Data Modelling for Socially-Based Routing in Opportunistic Network

Radu-Ioan Ciobanu, Ciprian Dobre, and Fatos Xhafa

Abstract Opportunistic networks are the next step in the evolution of mobile networks, especially since the number of human-carried mobile devices such as smartphones and tablets has greatly increased in the past few years. They assume unselfish communication between devices based on a store-carry-and-forward paradigm, where mobile nodes carry each other's data through the network, which is exchanged opportunistically. In this chapter, we present opportunistic networks in detail and show various real-life scenarios where such networks have been successfully deployed or are about to be, such as disaster management, smart cities, wildlife tracking, context-aware platforms, etc. We highlight the challenges in designing successful data routing and dissemination algorithms for opportunistic networks, and present some of the most important techniques and algorithms that have been proposed in the past few years. We show the most important issues for each of them, and attempt to propose solutions for improving opportunistic routing and dissemination. Finally, we present what the future trends in this area of research might be, from information-centric networks to the Internet of Things.

Radu-Ioan Ciobanu
University Politehnica of Bucharest, Splaiul Independentei 313, Bucharest, Romania, e-mail: radu.ciobanu@cti.pub.ro

Ciprian Dobre
University Politehnica of Bucharest, Splaiul Independentei 313, Bucharest, Romania, e-mail: ciprian.dobre@cs.pub.ro

Fatos Xhafa
Universitat Politecnica de Catalunya, Girona Salgado 1-3, 08034 Barcelona, Spain, e-mail: fatos@lsi.upc.edu

1 Introduction

Opportunistic networks are extensions of the legacy Mobile Ad-Hoc Networks (MANETs) concept. Legacy MANETs are composed of mobile nodes that collaboratively set up a network plane by running a given routing protocol. Therefore, the sometimes implicit assumption behind MANETs is that the network is well connected, and nodes' disconnection is an exception to deal with. Most notably, if the destination of a given message is not connected to the network when the message is generated, then that message is dropped after a short time (i.e. the destination is assumed to not exist). Opportunistic networks are mobile wireless networks in which the presence of a continuous path between a sender and a destination is not assumed, since two nodes may never be connected to the network at the same time. The network is assumed to be highly dynamic, and the topology is thus extremely unstable and sometimes completely unpredictable. Nevertheless, the network must guarantee end-to-end delivery of messages despite frequent disconnections and partitions.

The opportunistic networking paradigm is particularly suitable to those environments which are characterized by frequent and persistent partitions. In the field of wildlife tracking, for example, some kinds of sensor nodes are used to monitor wild species. In these cases it is not easy (nor possible sometimes) to have connectivity among a source sensor node and a destination data collector node. This happens because the animals to be monitored move freely and there is no possibility to control them in such a way to favor connectivity. Opportunistic networks may also be exploited to bridge the digital divide. In fact, they can support intermittent connectivity to the Internet for underdeveloped or isolated regions. This can be obtained by exploiting mobile nodes that collect information to upload to the Internet as well as requests for Web pages or any kind of data that needs to be downloaded from the Internet. Both data and requests are uploaded to and downloaded from the Internet once the mobile data collector node reaches a location where connectivity is available.

This chapter focuses on the particular problem of data gathering in such challenged networks. We describe different alternatives and solutions to routing the data from source to its destination. Today different techniques could be employed for mobile data gathering. A basic strategy would be to only allow data delivery when mobile devices are in direct proximity of the sinks. This technique has very little communication overhead, given that messages are only sent directly from the sensor node generating messages to the sinks. However, depending on how frequently mobile nodes meet the sinks, the delivery of the data might be very poor. This is particularly true if the sinks are very few and spread out. More refined techniques include epidemically-inspired approaches, which would randomly spread the data over the network, so that eventually a sink could be reached. We analyze both these worlds, highlighting specific problems and solutions to solve them in concrete case scenarios.

