Mobile Social Networks: Design Requirements, Architecture and State of the Art

Zhifei Mao[a], Yuming Jiang[a], Geyong Min[b], Supeng Leng[c], Xiaolong Jin[d]

[a] *Norwegian University of Science and Technology, Trondheim, Norway*
[b] *University of Exeter, Exeter, UK*
[c] *University of Electronic Science and Technology of China Chengdu, China*
[d] *Institute of Computing Technology of the Chinese Academy of Sciences, Beijing, China*

Abstract

Mobile social networks are new infrastructures for people to share content, communicate and interact anywhere, anytime. It brings together social computing and mobile networking techniques, and thus yields a plethora of research problems in community detection, information dissemination, privacy preservation and so forth. In this paper, we summarize the characteristics of MSNs, and outline several requirements that should be taken into consideration during the design of MSNs. A functional reference architecture of MSNs is designed, and its building blocks are described. We classify the literatures falling under MSNs according to the building blocks of the architecture, and give a detailed survey for each. At last, a conclusion on the future development of MSNs is presented.

Keywords: mobile networks, social networks, network architecture, survey

1. Introduction

The explosive evolution of information and communication technologies enables online social networking (OSN) systems such as Facebook, Twitter, LinkedIn, etc, to connect active users of similar interests, conversing with one another and forming virtual communities. Nowadays, social networks are increasingly accessed via mobile devices thus rendering a new research field of mobile social networks (MSNs). A few interesting research and development results about MSNs have been reported in the literature. The majority of them, however, focus on user applications running on mobile devices and pay little attention to the underlying mobile communication networks. These work concerns the way that the specific features of mobile phones can be utilized to augment the OSN systems from traditional stationary PCs to mobile devices. Moreover, the data collected by mobile phones and then mined by data-mining or pattern recognition engines are also used to benefit the further development of OSN systems. Nevertheless, in these cases, the wireless mobile networks on which these mobile devices operate are not taken into consideration. Since the underlying wireless networks play an equally important role in the success of OSN

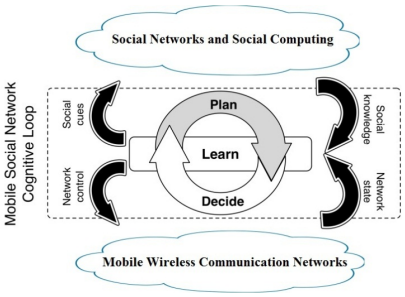

Figure 1: Two Dimensions of Management of Mobile Social Networks.

systems, MSNs are regarded in this proposal as a marriage of the traditional wired-network-based social networks, e.g., Facebook, with mobile wireless communication networks.

Taken together, the two interactive aspects, network-aware social computing and socially-aware mobile wireless communication networks, create a so-called cognitive loop, as illustrated in Figure 1, which means that the networks become self-aware of themselves and surroundings and, accordingly, are able to learn, decide and act based on particular high-level goals. Well-known cognition techniques (e.g., machine learning and game theory) can be used to control this loop. In conventional cognitive loops, such as in cognitive radio networks (CRNs), the operating domain of the loop is in a single contextual environment, e.g., the electromagnetic spectrum. This means that only a single set of sensing parameters and actuators are necessary. In the CLIMBER project[1], we envisage the creation of a dual-domain cognitive system wherein a learning system incorporates sensing (measuring) information from both a social user context and the communication network state experienced by the user device. This information is fed into the learning system that processes it, providing as output changes in the network control parameters and indicators or cues for social interaction.

In MSNs, users are not only able to use existing OSNs via mobile devices, but also able to have new social services enabled by the powerful communication and sensing capabilities of the smart devices. In fact, the convergence of social computing and smart devices is indeed happening. There have been several initiatives (e.g., Foursquare) focusing on enriching users social interaction by leveraging their physical location information on their smart devices. However, the convergence of social networking and mobile computing can proceed much further beyond this. First, the context information can be utilized is no limited to locations, since the smart devices are equipped with multiple sensing capabilities other than positioning. And in the short future, wearable devices (e.g., Google glasses, iWatch, and Nike Fuelband) and smart devices on vehicles are bound to gain massive popularity, which are new network infrastructures and

[1] www.fp7-climber.eu

rich sources of context information as well. Second, most smart devices have wireless interfaces (e.g., bluetooth, WiFi) for local short-range communication, which are rarely explored in reality. However, the need for sharing information among people within vicinity offers a unique opportunity to explore this local communication paradigm. People can exchange information whenever their devices are within each other's transmission range, in this way, they form a delay tolerant network [1], where the connectivity between users is intermittent mainly due to their mobility and limited transmission range. Interesting but useful services can be designed under the local communication paradigm, such as proximity-based matching.

As we described, MSNs have the potential to make social networking pervasive, anywhere and anytime. However, to realize this vision, a great deal of research work has to be done. First of all, in the delay tolerant setting, due to the intermittent connectivity, it is impossible for users to maintain end-to-end communication using current TCP/IP paradigm, they have to take opportunistic encounters to transmit information. This poses a great challenge for routing mechanism since the encounters are not fully predictable. Studies have shown that the underlying social relationships among people are much stabler than temporal topology, and people's mobility also follows certain distribution and thus it is predictable to some extent. These findings have been utilized to design routing protocols, but the results are far from satisfactory, they either have low delivery ratio or have large communication overhead. Another major issue in MSNs is users' privacy and security concern. The lack of central infrastructure makes most existing schemes infeasible. Besides, other issues also call for close attention, such as the integration of delay tolerant with the Internet, battery power conserving for context-aware computing and name service for both device and information.

The remainder of this paper is structured as follows. In Section 2, we provide analysis on the system requirements of MSNs and present a case study to illustrate. In Section 3, a detailed overall system architecture for MSNs is introduced. In Section 4, we provide a state-of-the-art study for each of the constructing modules of the MSN architecture. Finally, in Section 5, we conclude this paper with a discussion about potential evolution of MSNs.

2. MSN: Definition, Characteristics, Design Requirements and Architecture

Before introducing specific research issues, in this section, we provide a comprehensive overview of mobile social networks, which outlines the big picture of MSNs, including its characteristics, underlying networks, design factors and architecture.

2.1. Definition of MSN

We consider a mobile social network to be an overlay network, that is, a social network overlaying on top of one or several types of mobile networks. A social

tie in the social network may correspond to several physical connections in the underlying mobile networks. It is a social networking platform able to connect people's mobile devices, and allow these users to converse, share information and form communities. In other words, the functions of the social networking platform are built around the functionalities of the underlying mobile networks.

2.2. MSN Characteristics

Several characteristics of mobile social networks are summarized as follows.

2.2.1. Social Relationship

One of the main factors driving the development of mobile social networks is the underlying social relationship among users. On one hand, MSNs are designed for the purpose of facilitating social interaction between users. On the other hand, social relationship can also be leveraged to help design communication protocols, for instance, social-aware routing mechanism.

2.2.2. User Mobility

Mobility characterizes user's movement, which is another major driven factor and concern in MSNs. People may move frequently during a day due to various reasons such as working, shopping, and travelling. Since mobile devices in MSNs are attached to people, their mobility introduces high dynamics to the network topology. On the other hand, user's mobility seems free in time and duration, but in fact, it is considerably correlated with geography and social relationship [2]. This implies the high potential of social networking services based on location and proximity.

2.2.3. Opportunistic Communication

In some MSNs where wireless access networks (cellular network, etc.) are not used or unavailable, nodes have to be networked in a self-organized manner via short-range wireless technology like Bluetooth and WiFi, they can only exchange information when they are within each other's transmission range. Besides, the network topology tends to be sparse and dynamic, nodes are not able to maintain their connection all the time, and routes between data source and destination also seem unmaintainable. Thus, to deliver information from source to destination, nodes should take the opportunity of encountering some node knowing the destination to forward the information.

2.2.4. Overlay Network

Mobile social networks are overlay networks on top of existing mobile networks. As illustrated in Fig 2, mobile nodes in the underlying mobile network are connected by physical wireless link, while nodes in the MSN overlay network are connected by social relationship, which can be regarded as a virtual link consisting of several physical links in the underlaying mobile network. Each node in the MSN maintains her social relations with other nodes, and the information exchange between them can be fulfilled through opportunistic networking schemes when the Internet is unavailable.

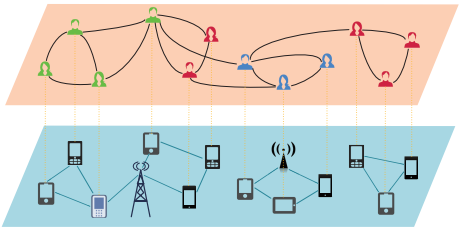

Figure 2: Mobile social network as an overlay network.

2.2.5. Heterogeneity of Mobile Devices

Due to the diversity of user's preference and device utilization, mobile devices in MSNs are highly heterogenous. First, they may have different model and hardware, and thus their computation power, memory, and storage are also different. Second, the operating system running on them can also be different, for instance, both Andriod and iOS, two leading operating systems on mobile devices, have a great bunch of supporters all around the world. Third, the running condition of these devices varies, for example, A's battery is going to run out while B just has had her phone charged.

2.2.6. Differences with Online Social Networks

Online social networks (OSNs) normally adopt a client/server architecture. Users (clients) can access the social networking services by requesting the servers whenever they have Internet connection. In contrast, mobile social networks do not necessarily adopt client/server model or rely on Internet connection. With the capability of short-range wireless communication, mobile devices can compose a mobile social network in an ad hoc manner. And in MSNs for local use, the Internet connection is not a necessity. Due to this architectural difference, some supporting mechanisms (e.g., routing, and security) are therefore different.

2.3. Underlying Mobile Networks

Mobile social networks can be built on top of various existing network infrastructures, such as mobile access networks, ad hoc networks, delay tolerant networks, and sensor networks.

- ⋆ ***Mobile Access Networks***. Currently, the most common way that mobile users exchange information is through mobile access networks, including cellular networks, WiFi networks, and WiMAX networks. Cellular and WiMAX networks can offer mobile users seamless Internet access on the move, since the range covered by the infrastructures is wide. Though the coverage of a WiFi hotspot is relatively small, it can provide a high data rate.

- ⋆ ***Wireless Ad hoc Networks***. A wireless ad hoc network is formed by a set of devices that are inter-connected by wireless without the help of any infrastructure such as routers or access points. End-to-end paths

can be maintained via multi-hop communication. This type of network is usually deployed on the fly for local and temporary use. For example, in a conference environment, participants can share presentation slides or documents through an ad hoc network formed by their laptops.

- ⋆ ***Delay Tolerant Networks***. A delay tolerant network is a network of several disconnected small networks. The disconnection can be a result of frequent movement and sparsity of mobile users, or the short transmission range and energy constraint of mobile devices. In DTNs, end-to-end paths are difficult to establish due to the disconnection. Instead, store-and-forward is used to enable communication among disconnected parts.

- ⋆ ***Sensor Networks***. Sensor networks can provide rich context information for mobile users to share with each other and augment their experience of social interaction. For example, a group of students want to find an empty and quiet room in a building to discuss their homework. They do not have to spend too much time on finding one, if they can obtain relevant information from or enquire a sensor network that monitors the condition (e.g., noise level, temperature, and occupancy) of all rooms in this building.

Depending on the type of mobile network that the MSN is on top of, it has three operating modes:

1. ***Internet-based mode***. Before mobile devices can communicate with each other, they must connect to the Internet first, either through cellular, WiFi or other access networks. In this mode, it is essentially an online mobile social networks. Representatives are Whatsapp, Waze, and Forthquare.

