropped messages is the number of messages discarded due to buffer overflow. For the later simulation results, following the previous work, we show the cumulative average values over time.

For purposes of evaluation, we compare HORSE with the following five representative protocols in the ONE simulator. These protocols are commonly used for comparison or benchmarking purposes.

- **Epidemic** – Nodes deliver messages to all nodes that are encountered (i.e., flooding). Under ideal situations (e.g., infinite buffer size and infinite bandwidth), this approach sets the upper bound for the delivery ratio.
- **Direct Delivery** – A sending node only delivers a message to the destination node directly (i.e., only when they meet). Obviously, the overhead cost is the lowest. It generally sets the lower bound for the delivery ratio as well.

- **PROPHET** – Nodes use a history of past encountered nodes to predict whether an encountered node can deliver a message to a destination. PROPHET has been evaluated against many other algorithms before. In general, it achieves a very good performance.
- **Spray & Wait –** A sending node generates *L* copies of a message and sends half of the remaining copies to an encountered node, which does not have the message. Other nodes follow the same method of transmission. When a node transfers all copies to others, it waits for a direct opportunity to transmit the copy to the destination.
- **Bubble Rap -** A message is forwarded to a node that interacts with more nodes until reaching a node that belongs to the same community as the destination node. The message is then forwarded to nodes with higher centrality in the community until the destination node is reached. Bubble Rap is a representative social-based forwarding protocol. In this chapter, the code used for simulating Bubble Rap is based on [80].

Fig. 15 shows the delivery ratio. It can be seen that Epidemic, PROPHET and HORSE have the highest delivery ratio, delivering about 60% of the messages after two weeks. Bubble Rap and Spray & Wait can achieve a comparable delivery ratio (i.e., close to 50% of the messages are delivered after two weeks). As expected, Direct Delivery has the lowest delivery ratio, just about 15%.

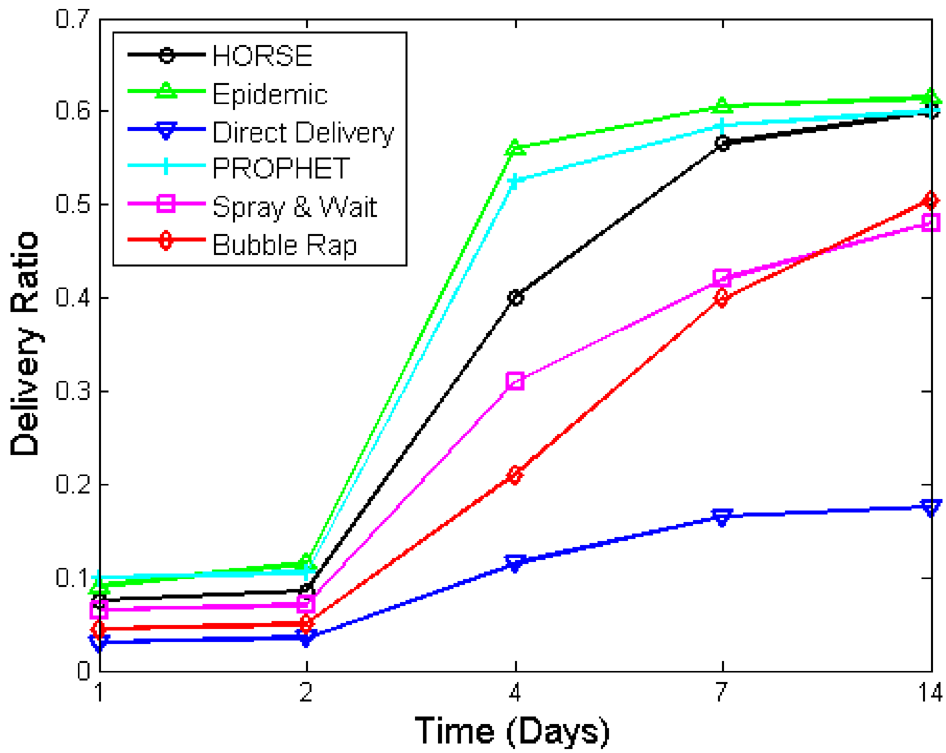


Fig. 15 Delivery ratio

Fig. 16 shows the delivery cost. Although Epidemic and PROPHET can achieve a higher delivery ratio than that of HORSE, their delivery costs are about ten times higher than that of HORSE after two weeks. The delivery cost of Epidemic is the highest, obviously because of its flooding mechanism. However, Epidemic cannot achieve much higher delivery ratio than PROPHET and HORSE because many unnecessary messages are generated in the network, which generally affects the opportunity for sending messages to appropriate nodes. It can be seen that HORSE can achieve a high delivery ratio with a low delivery cost because it makes use of social relationships to make forwarding decisions. Its delivery cost is low because the number of messages relayed is limited. A message will only be forwarded to people in the friend chain of the destined node, a highly sociable node, or an exceptional stranger. Moreover, after a node forwards the message to a node in the friend chain, it will no longer forward the message to a highly sociable node or to an exceptional stranger so as to limit the number of messages sent to the network. As expected, Direct Delivery has the lowest delivery cost but it can only deliver about one fifth of the messages to the destination after two weeks. This is because it is not easy for the source node to meet the destination node directly. Epidemic and Direct Delivery generally provide upper and lower bound delivery ratio for purposes of evaluation, respectively.

Although Bubble Rap and Spray & Wait can provide a low delivery cost, their delivery ratio is lower.

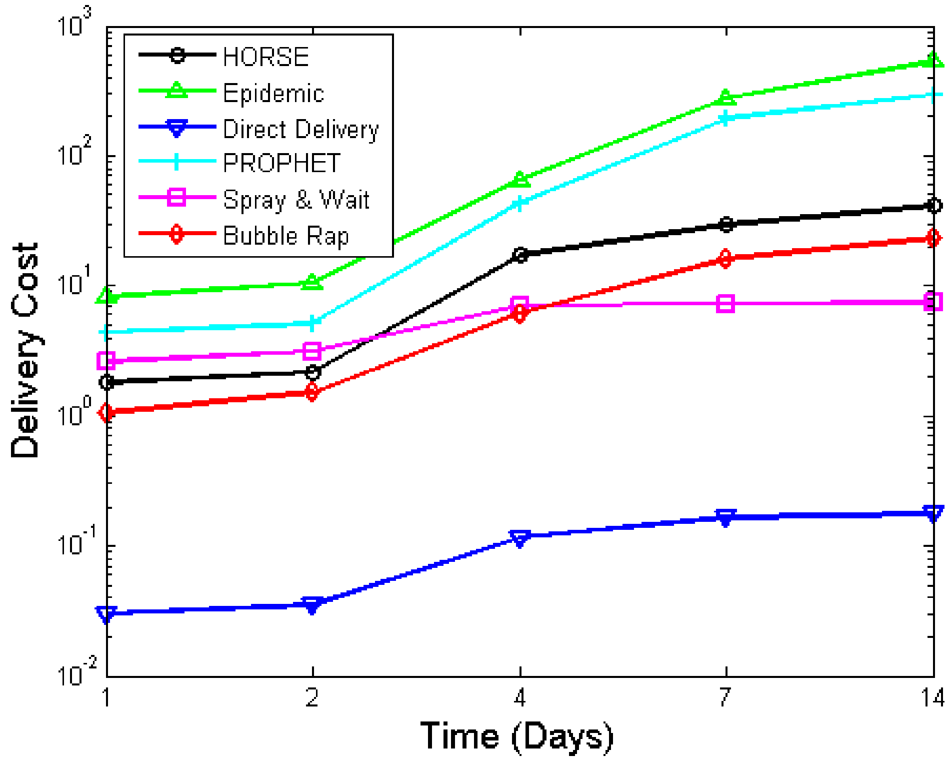


Fig. 16 Delivery cost

Fig. 17 shows the overhead. Obviously, the overhead of Direct Delivery is zero. Again, similar to the delivery cost, PROPHET and Epidemic have the highest overheads. The overhead of HORSE is relatively low. Taking into consideration both delivery ratio and overhead, HORSE is the most efficient approach based on the simulation results. Although the overhead of Bubble Rap and Spray & Wait is lower, their delivery ratio is also lower. Note that the overhead of Spray & Wait decreases slightly over time. This is because the number of messages relayed remains unchanged due to the limited copies involved, but the number of messages delivered to the destination nodes increases over time.