The rest of this chapter is organized as follows. We first introduce in more detail the domain of opportunistic mobile communication. We present case studies where opportunistic networks are already being successfully deployed and used, and highlight specific issues regarding their implementation in particular scenarios. In Section 3 we perform an analysis of the latest advances in data routing and dissemination solutions based on the use of opportunistic mobile networks. We highlight the main issues of each proposal, and attempt to give solutions to some of them in Section 4. Section 5 presents future trends, and Section 6 concludes our study.

2 Opportunistic Networks

This section presents a detailed view regarding opportunistic networks (including definitions, benefits and challenges), as well as a presentation of real-life use cases where they can bring a significant contribution.

2.1 Definition

Opportunistic networks (ONs) are a natural evolution of MANETs, where most (or sometimes all) of the nodes are mobile wireless devices. These devices range from small wireless-capable sensors to smartphones and tablets. The evolution from MANETs to ONs was necessary because opportunistic networks help transmit data horizontally, i.e. using costless inter-device transmissions, taking advantage of the already-existent device interaction. Moreover, ONs help disseminate data and decongest currently existing backend protocols by using short-range communication over IEEE 802.11, Bluetooth, ZigBee, etc. They are also considered to be the solution that will offer vehicle-to-vehicle communication for future Vehicular Ad-Hoc Networks (VANETs). The composing nodes of an ON have no knowledge of the shape of the network when they join it. They are only aware of other nodes that they come in close proximity to, depending on the radius of their communication mechanism (e.g. WiFi, Bluetooth, NFC, etc.). Thus, no assumptions are made regarding the existence of paths between nodes, since a network topology isn't known by any node (and even if it would be known, it would much too dynamic to be used in routing mechanisms, since nodes are highly mobile and they hardly stay in the same place for long periods of time).

ONs are based on a paradigm entitled store-carry-and-forward [39]. This implies that a node begins by storing some local data, which was either generated by itself, or was received from another node. Since ONs are characterised by a high degree of mobility, nodes move around a lot, thus carrying the stored data around the network. Finally, when the destination for a particular data item[1] is encountered, a forwarding process is started in order to send the data to the destination. However, encountering a message's destination is not the only situation where a data forward occurs. Opportunistic networks are based on the altruism of nodes: it is not enough for a node

[1] In opportunistic networks, data items are generally called messages. From this point on, we will refer to them thusly, making a distinction only where the situation requires it.

to see only to its interest, instead it must help other encountered nodes transfer their data (in exchange for them carrying the node's messages as well). This mutual help is the key to ONs and helps ensure that the data is spread through the network as much as possible, thus increasing the probability of a message to reach its intended destination, and decreasing the time it takes to do so.

There are two important parameters that describe the shape and behavior of an opportunistic network: contact time and inter-contact time [9]. The contact time is the duration of a contact between two encountering nodes, and it represents the time window when the nodes may exchange data. Higher contact times lead to the opportunity of exchanging more data between the nodes, but may also mean that the two nodes are static and the chance of them encountering other nodes and spreading the data is lower. Inter-contact time is the duration between two consecutive contacts of the same pair of nodes, and it offers an indication of the familiarity between the two nodes. If they have low inter-contact times, they meet each other very often, so they probably move around in the same geographical areas. The two parameters can be extended to the entire network, as any-contact time and inter-any-contact time [28]. The any-contact time is the duration of a node's encounter with any other node, whereas the inter-any-contact time represents the time between a node's sighting of any two nodes. The sparser the network, the lower the inter-any-contact time is, which leads to few forwarding opportunities, so the messages have a lower chance of reaching their intended destinations.

Opportunistic networks have been thoroughly studied in [39]. The authors offer a definition of ONs and present several realistic case studies of opportunistic networks that have been successfully deployed, including pocket-switched networks (PSNs) in the Haggle project[2], wildlife monitoring and ONs for intermittent Internet connectivity (which we'll discuss in more detail in Section 2.3). Furthermore, the paper also analyzes several ON routing and forwarding algorithms, while proposing a taxonomy used to classify them. They are split into algorithms with and without infrastructure. The infrastructure-based algorithms may be based on dissemination or on context, whereas the algorithms without an infrastructure are split based on their infrastructure type (fixed or mobile). Another detailed study of ONs has been performed by Conti et al. [17]. Their paper describes opportunistic networks, stating that the understanding of human mobility is paramount to designing efficient protocols for opportunistic networking. Conti et al. also discuss ON architecture, forwarding algorithms, data dissemination, security and applications, and conclude their work by observing that there is a strong link between opportunistic networking and mobile social networks. The authors also show that ONs can be used for both point-to-point communication, as well as data dissemination.