2. ***Self-organized mode***. In this mode, the Internet is unavailable for all the mobile users. They are connected by short-range wireless technologies such as WiFi and Bluetooth. The data transmission among users relays on ad-hoc or delay-tolerant communication.

3. ***Hybrid mode***. This is a hybrid of the first two modes, that is, some mobile devices have Internet connection, while the others do not. Mobile devices without Internet connection operate on self-organized mode, while mobile devices with Internet connection operate on Internet-based mode.

2.4. Design Requirements

Emerging social computing and mobile networking would give birth to diverse interesting mobile social networks. However, to develop a successful MSN, social computing and mobile networking, two basic functional requirements of all MSNs, should not be the only consideration. In fact, each mobile social network has different requirements, constraints and challenges that affect its development. The factors presented below are among the most common and influential.

— ***Decentralization***. Existing centralized application architecture, where users are required to upload their profiles and data to some central server, allows the social networking service provider (SNSP) to host all data of users and present it according to the application's design. This centralization give rise to two major problems. The primary one should be users' constant concern about their privacy, since they do not hold the ownership of their personal data which would be disseminated by the SNSP in a way against their willingness [3]. The other problem is traffic, which is caused by too many users simultaneously requesting the service from the server, and it would grow when the MSN gets popular. One method to resolve these problems is decentralization — users themselves decide to put their data either locally or onto a trusted server, and manage access control policies to allow retrieval of their data to selected users [4].

— ***Context Awareness***. A major advantage of mobile devices over desktop PCs is the capability of context-awareness of users, environment and themselves, which contributes to the shift from web-based social networks to mobile social networks [5]. This capability is granted by the diverse sensors equipped on the mobile devices and the always-with-user fact of mobile devices. To turn capability into benefit, MSNs shall capture contextual information related to users without too much user intervention. For example, when people is running, MSNs can automatically invoke the GPS and gravity-controlled gyroscope on her mobile device to detect her current moving status including location, speed, acceleration and route, and then share it with her friends.

— ***Inter-Operability***. To provide better service for users, mobile social networking applications may communicate with each other. For example, a book recommendation system may request a friending application for users' social profile possibly including reading preference. In addition, due to various preference, users in a MSN may use different software platforms (Android, iOS, etc.). We need to develop some scheme to support the inter-operability between different applications and heterogeneity of different software platforms on users' mobile devices.

— ***Internet Connectivity***. Internet connectivity significantly affects the applicability and application architecture of a MSN. On one hand, MSNs that do not require Internet connectivity are restricted to local and temporary use. In this case, fast formation of the MSNs is required. On the other hand, MSNs relying on Internet connectivity can adopt a client/server architecture to allow users from different geographical areas to interact and access the services whenever the Internet is available.

— ***Resource Constraint***. During the information dissemination in MSNs, a variety of resources on user's mobile device are closely/significantly involved. First, radio is needed to send/receive data, which also consumes a considerable amount of battery power when the data is of large size.

And, storage space is required to temporarily store data while its destination is unavailable. Also, resources like GPS and sensors are used to generate abundant contextual information to share with others. These resources are limited, and they might be entirely or partially in use for other purpose beside the mobile social networking service. These resource constraints should be considered in the design of routing protocols and information dissemination schemes for efficient use.

— ***Distributed Storage***. In mobile social networks with limited or without Internet connectivity (mode 2 and 3), users have to communicate with each other via short-range wireless technology like Bluetooth and WiFi Direct, they can only exchange data when they are within each others transmission range. To increase data availability, a data piece may have to be copied and stored in several end users' devices. Then the data is more likely to be delivered with good quality of service. In addition, the absence of a central infrastructure also necessitate distributed storage on end users' devices. Since wireless devices have much limited storage space and power supplies, proper mechanisms are required to solve problems like which devices to store data, how many data should be stored on a given device for how long. By exchanging data during opportunistic contacts, distributed storage also helps reduce the overall load on the infrastructure during peak time[6].

— ***Privacy***. With the purpose of facilitating social interaction, mobile social networks encourage people to disclose personal and private information such as current physical location, preference, and social relationship. However, this information is valuable to advertisers and even frauds – the major reason why people always show particular concerns about their privacy while using social networking services. Thus, for MSN designers, one challenging task is to develop useful social networking function without compensation of user's privacy. Moreover, designers should also consider the diversity of different user's privacy concern, since what is viewed private differs among individuals and cultures.

2.5. Case Study

To illustrate the above discussed requirements, an explanative example case study is presented in this subsection. Consider a mobile social network operating on mode 3 (see Fig. 3), that is only few of the users in the network have direct Internet connection. Suppose Leo is traveling in Africa with a WiFi-direct enabled camera. When seeing the legendary Victoria Falls with huge volume of water falling down and a rainbow circling around, he wants to share this spectacular moment immediately with his girlfriend Kate who is now in MIT. Since Leo cannot access the Internet directly, he has to find another way to send the video. Fortunately, his Zambian travel guide, Lungu, is in proximity to him, who happens to have 3G connection on his smart phone.

Before sending the video, Leo's camera has to find, and connect to Lungu's phone. Suppose each device in the network can be addressed simply using a

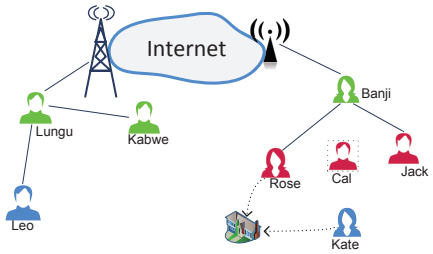

Figure 3: A case study.

unique community identity(ID) plus a unique user-name (In practice, the addressing method can be very complex). By broadcasting a user profile, including a user ID, physical address or other information if needed, Lungu's phone can be found by Leo's camera. After finding Lungu's phone, Leo sends out a connection request. Lungu accepts Leo's request when Leo's identity is authenticated by certain authentication scheme such as public-key cryptography.

After the two devices are connected, Leo sends the video to Lungu. To prevent the video from being watched by some intermittent forwarder, it is encrypted before sent. Now, it is Lungu's turn to find next forwarder. He has two candidates to forward the video - Kabwe, a hotel manager nearby, and Banji, who is studying in MIT and currently online. Learning from their profiles that Banji is in Rome while Kabwe is in Zambia, Lungu decides to send the video to Banji since she is much closer to Kate. The transmission between Lungu and Banji simply follows the procedures of TCP/IP protocol. Banji now has to choose a forwarder between Jack and Rose (In practice, depending on the forwarding algorithm being implemented, multiple forwarders may be chosen to help forward). Except Jack and Rose, there is also another user, Cal, in proximity to Banji. However, Cal does not want to be discovered and help others forward data, because his mobile device is running out of battery. From the history of contact with Kate, Rose supposes she will meet Kate later at dinner, whilst Jack thinks he will not contact Kate during this week. Banji has learned this information during the neighbor discovery, so she sends the video to Rose since she has higher probability to meet Kate. After receiving the video, Rose stores it in her phone's storage, and delivers it to Kate when they meet in canteen.

Finally, Kate can enjoy watching the video after it is decrypted. Note that, all the decisions such as neighbor discovery, forwarder selection, data encryption, and hiding form being discovered, are made by the protocols implemented on users' devices, completely transparent to users.

3. Overall System Architecture Design

In the literature[7, 8], MSN architectures have been discussed. However, the architectures in them are mainly from the applications point of view without

outlining what functionalities a MSN must have. In addition, in those architectures, the Internet-based operating mode is implicitly assumed. MSNs are overlay networks on top of various wireless networks (e.g., WiFi, Bluetooth, and cellular), therefore its functionalities should also be built on top of those network interfaces. In this section, we present a reference system architecture for MSNs. In this design, the characteristics and design requirements of MSNs are taken into consideration.

Figure4 illustrates the overall MSN architecture. In brief, the architecture consists of five components, namely the Application Component, the Social Mining Component, the Networking Mechanisms Component, the Local Resource Management Component, and the Privacy & Security Component. In the following, the functionalities and their interactions of these components are introduced. As part of the design, the states-of-the-art of the five components are reviewed in the next section.

- ⋆ ***Application***. An application is composed of several protocols that provide different visible services to MSN users, such as posting, searching, matching and recommendation. The structure of this application over mobile systems is determined by its architecture which can be client/server, peer-to-peer or hybrid of the two formers. Different application architectures have distinct impact on the underlying networking mechanism, since information exchange among MSN users in client/server architecture is controlled by central server, while in peer-to-peer architecture, it is executed via direct communication among peers. The application resided on user's mobile devices should be able to leverage their sensing capabilities in order to obtain abundant context information that can be used to share with other users or to improve the application's performance. For example, GPS and gyro can detect your location and moving state (walking, running, etc.), if you are running and not convenient to interact with the screen, the application should be able to push incoming message by audio instead of text.

- ⋆ ***Social mining***. This component collects information of user, device, and the environment, and provides knowledge on social properties, mobility pattern, and individual preference. There are several social properties applicable in MSNs, including social tie, centrality, community, and edge expansion [8]. Mobility pattern contains user's mobility trace and the distribution of contact with other users, such as the distributions of inter-contact time, and contact duration. Finally, individual preference records user's time-varying preference on resource contribution to the network.

  As shown in Fig. 4, the social mining has two fundamental functionalities - profiling and data aggregation & learning. The profiling collects information of user, device, and make it into profiles. More specifically, user profile includes a user's social contacts, visited places, activities and so on. Device profile captures the capabilities of the device, including network interfaces, storage space, battery, and sensors. The information

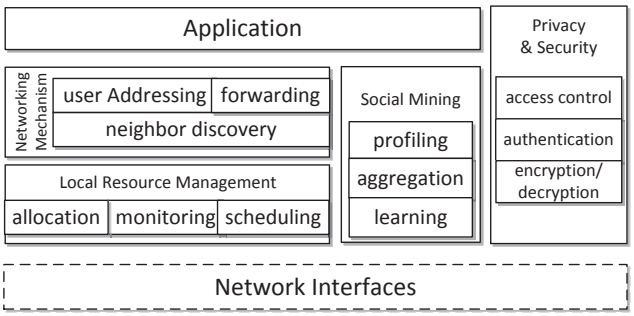

Figure 4: Building blocks of an MSN architecture.

comes from other components such as application, networking mechanism, local resource management and resources. Then by data aggregation and learning, some social knowledge mentioned above will be discovered. This knowledge can be used in decision-makings in local resource management, networking, security strategy, and application.

- ***Networking mechanism***. This networking mechanism in MSNs is responsible for information dissemination among users in the MSN, which are capable of supporting the three MSN operational modes if needed by applications. Basically, it contains user addressing, neighbor discovery, and routing. Each device in the network should have a unique address so that the data can be addressed and then delivered to the right recipient. In a MSN, different types of addresses may be used in combination. For example, devices with Internet connection may use IP addresses, while devices without Internet connection have to use other type of addresses. Neighbor discovery allows a user to find and connect to other users in vicinity through short-range wireless communication. The routing function determines the path, a sequence of addresses of intermediate devices, that data needs to traverse from a sender to a receiver.

  For MSNs that are highly integrated with the Internet, people can share their information via various access networks (cellular, WiFi, etc.). However, for decentralized MSNs where access networks are unavailable, opportunistic networking should be employed, thanks to the sparseness and dynamic property of the network topology. For example, when a person wants to send information to a friend that is not within her transmission range, she has to store and carry the data until she meets her friend, or just sends to an intermittent user who might meet her friend in the future and then forwards the data. The opportunistic networking mechanism shall be aware of the limitedness of the resources on users' mobile devices, and be able to leverage social pattern such as community property and user mobility pattern for helping information dissemination.