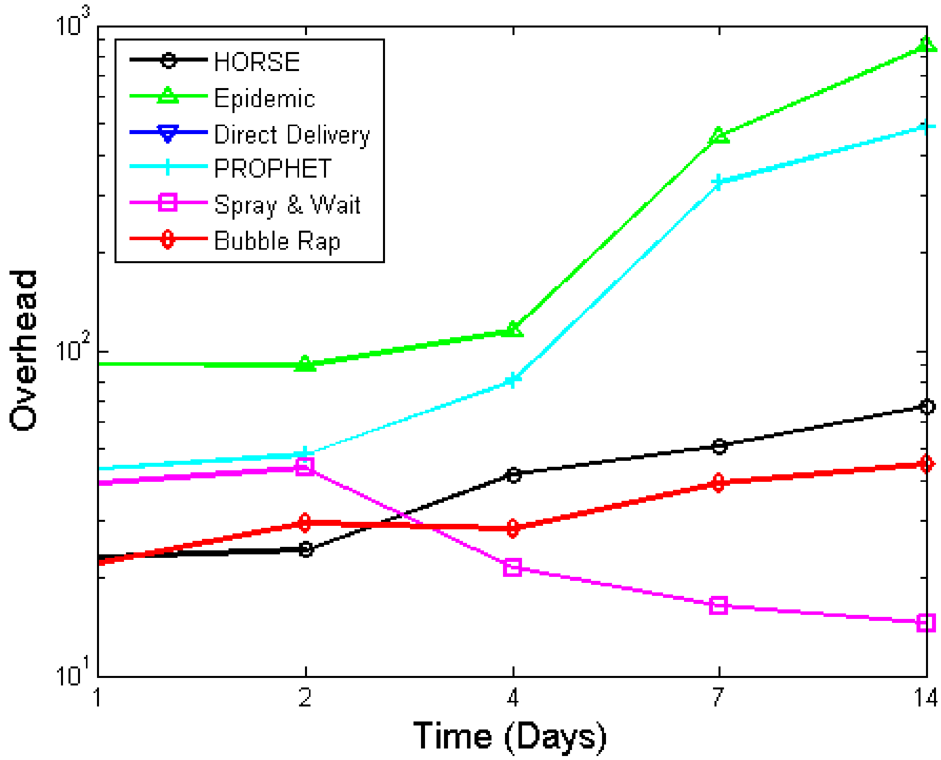


Fig. 17 Overhead

Fig. 18 shows the hop count. Obviously, the hop count of Direct Delivery is always one because the source node will always deliver messages to the destination node directly. Spray & Wait has a low hop count of less than three because after a node transfers all copies to others, the node will wait for an opportunity to transmit the message directly to the destination. Therefore, the number of hops that is used to relay messages is limited. Bubble Rap can achieve a lower hop count than that of Spray & Wait because it selects relay nodes very carefully. Compared with Epidemic and PROPHET, HORSE has a lower hop count of about two to four. For HORSE, messages are mostly sent through a friend chain. If a message is sent through a friend chain, the hop count is two or three. However a message is sometimes relayed to the friend chain through a highly sociable node or an exceptional stranger, the average hop count becomes slightly higher. The hop count of both Epidemic and PROPHET are about four to six. More relay nodes are therefore required to deliver a message. In other words, messages have been sent to more unnecessary nodes.

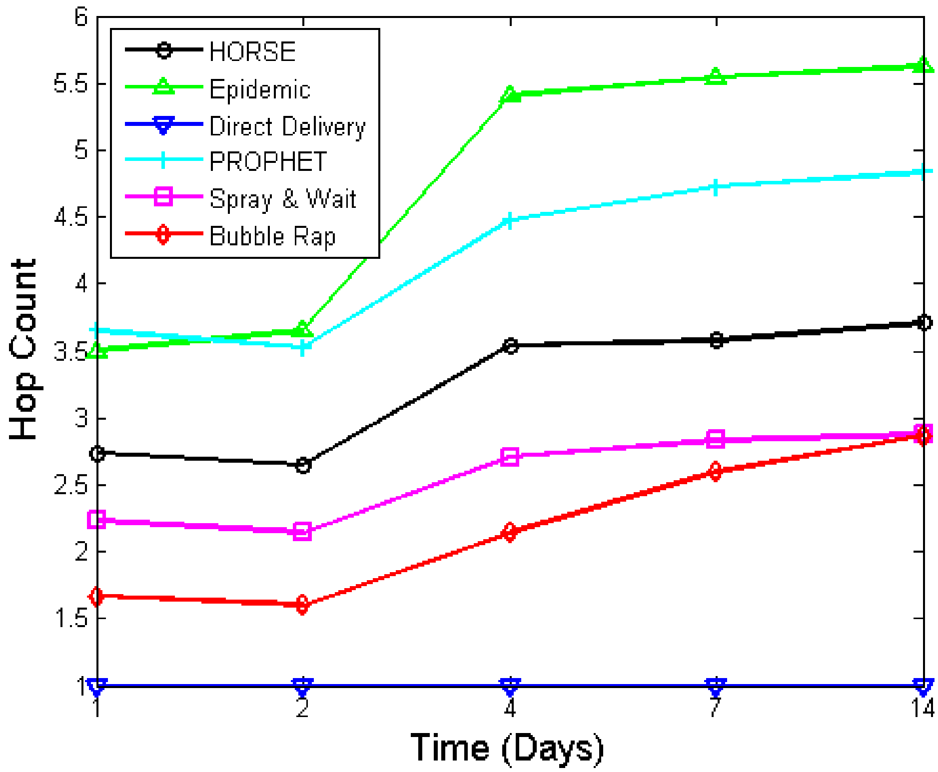


Fig. 18 Hop count

Fig. 19 shows the number of dropped messages. As the buffer size is limited, messages may be dropped due to buffer overflow. Direct Delivery does not have any dropped messages because the buffer size is large enough to hold all messages generated during the simulation. Similarly, Spray & Wait does not have any dropped messages because the buffer size set in the simulation environment is big enough to accept all copies of the message that have been generated by source nodes. Epidemic has about 100,000 dropped messages because nodes try to send all messages to all encountered nodes. PROPHET has about 50,000 dropped messages because nodes send messages to others that are more likely to deliver the messages. Hence, each node may receive a large number of messages. In the case of HORSE and Bubble Rap, the number of dropped messages is less than 1,000 because messages are sent through a more controllable manner (i.e., In HORSE, messages are sent to nodes in the corresponding friend chain, highly sociable nodes, and exceptional strangers).

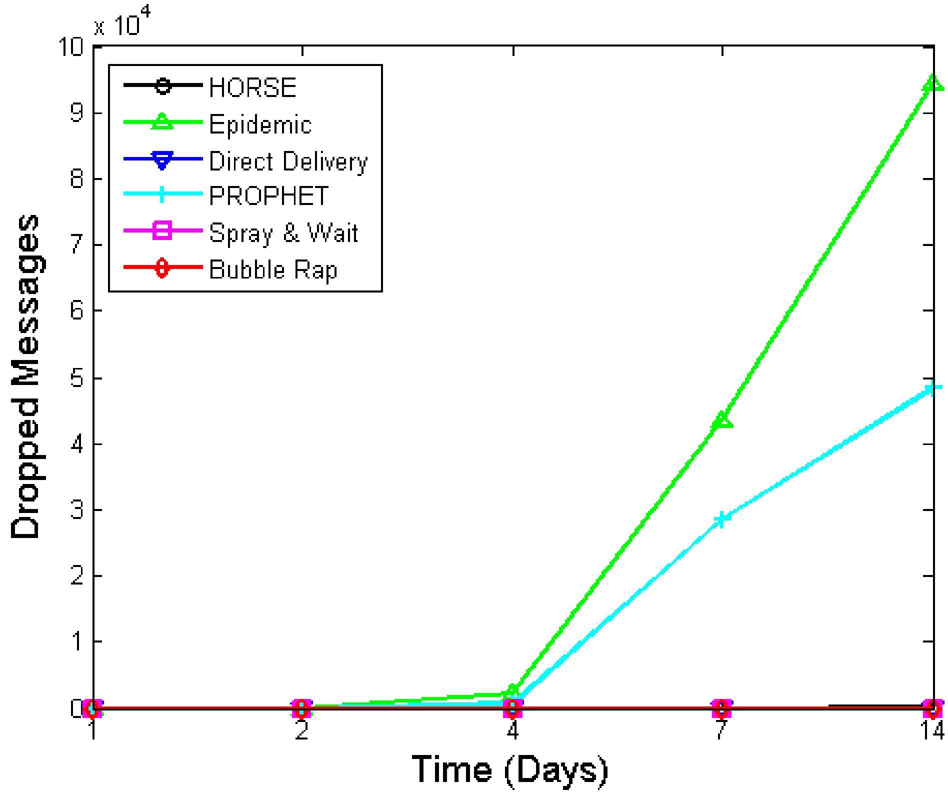


Fig. 19 Number of dropped messages

5 Smart Shopping System

With the popularity of smart phones, mobile marketing is becoming increasingly popular. Mobile marketing provides context-aware information [81] to customers as an effective marketing communication method, thereby generating value for customers and sellers [82]. In this chapter, we present a Smart Shopping System using social vectors for MSNs. The Smart Shopping System is able to sends real-time customized marketing messages (i.e., e-coupons or mobile advertisements of products purchased by similar customers) to customers when they have just entered a shop. There are two issues in the development of the Smart Shopping System. (1) How to identify the customers effectively (e.g. when they have just entered the shop)? (2) How are the customized marketing messages to be generated?

To address the first issue, radio frequency identification (RFID) technology is employed so that when any customers come into a shop, the Smart Shopping System can be notified instantly. RFID technology [83][84] has been widely employed by various industries. It provides indoor location tracking, which the Global Positioning System (GPS) cannot provide [85][86]. It uses radio-frequency electromagnetic fields to retrieve data from an RFID tag that is attached to an object, for the automatic identification and tracking of the object. It is envisioned that in the future any product in a shop can be tracked automatically using RFID tags. In the Smart Shopping System, we use RFID technology to track the location of customers[2]. By equipping each shop with an RFID reader and each customer with an RFID-enabled smart card, customers can be identified by the Smart Shopping System as soon as they enter a shop.