[2] http://www.haggleproject.org.

2.2 Challenges

Aside from the benefits of opportunistic networks and their applicability in real-life (presented in Section 2.3), there are several challenges that must be taken into consideration when designing an ON-based solution. The first and most important caveat of ONs is that the lack of connectivity at all times leads to a potential lack of end-to-end paths. In other words, deploying an opportunistic network means accepting the fact that not all messages may successfully reach their destinations, or when they do, reach them with high delays. As stated in Section 2.1, there are many factors that can affect an ON's hit rate[3], ranging from the number of devices in the network to the behavior and social grouping of the device's owners (if we are dealing with an ON where the nodes are humans carrying mobile devices). Opportunistic network administrators must be aware of this and only use such networks where delays and loss of messages are acceptable. The purpose of every ON routing or dissemination algorithm is to increase the hit rate, because higher hit rates make ONs much more likely to be successfully used in real-life scenarios.

Closely related to the first challenge is the decision of selecting a message's next hop. Ideally, each node should have access to the future behavior of the entire network, and thus choose the shortest path between it and a message's destination, similar to what is done for classical static networks. Unfortunately, this is not the case for opportunistic networks, and this is why researchers are still proposing new methods of deciding whether a message should be exchanged when two nodes are in range of each other. Aside from selecting the next hop, decisions should be made regarding the amount of copies a message should have in the network, and whether it should be kept or deleted by the originating node. Various methods for routing or dissemination algorithms have been proposed over the years, and the most successful ones are presented in Section 3, along with their (still existing) issues.

Another aspect that should be taken into consideration when deploying an opportunistic network is that, since the nodes are generally mobile devices, they have a limited life until they need to be recharged. The more data transfers are performed in a short period of time, the quicker a device's battery consumes, which in turn leads to removing it from the network for a certain period of time (until it is recharged). Furthermore, congestion also leads to quicker energy consumption, since a node has to retry sending messages when the receiving node is flooded with data forwards. Asymmetric data rates can also cause needless power consumption, because the node with the faster connection is blocked by the slower node until all the data is exchanged, so it's working at a slower rate that it actually can.

An area of opportunistic networks that hasn't been researched too much is security. Along with privacy, they are two key conditions to people accepting opportunistic networks where their devices are nodes. This would imply that a message sent by a node can only be decrypted by the intended recipient, and that nodes can't enter the network and perform malicious deeds (such as flooding nodes with data,

[3] The percentage of messages that reach their destinations, out of the total number of messages sent in the network.

reporting false information, etc.). Moreover, nodes should be stopped from being selfish, using incentive mechanisms.

By taking all these challenges into consideration, the main goal of opportunistic networks is to achieve real mobile computing without the need for a connected network. Researchers are not there yet, but the algorithms and solutions proposed have been better and better over the past few years, so the research is heading into the right direction. It probably won't be too long until we will be able to see and use opportunistic networks for various purposes.

2.3 Use Cases

This subsection presents real-life use cases for ONs. It highlights various area where opportunistic networks have been or are soon to be deployed.

2.3.1 Disaster Management

One scenario where opportunistic networks may prove to be very important is disaster management. When a disaster such as an earthquake, a tsunami, or an explosion occurs, legacy communication might collapse, due to potential damage to the physical components of the network, such as switches or cables. Therefore, need arises for a method of ensuring more efficient and dependable solutions that can be employed for security missions. One such solution is proposed by Bruno et al. [8], and it implies using ONs to create an overlay infrastructure for rescue and crisis management services. The proposed solution uses the unaffected components of the static infrastructure (i.e. the ones that have not been damaged by the disaster), by making them act as nodes in opportunistic networks. Aside from these ad-hoc nodes, special networks and connectors are also deployed, and even singular mobile devices belonging to the survivors or to people nearby the disaster spot may be used. The main goal is to offer connectivity where otherwise there would be none, which leads to a higher efficiency in finding survivors or organizing the rescue efforts. Moreover, another goal is to lower the congestion rate, since regular network infrastructures tend to become very crowded during such incidents.