- ***Local resource management***. Resources refers to both hardwares and

non-hardware features of each user's mobile device, which support proper functioning (social computing service, mobile networking, etc.) of mobile social networks. The hardware resources are mainly for mobile networking purpose, including radios, sensors, battery, memory, storage and so on. A message from its source to destination may traverse several intermediate nodes, and thus use their radios, occupies amount of their storage spaces and consume their batteries. Note that, some of them like storage and sensors can also be directly shared with other users. For example, people who wants to know her exact physical location but her GPS is under malfunction may turn to nearby user who has GPS signal for help. The non-hardware resources, which are for social networking purpose, can be any shareable function enabled on user's mobile device. Take Internet access for example. What can he/she does, when a person happens to run out of mobile balance but still wants to check email? One option is to ask users around if they can access the Internet and forward message.

Most of the resources mentioned above are considerably limited, which necessitates effective local resource management (LRM) to make the resource consumption as efficient as possible. What is more important, the LRM should be carried out on an individual user's basis, since the devices having these resources normally belong to individual users rather than a network-wide owner. Generally, LRM provides three basic functions – resource monitoring, allocation and scheduling. First of all, the LRM should be able to obtain the resources' availability and track their usage, such as how much resource (e.g, bandwidth) is in use and by which task. Second, when a new task demands this resource, the LRM decides how much resource should be allocated to the task. If there are several tasks requesting the same resource, the LRM should determine to which order each task should be executed.

- ⋆ ***Privacy and security management***. Privacy and security management addresses three main issues. First, it provides anonymity scheme for user's identity and location to keep user away from being identified by others with improper purpose. Second, it specifies fine-grained access control policy for user's personal information (e.g., who can view which part of user's information). Third, it provides comprehensive security mechanism to prevent from attacks such as eavesdropping, spoofing, replay, and wormhole attack, which are major threats to decentralized mobile social networks [9].

4. MSN State-of-the-Art

In the MSN architecture introduced in the previous section, we have discussed its functionalities and their relation in the architecture. To provide a better understanding of the MSN architecture, this section is devoted to providing a review of the state-of-the-art of each of the five components in the designed architecture.

4.1. Applications

Smart devices have been widely accepted by people due to their features of GPS, high resolution camera, WiFi and so on. Application developers have realized that it is a unique opportunity to make social computing pervasive by leveraging those features of smart devices. Traditional OSNs like Facebook and Twitter have already made their PC-based services available on mobile devices. Meanwhile, services purely for mobile users have also been developed to meet people's variant requirements. Some of them have gained tremendous success, such as Foursquare and Waze. We classify them into four categories: location-based, proximity-based, participatory sensing, and vehicular social networks, and for each category, we introduce several representatives in brief.

4.1.1. Location-Based Social Networks

At present, mobile user's location can be easily tracked. Enabling methods include GPS system, wireless access (WiFi, cellular network, etc.), and user self-report. And the extensive deployment of mobile devices has given birth to a plethora of location-based social networking services or geo-social networking services, including Gowalla, Foursquare, Brightkite, Loopt, Google Latitude, and Facebook Places.

Among those, Gowalla and Foursquare are the pioneers, which were both launched in 2009. Before acquired by Facebook, Gowalla had attracted millions of active users by its check-in feature. Basically, Gowalla allows user to 'check-in' while visiting a venue (e.g., cafe, cinema) through a mobile application and share this check-in to friends on Gowalla. Using location data provided by GPS system or WiFi connection, Gowalla verifies whether the user is actually located at the checked-in venue or not. As a rival of Gowalla, Foursquare shares the same feature of check-in. However, unlike Gowalla's verification on user's location, Foursquare allows a user to check-in venue even this user is not physically located at this place. In addition, Foursquare emphasizes more on social interaction among users by making their tips on checked-in venue visible to future visitors.

Geo-recommendation is another favorable feature of LBSs. GeoLife [10], for example, is a GPS-data-driven recommender system. People using GeoLife can keep tracking their locations when they are on the move, and record the visited locations as trajectories. For each location, multimedia content such as a photo can be associated and shared with friends. For each user, GeoLife computes similarity between this user and other users using the information provided by their trajectories (i.e., time, location), and people with high similarity are recommended as potential friends.

4.1.2. Proximity-Based Social Networks

Proximity-based social networking applications use geo-proximity as the primary filter in determining who is discoverable on the social network[11]. Several methods enable user with mobile device to discover one another in geo-proximity, such as measuring distance with GPS locations, and sensing through short-range wireless communication technologies (WiFi, Bluetooth, Near Field Communication, etc.). WeChat, a mobile messaging application that claims to have 600

million monthly active users [12], provides a social service called 'Look Around' for users to chat with other WeChat users nearby. Once a request for 'Look Around' is initiated, it returns a filtered list of candidates with visible profile and distance to the service requestor. Others like Badoo and WhosHere have similar feature.

E-Shadow [13] is a system that provides information publishing and ownership matching for users within vicinity. Each user maintains a local user profile containing user name, interests and so on. The visibility of this profile to other nearby users depends on distance, i.e., the closer they are the more detailed profile they can obtain and vis versa. If someone is interested into another person's profile and desires more information, E-shadow provides a function called direction-driven localization for him/her to walk toward the target.

Reflex framework [14] enables local spontaneous interactions among people using the same application over Bluetooth or WiFi interface. Basically, to interact in a mobile application, people only needs an anonymous identity to start and join an interactive session, registration and login are no more required. Thus, people without internet connection are also possible to join in the interactive session spontaneously. Above this framework, MusicScore is developed for users that are connected via Bluetooth or WiFi (peer mode) to compose music simultaneously.

4.1.3. Participatory Sensing

Micro-Blog [15] takes advantage of smartphone sensors, and encourages users to record multimedia contents named microblogs on-the-fly and share them with other users. For a particular microblog, the creator can associate location, time and other information with it. Micro-Blog also keeps update user's location periodically, so that when this user query about his current location, relevant micro-blogs can be served immediately.

CenceMe [16] is a people-centric sensing application that automatically senses user's presence with the sensor-enabled mobile phone. It can detect the mobile phone carrier's location, mobility mode (sitting, walking, running, etc.) , surroundings (noise level, temperature, etc.), and whether he is in a conversation or not. With these rich contexts, CenceMe provides a bundle of useful functionalities to its users, such as presence sharing, and self history tracking.

4.1.4. Vehicular social networks

Waze offers community-based GPS navigation with turn-by-turn directions. Keeping Waze open while driving, users can update traffic information such as traffic jam, road hazard, or station offering cheap gas, in a realtime manner. Other drivers in the same area will be alerted by receiving this crowdsourced traffic information and thus improve their commute. Waze also allows Facebook friends who are both driving to see each other on the map, so that their arrival time can be coordinated.

Similar to Waze, Drive and Share (DaS) [17] provides a platform for drivers to share realtime traffic and personal information. Besides, it also enables

vehicle-to-vehicle communication for drivers to exchange vehicle-detected information such as vehicle density and speeds. RoadSpeak [18] is designed for group voice-chat among commuters on roadways. People in RoadSpeak can join talk groups that are active between a certain period. When a group becomes active, group members will receive an alert from RoadSpeak, and during this active period, they can communicate with each other freely.

4.1.5. Middleware for mobile social networks

MobiSoC [19] is a social computing middleware that explores social state information of physical communities. A community is formed by people with certain relationships, be they friends, family or colleagues. And the social state information refers to people profiles, social ties among people, place profiles, and people-place association. It employs a client/server architecture to manage the collected information - stores the information on central server and performs learning algorithms to discover emergent geo-social patterns including people-people affinities and people-place affinities. To deal with the frequent change of social state, an event manager is enabled by MobiSoC to register those changes with applications. For the privacy concern, users are allowed to issue a privacy statement which specifies which user can access his data. When an application requests information regarding to this user, MobiSoC will verify the associated privacy statement before the information is accessible.

Authors in [20] proposed a mobile social middleware MobiClique to support opportunistic ad hoc communication without relaying on central server. Users' social profiles including full user profile, list of friends, groups and events that the user is attending, are retrieved from an existing OSN service (Facebook) via open API, and occasionally synchronized with Facebook when Internet connection is available. MobiClique leverages Bluetooth interface to enable opportunistic communication among users. When users coming into proximity, they can discover each other, exchange identity and send message via bluetooth equipped on their devices. Two types of message forwarding are enabled by MobiClique, i.e., unicast between a pair of users and flooding among a group of users. Furthermore, MobiClique supports opportunistic messaging between devices through a destination-oriented messaging abstraction. In other words, if the destination is currently unavailable, the message will be forwarded through friends of the destination.

Yarta [21], developed by A. Toninelli et al., does not only support information exchange among users, but also supports knowledge exchange among applications built above it. The component storing user's data is called Knowledge Base. The base data model of Yarta is represented using the Resource Description Framework (RDF). Based on RDF, applications can extend the data model according to specific requirement. Since the extensions have clear semantic relation with base classes, the data interoperability is enabled. Yarta decouples data access policy from application logic and Knowledge Base management, and defines customizable rules for users to specify their own preferences on data accessibility. In addition, Yarta does not relay on centralized server to collect

user's data, and support and supports data transfer over heterogeneous network interfaces.

4.2. Social Mining

Mobile social networks are overlay networks, that is, social networks overlaying over the mobile networks, as described in Figure 2. Therefore, MSNs exhibit some properties as traditional social networks do. First, MSNs have community structure, where a community is a group of tight-knit nodes having more internal connections than external connections [22]. People may form a community if they have common attributes, e.g., in the same club, and people in the same community interact more frequently with each other than with people outside. Second, MSNs are scale-free networks [23]. The degrees of the nodes in the network follow a power-low distribution. In other words, few nodes called hubs have the largest degrees, while most other nodes are connected through them.

Understanding these properties is of great advantages for mobile networking problems such as routings and worm containments [24]. However, it has not been an easy task in dynamic networks like MSNs where users move about and sometimes loss connection with each other. Nevertheless, a number of works have been done targeting on community detection and influential user identification in the context of MSNs.

4.2.1. Community Detection

Community detection is a mechanism that clusters nodes in the network into groups such that nodes in each group are more densely connected internally than externally. The community detection can be either centralized or distributed, depending on whether the whole network graph is required or only part of it is required by the entity that performs the community detection. A summary of the various community detection mechanisms is presented in Table 1.

Centralized Community Detection. Nguyen et al. proposed Quick Community Adaptation (QCA), an adaptive modularity-based method for identifying community in dynamic networks [25]. QCA samples a dynamic network into a sequence of snapshots over time, where each snapshot can be considered as a static network. Initially, QCA uses another static modularity-based algorithm to find the basic community structure from the first snapshot. And after that, QCA only deals with the network changes and detect new community structure by modifying the previously detected one. When a new snapshot is taken, four events that affect the network structure may happen, namely, node addition, node removal, edge addition and edge removal. For each type of event, QCA updates the community structure according to maximizing the overall modularity. Though QCA is fast and effective, it fails to detect overlapping network communities.

To detect overlapping communities, the authors of QCA designed AFOCS[24]. With the similar idea of slicing the network into time-dependant snapshots,

AFOCS detects the initial overlapping community structure from the first network snapshot and then updates it when taking the next snapshot. First, local communities are located around edges. A local community $\mathcal{C}$ around edge (u, v) is formed by u, v and their common neighbors, if the internal connections of $\mathcal{C}$ is larger than its density function [26]. And then, local communities that have a overlapping score lager than a threshold are merged as one community.