To address the second issue, social vectors are employed to identify similar customers. We will introduce a rule-based approach and a comparison-based approach based on social vectors to find similar customers in following sections. The product recommendations and the customized marketing messages for customers are generated based on the products purchased by most similar customers of the shop. With the ability to determine whether a customer is similar to another customer or a group of customers, the Smart Shopping System can provide product recommendations to customers based on the purchase records of a similar customer or most

[2] I would like to thank Felix Hui for his help on developing the RFID parts of the system.

similar customers. For example, if a customer C_1 is similar to another customer C_2 or to a group of customers G, customer C_1 would be recommended the products that customer C_2 or most customers in the group of customers G had just purchased.

5.1 Rule-Based Approach for Finding Similar Customers

A rule-based approach can be used to find similar customers based on social vectors. Basically, this approach can be divided into two parts. The first part is to classify customers into various types with respect to a shop. The second part is to determine similar customers based on their preferences in each customer type. For purposes of illustration, we define a simple social vector with *m* attributes. $\overrightarrow{V_{S,C}}(1)$ defines the visit duration to a shop S last month. $\overrightarrow{V_{S,C}}(2)$ defines the number of visits to the shop S last month. $\overrightarrow{V_{S,C}}(3)$ defines the number of purchases in the shop S last month. $\overrightarrow{V_{S,C}}(i)$ (for $i > 3$) are Boolean attributes, indicating whether the customer has a preference for any product in the shop S. For example, if a person likes sports, then a_4 is set to 1 for the social vector between the customer and the shop that sells sports products. Otherwise a_4 is set to 0, indicating that the customer does not have such a preference.

Fig. 20 Four types of customers

Every shop can classify its customers into various customer types based on social vectors with customized rules. Based on $\overrightarrow{V_{S,C}}(1)$, $\overrightarrow{V_{S,C}}(2)$, and $\overrightarrow{V_{S,C}}(3)$, customers can first be classified into four types, as shown in Fig. 20. The shop S can define that a loyal customer has $a_1 \geq 10$ hours, $a_2 \geq 10$ times, and $a_3 > 15$ units. A discount customer has $a_1 > 10$ hours, $a_2 > 10$ times, and $a_3 \leq 15$ units. A new customer has $a_1 < 1$ hours, $a_2 < 2$ times, and $a_3 < 3$ units. A general customer is a customer that does not belong to the above three types.

Given a customer C with a social vector $\overrightarrow{V_{S,C}}$ = [15, 20, 30, 1, 0, 0, 1, 0], we can find that customer C is a loyal customer based on above rules. Then, by making use of the XNOR operator of the social vector, we can determine which customers under the same type are more similar to customer C. For example, suppose there are five loyal customers V, W, X, Y, and Z for the shop S. The social vectors between the shop S and these customers are shown as follows.

$$\overrightarrow{V_{S,V}} = [10, 15, 30, 1, 0, 0, 1, 1]$$

$$\overrightarrow{V_{S,W}} = [25, 10, 20, 1, 1, 1, 0, 0]$$

$$\overrightarrow{V_{S,X}} = [15, 18, 30, 1, 1, 0, 1, 0]$$

$$\overrightarrow{V_{S,Y}} = [12, 10, 30, 1, 0, 1, 0, 1]$$

$$\overrightarrow{V_{S,Z}} = [13, 25, 30, 1, 1, 1, 1, 0]$$

By using the XNOR operator, we can find that

$$\overrightarrow{V_{S,C}} \text{ XNOR } \overrightarrow{V_{S,V}} = 4$$

$$\overrightarrow{V_{S,C}} \text{ XNOR } \overrightarrow{V_{S,W}} = 2$$

$$\overrightarrow{V_{S,C}} \text{ XNOR } \overrightarrow{V_{S,X}} = 3$$

$$\overrightarrow{V_{S,C}} \text{ XNOR } \overrightarrow{V_{S,Y}} = 2$$

$$\overrightarrow{V_{S,C}} \text{ XNOR } \overrightarrow{V_{S,Z}} = 3$$

Since $\overrightarrow{V_{S,C}}$ XNOR $\overrightarrow{V_{S,V}}$ gives the highest value, customers C and V are therefore more similar. Note that in the above example, the value of continuous or discrete attributes of social vectors is discarded by setting their corresponding weight to zero so that the large values of these attributes cannot dominate the results of the XNOR operation.

5.2 Comparison-Based Approach for Finding Similar Customers

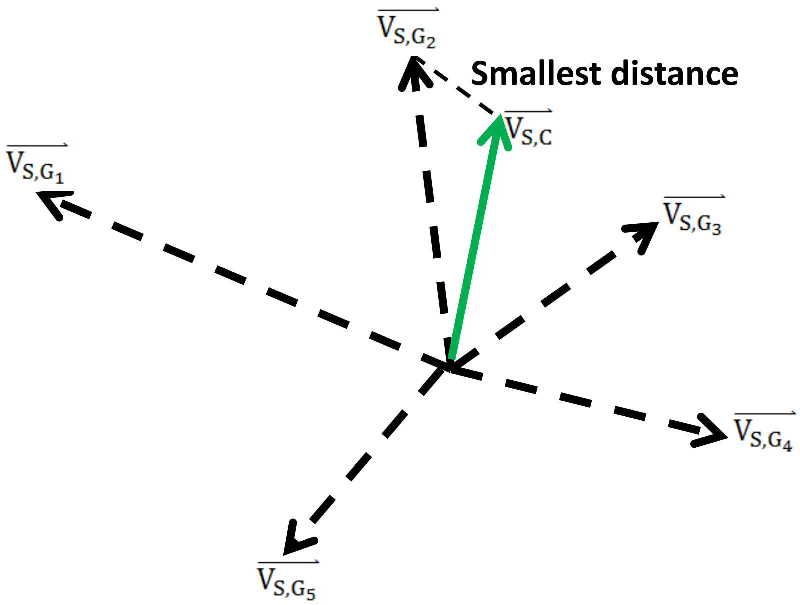


Fig. 21 Finding a group of similar customers using social vectors

Based on their visit duration, number of visits, purchase records, and personal preferences, customers can be categorized into different groups using various clustering algorithms such as the K-Means Clustering Algorithm and the Genetic Algorithm. (Note that in the Smart Shopping System, customers' preferences can be collected in an initial registration procedure through their smart phones.) Each group can be represented by a group social vector. Fig. 21 shows an example of how a group of similar customers is found using social vectors. Given some customers of a shop S, they can be grouped into five groups $\mathrm{G_1}$, $\mathrm{G_2}$, $\mathrm{G_3}$, $\mathrm{G_4}$, and $\mathrm{G_5}$ and are represented as five group social vectors $\overrightarrow{\mathrm{V_{S,G_1}}}$, $\overrightarrow{\mathrm{V_{S,G_2}}}$, $\overrightarrow{\mathrm{V_{S,G_3}}}$, $\overrightarrow{\mathrm{V_{S,G_4}}}$, and $\overrightarrow{\mathrm{V_{S,G_5}}}$. For each group, the group social vector gives the centroid of the community (see (6)). When there is a new social vector $\overrightarrow{\mathrm{V_{S,C}}}$ (i.e., the social vector between the shop S and the customer C), we can determine whether $\overrightarrow{\mathrm{V_{S,C}}}$ is more similar to a group by comparing the distance to the centroid of the two groups. In the example, the new social vector $\overrightarrow{\mathrm{V_{S,C}}}$ should be closer to $\overrightarrow{\mathrm{V_{S,G_2}}}$. In other words, the new customer C is similar to those of group $\mathrm{G_2}$. Note that, in some cases, some attributes may be more important than others. Therefore, a weight vector can be employed to highlight the important attributes (see (10)). For example, a shop that sells sports equipment would probably set a higher weight for the attribute that indicates that the customer like sports than give all attributes the same weight.

5.3 Overview of the Smart Shopping System

The Smart Shopping System provides customers with a smart shopping experience with the help of social vectors and RFID technology. To use the system, customers first need to download a mobile application and pick up an RFID-enabled smart card from a participating shop. They then need to follow a registration procedure, which requires them to input the identity number of their RFID-enabled smart card and their personal preferences to their smart phones. Once the registration is completed, the customized marketing messages will automatically be sent to the customers' smart phones as soon as the customer comes into a participating shop. Real-time reviews from similar customers can also be received. Besides, customers can check recommended products and exchange e-coupons or product information with their friends using NFC technology anytime and anywhere.