A similar solution is proposed by Lilien et al. [35], where systems that were not originally nodes of an opportunistic network dynamically join it, with the purpose of aiding with communication in disaster situations. MAETT (Mobile Agent Electronic Triage Tag) and Haggle-ETT [37] are two similar methods that are used to collect the triage data (i.e. location) of a disaster victim, when regular communication systems are down. They allow it to be collected and represented in an electronic format, which can then be transmitted to coordination points where it is processed and made available for the rescue missions. The difference between the two algorithms is that MAETT uses mobile agents for storing the electronic triage tag, whereas Haggle-ETT is based on the Haggle architecture.

2.3.2 Smart Cities

Cities are areas where Big Data is having a real impact. Town planners and administration bodies just need the right tools at their fingertips to consume all the data points that a town or city generates and then be able to turn that into actions that improve people's lives. In this case, Big Data is definitely a phenomenon that has a direct impact on the quality of life for those that choose to live in a town or city. Smart cities of tomorrow will rely not only on sensors within the city infrastructure, but also on a large number of devices that will willingly sense and integrate their data into technological platforms used for introspection into the habits and situations of individuals and city-large communities. Predictions say that cities will generate over 4.1 terabytes per day per square kilometre of urbanized land area by 2016. Efficiently handling such amounts of data is already a challenge.

Smart cities monitor and integrate the conditions of all their critical infrastructures (such as roads, bridges, tunnels, rails, subways, airports, sea-ports, communications, water, power, etc.) in order to better optimize their resources and plan their preventive maintenance activities [10]. They connect the physical, IT, social and business infrastructures, for the purpose of leveraging the collective intelligence of the cities. Opportunistic networks are the logical means of achieving at least a part of a smart city infrastructure, since they can be employed to perform communication between various parts of a smart city. For example, the traffic lights system can be opportunistically connected to a service that offers information about traffic jams, crowded roads, accidents, etc., so it can adapt to the conditions of the environment.

Moreover, mobile devices belonging to a smart city's citizens may also be opportunistically employed as sensor nodes, as shown by Le et al. [34]. The authors propose a new routing algorithm for a heterogeneous architecture composed of various types of nodes, as opposed to existing routing algorithms, which are based on the idea that the nodes and technologies are homogeneous and can easily work together. Using mobile devices as nodes in a smart city leads to a much better knowledge of the conditions in various parts of the town, ranging from traffic information, to data such as temperature or weather conditions. The more information is available, the better the decisions made by the various subsystems of the smart cities are.

2.3.3 Floating Content

Another potential practical use of opportunistic networks is in regard to floating content in areas such as open city squares [18], which are geographical zones (also known as anchor zones) where mobile nodes enter, spend a certain amount of time, and leave. In this case, the nodes are mobile devices belonging to humans, and the anchor zones are relatively small-sized areas where many people congregate, which represent the boundaries of the ad-hoc opportunistic network. While inside the anchor zone, the nodes may copy the data either if they need it for themselves, or if they transport it for the benefit of other nodes. If a node is interested in a certain data item floating around the anchor zone, it replicates it. Content availability in the anchor zone is probabilistic and best-effort, since the information can disappear from the area (i.e. it can sink) if unoptimal data sharing algorithms are employed. Nodes exiting the open city square delete the zone-specific content, since it is of no relevance outside the area in which it was generated, which means that the availability of the floating content is probabilistic. When dealing with floating content, the main requirement is that the content should easily be available, which happens when a certain fraction of the nodes in the anchor zone carry it. This is why ON-specific algorithms are employed and adapted for making the replication decisions. The best nodes should be chosen for replicating data, while at the same time congestion should be avoided. Desta et al. [18] attempt to increase the proportion of nodes having the content, and to have a high probability of bootstrapping the system (i.e. avoiding early extinction of the content during the initial phase).