Chen et al. designed a community detection method that can identify overlapping communities, based on a game-theoretic approach [27]. They suppose that the structure of social network is given as prior knowledge, that is, each node has fixed social connections with others. Basically, nodes are selfish players who want to maximize their own utilities when deciding whether to join a community or not. The utility of each player is the difference of gain and loss of joining a community, where the gain of joining a community is given by a modified modularity called personalized modularity function, and the loss simply represents fixed cost (e.g., membership fee). Overlapping communities will be discovered if the game has Nash equilibrium.

Distributed Community Detection. When the community detection is performed by each node in the network, who only knows part of the graph, distributed community detection should be used. Clauset proposed a greedy algorithm to find local community structure [28]. For a node s who wants to find its local community, the whole network graph consists of two complementary subsets, i.e., known potion of the graph $\mathcal{C}$ and unknown portion of the graph $\mathcal{U}$. In $\mathcal{C}$, there is a set of nodes that have one or more neighbors in $\mathcal{U}$. This set of nodes $\mathcal{B}$ is called the boundary of $\mathcal{C}$. To measure the quality of a local community, local modularity is defined as the fraction of the number of edges with one end point in $\mathcal{B}$ and the other not in $\mathcal{U}$ to the number of edges with one or more end points in $\mathcal{B}$. To detect its local community, initially node i put itself in $\mathcal{C}$, and its neighbors in $\mathcal{U}$. Then it adds to $\mathcal{C}$ the neighbor that brings the largest increase in the local modularity. This step repeats until it either has gathered a given number of nodes k or has discovered the entire known graph.

Rather than forming community based on social relationship among users, authors in [29] proposed to use contact rate and duration. They define 'temporal community' as a cluster of nodes that contact frequently within a long period. Basically, the dynamic network is snapshotted into a sequence of static networks. For each snapshot, they first use Louvain algorithm [30], a static modularity-based community detection method, to partition the network into dense and disjoint clusters. And then, clusters that have Jaccard distances no exeeding a given δ [31] between each other are aggregated into a temporal community.

Hui et al. proposed three distributed community detection algorithms, namely, SIMPLE, K-CLIQUE and MODULARITY [32]. Basically, each node in the network needs maintain a list of encountered nodes and their contact durations, a familiar set, and a local community. The familiar set of a node comprises encountered nodes that have a cumulative contact duration with this node longer than a threshold, while the local community of this node is the union of its familiar set and nodes selected by the algorithms. For K-CLIQUE

and MODULARITY, each node also needs to know the local community of every encountered node. These three algorithms have similar procedures to go through to detect communities - when a node meets another node, it has to decide 1)whether to include this encountered node into familiar set and/or local community, and 2) whether to merge the local communities of these two nodes. The difference between SIMPLE, K-CLIQUE and MODULARITY lies in the admission criteria. Based on K-CLIQUE, DiBuBB [33] allows each node to compare with every encountered node on how many unique nodes they have met, to get its centrality ranking within the local community.

Li and Wu proposed a LocalCom scheme which can detect the community structure using limited local information [34]. Nodes are required to have the history of contacts with other nodes. To depict the relationship between each pair of nodes, a new similarity metric is defined using the statistic properties (i.e., average length and variance) of the separation period between a pair of nodes. A community is formed by a clique of nodes that the similarity weight between any two nodes is larger than a threshold w. Two nodes that are not direct neighbors can also be in the same community, if there is path (called virtual link) with path length no longer than k and similarity weight larger than the threshold w. First, each node detects its neighbors and virtual links to them. Node selects itself as the initiator of community detection, if it has the largest degree within its neighborhood. The initiator constructs a community by adding members to the community one by one. A node will be added to the community, if it has similarity weight larger than the threshold with all other nodes in this community.

4.2.2. Identifying K Influential Users

iWander is a distributed protocol for identifying influential users using random walks in MSNs[35]. The influential users refer to as the users that have high centrality in their social-contact graphs. The iWander is inspired by a previous finding that most people have fewer friends than their friends have [36]. This finding suggests that users with high centrality can be encountered by random selected users with a high probability. In iWander, for every $\triangle T$ period, each smartphone generates a probing message with a probability p. The probing message only contains a time-to-live field (TTL). When a user having a probing message with positive TTL encounters another user, it sends the message to the encountered user who then decreases the TTL by 1 after received the message. Message with $TTL = 0$ will be discarded and not forwarded. Each user counts how many probing messages has been collected within $\triangle T$. Finally, the k users with the most messages are selected the k influential users.

4.3. Mobility Pattern

Human mobility measures how people move over time. Surely, the mobility of one person is distinct from that of another. Nevertheless, statistical results of human mobility have been discovered based on large-population mobility data [37, 38, 39, 40, 41]. And these results have extensive applicability in urban planning, epidemiology, and mobile computing [42, 43, 44, 45, 46, 47, 48]. Recently,

Table 1: Summary of community detection methods

Method	Concept used	Dynamic	Overlapping	Complexity	Applicability
QCA	Modularity	Yes	No	unknown	Mode 1 and 3
AFOCS	Density function	Yes	Yes	$\mathcal{O}(n^2)$	Mode 1 and 3
Game	Modularity & utility	No	Yes	$\mathcal{O}(m^2)$	Mode 1 and 3
Temporal	Modularity	Yes	No	$\mathcal{O}(N)$	Mode 2
SIMPLE	Modularity	Yes	Yes	$\mathcal{O}(n)$	Mode 2
K-CLIQUE	Modularity	Yes	Yes	$\mathcal{O}(n^2)$	Mode 2
MODULARITY	Modularity	Yes	Yes	$\mathcal{O}(n^4)$	Mode 2
DiBuBB	Modularity	Yes	Yes	$\mathcal{O}(n^2)$	Mode 2
Greedy	Modularity	Yes	No	$\mathcal{O}(k^2 d)$	Mode 2
LocalCom	Similarity	Yes	No	unknown	Mode 2

human mobility characterization has received great research effort, mainly due to the increasing popularization of mobile devices. People's mobility information can be obtained from either infrastructure (WiFi access spot, cellular base station, etc.) association, or GPS on their mobile phones, or proximate device discovery. Generally, there are three types of dataset that has been exploited in literature:

Call Detail Records. Call Detail Records are collected by telephone service providers, for purposes like billing and traffic management. Each call detail record (CDR) contains information of a telecommunication transaction (voice, SMS, etc.) except the content, such as the phone numbers of the calling party and called party, the starting and ending time of the call, and the associated cellular antennas. CDRs datasets usually have a large volume, wherein millions or billions of mobile users are recorded. In addition, obtaining such data does not incur substantial cost. However, CDRs are coarse-grained since a record can be made only when there is a call[42].

Location Based Service Check-ins. Location-based social networking services such as Foursquare allow people to 'check-in' their current location and share tips with friends using their mobile devices. Check-in data can be collected via public API. For example, Foursquare API[2] gives access to all data used by the Foursquare mobile application. Compared to the CDRs, data from LBSNs can provide specific user locations (shop, cinema, etc.) and explicit friendship among users. Besides, semantic information such as pictures shared with each other is also available.

Experimental Data. To study the characteristics of human mobility, several experimental projects have been developed by research groups. Table 2 provides a list of project datasets that have been widely used in the literature. These projects typically asked a number of volunteers (mostly students and researchers) to regularly report their positions or contact records with other participants. For example, GeoLife collected GPS trajectories of 182 users over three years from April 2007 to August 2012. And most of the trajectories consist of a frequent (say per 5 seconds) position logins. Compared to CDRs and LBSN data, experimental data is more fine-grained in time and space, but with a much smaller scale (tens or hundreds of users involved).

The data sources discussed above provide real-life data of human mobility in different scale and granularity. Apart from those, there are also studies of user mobility in virtual world. La and Michiardi in [58] presented a measurement study of user mobility in Second Life (SL)[3], an online virtual world allows users (called avatars or residents in SL) to build and trade property and socialize. They found that contact-time distribution in SL is similar to that of real humans.

[2] developer.foursquare.com
[3] secondlife.com

Table 2: Experimental Project Datasets

Dataset	Data Type	Duration	Report Frequency	Device Type	Participants
Reality[49]	call logs, etc.[a]	9 months	300 seconds	mobile phone	100
Infocom05[50]	contact[b]	4 days	120 +/− seconds	iMote with Bluetooth	54
Cambridge[51]	contact	23 days	120 seconds	iMote with Bluetooth	36
Infocom06[52]	contact	4 days	120 seconds	iMote with Bluetooth	98
HongKong[33]	contact	5 days	120 seconds	iMote with Bluetooth	37
Dartmouth[53]	AP association	17 weeks	300 seconds	Laptop/PDA with WiFi	7000
UCSD[54]	AP association	7 weeks	120 seconds	PDA with WiFi	275
Geolife[55]	GPS trace	65 months	1 5 seconds	GPS receiver/phone	182
Levywalk[56]	GPS trace	5 months	1 5 seconds	GPS receiver	44
Taxicab[57]	GPS trace	6 months	10 seconds	GPS receiver	50

Both Infocom05, Infocom06, Cambridge and HongKong belong to the Haggle Project (www.haggleproject.org).

[a] The data collected in Reality includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (charging and idle).

[b] A contact means a proximate Bluetooth discovery, and it is represented by a tuple (MAC address, start time, and end time).

Table 3: A Classification of Human Mobility Patterns

Classification	Pattern	Definition
Social	Inter-contact time[46, 60, 61]	The time elapsed between two consecutive contacts of the same pair/group of people.
	contact duration[50]	The time interval for which two network devices can communicate when they come into range.
	Mobility similarity[40]	The overlapping degree of two trajectories.
Spatial	Flight length[62, 63]	The distance of straight line trips without directional change or pause.
	radius of gyration[42]	The linear size occupied by each user's trajectory.
Temporal	Return probability[37, 60]	The probability that a user returns to the location where he/she has visited before.
	pause time[64, 65]	The time interval a user spends at one location.
	Visit frequency[37]	The frequency of a user visiting the same location.

Hitherto, considerable conclusions on human mobility pattern have been drawn from the rich data. D Karamshuk et al. have classified them into three categories: spatial, temporal and social[59], as shown in Table 3. The followings present two fundamental statistical findings of human mobility.

Inter-contact time. As defined in Table 3, the inter-contact time measures how frequently two users meet. Authors in [46] found that the inter-contact time has a power-law distribution for the timescale of interest of $[10minutes, 1day]$. Further, it is discovered that there is a dichotomy in the CCDF (complementary cumulative distribution function) of inter-contact time, that is, the CCDF follows power-law until a characteristic time, and beyond that, the CCDF has an exponential decay[60].

Flight length. People usually travel only over short distances, and occasionally take long trips. By studying outdoor GPS traces of 44 people, I. Rhee et al. in [39] show that human walk can be described by a truncated Levy walk[66]. They aggregated altogether the flight length samples, and found that the CCDF of flight length quite fits to the truncated Pareto distribution[67]. This distribution indicates that people have a high probability to take long trip in a single step. The flight truncation is caused by geographical constraints such as boundaries and physical obstructions.

The discovered human patterns have the following two significant applications in mobile social networks.

Mobility modeling. The performance of a MSN is significantly depends on the mobility patterns of mobile device carriers. Before the MSN reaches the end users, its performance must be evaluated. Deploying a real network is not recommended since it introduces high cost. By contrast, simulation or theoretical analysis using mobility model is widely adopted. To get meaningful results, the mobility model must be able to reproduce essential human mobility patterns. Fine solutions for mobility modeling have been proposed like SLAW[68], SWIM[69], HCMM[63], and GeSoMo[70], just to name a few. [59] and [8] provide comprehensive reviews of recently proposed human mobility models.