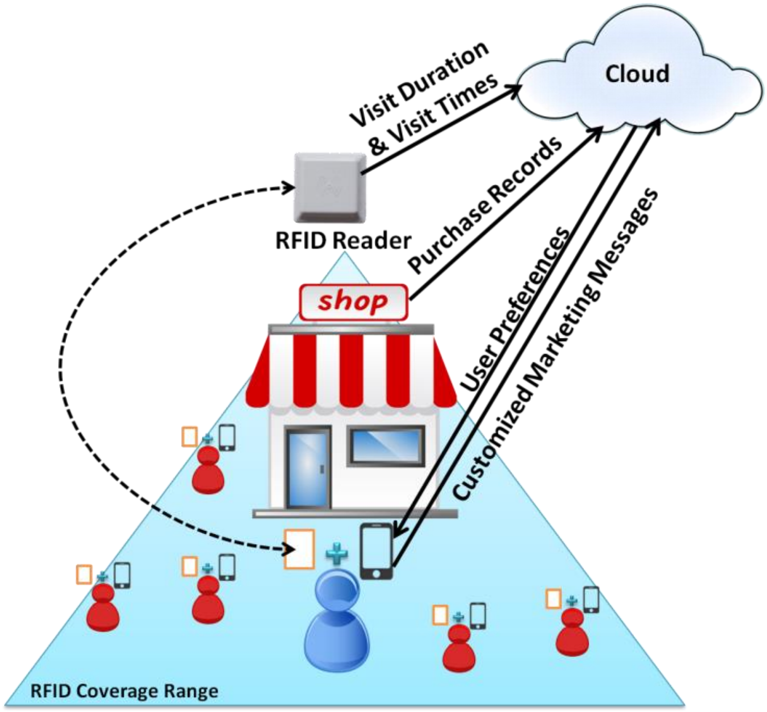


Fig. 22 The architecture of the Smart Shopping System

5.3.1 Architecture

The Smart Shopping System is comprised of several main components: RFID readers, RFID-enabled smart cards, web-based terminals in shops and customers' smart phones with the mobile application of the Smart Shopping System and the cloud server of the Smart Shopping System. The general architecture of the Smart Shopping System is shown in Fig. 22.

RFID Reader and RFID-enabled Smart Card

The RFID reader is able to read data from RFID tags through electromagnetic transmission. The Smart Shopping System employs active RFID technology to track the location of customers (Note that active RFID technology provides a longer coverage range compared with passive RFID technology).

Under the Smart Shopping System, each shop has an RFID reader while each customer has an RFID-enabled smart card (i.e., the smart card is equipped with an RFID tag). Whenever a customer brings that smart card to the shop, the RFID reader can inform the cloud server of the Smart Shopping System and update to the cloud server the visit duration and the number of visits made by that customer.

Smart Phone with Mobile Application

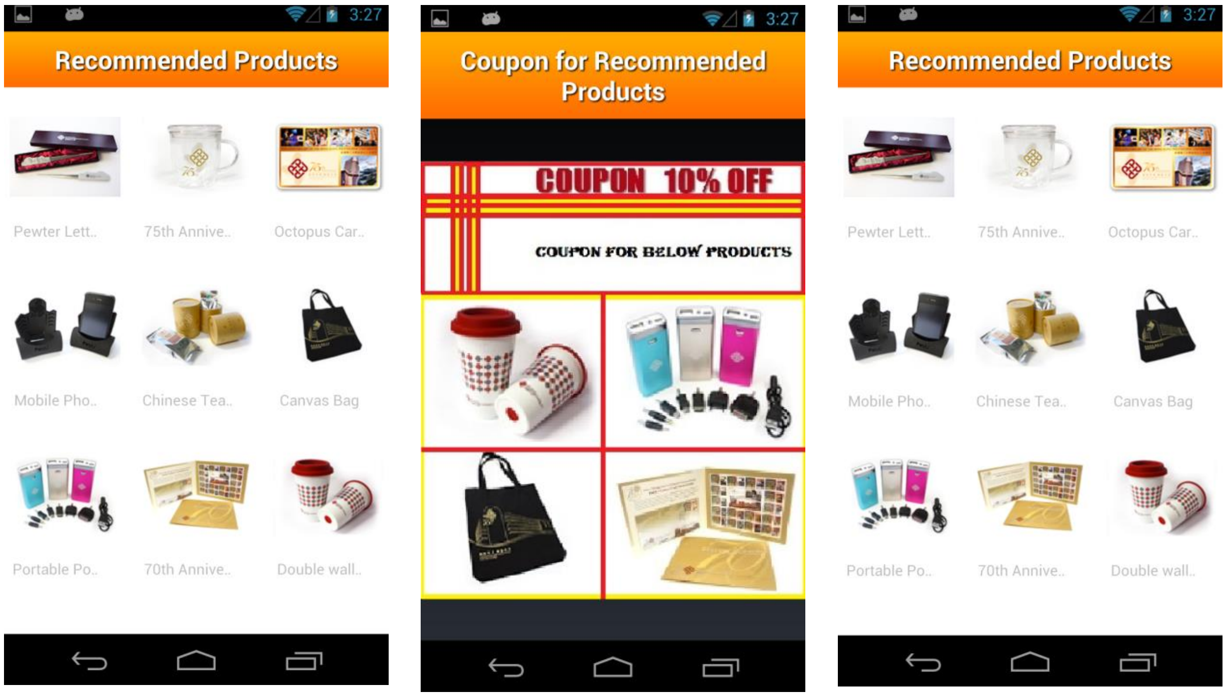
Customers who use the Smart Shopping System need to input the identity number of an RIFD-enabled smart card and their personal preferences (e.g., Sports, Education, Game, Music, Lifestyle, etc.) to the mobile application of the Smart Shopping System in the registration stage. After the registration stage, the mobile application can display customized marketing messages received from the cloud server as soon as the customers enter a participating shop. In addition, through the mobile application, product recommendations can be provided to customers anytime and anywhere.

Web-based Terminal

Each shop needs to provide purchase records of customers to a web-based terminal of the Smart Shopping System. The web-based terminal will then update the customers' purchase records to the cloud server immediately whenever the customers purchase any product. In this thesis, the Smart Shopping System records the total number of purchases and the total value of the purchases for each customer.

Cloud Server

The visit duration and the number of visits of customers with respect to a shop are updated whenever the RFID readers of the shops inform the cloud server. In addition, the customers' purchase records are updated on the cloud server by the web-terminals of the shop, and user registration information is sent from customers' smart phones to the cloud server after the registration stage. Whenever the RFID reader in a shop notifies the cloud server that a customer has entered the shop, the cloud server will send customized marketing messages based on social vectors to that customer's smart phone.



a b c

Fig. 23 Screen captures of the mobile application in the Smart Shopping System

5.3.2 Functionalities

The Smart Shopping System has four main functionalities: 1) recommending products purchased by a similar customer or most similar customers 2) delivering real-time customized marketing messages, 3) providing and receiving real-time product reviews, and 4) exchanging e-coupons or product information.

Recommending Products Purchased by Similar Customers

Based on social vectors, similar customers can easily be identified in the Smart Shopping System. Products purchased from a similar customer or a group of similar customers can be recommended to the customers anytime and anywhere. Fig. 23a shows the screen capture of product recommendations in a customer's smart phone.

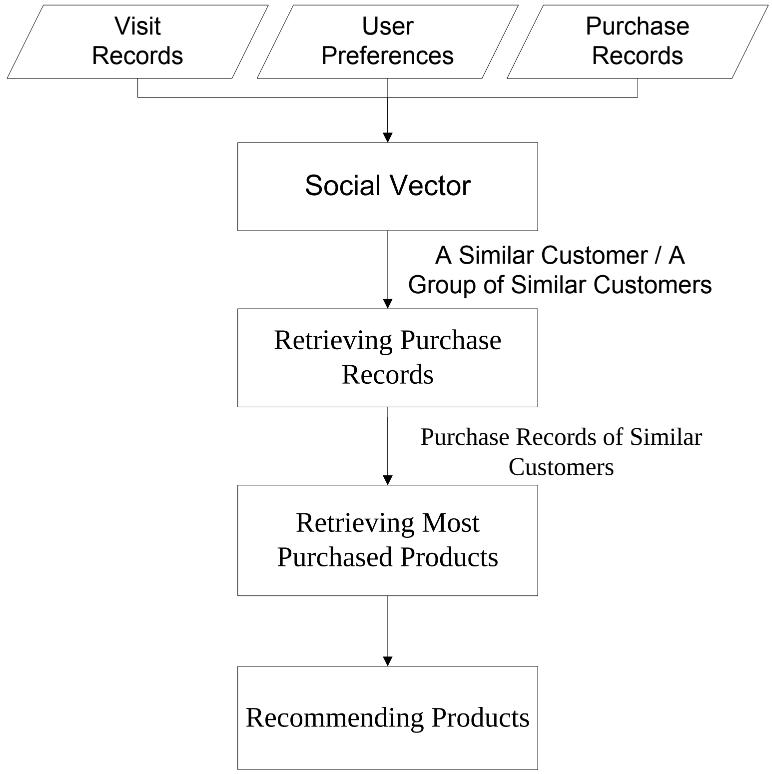


Fig. 24 Flow chart of product recommendations

Delivering Targeted Marketing Messages

Real-time customized marketing messages (i.e., mobile advertisements or e-coupons of recommended products) can be delivered to customers' smart phones to motivate customers to make purchases as soon as they come to a shop or are near the shop. For example, the Smart Shopping System will send an e-coupon to discount customers that other similar discount customers have already purchased, in order to motivate the customers to make a purchase. Fig. 23b shows the screen capture of a real-time marketing message in a customer's smart phone.