2.3.4 Advertising

More recently, ONs have been used for other purposes, such as advertising. One such example is MobiAd [25], an application that presents the user with local advertisements in a privacy-preserving manner. The ads are selected by the phone from a pool of advertisements which are broadcast on the local mobile base station or received from local WiFi hotspots. Information about ad views and clicks is encrypted and sent to the ad channel in an opportunistic way, via other mobile devices or static WiFi hotspots. This helps ensure privacy, since the other nodes won't discover which ads were viewed, and the ad provider isn't able to know which user saw what ad.

Another way of applying opportunistic networking to advertising was proposed by Heinemann and Straub [26]. They base their proposition on word-of-mouth (WOM) communication or viral marketing, which means facilitating advertising where none of the participants are marketing sources. In classical WOM communication, a user A passes next to a store and sees a promotion for a certain item. Knowing that a friend B is interested in those types of items, A lets B know about the offer, thus advertising the product through word-of-mouth, even though there is no commercial interest from user A. Heinemann and Straub propose using this mechanism over an ON infrastructure. Instead of showing their promotions in the window, stores have mobile devices that broadcast information about promotions and offers on a certain radius. When people with mobile devices pass near the store, their own device receives the offer and attempts to match it with the current user's preferences. If the user is interested, then an alert is shown. If not, the information is stored for opportunistic transmission. Assuming that node A received an offer it is not interested in, it carries it around until it encounters node B. If node B is interested, then the offer is forwarded to it, and its owner is notified. This mechanism would imply having an application where users can store their preferences, in order to help match the received offers.

2.3.5 Sensor Networks

Sensor networks [4] are the backbone of infrastructures such as smart cities or smart houses. They are composed of various types of sensors which are able to monitor environmental conditions, such as temperature, light, sound, etc. The data thus collected is sent to a main location, which analyzes it and may decide to act upon it (e.g. when the temperature is too high, the cooling system is started, or when the light is too low, the lights are turned on, etc.). Applications of sensor networks in real-life include environmental monitoring (air quality/pollution, forest fires, landslide, water quality, natural disasters, etc.), industrial monitoring (machine health, data logging, water, waste) or agriculture. When dealing with complex sensor network, where there are many heterogeneous sensors, opportunistic networks are employed in order to improve the data collecting process. Instead of every sensor sending data to a singular central processing unit, the data is collected by various mobile collectors, which pre-process the data before sending it to the central unit. This way, there are much fewer data transfers, and the central unit becomes less of a bottleneck. Moreover, it is able to respond more quickly to changes in the environment, leading to a more optimal behavior of the network.

2.3.6 Interplanetary Internet

Interplanetary Internet [1] is a specialized type of opportunistic network that would allow fast communication between the Earth and other planets (or even spaceships or probes sent way beyond the planets in our solar system). The first point of focus is on fast Mars communication, and this network has three basic components: NASA's Deep Space Network (DSN)[4], a six-satellite constellation around Mars together with a large Marsat satellite placed in low Mars orbit, and a new protocol for transferring data. The DSN would act as the portal between Earth and the Interplanetary Internet, the six smaller satellites and the Marsat would provide a full-time connection between the DSN and Mars, and the new protocol would be used to ensure that data is transferred optimally through the satellites. Communication with Mars would be the first step, with the planet acting as a node in the network, passing data further on to other planets. One proposed idea for the communication protocol is the parcel transfer protocol (PCP), which assumes that data is transferred using the six smaller satellites opportunistically, by selecting the appropriate one according to the position of the satellites in orbit and the position of the point that communication must be performed with. Moreover, the satellites can relay data between each other before it reaches Mars.

[4] An international network of antennas that is able to track data and control the navigation of interplanetary spacecraft, used by NASA.