Mobility prediction. It is reasonable to suppose that human mobility is predictable, since it has spatial and temporal regularity to some extent. As described in [71], a mobility prediction process consists of four steps: 1) positioning and tracking. Discover the position of the user by a positioning system. 2) track logging. Record and sequence the positions as traces. 3) data of interest extraction. Learn meaningful information (e.g., point of interests) from the raw trace data. 4) location prediction.

There have been proposed several prediction algorithms based on Markov model[72, 73, 74]. They predict the next location of a user based on the previously locations that he/she visited. Authors in [73] first discover user's POIs (points of interest) from GPS trajectories using a k-means clustering algorithm [75]. Then, they predict user's next location with a Markov model, where each state denotes a POI and the transition between two states means the probability of moving from one POI to another. S. Gambs et al. proposed a similar Markov-model based prediction called $n - MMC$[72]. It uses Density-Joinable cluster[76] to extract POIs, and it keeps track of n previous locations instead of one to improve the prediction accuracy. In [72] and [73], a state represents only one location. In [71], a $K - to - 1$ past model is presented, where a state can represent various size of consecutive locations from K to 1. The $K - to - 1$ past model tries to predict the next K consecutive locations using the previous K consecutive locations. If it fails (the transition probability is smaller than a threshold), it will try to predict the next $K - 1$ locations using the previous $K - 1$ locations until $K = 1$. A location prediction based on Hidden Markov Models (HMMs) is presented in [77]. Before the prediction, the location histories are discretizated into discrete codes associated to specific locations using hierarchical triangular mesh[78], in order to make the learning of HMMs efficient. Then a HMM is used to compute the probability of each sequence of locations, composed of several locations already visited and a potential next location to be visited. Finally, the potential next location in a sequence with the highest probability is the result of prediction.

There are also several works predict human mobility using pattern matching [79]. Basically, user's movement history is observed and recorded, and patterns

are mined from it. By matching the current section of movement with the patterns, future location is predicted. Al-Hattab et al. in[79] define a 'query' as a time series of locations starting at the current point in time and goes back for certain points, and 'Location Time Series (LTS)' as a time series of locations for a given period. Then, they use normalized cross correlation[80] to compute the similarity between the current query and LTSs that have been stored. The predicted next location is given by the LTS having the largest similarity with the current query. Based on the assumption that individuals tend to follow common paths, A. Monreale designed Wherenext[81] that uses movements of all objects in a certain area to predict an individual's next location. In Wherenext, global frequent movement patterns, named T-patterns, are extracted from all individuals' trajectories by a Trajectory Pattern algorithm. Each T-pattern describes a sequence of regions visited frequently by individuals and typical durations of movements between regions. Given a set of T-patterns, a T-pattern Tree can be constructed using association rules learning[82], where a node represents a location and each edge is labeled with a time interval. Afterwards, Wherenext predicts the next location by finding the best match of a trajectory (used to predict next location) with all admissible paths on the T-pattern Tree, and computes matching scores. Authors in [83] show that people's location can be discovered even it is kept private. They designed a location prediction method, Flap, to guess people's past and future location using public locations of his/her friends (on Twitter). From check-in datasets of location-based social networks, it is found that human movement is partly periodic (50% to 70%) and partly correlated with friendship (10% to 30%)[2]. Based on the empirical finding, a Periodic & Social Mobility Model (PSMM) is developed to predict people's future location, which consists of three sub-models for spatial frequently visited locations, temporal movement between locations, and movement driven by social relationship, respectively.

4.4. Networking Mechanisms And Local Resource Management

Information Dissemination is a fundamental problem in all types of mobile social networks. In online mobile social networks, though the information is transmitted via the Internet infrastructure, how to cost-efficiently distribute the information requested by large number of users is still challenging. In delay-tolerant mobile social networks, the intermittent connectivity among users makes end-to-end communication impossible. To address those challenges, a myriad of networking and local resource management mechanisms have been proposed recently (e.g., [84, 33, 85]), by leveraging the social pattern and mobility pattern discussed in above.

4.4.1. Information Dissemination in Online MSNs

In online mobile social networks, information is normally disseminated to mobile users via the Internet. But with the proliferation of smart phones, the cellular networks are severely overloaded by mobile data traffic. And online social networking services like Youtube and Twitter account for a large portion

Table 4: Summary of mobility prediction methods

Method	Technique used	Input	Accuracy
$n-$MMC [72]	Mobility Markov Model and Density-Joinable cluster	n previous visited locations	70% to 95% as soon as $n = 2$
Wearable[73]	Markov model and $k-$means clustering	Current location	Medium
AKMM[71]	All-K^{th} Markov Model, $K - to - 1-$Past* Model	K previous visited locations	High
HMM[77]	Hidden Markov chain and hierarchical triangular mesh	A set of sequences	Low
LTS[79]	Normalized cross correlation	A time series of locations	Proportional to the length of the prediction series
Wherenext[81]	Trajectory pattern extraction and association rules learning	Sequences of regions frequently visited with a typical travel time	Medium
Flap[83]	Supervised and unsupervised learning and Viterbi decoding	A sequence of locations visited, along with corresponding time information	High

of the traffic. How to alleviate the traffic load and reduce operational cost has become an issue. Some information dissemination methods like traffic offloading and proactive seeding seem to be premising solutions.

One way to offload the traffic is to exploit the capacity of opportunistic communications among mobile devices. S. Ioannidis considered a scenario where mobile users subscribe to a dynamic-content service[84]. The injection of new contents (updates) to the network is modeled as a Poisson process with average injection rate μ. Each update is only pushed to a user with a certain probability, other users get the update by opportunistic contact with user who has the update. The problem of allocating injection rate to mobile users is formulated as an optimization problem where the objective is to maximize the update dissemination speed. It is proved that the optimal rate allocation can be found using gradient descent. In [6], a K-user selection problem is formulated to minimize the cellular data traffic. The information is initially delivered by service provider to the targeted K users who will forward it to the rest users via opportunistic communications. And if a user does not receive the information before a delay threshold, the service provider will send the information to this user directly. An information dissemination function is built as a mapping of the target set (K users) to the expected number of users that would receive the information by opportunistic contact. It is proved that the information dissemination function is submodular and can be solved by greedy algorithm. In [86], authors proposed Proactive Seeding to reduce the peak load in cellular networks. They classify the data traffic into predictable (e.g., Twitter posts) and unpredictable (e.g., voice call), based on the assumption that people tend to get information that their friends are interested in or recommend. The traffic is smoothed by pushing some predictable traffic of peak hours to mobile users before they request it.

4.4.2. Information Dissemination in Delay-Tolerant MSNs

BUBBLE Rap is a social based forwarding algorithm based on community and betweenness centrality [33]. There are two underlying assumptions: 1) users are structured into communities, and each user belongs to at least one community; 2) each user has two types of centrality ranking, namely, global ranking across the network and local ranking within its community. The knowledge of community structure and centrality rankings is known to BUBBLE Rap on each user's device. When a node has message for another node, it first sends the message to any encounter having higher global ranking than it, and the encounter(s) also does this, until the information reaches the destination or node in the same community with the destination. When the message reaches the community that the destination belongs to, node who has this message sends it to its encounter(s) having higher local ranking, until the message finally reaches the destination.

SimBet, proposed by Daly and Haahr [87], takes advantage of egocentric betweenness centrality and similarity [88, 89] to help deliver message. Betweenness is used to identify bridge nodes which can broker information exchange among disconnected clusters of nodes, while similarity is used to predict future encounter of two nodes. When two nodes encounter, each of them sends a list

of nodes it has encountered to the other. With the information collected from its encountered nodes, each node can compute its betweenness and similarity locally. Two encountering nodes also exchange a vector containing a list of destination nodes they are carrying message for. For each destination, both of two nodes compute a SimBet utility based on the betweenness and similarity, and then compare with each other. Node with higher SimBet utility will be the next message carrier for this destination.

SocialCast is a socially-aware routing protocol for MSNs with publish/subscribe messaging pattern [90]. Since in the publish/subscribe paradigm, information publishers and subscribers are agnostic of each other, it is employed to decouple the communicating nodes who may become disconnected during at the message delivery. When a message is published, 'interest' tags instead of node identifiers are specified in the message, which are visible in routing layer. In SocialCast, each node maintains a list of its interests. And for each interest, it computes a utility based on the Kalman filter [91] to measure the suitability of this node to be a carrier for messages matching this interest. The utility is a function of 1) the probability of a node to be co-located with another sharing this interest, and 2) the change degree of connectivity of this node. Each node periodically broadcasts a message containing its interests and corresponding utility values to its one-hop neighbors. After the interest dissemination, for each interest, nodes compare the utility against each other. Node with the highest utility will be the carrier for the messages matching this interest.

In [85], authors proposed a forwarding strategy based on differentiated friendship. To simultaneously capture the features of friendship including frequency, longevity and regularity, a matric called social pressure metric (SPM) is introduced. SPM between two nodes i and j is essentially the average time it takes node i to encounter node j if i has a message for j at any time unit. The inverse of SPM measures the closeness of the friendship between i and j. Each node constructs a friend community out of its encounter history, which incorporates nodes that have friendship higher than a threshold with itself. To account indirect friendship between nodes, e.g., node i and k never meet but have a close common friend j, a conditional SPM (CSPM) betweeb i and k is defined as the average time it takes j to encounter k and send it a message received from i. Then the indirect friendship between i and k is defined as the inverse of the sum of SPM between i and j, and CSPM between i and k. If both the indirect friendship of i and k and friendship of i and j are higher than the threshold, then k is incorporated in the friend community of i. For different period of the day, each node computes a different friend community, since there is a temporal difference in the strength of friendships. When a node i has a message for node d encounters j, it forwards the message to j only if j's current friendship community includes d and j has a closer friendship with d than i.

Habit[92] utilizes information of user co-location at the physical layer and social relation at the application layer to help content dissemination. At the physical layer, each node detects its familiar strangers that it meets regularly, and maintains a regularity table containing the regularity weight for each familiar stranger. At the application layer, each node keeps a list of source nodes

whose content it is interested in. The list and the regularity table are exchanged with other nodes upon contact up to a certain number of hops (maxHops). Then, each node can construct an interest graph and a regularity graph using the collected information. When a source has message to send, it first select nodes in the interest graph that are interested in its content as recipients. For each recipient, the source computes all paths from the regularity graph, and select the path with least nodes that are uninterested in its content. If multiple paths exist, the source node computes the regularity weight (the minimum among all edges) for each path, and selects the path with the highest regularity weight. For possible recipients that are not in the source's interest graph, nodes who have received the message check if there is any node in their interest graphs is interested in the message. If yes, these nodes follow the steps that the source did, and send the message to the recipients.

PrefCast is a preference-aware content dissemination protocol for MSNs [93]. The authors consider that multiple content objects are shared among users via opportunistic contacts. In PrefCast, time is slotted, and during each time-slot, only an object can be transmitted by a user to all its neighbors. Users has different utility on different content object. For each user, it may hold multiple content objects, the problem of when to forward which object is formulated as an optimization problem, where the objective is to maximize the total utility of all mobile users. For each object it holds, a user estimates a future utility contribution which is determined by three factors: 1) the probability that this user meets others who do not have this object, 2) the utilities of its contacts on this object, and 3) the probability that the user drops this object due to limited buffer space. Using greedy approximation, the user can compute an approximate optimal schedule for all object transmissions so that the total future utility contribution is maximized.