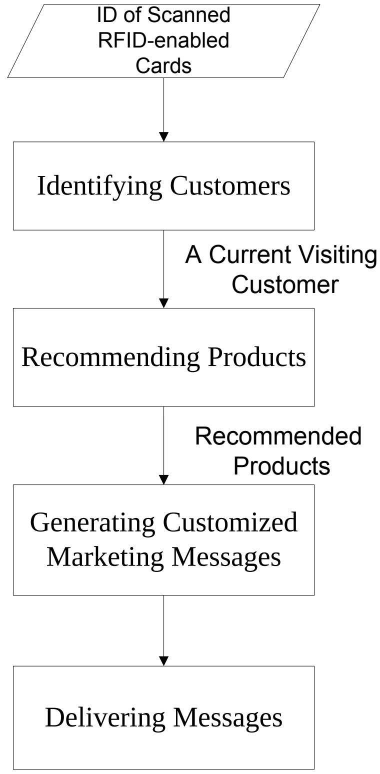


Fig. 25 Flow chart of real-time marketing message delivery

Providing and Receiving Real-Time Reviews

Customers can provide product reviews for a shop so that other customers can receive the product reviews from similar customers who have just visited the shop as soon as they come into that shop. The real-time reviews are a collection of human intelligence to help customers make better purchase decisions. Fig. 23c shows the real-time product reviews in a customer's smart phone.

Exchanging E-Coupons or Product/Service Information

Customers can exchange e-coupons or share product information anytime and anywhere. Whenever a customer gets a new e-coupon or product information from the Smart Shopping System, the customer can share this with their friends by physically tapping the devices together. This interaction is enabled by NFC technology. Unlike other wireless technologies such as Bluetooth or WiFi-Direct, it does not require any manual device discovery or pairing. The e-coupon or product or service information will be automatically exchanged when two devices come into range.

5.3.3 Scenarios Testing

We used an RFID reader, four RFID active tags, and four Android phones to build a prototype of the Smart Shopping System. The RFID reader, the RFID active tags, and the Android phones that we used are shown in Fig. 26. Some scenario tests were executed to test the prototype of the Smart Shopping System.

Fig. 26 RFID reader, RFID tags and Android phones used in the prototype

In our scenarios testing, there are 12 customers A, B, C, D, E, F, G, H, I, and J. The attributes of their social vector are shown in Table 5. Note that customers A, D, G, and J have real Android devices and RFID active tags. Others are virtual customers, who do not have real Android devices and RFID active tags. In general, the results were satisfactory. In particular, most customers were able to receive the real-time customized marketing messages within 30 seconds after entering the coverage area of the RFID reader. Two important scenario tests are described below for checking the correctness of the rule-based social vector and the comparison-based social vector approach, respectively.

Table 5 Attributes of social vectors for the scenario tests

	a1	a2	a3	a4	a5	a6	a7	a8	a9
A	15	15	20	1	0	1	0	1	0
B	10	10	15	1	1	1	0	0	0
C	20	10	18	1	1	1	1	0	1
D	15	15	5	1	1	0	1	1	1
E	10	10	1	0	0	0	1	1	1
F	20	10	8	0	1	0	1	0	1
G	10	15	1	1	0	1	1	1	1
H	5	2	20	1	1	1	1	0	0
I	1	1	0	0	0	0	0	0	1
J	1	1	2	1	0	1	0	0	0

Scenario Test for Verifying the Rule-Based Social Vector

Based on the customized rule we just mentioned, customers can first be divided into four types. Under the testing of the Smart Shopping System, customers A, B, and C can successfully be classified into loyal customers; customers D, E, and F can be classified into discount customers; customers G and H can be classified into general customers; and customers I and J can be classified into new customers.

In addition, customer A can receive a customized marketing message (i.e., the e-coupon of the products purchased by B) instantly when entering the coverage area of the RFID reader. Note that customers A, B, and C are loyal customers but customers A and B are more similar than customers A and C because $\overrightarrow{V_{S,A}}$ XNOR $\overrightarrow{V_{S,B}}$ is larger than $\overrightarrow{V_{S,A}}$ XNOR $\overrightarrow{V_{S,C}}$. Customers D, G, and J can also receive a customized marketing message (the e-coupon/advertisement of the products purchased from the most similar customers under same customer type).

Scenario Test for Verifying the Comparison-Based Social Vector

Based on K Means clustering, customers B, C, D, E, F, G, H, I, and J can be divided into three groups: Group X (comprised of customers B, C, and D), Group Y (comprised of customers E, F and G), and Group Z (comprised of customers H, I and I). Since the social vector of customer A is closest to the social vector of Group X, under the scenario test customer A can receive customized marketing messages (i.e., the e-coupons of the products purchased by customers B, C, and D) when customer A enters the coverage area of the RFID reader.

6 Dynamic Signage System

With the advance in consumer electronics, more sophisticated television or digital display systems have become available, providing many innovative functions. For example, [87] presented a FamTV system, which can provide presence-aware and personalized television services, and [88] presented an interactive television system to enhance user interactivity. Another recent example is Mobiature [89], which allows users to manipulate three-dimensional objects on large displays through their mobile phones. The aforementioned technologies can play an important role in developing advanced or next-generation signage systems. In this chapter, we present a Dynamic Signage System that can support dynamic content generation based on our previous work [90] and a social vector-based dynamic playlist mechanism.

6.1 Overview of Dynamic Signage System

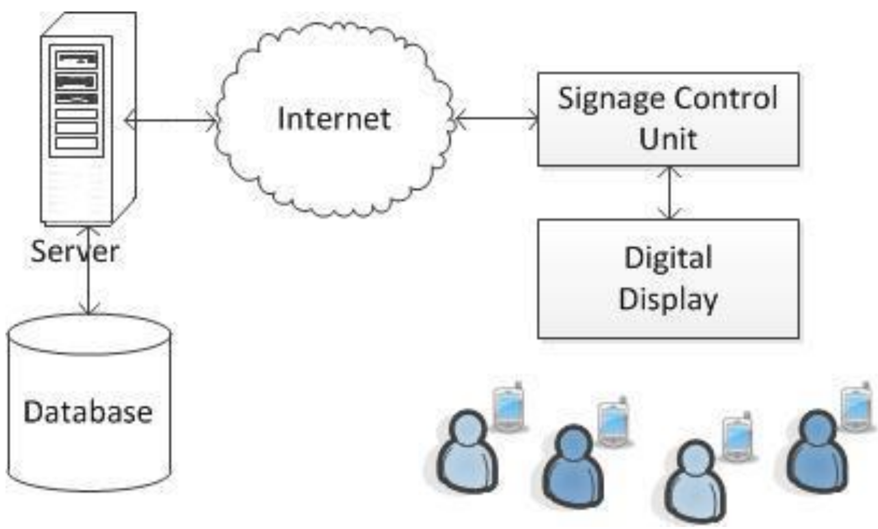


Fig. 27 Basic system architecture of Dynamic Signage System

Fig. 27 shows the basic system architecture, which comprises four major components: signage control unit, digital display, server and database, and mobile phones. The signage control unit provides the major coordination function. The digital display such as a digital television is connected to the signage control unit. Multimedia contents can be obtained either locally or remotely for display onto the digital display. In the next section, we will present a social vector-based dynamic playlist mechanism for generating dynamic contents. The signage control unit and server can be communicated over the Internet so that remote contents can be retrieved. As a future work, a cloud-based content management system can also be developed to facilitate content management.

6.2 Dynamic Playlist Mechanism

Here we present the dynamic playlist mechanism for showing dynamic contents. Basically, contents are displayed based on the interests of users, as well as on how many times the contents have been shown. We assume that the user interest for content C_k can be known (e.g., through a registration system outside the scope of this thesis). Denote $I_{i,j}$ as the interest index for user *i* after seeing C_k for *j* times as follows:

$$\mathrm{I}_{i,j}(C_k) = \begin{cases} j^{-n} & \text{if user } i \text{ is interested in } C_k \\ 0 & \text{otherwise} \end{cases}$$

Note that the interest index is between 0 and 1, where 1 means that a user is fully interested in the content. The above formula means that if a user is interested in C_k, the interest index will depreciate exponentially based on how many times that user sees C_k. As an example, we assume that *n* = 2. Basically when *n* is larger, the interest will depreciate more rapidly. This should be a reasonable assumption because even if a user is interested in the content, he/she will lose the interest after viewing the content for many times. Of course, the above function is just an example, other functions can also be used. On the other hand, if a user is not interested in C_k, the interest index will always stay at zero. With above settings, an *m*-dimensional social vector $\overrightarrow{\mathrm{V}_{C_k,G}}$ = [$\mathrm{I}_{1,j}(C_k)$, $\mathrm{I}_{2,j}(C_k)$,…,$\mathrm{I}_{m,j}(C_k)$] can be defined to quantify the social relationship between content C_k and a group of users *G*, where *k* = 1, 2, …, *m*. Based on *j* and the interest index of all users, the display value for C_k can be found by as follows:

$$Display\,Value = \sum_{i=1}^{m} \overrightarrow{\mathrm{V}_{C_k,G}}\,(i)$$

The signage then selects C_k, with the highest display value for display to the users. To quantify the effectiveness of the dynamic playlist mechanism, we have conducted the following simulations. We assume that a discrete time system is employed and potential users arrive to a signage display according to a Poisson distribution with an arrival rate of λ people per time slot. The signage area can accommodate a maximum of M people. A potential user can join if space is available. There are N content pages to be selected for display. For computing the interest index, each user may be interested in a content page with probability p. At each time slot, a content page is displayed. At the end of a time slot, a user may leave at probability q (i.e., he or she may stay with probability $1 - q$) to view the content page at the next time slot. The objective of the simulation is to compare the average display value per time slot of the dynamic playlist mechanism and of the random display mechanism (i.e., each content may be displayed with probability $1/N$). Note that this is just an example simulation model.

6.3 Simulation Results and Discussion

As an example, unless otherwise specified, we set $\lambda = 10$, $M = 20$, $N = 50$, $p = 0.3$ and $q = 0.4$ in the following simulations. Fig. 28 shows the mean display value when N is varied and $\lambda = 10$. It can be seen that the mean display value for the random selection method is almost constant because according to the aforementioned simulation model, each random content page is of interest to a user with an equal probability. For the dynamic playlist, the mean display value increases slightly as N increases because with more contents, the system can find a better content page that can satisfy the overall interest of the users. The figure also shows that when q is smaller, the mean display value is higher because more people may stand around the signage.

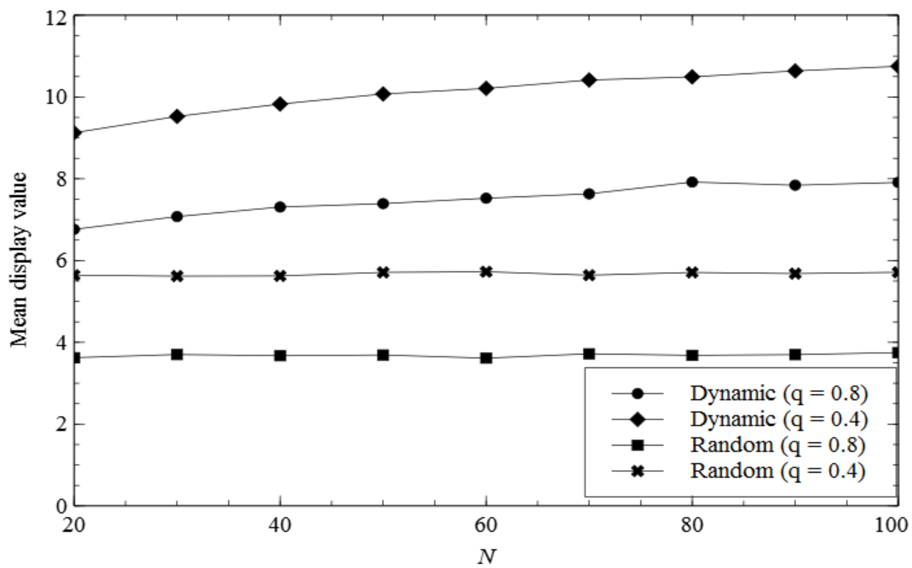


Fig. 28 Mean display value when *N* is varied (λ = 10)

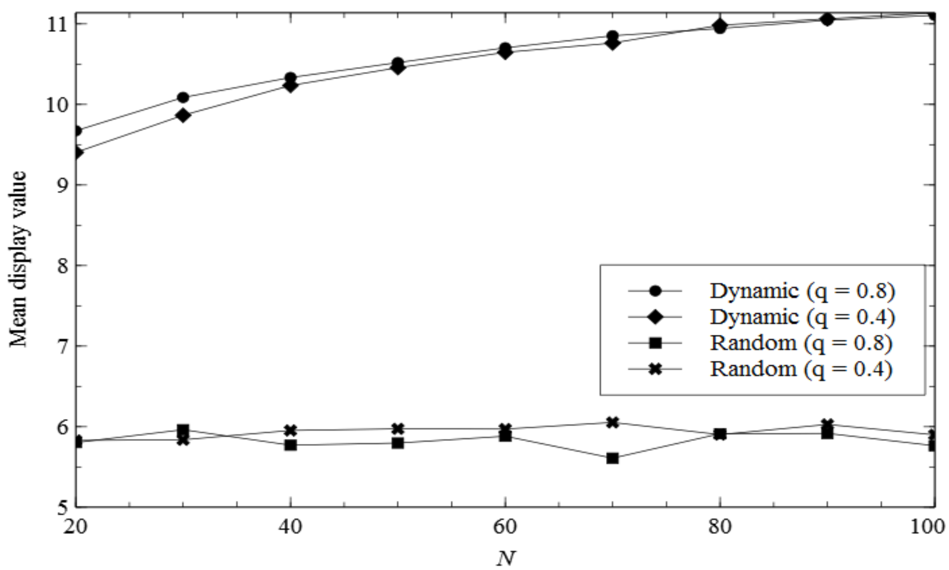


Fig. 29 Average display value when p is varied (λ = 30)

Fig. 29 shows the mean display value when N is varied and λ = 30. In this case, the mean display value is less affected by q due to the high arrival rate (i.e., most of the time, there are M people standing around the signage). Fig. 30 and Fig. 31 show the mean display value when p is varied for λ = 10 and λ = 30, respectively. As expected, the mean display value increases as p increases. Again, the dynamic playlist can outperform the random selection. Similar to above, when λ = 30, the mean display value is almost unaffected by q. In this case, there are M people standing around the signage most of the time. On the other hand, when λ = 10 (see Fig. 30), the mean display value for q = 0.4 is higher than that for q = 0.8 because there are likely more people standing around the signage in the former case. In summary, the above simulation results indicate that the dynamic playlist mechanism can outperform the random selection method in general as expected. In particular, it is more attractive to use the dynamic playlist mechanism for certain situations (e.g., when N is high and p is moderate).

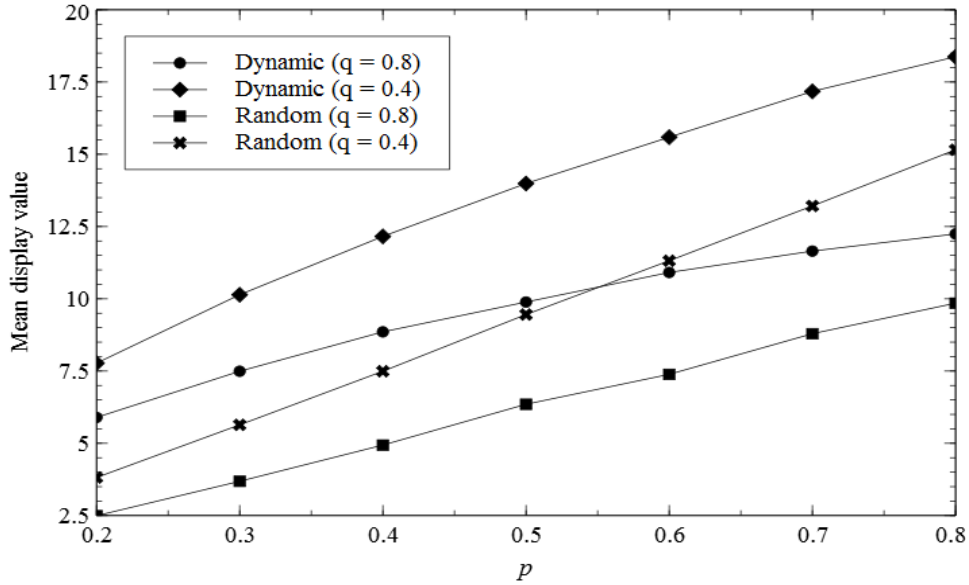


Fig. 30 Mean display value when p is varied (λ = 10)

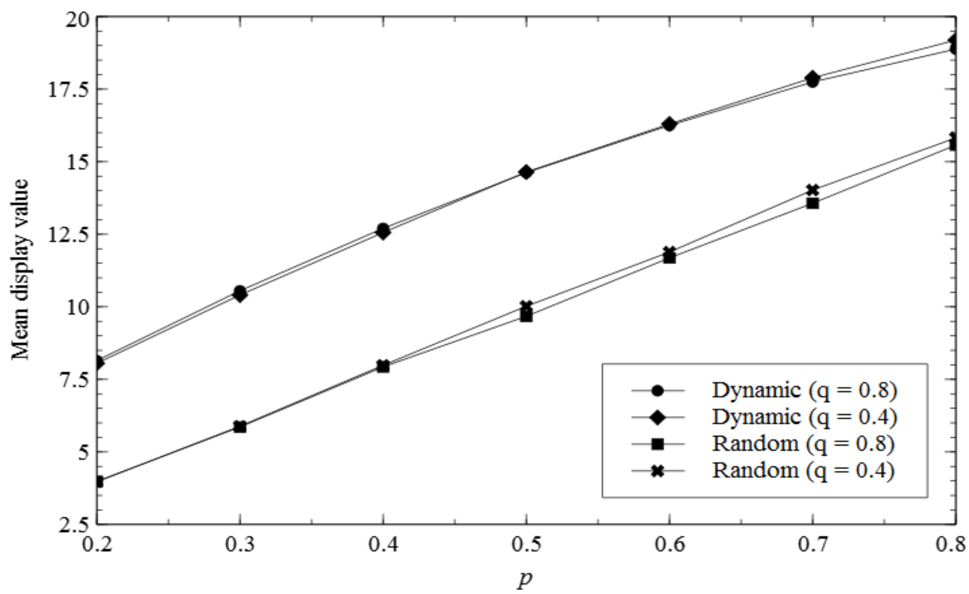


Fig. 31 Mean display value when p is varied (λ = 30)

7 Conclusions and Future Work

In this thesis, we first proposed a social network framework based on a new concept called social vectors. Social vectors can be used to define and quantify social relationships between social entities (e.g., two people, and a person and a set of places etc.). In some cases, some attributes may be more important than others. A weight vector can therefore be employed to highlight the important attributes. This new concept can open up some interesting areas of research such as social vector algebra and social vector calculus. Social vectors can be employed to facilitate the development of social network protocols and social computing systems in general. For future work, we plan to develop an application programming interface (API) for social vectors so as to facilitate researchers to build MSN systems and applications using social vectors. Besides, we also plan to investigate a social network framework from the social scientist's point of view, simulating the social network in the real world more accurately.