In [94], Gao et al. propose a relay selection mechanism for multicasting in MSNs, based on social community structure and ego-centric centrality. The contact process of each node pair is formulated as a Poisson process. And based on the Poisson modeling, a centrality metric called cumulative contact probability is defined to represent the probability of a node to contact others within a given time. Basically, data source selects some nodes it contacts as relays, which further deliver the data to destinations. The relay selection is formulated as a knapsack problem, where the objective is minimizing the number of relays subject to that the average ratio of data being delivered to destinations is higher than a given p. Two multicast scenarios are considered, namely, single-data multicast and multiple-data multicast. For single-data multicast, the destinations are assumed to be uniformly distributed, and the data source selects the relays such that all other nodes in the network can be contacted by the selected relays. For multiple-data multicast, each node is assumed to belong to at least one community, and it needs to maintain social forwarding paths with the highest forwarding probability to all other nodes in the same community. The knapsack problem of the relay selection is solved by a two-stage heuristic. The selected relays will send the data to the destinations within the same community. Destinations in other communities can get data through gateway nodes which belong

to multiple communities.

FairRoute is proposed to overcome unfair load distribution in routing - most of the traffic goes through a small subset of users with high centrality [95]. This unfairness can drain the limited resources of these users' devices and degrade the robustness of the network due to random failures and attacks. First, a heuristic forwarding mechanism is a designed based on the notion of perceived interaction strength [96], message is forwarded to nodes such that the forwarding utility is greedily maximized. Then, the problem of unfair load distribution is handled by an assortative-based queue control performed by each node. The assortativeness (or homophyly)[97] refer to the behaviour that people tend to interact with others of similar characteristics as themselves. In the assortative-based queue control, every node uses the size of its queue length to represent its social status, and it only accepts forwarding request from nodes that have equal or higher social status. This limits the number of messages that nodes with high social status have to carry and forward. Nodes with low social status have to find alternative paths in the network.

DAC is a cooperative caching technique aiming to improve the data accessability [98]. It is assumed that the contact duration between nodes follows Pareto distribution. Since the contact duration is limited, a complete data may not be transmitted during a contact. Therefore, the data is divided into s packets using the random linear network coding, so that the receiver can recover the data using any s linearly independent coded packets. The distributed k-clique algorithm [32] is used to identify the communities in the network. For each community, a cooperative caching problem is formulated as an optimization problem, that is, each node decides how many packets of the data should be cached so that the total caching benefit of the community is maximized, subject to the caching cost and caching space limit. The caching benefit of a node is defined as the expected amount of data that this node can send to a requester before the request becomes invalid. Based on the caching benefit, a new centrality called marginal caching benefit is defined as the gain in the caching benefit from caching an additional packet. It is proved that when the contact duration is limited, node's centrality decreases with the increase of packets it caches. The optimization problem is solved by a greedy algorithm. When two nodes meet, the node with higher centrality caches the packets until it reaches the caching space limit.

Give2Get is a mechanism that motivates selfish individuals to truthfully relay messages for others in epidemic forwarding and delegation forwarding [99]. This mechanism is based on public key cryptography[100]. Each user has a public key signed by a trusted authority, and its corresponding private key. In Give2Get, the destination of a message is hidden to every possible relay except the destination. When a sender S finds a candidate B to relay message m, it first request B if it has handled a message with hash $H(m)$. B would not lie on this, since it does not know the destination of the message, which might be itself. If B agrees to relay the message, it will reply S with a proof of relay. In return, S send B the key for the message m, which allows B to know the destination of the message. If B finds it is not the destination, it will relay the

message rather than cheat S by discarding the message, because it is asked by S to provide two proofs of relay during next contact. If B fails to do so, S can broadcast a proof of misbehavior message to the whole network. Other nodes who receives this message will exclude B from future forwarding path.

A summary of the above-discussed networking and resource management algorithms for mobile social networks is presented in Table 5.

4.4.3. Local Resource Management

Battery, bandwidth, storage and such are limited resources on mobile devices. The way those resources being utilized has significant impact on the overall performance of the network. Since mobile devices are normally private properties of their masters, they have full authority to manage these resources on their own devices. A local resource management module is necessary to monitor, allocate, and scedule the usage of these resources on individual user's behalf.

In some previous work, resource (in particular, storage) mamagement was incorporated in the forwarding mechanism [95, 101]. For example, in the FairRoute [95] presented above, authors designed a queue control mechanism to avoid users carry out too many data forwards and handovers which may quickly deplete the battery and storage. In this mechanism, social status of a node is defined as its queue length. Upon arrival of a forwarding request, nodes check the social status of the requester, and only accept it if the requester has equal or higher status. Thus, the volume of data forwards is significantly reduced by comparing nodes social status.

Guardalben et al. considered the chances of MAC layer association as limited local resource, and proposed a social metric based model to improve MAC layer association among users [102]. It is assumed that nodes in the network are grouped in communities, and nodes within each community are either directly associated or have multi-hop path between them. To form a community, nodes detect and identify other nodes nearby, and then create association between them. Different from node association based on the received signal strength indication and signal to noise ratio, several social metrics, such as neighborhood nodes friendship, associated nodes friendship and community nodes friendship, are adopted to be the association criteria. Specifically, the social-based association model is composed of two main blocks - local repositories and cooperation mechanisms. In the local repositories, Partial View stores information about neighbor nodes in vicinity, while Known Nodes stores information of nodes detected through other nodes. The cooperation mechanisms consist of Bootstrapping and Discovery. The Bootstrapping mechanism prepares basic information of a node before the association process, including identifier, MAC address, BSSID, hardware capabilities and social metric value. Then the Discovery starts exchanging MAC packets containing above information among neighbor nodes. After the Discovery process, each node associates with a neighbor node having the highest social metric.

Table 5: Summary of routing algorithms

Method	Communication type	Metrics used	Data cache	Communication Overhead
BUBBLE Rap[33]	Point to point	Betweenness centrality	No	Community detection and ranking exchange
SimBet[87]	Point to point	Egocentric betweenness centrality and similarity	No	Information (a list of encountered nodes and a list of destination nodes) exchange
SocialCast[90]	Publish/subscribe	Interest and utility	Yes	Periodical interests and utility broadcast
SPM[85]	Point to point	Social pressure	No	Friend discovery
Habit[92]	Point to point	Interest and regularity	No	Familiar stranger detection
PrefCast[93]	Broadcast	Future utility contribution	Yes	Preference exchange
Ego[94]	Multicast	Ego-centric centrality	No	Contact discovery and centrality request
FairRoute[95]	Point to point	Interaction strength and assortativeness	No	Update of perceived interaction strengths
DAC[98]	Point to point	Marginal caching benefit	Yes	Community Detection, and exchange of local information and Community Information Table
Give2Get[99]	Point to point	Nash equilibrium	No	Negotiation of cryptographic session key and proof collection

4.5. Privacy & Security Management

Users in MSNs tend to expose an astonishing amount of personal information, such as physical location and interests, either for sharing information with friends or for making new connections. In online MSNs, those personal data are stored in the servers of service providers, which empower them to share the information with, for example, advertisers and benefit form it [103]. While in decentralized MSNs, the lack of central infrastructure, for example, trusted third party issuing key materials, makes security management a harder task. Moreover, the broadcast nature of wireless communication also render the network susceptible to attacks like eavesdropping[104]. In this section, we present the latest work on privacy preserving and security mechanisms in MSNs.

Dong et al. developed a secure proximity computation protocol for secure friend discovery in MSNs [105]. When two users are in physical vicinity, they can exchange social coordinates (e.g., a user's attributes) and compute social proximity between them by doing a dot product operation on the vectors of their coordinates. A social proximity exceeding a threshold indicates that they are potential friends. Authors identified several potential attacks on the friend discovery, including user fingerprinting based on its coordinate or proximity, user tracking based on its coordinate, and proximity falsifying. To prevent the attack of user tracking, each user is allowed to use a virtual ID, which is assigned by trusted server and valid for a short term. To preserve the privacy of user's social coordinate and proximity, a proximity pre-filtering is used to efficiently identify potential friends from all users. The pre-filtering can quickly compute whether the dot product (proximity of users) is above a threshold or not without revealing the value of the dot product to them. If the computed proximity is above the threshold, the validity of social coordinates and proximity is checked using a homomorphic cryptography based private and verifiable proximity computation, to see whether they are genuine or forged.

MobID is a decentralized defence mechanism against sybil attacks in networks formed by portable devices [106]. A malicious node can defraud honest nodes of their trust, and by creating bogus identities, it can disrupt the services (e.g., file sharing) provided by the network without trace, this is what called sybil attack. MobID defences against sybils by excluding the bogus identities created by malicious individuals rather than excluding malicious individuals. Mobile users running MobID identify themselves using public keys, and they exchange their keys only with friends. Suppose an honest node A has received a friending request from node B, it has to decide whether to accept it or not. First, A requests B's list of friends, which includes each of its friends' identifier and signature on the relationship with B. Upon receiving B's list, A updates its network of friend by absorbing B's list of friends, and computes a normalized random-walk betweenness of B as B's GoodRank in the network. The higher its GoodRank, the more likely that B is honest. In case that B is malicious and some node in A's network of friend has accepted B as friend, which means B can boost its GoodRank by creating several bogus identities, A also incorporates B's list of friends into the network of foes and computes a BadRank of B. With the GoodRank and BadRank of B, A can group B into a 'sybil set' or

'honest set' by applying the K-means clustering [107]. If B is in the 'sybil set', it will not be accepted by A.

In [108], He et al. presented an investigation on location cheating in location-based MSN services. Location-based MSN services such as Foursquare rewards user who checks in a venue frequently enough. Some dishonest users may check-in the venue without truly being there, in order to obtain the reward. The authors introduced several novel location cheating attacks to achieve this. One possible attack is location cheating against GPS verification provided by location-based services (e.g., cheater code of Foursquare). Attackers can provide fake GPS coordinates and fool the client application by modifying the GPS module or GPS APIs, and using device emulator. Another more sophisticated attack is called automated cheating. First, attackers need to crawl data of users' profiles and venues' profiles from Foursquare. Then, attackers create a number of fake users, select a list of venues and organize them into a schedule in order to let the fake users check-in the venues automatically. To address those attacks, several possible solutions are suggested, including distance bounding [109], address mapping[110], venue side location verification[111], access control for crawling, and hiding information from profiles.

Krishna and Zhao designed an approach for preserving users' location privacy in location-based social networks [112]. They argue that LBSN fails to protect location privacy because applications running on servers use users' location data in plain-text in order to provide services. The basic idea of their approach is to move the application functionality (e.g., compute the distance between two location) to the client devices and treat the untrusted third-party servers as encrypted data stores. There are two building blocks in the proposed approach, namely, friendship proofs (FProofs) and transaction proofs (TProof). Users identify their friends by exchanging FProofs offline (e.g., via Bluetooth). The FProof that a user A gives user B consists of some content and A's signature on the harsh of the content. The content contains A's public key, B's public key, time of issue, and A's symmetric session key. Unlike FProof, the TProof is stored on the server, it consists of a message encrypted with the session key, and A's signature on the harsh of the message. A stranger to user A is not able to decrypt the TProof, since it does not have public key and session key of A, which are shared only within friends.