Based on social vectors, we then proposed an opportunistic routing protocol called HORSE for forwarding messages between social entities over MSNs. HORSE seeks to combine the advantages of the previous opportunistic routing protocols. We have evaluated the performance of the proposed HORSE protocol based on simulations using the Opportunistic Networking Environment (ONE) simulator [77]. Simulation results using real-world data traces are employed to evaluate the performance of HORSE in comparison with other schemes. The performance of these schemes is evaluated based on five common performance indicators: delivery ratio, delivery cost, overhead, hop count, and number of dropped messages. Simulation results show that by combining these advantages, HORSE can achieve a high delivery ratio while maintaining a low overhead cost. As an extension, HORSE can be used to forward messages based on social relationships between persons and places as defined by social vectors. For example, if X wants to send a message to Z and both Y and Z visit a place P frequently, X will forward the message to Y when X meets Y. The rationale is that Y will likely meet Z or friends of Z at place P in the near future. For future work, we plan to extend HORSE to support data dissemination services (i.e., one-to-many and many-to-many communications).

For illustration purposes, we have presented two social vector-based applications. The first one is the Smart Shopping System. It first identifies customers by scanning the RFID-enabled smart cards available in the coverage range of the RFID reader. Then it finds similar customers by using social vectors with the purpose of sending customized marketing messages when they enter a shop or are close to a shop. For future work, the Smart Shopping System may provide a cross-shops marketing service to customers. For example, it may offer the e-coupons of restaurants or cafes to customers when it thinks that the customers are getting tired of shopping, based on social vectors.

The second application is the Dynamic Signage System. It employs a social vector-based dynamic playlist mechanism to display contents dynamically based on user interests. Simulation results are presented to show its effectiveness. Currently, we use social vector to represent the social relationships between a person and a set of content. In the future, we can investigate how to use social vector to represent the social relationships between a community and a set of content so that better performance of the system can be provided. Besides, the proposed system can also be extended or enhanced in many other ways. For example, it can be integrated with the Smart Shopping System so that customized marketing messages can be displayed to a user on a signage whenever a user enters a shop. Furthermore, it can also work with image or video processing technologies to provide more intelligent services. For example, by equipping with a video camera, the facial expression of a user can possibly be recognized so that more effective contents can be displayed.

8 References

[1] K. Xu, J. Li, S and S. Liao, “Sentiment community detection in social networks,” *in Proc. iConference*, pp. 804-805, 2011.

[2] R. Chen, E. K. Lua, and Z. Cai, “Bring order to online social networks,” *in Proc. INFOCOM 2011 (Mini-Conference)*, pp. 541-545, 2011.

[3] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks,” *in Proc. KDD 2012*, pp. 1267-1275, 2012.

[4] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, “Over-exposed?: privacy patterns and considerations in online and mobile photo sharing,” *in Proc. ACM SIGCHI*, pp. 357-366, 2007.

[5] X. Cheng and J. Liu, “NetTube: Exploring social networks for peer-to-oeer short video sharing,” *in Proc. IEEE INFOCOM*, pp. 1152-1160, 2009.

[6] C. Tong, “Analysis of some popular mobile social network system,” Helsinki University of Technology, 2008.

[7] B. Han and A. Srinivasan, “Your friends have more friends than you do: identifying influential mobile users through random walks,” *in Proc. MobiHoc 2012*, pp. 5-14, 2012.

[8] S. Gaito, G. Grossi, and F. Pedersini, "A two-level social mobility model for trace generation," *in Proc. MobiHoc 2008*, pp .457-458, 2008.

[9] J. H. Kim et al., “MobiSNA: a mobile video social network application,” *in Proc. MobiDE 2009*, pp. 53-56, 2009.

[10] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge Univ. Press, pp. 505–555, 1994.

[11] Ogatha, “Computer supported social networking for augmenting cooperation,” *Computer Supported Cooperative Work*, vol. 10, pp. 189–209, 2001.

[12] L. Backstrom, D. Huttenlocher, and J. Kleinberg, “Group formation in large social networks: Membership, growth, and evolution,” *in Proc. Knowledge Discovery Data Mining*, pp. 44–54, 2006.

[13] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, “Natural communities in large linked networks,” *in Proc. Knowledge Discovery Data Mining*, pp. 541–546, 2003.

[14] M. E. J. Newman, “Modularity and community structure in networks,” *in Proc. Nat. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[15] C. A. Hidalgo and C. Rodriguez-Sickert, “The dynamics of a mobile phone network,” *Phys. A*, vol. 387, no. 12, pp. 3017–3024, May 2008.

[16] W. Teng andM. Chou, “Mining communities of acquainted mobile users on call detail records,” *in Proc. 22nd Annu. ACM Symp. Appl. Comput.*, pp. 957–958, 2007.

[17] G. Palla, A. Barabasi, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, Apr. 2007.

[18] J. P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A. L. Barabasi, “Structure and tie strengths in mobile communication networks,” *in Proc. National Academy of Sciences*, vol. 104, no. 18, pp. 7332–7336, May 2007.

[19] M. Kurucz, A. Benczur, K. Csalogany, and L. Lukacs, “Spectral clustering in telephone call graphs,” *in Proc. Joint 9th WEBKDD and 1st SNA-KDD Workshop*, pp. 82–91, 2007.

[20] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos, “Using ghost edges for classification in sparsely labeled networks,” *in Proc. 14th ACM SIGKDD Conf.*, pp. 256–264, 2008.

[21] L. Wang, Y. Jia, and W. Han, “Instant message clustering based on extended vector space model,” *Adv. Comput. Intell.*, vol. 4683, *Springer-Verlag Lecture Notes Computer Science*, pp. 435–443, 2007.

[22] P. Koutsourelakis, “Unsupervised group discovery and link prediction in relational datasets: A nonparametric Bayesian approach,” Lawrence Livermore Nat. Lab. (LLNL), Livermore, CA, Tech. Rep. UCRL-TR-230743, 2007.

[23] N. Eagle, A. Pentland, and D. Lazer, “Inferring social network structure using mobile phone data,” *in Proc. National Academy of Sciences*, vol. 106, no. 36, pp. 15274–15278, Sep. 2009.

[24] N. Eagle and A. Pentland, “Eigenbehaviors: Identifying structure in routine,” Behavioral Ecology and Sociobiology, vol. 63, no. 7, pp. 1057–1066, May 2009.

[25] N. Eagle and A. Pentland, “Reality mining: Sensing complex social systems,” Pers. Ubiquitous Comput., vol. 10, no. 4, pp. 255–268, May 2006.

[26] J. Lawrence, T. R. Payne, and D. De Roure, “Co-presence communities: Using pervasive computing to support weak social networks,” *in Proc.WETICE*, pp. 149–156, 2006.

[27] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” *in Proc. MobiArch*, Kyoto, Japan, 2007.

[28] T. A. Dang and E. Viennet, “Community Detection based on Structural and Attribute Similarities,” *in Proc. Digital Society (ICDS)*, pp. 7–12, 2012.

[29] P. Hui, J. Crowcroft, and E. Yoneki “BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks,” *in Proc. ACM MobiHoc*, Hong Kong, pp. 241-250, May 2008.

[30] E. Yoneki, P. Hui, S. Chan and J. Crowcroft "A Socio-Aware Overlay for Publish/Subscribe Communication in Delay Tolerant Networks," *in Proc. MSWiM*, pp. 225-234, Oct. 2007.

[31] Amin Vahdat and David Becker, “Epidemic routing for partially connected ad hoc networks,” *Technical Report CS-200006*, Duke University, Apr. 2000.

[32] T. Spyropoulos, K. Psounis et al., “Spray and wait: An efficient routing scheme for intermittently connected mobile networks,” *in Proc. WDTN*, 2005.

[33] K.-I. Kawarabayashi, F. Nazir, and H. Prendinger, “Message duplication reduction in dense mobile social networks,” *in Proc. ICCCN*, pp. 1–6, Aug. 2010.

[34] A. Lindgren et al., “Probabilistic routing in intermittently connected networks,” *in Proc. SAPIR*, 2004.

[35] E. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” *in Proc. MobiHoc*, pp. 32-40, 2007.

[36] R. Cabaniss, S. Madria, G. Rush, A. Trotta, and S. S. Vulli, “Dynamic social grouping based routing in a mobile ad-hoc network,” *in Proc. MDM*, pp. 295–296, May 2010.

[37] B. R. Karki, A. Hamalainen, and J. Porras, “Social networking on mobile environment,” *in Proc. Middleware*, pp. 93–94, 2008.

[38] A.-K. Pietil¨ainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “Mobiclique: middleware for mobile social networking,” *in Proc. ACM WOSN*, pp. 49-54, 2009.

[39] M. Chuah and P. Yang, “Cooperative User Centric Information Dissemination in Human Contact-Based Networks,” *in Proc. IEEE ICPADS*, pp. 794 – 799, 2010.

[40] J. Rana, J. Kristiansson, J. Hallberg and K. Synnes, “Challenges for Mobile Social Networking Applications, Communications Infrastructure, Systems and Applications in Europe,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 16, 2009.

[41] M. Conti, M. Mordacchini and A. Passarella, “Data Dissemination in Opportunistic Networks using Cognitive Heuristics,” *in Proc. IEEE WoWMoM*, 2011.

[42] C. Boldrini, M. Conti, and A. Passarella, “Contentplace: Social-aware data dissemination in opportunistic networks,” *in Proc. MSWiM*, pp. 203-210, 2008.

[43] D. Görgen, H. Frey and C. Hiedels, “JANE-The Java ad hoc network development environment”, *in Proc. IEEE ANSS*, pp. 163-176, 2007.

[44] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," *in Proc. Simutools*, Mar. 2009.

[45] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *ACM Journal Personal and Ubiquitous Computing*, v. 10, no. 4, pp. 255–268, May 2006.

[46] J. Su et al, “Haggle: Seamless networking for mobile applications,” *in Proc. Ubiquitous Computing*, 2007.

[47] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, “User mobility for opportunistic ad-hoc networking,” *in Proc. Mobile Computing Systems and Applications*, 2004.

[48] C. Chiang and M. Gerla, “On-demand multicast in mobile wireless networks,” *in Proc. IEEE ICNP*, pp. 262-270, 1998.

[49] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” *Mobile Computing*, Kluwer Academic Publishers, pp. 153–181, 1996.

[50] A. Keranen and J. Ott, “Increasing reality for DTN protocol simulations,” *Tech. rep.*, Helsinki University of Technology, Networking Laboratory, July 2007.

[51] F. Ekman, A. Ker¨anen, J. Karvo, and J. Ott, “Working day movement model,” *in Proc. ACM Sigmobile*, pp. 33–40, 2008.

[52] M. Musolesi and C. Mascolo, “Designing mobility models based on social network theory,” *ACM SIGMOBILE Mob. Comp. Commun. Rev.*, vol. 11, pp. 59-70, 2007.

[53] C. Boldrini and A. Passarella, “HCMM: Modeling spatial and temporal properties of human mobility driven by users’ social relationships,” *Computer Commun.*, vol. 33, pp. 1056–1074, June 2010.

[54] N. Vallina-Rodriguez, P. Hui, and J. Crowcroft, “Has anyone seen my Goose? – Social network services in developing regions,” *in Proc. IEEE CSE*, pp. 1048 – 1053, Aug. 2009.

[55] B. Chelly and N. Malouch, “Movement and connectivity algorithms for location-based mobile social networks,” *In Proc. IEEE WIMOB*, Oct. 2008, pp. 190–195.

[56] S. M. Allen, G. Colombo, and R. M. Whitaker, “Uttering: social microblogging without the internet,” *in Proc. ACM MobiOpp*, pp. 58-64, 2010.

[57] M. Motani, V. Srinivasan, and P. S. Nuggehalli, “Peoplenet: engineering a wire-less virtual social network,” *in Proc. ACM MobiCom*, pp. 243-257, 2005.

[58] N. Eagle and A. Pentland, “Social serendipity: Mobilizing social software,” *Pervasive Computing*, v. 4, no. 2, 2005.

[59] SOCIALNETS Workshop. wireless market report and exploitation assessment. [Online]. Available: http://www.social-nets.eu/deliverables_pdfs/D5.4.pdf

[60] H. Zhao, M. Pan, X. Liu, X. Li and Y. Fang, "Optimal resource rental planning for elastic applications in cloud market," *in Proc. IEEE IPDPS*, pp. 808-819, 2012.

[61] E. Fehr and B. Rockenbach, “Detrimental effects of sanctions on human Altruism,” *Microeconomics 0305007*, EconWPA, May 2003.

[62] E. Fehr and U. Fischbacher, “The nature of human altruism,” *Nature*, vol. 425, pp. 785-791, Oct. 2003.

[63] L. Buttyan and J. P. Hubaux, “Enforcing service availability in mobile ad-hoc WANs,” *in Proc. IEEE MoBiHoc*, pp. 87–96, 2000.

[64] T. Ning, Z. Yang, X. Xie, and H. Wu, “Incentive-aware data dissemination in delay-tolerant mobile networks,” *in Proc. SECON*, pp. 539 – 547, 2011.

[65] S. Zhong, J. Chen, and Y. R. Yang, “Sprite: a simple, cheatproof, credit-based system for mobile ad hoc networks,” *in Proc. IEEE INFOCOM*, vol. 3, pp. 1987-97, Apr. 2003.

[66] L. Buttyan, L. Dora, M. Felegyhazi, and I. Vajda, “Barter trade improves message delivery in opportunistic networks,” *ACM Journal Ad Hoc Networks*, vol. 8, no. 1, pp. 1-14, Jan. 2010.

[67] L. Peterson and T. Roscoe, “The design principles of planetlab," *ACM SIGOPS Operating Systems Review*, v. 40, pp. 11-16, 2006.

[68] K. Balakrishnan, J. Deng, and P. K. Varshney, “Twoack: preventing selfishness in mobile ad hoc networks,” *in Proc. IEEE WCNC*, pp. 2137-2142, 2005.

[69] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *in Proc. ACM MobiCom*, pp. 255-265, 2000.

[70] S. Madhavi and T. H. Kim, "An intrusion detection system in mobile ad hoc networks," International Journal of Security and its Application, vol. 2, no. 3, pp. 1-16, 2008.

[71] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," *in Proc. INFOCOM*, pp. 857-865, 2010.

[72] P.-A. Chen and D. Kempe, "Altruism, selfishness, and spite in traffic routing," *in Proc. ACM EC*, pp. 140–149, 2008.

[73] P. Hui, K. Xu, V. Li, J. Crowcroft, V. Latora, and P. Lio, "Selshness, altruism and message spreading in mobile social networks," *in Proc. IEEE INFOCOM*, pp. 284-289, Apr. 2009.

[74] E. Jaho, M. Karaliopoulos, and I. Stavrakakis, "Iscode: a framework for interest similarity-based community detection in social networks," *in Proc. IEEE INFOCOM (Workshop on NetSciCom)*, pp. 551-555, 2011.

[75] H. Gao, G. Barbier, and R. Goolsby, "Harnessing the crowdsourcing power of social media for disaster relief," *IEEE Intelligent Systems*, vol. 26, no. 3, pp. 10–14, 2011.

[76] A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36-44, 2012.

[77] A. Keranen, J. Ott and T. Karkkainen, "The ONE simulator for DTN. protocol evaluation," *in Proc. SIMUTools*, Rome, 2009.

[78] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, May 2006.

[79] A. Zubow, M. Kurth, J.P. Redlich, "Opportunistic protocols in multirate environments," *in Proc. SENSORCOMM*, pp.743–751, 2008.

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

[81] K. Heikki and L. Matti, "Factors influencing consumers' willingness to accept mobile advertising: a conceptual model," *Int. J Mobile Communications*, vol. 3, no. 3, p. 198, 2005.

[82] M. Leppäniemi, "Mobile marketing communications in consumer markets," Faculty of Economics and Business Administration, Department of Marketing, University of Oulu, p. 21, 2008.

[83] M. Chiesa, R. Genz, F. Heubler, K. Mingo, and C. Noessel, RFID, (March 2002). [Online]. Available: http://people.interactionivrea.it/c.noessel/RFID/research.html

[84] S. B. Miles, *RFID Technology and Applications*, Cambridge University Press, London, 2011.

[85] J. Hightower, R. Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," *Univ. Washington, Seattle, Tech. Rep*. UW CSE 2000–02-02, Feb. 2000.

[86] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Network*, vol. 10, no. 6, pp. 701–710, 2004.

[87] P.A. Cabarcos, et al., "FamTV: An architecture for presence-aware personalized television," *IEEE Trans. on Consumer Electronics*, vol. 57, no. 1, pp. 6-13, Feb. 2011.

[88] L.C. Miranda, H.H. Hornung and M.C.C. Baranauskas, "Adjustable Interactive Rings for iDTV," *IEEE Trans. on Consumer Electronics*, vol. 56, no. 3, pp. 1988–1996, Aug. 2010.

[89] D. Yoon, et al., "Mobiature: 3D model manipulation technique for large displays using mobile devices," *in Proc. IEEE Consumer Electronics*, Jan. 2011.

[90] Kong H. K. Choi, Terry H. S. Chu, and Henry C. B. Chan, "Dynamic and Interactive Intelligent Signage System," *in Proc. IEEE International Conference on Consumer Electronics*, pp.672-673, 2012.