FindU is a privacy-preserving scheme for personal profile matching in MSNs [113]. Each user's profile consists of a set of attributes, such as hobbies and places. The more two users' attribute sets intersect, the better these two profiles match. Using profile matching, users can make new connections with other users in proximity. To meet users' different privacy requirements, three levels of privacy are defined. In privacy level 1, matching initiator and a candidate can learn their intersection attributes. In privacy level 2, matching initiator and a candidate can only learn the size of intersection, i.e., how many attributes are intersected. However, in privacy level 3, the matching initiator and each candidate only know its own rank of the intersection size among all candidates. It is assumed that communication channel used by users is secured, such as by using public/private key pair if possible. The ideas to achieve the proposed

privacy levels is based on private set-intersection (PSI) techniques [114]. To improve the efficiency, FindU also adopts Shamir secret sharing scheme [115] and secure multi-party computation.

A summary of the above-reviewed privacy and security mechanisms for MSNs is presented in Table 6. For a more comprehensive overview on the challenges and solutions for security and privacy in MSN as well as most recently developments, see [116] and references therein and [117, 118].

5. Concluding Remarks

Mobile social networks are highly likely to be an indispensable part of people's social life. Unfortunately, there are still problems not well addressed. There is a common assumption made by most of the reviewed papers for delay tolerant MSNs, that is, somehow, the source of the information knows the destination(s) of the information. However, this assumption has never been justified. In such a source-destination based conversation model, we believe that a user most likely only send information to other users whose identities have been verified during physical encounter. Although users' identifiers can be obtained from intermediate nodes, but she can not be sure whether this is her really friends. Some studies [112] suggest that the identifiers can be exchanged by other means such as email and use identifiers from existing OSNs [20], however, this is impossible in fully delay-tolerant setting.

This source-destination based conversation model not only restricts the usability of delay tolerant MSNs in large scale but also generate useless duplicate messages in the network. An intermediate node tend to store messages for different destinations, these messages are likely to be the copies of the same message if a source wants to send a message to multiple friends or this message is so popular that many users would like to share it with others. But the intermediate node is not able to distinguish the content of these messages, if the messages are concealed in several envelops with specified destinations. In this case, the store space or the intermediate nodes are severely wasted.

Perhaps we should treat MSNs as information-centric networks (ICNs)[119, 120]. In ICNs, requests and responses are decoupled in both space and time by employing the publish/subscribe paradigm. The content is cached at each level of the network (e.g., routers and hosts) to decrease the transmission traffic and response time. Most importantly, each piece of the content has a unique name to describe itself. To authenticate the content, the name also includes a cryptographic hash. Someone may have noticed that the store-and-forward technique used in MSNs is essentially the same with the in-network caching. But we also believe that the network performance can be improved if we adopt the publish/subscribe paradigm and content naming as well. Using the publish/subscribe paradigm, the distribution of the content will not be constrained, any user having the interest matching the attribute of the content can get it. If the messages are named, the intermediate nodes can distinguish the messages, keep only one copy of the same messages having the same content and thus save storage space. Though it seems to be a fine solution, there are a number

Table 6: Summary of privacy and security mechanisms

Mechanism	Type /Attack model	Technique used	Third party	Applicability
Secure Friend [105]	Secure proximity computation, fingerprinting, and user tracking	Virtual ID, digital signatures, proximity pre-filtering, and homomorphic encryption	Yes	Mode 1 and 3
MobID[106]	Sybil attack	Public key authentication and normalized random-walk betweenness	No	Mode 2
Location[108]	Location cheating	Distance bounding, address mapping, venue side location verification, access control for crawling, and hiding information from profiles	Yes	Mode 1
Proofs[112]	User location privacy	Offline exchanging of FProofs (public key, time of issue, and session key) and signature	Yes	Mode 1 and 3
FindU[113]	Privacy of personal profile	Private set-intersection, Shamir secret sharing scheme and secure multi-party computation	No	Mode 2

of following issues to be addressed, such as how to authenticate the message without trusted authority, do the intermediate nodes have to store the whole file or only part of it.

Acknowledgments

The work is supported in part by the European FP7 project: CLIMBER (PIRSES-GA-2012-318939). The authors would also like to thank our project partners Kun Yang (UEssex), Hao Yin (THU), Xueqi Cheng (ICT-CAS), Minyi Guo (SJTU), and Bin Yao (SJTU) for discussion and help in preparing the paper.

References

[1] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, ACM, 2003, pp. 27–34.

[2] E. Cho, S. A. Myers, J. Leskovec, Friendship and mobility: user movement in location-based social networks, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 1082–1090.

[3] C.-m. A. Yeung, I. Liccardi, K. Lu, O. Seneviratne, T. Berners-Lee, Decentralization: The future of online social networking, in: W3C Workshop on the Future of Social Networking Position Papers, Vol. 2, 2009, pp. 2–7.

[4] T. Paul, S. Buchegger, T. Strufe, Decentralizing social networking services, in: International Tyrrhenian Workshop on Digital Communications, 2010, pp. 1–10.

[5] N. Jabeur, S. Zeadally, B. Sayed, Mobile social networking applications, Communications of the ACM 56 (3) (2013) 71–79.

[6] B. Han, P. Hui, V. Kumar, M. V. Marathe, G. Pei, A. Srinivasan, Cellular traffic offloading through opportunistic communications: a case study, in: Proceedings of the 5th ACM workshop on Challenged networks, ACM, 2010, pp. 31–38.

[7] N. Kayastha, D. Niyato, P. Wang, E. Hossain, Applications, architectures, and protocol design issues for mobile social networks: A survey, Proceedings of the IEEE 99 (12) (2011) 2130–2158.

[8] N. Vastardis, K. Yang, Mobile social networks: Architectures, social properties, and key research challenges, Communications Surveys & Tutorials, IEEE 15 (3) (2013) 1355–1371.

[9] A. Beach, M. Gartrell, R. Han, Solutions to security and privacy issues in mobile social networking, in: Computational Science and Engineering, 2009. CSE'09. International Conference on, Vol. 4, IEEE, 2009, pp. 1036–1042.

[10] Y. Zheng, Y. Chen, X. Xie, W.-Y. Ma, Geolife2.0: A location-based social networking service, in: Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on, 2009, pp. 357–358.

[11] Proximity based mobile social networking, http://www.researchandmarkets.com/reports/1945613/proximity_based_mobile_social_networking.

[12] Tecent q2 2013 earnings, http://www.tencent.com/en-us/content/ir/news/2013/attachments/20130814.pdf.

[13] J. Teng, B. Zhang, X. Li, X. Bai, D. Xuan, E-shadow: Lubricating social interaction using mobile phones, in: Distributed Computing Systems (ICDCS), 2011 31st International Conference on, 2011, pp. 909–918.

[14] Z. Liu, Y. Feng, B. Li, Socialize spontaneously with mobile applications, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1942–1950.

[15] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, A. Schmidt, Micro-blog: sharing and querying content through mobile phones and social participation, in: Proceedings of the 6th international conference on Mobile systems, applications, and services, MobiSys '08, ACM, New York, NY, USA, 2008, pp. 174–186.

[16] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, A. T. Campbell, Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application, in: Proceedings of the 6th ACM conference on Embedded network sensor systems, SenSys '08, ACM, New York, NY, USA, 2008, pp. 337–350.

[17] I. Lequerica, M. G. Longaron, P. M. Ruiz, Drive and share: efficient provisioning of social networks in vehicular scenarios, Communications Magazine, IEEE 48 (11) (2010) 90–97.

[18] S. Smaldone, L. Han, P. Shankar, L. Iftode, Roadspeak: enabling voice chat on roadways using vehicular social networks, in: Proceedings of the 1st Workshop on Social Network Systems, SocialNets '08, ACM, New York, NY, USA, 2008, pp. 43–48.

[19] A. Gupta, A. Kalra, D. Boston, C. Borcea, Mobisoc: a middleware for mobile social computing applications, Mob. Netw. Appl. 14 (1) (2009) 35–52.

[20] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, C. Diot, Mobiclique: middleware for mobile social networking, in: Proceedings of the 2nd ACM workshop on Online social networks, WOSN '09, ACM, New York, NY, USA, 2009, pp. 49–54.

[21] A. Toninelli, A. Pathak, V. Issarny, Yarta: A middleware for managing mobile social ecosystems, in: J. Riekki, M. Ylianttila, M. Guo (Eds.), Advances in Grid and Pervasive Computing, Vol. 6646 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 209–220.

[22] B. Furht, B. Furht, Handbook of social network technologies and applications, Springer, 2010.

[23] M. O. Jackson, Social and economic networks, Princeton University Press, 2010.

[24] N. P. Nguyen, T. N. Dinh, S. Tokala, M. T. Thai, Overlapping communities in dynamic networks: their detection and mobile applications, in: Proceedings of the 17th annual international conference on Mobile computing and networking, ACM, 2011, pp. 85–96.

[25] N. Nguyen, T. Dinh, Y. Xuan, M. Thai, Adaptive algorithms for detecting community structure in dynamic social networks, in: INFOCOM, 2011 Proceedings IEEE, 2011, pp. 2282–2290.

[26] S. Fortunato, C. Castellano, Community structure in graphs, in: Computational Complexity, Springer, 2012, pp. 490–512.

[27] W. Chen, Z. Liu, X. Sun, Y. Wang, A game-theoretic framework to identify overlapping communities in social networks, Data Mining and Knowledge Discovery 21 (2) (2010) 224–240.

[28] A. Clauset, Finding local community structure in networks, Physical review E 72 (2) (2005) 026132.

[29] A.-K. Pietiläinen, C. Diot, Dissemination in opportunistic social networks: the role of temporal communities, in: Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing, ACM, 2012, pp. 165–174.

[30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.

[31] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75–174.

[32] P. Hui, E. Yoneki, S. Y. Chan, J. Crowcroft, Distributed community detection in delay tolerant networks, in: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture, ACM, 2007, p. 7.

[33] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: Social-based forwarding in delay-tolerant networks, Mobile Computing, IEEE Transactions on 10 (11) (2011) 1576–1589.

[34] F. Li, J. Wu, Localcom: a community-based epidemic forwarding scheme in disruption-tolerant networks, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on, IEEE, 2009, pp. 1–9.

[35] B. Han, A. Srinivasan, Your friends have more friends than you do: identifying influential mobile users through random walks, in: Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing, ACM, 2012, pp. 5–14.

[36] S. L. Feld, Why your friends have more friends than you do, American Journal of Sociology (1991) 1464–1477.

[37] M. C. Gonzalez, C. A. Hidalgo, A.-L. Barabasi, Understanding individual human mobility patterns, Nature 453 (7196) (2008) 779–782.

[38] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Pocket switched networks and human mobility in conference environments, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005, pp. 244–251.

[39] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, S. Chong, On the levy-walk nature of human mobility, IEEE/ACM Transactions on Networking (TON) 19 (3) (2011) 630–643.

[40] D. Wang, D. Pedreschi, C. Song, F. Giannotti, A.-L. Barabasi, Human mobility, social ties, and link prediction, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 1100–1108.

[41] B. Jiang, J. Yin, S. Zhao, Characterizing the human mobility pattern in a large street network, Phys. Rev. E 80 (2009) 021136.

[42] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, C. Volinsky, Human mobility characterization from cellular network data, Commun. ACM 56 (1) (2013) 74–82.

[43] A. Wesolowski, N. Eagle, A. J. Tatem, D. L. Smith, A. M. Noor, R. W. Snow, C. O. Buckee, Quantifying the impact of human mobility on malaria, Science 338 (6104) (2012) 267–270.

[44] B. D. Dalziel, B. Pourbohloul, S. P. Ellner, Human mobility patterns predict divergent epidemic dynamics among cities, Proceedings of the Royal Society B: Biological Sciences 280 (1766).

[45] S. K. Fayazbakhsh, Modeling human mobility and its applications in routing in delay-tolerant networks: a short survey, arXiv preprint arXiv:1307.1926.

[46] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on opportunistic forwarding algorithms, Mobile Computing, IEEE Transactions on 6 (6) (2007) 606–620.

[47] S. Merler, M. Ajelli, The role of population heterogeneity and human mobility in the spread of pandemic influenza, Proceedings of the Royal Society B: Biological Sciences 277 (1681) (2010) 557–565.

[48] P. Hui, J. Crowcroft, Human mobility models and opportunistic communications system design, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 366 (1872) (2008) 2005–2016.

[49] N. Eagle, A. (Sandy) Pentland, Reality mining: sensing complex social systems, Personal Ubiquitous Comput. 10 (4) (2006) 255–268.

[50] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Pocket switched networks and human mobility in conference environments, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005, pp. 244–251.

[51] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding, Tech. rep., Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Laboratory (2005).

[52] V. Erramilli, A. Chaintreau, M. Crovella, C. Diot, Diversity of forwarding paths in pocket switched networks, in: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC '07, ACM, New York, NY, USA, 2007, pp. 161–174.

[53] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, in: Proceedings of the 10th annual international conference on Mobile computing and networking, MobiCom '04, ACM, New York, NY, USA, 2004, pp. 187–201.

[54] M. McNett, G. M. Voelker, Access and mobility of wireless pda users, ACM SIGMOBILE Mobile Computing and Communications Review 9 (2) (2005) 40–55.

[55] X. X. Yu Zheng, Geolife: Building social networks using human location history, `http://research.microsoft.com/en-us/projects/geolife/` (August 2012).

[56] I. R. N. M. Shin, S. H. N. K. Lee, S. Chong, Human mobility patterns and their impact on routing in human-driven mobile networks.

[57] B. Jiang, J. Yin, S. Zhao, Characterizing the human mobility pattern in a large street network, Phys. Rev. E 80 (2009) 021136.

[58] C.-A. La, P. Michiardi, Characterizing user mobility in second life, in: Proceedings of the first workshop on Online social networks, ACM, 2008, pp. 79–84.

[59] D. Karamshuk, C. Boldrini, M. Conti, A. Passarella, Human mobility models for opportunistic networks, Communications Magazine, IEEE 49 (12) (2011) 157–165.

[60] T. Karagiannis, J.-Y. Le Boudec, M. Vojnović, Power law and exponential decay of intercontact times between mobile devices, Mobile Computing, IEEE Transactions on 9 (10) (2010) 1377–1390.

[61] T. Hu, B.-L. Wenning, C. Görg, U. Toseef, Z. Guo, Statistical analysis of contact patterns between human-carried mobile devices, in: Mobile Networks and Management, Springer, 2013, pp. 244–257.

[62] D. Brockmann, L. Hufnagel, T. Geisel, The scaling laws of human travel, Nature 439 (7075) (2006) 462–465.

[63] C. Boldrini, A. Passarella, Hcmm: Modelling spatial and temporal properties of human mobility driven by users social relationships, Computer Communications 33 (9) (2010) 1056–1074.

[64] C. Song, T. Koren, P. Wang, A.-L. Barabási, Modelling the scaling properties of human mobility, Nature Physics 6 (10) (2010) 818–823.

[65] M. Kim, D. Kotz, S. Kim, Extracting a mobility model from real user traces., in: INFOCOM, Vol. 6, 2006, pp. 1–13.

[66] G. Viswanathan, V. Afanasyevt, Lévy flight search patterns of, Nature 381 (1996) 30.

[67] I. B. Aban, M. M. Meerschaert, A. K. Panorska, Parameter estimation for the truncated pareto distribution, Journal of the American Statistical Association 101 (473) (2006) 270–277.

[68] K. Lee, S. Hong, S. J. Kim, I. Rhee, S. Chong, Slaw: A new mobility model for human walks, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 855–863.

[69] A. Mei, J. Stefa, Swim: A simple model to generate small mobile worlds, in: INFOCOM 2009, IEEE, 2009, pp. 2106–2113.

[70] D. Fischer, K. Herrmann, K. Rothermel, Gesomoa general social mobility model for delay tolerant networks, in: Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on, IEEE, 2010, pp. 99–108.

[71] F. Lassabe, D. Charlet, P. Canalda, P. Chatonnay, F. Spies, Predictive mobility models based on kth markov models, in: IEEE int conf on pervasive services, 2006, pp. 303–306.

[72] S. Gambs, M.-O. Killijian, M. N. n. del Prado Cortez, Next place prediction using mobility markov chains, in: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, MPM '12, ACM, New York, NY, USA, 2012, pp. 3:1–3:6.

[73] D. Ashbrook, T. Starner, Learning significant locations and predicting user movement with gps, in: Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on, IEEE, 2002, pp. 101–108.

[74] A. Asahara, K. Maruyama, A. Sato, K. Seto, Pedestrian-movement prediction based on mixed markov-chain model, in: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11, ACM, New York, NY, USA, 2011, pp. 25–33.

[75] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, An efficient k-means clustering algorithm: analysis and implementation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (7) (2002) 881–892.

[76] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, L. Terveen, Discovering personal gazetteers: an interactive clustering approach, in: Proceedings of the 12th annual ACM international workshop on Geographic information systems, ACM, 2004, pp. 266–273.

[77] W. Mathew, R. Raposo, B. Martins, Predicting future locations with hidden markov models, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 911–918.

[78] P. Z. Kunszt, A. S. Szalay, A. R. Thakar, The hierarchical triangular mesh, in: Mining the sky, Springer, 2001, pp. 631–637.

[79] M. Al-Hattab, M. Takruri, J. Agbinya, Mobility prediction using pattern matching.

[80] Cross-correlation, `http://en.wikipedia.org/wiki/Cross-correlation#cite_ref-2`.

[81] A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti, Wherenext: a location predictor on trajectory pattern mining, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 637–646.

[82] S. Kotsiantis, D. Kanellopoulos, Association rules mining: A recent overview, GESTS International Transactions on Computer Science and Engineering 32 (1) (2006) 71–82.

[83] A. Sadilek, H. Kautz, J. P. Bigham, Finding your friends and following them to where you are, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, 2012, pp. 723–732.

[84] S. Ioannidis, A. Chaintreau, L. Massoulié, Optimal and scalable distribution of content updates over a mobile social network, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 1422–1430.

[85] E. Bulut, B. K. Szymanski, Friendship based routing in delay tolerant mobile social networks, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, IEEE, 2010, pp. 1–5.

[86] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, U. C. Kozat, Proactive seeding for information cascades in cellular networks, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1719–1727.

[87] E. M. Daly, M. Haahr, Social network analysis for routing in disconnected delay-tolerant manets, in: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2007, pp. 32–40.

[88] P. V. Marsden, Egocentric and sociocentric measures of network centrality, Social networks 24 (4) (2002) 407–422.

[89] M. Everett, S. P. Borgatti, Ego network betweenness, Social networks 27 (1) (2005) 31–38.

[90] P. Costa, C. Mascolo, M. Musolesi, G. P. Picco, Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks, Selected Areas in Communications, IEEE Journal on 26 (5) (2008) 748–760.

[91] R. E. Kalman, et al., A new approach to linear filtering and prediction problems, Journal of basic Engineering 82 (1) (1960) 35–45.

[92] A. J. Mashhadi, S. Ben Mokhtar, L. Capra, Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks, in: World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a, IEEE, 2009, pp. 1–6.

[93] K.-J. Lin, C.-W. Chen, C.-F. Chou, Preference-aware content dissemination in opportunistic mobile social networks, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1960–1968.

[94] W. Gao, Q. Li, B. Zhao, G. Cao, Multicasting in delay tolerant networks: a social network perspective, in: Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing, ACM, 2009, pp. 299–308.

[95] J. M. Pujol, A. L. Toledo, P. Rodriguez, Fair routing in delay tolerant networks, in: INFOCOM 2009, IEEE, IEEE, 2009, pp. 837–845.

[96] M. S. Granovetter, The strength of weak ties, American journal of sociology (1973) 1360–1380.

[97] M. E. Newman, J. Park, Why social networks are different from other types of networks, Physical Review E 68 (3) (2003) 036122.

[98] X. Zhuo, Q. Li, G. Cao, Y. Dai, B. Szymanski, T. La Porta, Social-based cooperative caching in dtns: a contact duration aware approach, in: Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, IEEE, 2011, pp. 92–101.

[99] A. Mei, J. Stefa, Give2get: Forwarding in social mobile wireless networks of selfish individuals, Dependable and Secure Computing, IEEE Transactions on 9 (4) (2012) 569–582.

[100] A. Salomaa, Public-key cryptography, Vol. 23, Springer, 1996.

[101] Q. Li, S. Zhu, G. Cao, Routing in socially selfish delay tolerant networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[102] L. Guardalben, T. Gomes, P. Salvador, S. Sargento, Improving mac layer association through social-based metrics in mobile networks, Communications Magazine, IEEE 50 (6) (2012) 91–98.

[103] H. Gao, J. Hu, T. Huang, J. Wang, Y. Chen, Security issues in online social networks, Internet Computing, IEEE 15 (4) (2011) 56–63.

[104] R. Liu, H. V. Poor, Multiple antenna secure broadcast over wireless networks, arXiv preprint arXiv:0705.1183.

[105] W. Dong, V. Dave, L. Qiu, Y. Zhang, Secure friend discovery in mobile social networks, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 1647–1655.

[106] D. Quercia, S. Hailes, Sybil attacks against mobile users: friends and foes to the rescue, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–5.

[107] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (7) (2002) 881–892.

[108] W. He, X. Liu, M. Ren, Location cheating: A security challenge to location-based social network services, in: Distributed Computing Systems (ICDCS), 2011 31st International Conference on, IEEE, 2011, pp. 740–749.

[109] J. T. Chiang, J. J. Haas, Y.-C. Hu, Secure and precise location verification using distance bounding and simultaneous multilateration, in: Proceedings of the second ACM conference on Wireless network security, ACM, 2009, pp. 181–192.

[110] M. Balakrishnan, I. Mohomed, V. Ramasubramanian, Where's that phone?: geolocating ip addresses on 3g networks, in: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, ACM, 2009, pp. 294–300.

[111] A. Howard, S. Siddiqi, G. S. Sukhatme, An experimental study of localization using wireless ethernet, in: Field and Service Robotics, Springer, 2006, pp. 145–153.

[112] K. P. Puttaswamy, B. Y. Zhao, Preserving privacy in location-based mobile social applications, in: Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, ACM, 2010, pp. 1–6.

[113] M. Li, N. Cao, S. Yu, W. Lou, Findu: Privacy-preserving personal profile matching in mobile social networks, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 2435–2443.

[114] M. J. Freedman, K. Nissim, B. Pinkas, Efficient private matching and set intersection, in: Advances in Cryptology-EUROCRYPT 2004, Springer, 2004, pp. 1–19.

[115] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.

[116] X. Liang, K. Zhang, X. Shen, X. Lin, Security and privacy in mobile social networks: challenges and solutions, Wireless Communications, IEEE 21 (1) (2014) 33–41.

[117] X. Liang, X. Lin, X. S. Shen, Enabling trustworthy service evaluation in service-oriented mobile social networks, Parallel and Distributed Systems, IEEE Transactions on 25 (2) (2014) 310–320.

[118] K. Zhang, X. Liang, X. Shen, R. Lu, Exploiting multimedia services in mobile social networks from security and privacy perspectives, Communications Magazine, IEEE 52 (3) (2014) 58–65.

[119] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox, Information-centric networking: seeing the forest for the trees, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, ACM, 2011, p. 1.

[120] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, Communications Magazine, IEEE 50 (7) (2012) 26–36.