2.3.7 Crowd Management

Another situation where ONs and DTNs may be used is in crowded areas, e.g. at locations where groups of individuals get separated and need to locate each other. In such scenarios (e.g. an amusement park where a child gets lost from his parents, a concert where a group of friends split and they need to find each other, a football match, etc.), contacts between mobile devices happen often and for longer periods of time, but (due to the high number of mobile devices in a very small space), classic communication means such as 3G and mobile telephony may not work properly. However, since the ON created ad-hoc by leveraging the participants' mobile devices is very dense, the mobile devices may be used to opportunistically broadcast the location of mobile device owners to interested receivers. For example, a child at an amusement park might carry a smartphone that opportunistically broadcasts the child's encrypted location, by leveraging the neighboring devices. The child's parents have a smartphone of their own which receives the broadcast from encountered nodes, thus being aware of the child's location at any point in time. Similarly, non-emergency scenarios may also benefit from this solution, where events regarding the bands that will be playing or any other announcements can be disseminated to participants of a music concert or any other type of social event with high participation. This would gradually replace the classic printed event programmes, offering up the possibility for more detailed and interactive information for attendees.

An existing crowd management project is Extreme Wireless Distributed Systems (EWIDS)[5], which attempts to use wireless sensor technology to monitor people's behavior, using crowd management as their application domain. Proximity graphs are generated using body-worn sensors carried by thousand of people, and the data is extracted and processed, either in real-time or offline. The extracted data is a series of graphs that evolve, used to provide feedback to a crowd of people, leading to crowd management.

2.3.8 Context-Aware Platforms

Another area that benefits from ONs is represented by platforms that support generic context-aware mobile applications. CAPIM (Context-Aware Platform using Integrated Mobile services) [19] is such a framework, and it is designed to support the construction of next-generation applications. It performs an integration of context data-collecting services, such as location, user's profile and characteristics, environment, in order to offer a centralized framework for context information. The smart services are dynamically loaded by mobile clients, which take advantage of the sensing capabilities provided by modern smartphones, possibly augmented with external sensors. A framework such as CAPIM can be employed in academic environments as a means to facilitate the interaction between students, professors and other faculty members. The communication could be performed opportunistically, instead of

[5] http://www.distributed-systems.net/index.php?id=extreme-wireless-distributed-systems.

using fixed infrastructures such as WiFi or 3G, in order to limit the bandwidth used and to save battery life (since Bluetooth consumes less power than both WiFi and 3G). CAPIM would be used to disseminate announcements to students, signal their location, the classes they take, the interactions they have, etc. Other examples of context-aware platforms that might benefit from an ON-based framework include PACE [27], SOCAM [22] or CoWSAMI [3].

2.3.9 Wildlife Tracking

One of the first application of opportunistic networks in real-life has been wildlife tracking, i.e. recording the behavior and movement paths of animals in their natural habitat, while being non-intrusive. This assumes that animals are equipped with special tags that are able to communicate with other similar tags and with the researchers' base stations. The tags are able to record the geographical coordinates of the animal carrying it, as well as the encounters with other animals wearing similar tags. Having the tags able to communicate with the base stations means that researchers don't have to capture the animal in order to retrieve the collected information anymore. Instead, tracked animals exchange collected data between each other, and upload it whenever they are in range of the base stations. These stations may either be mobile (such as cars/ships belonging to researchers, caretakers from the national parks, fishermen, etc.), or fixed. Opportunistic routing protocols are employed when two tracked animals meet each other, in order to decide which data will be exchanged between the two. Generally, history-based algorithms are used, since flooding is not feasible due to lack to storage space.

Two examples of such wildlife tracking applications are ZebraNet [33] and Shared Wireless Infostation Model (SWIM) [41]. ZebraNet was deployed in the Kenyan savanna to help track zebras wearing special collars. In this case, the base station was mobile, namely the researchers' vehicle that periodically moved around the savanna to collect data. SWIM, on the other hand, was used to track whales' movement, and the base stations were either mobile (seabirds) or fixed (buoys).

2.3.10 Internet Access in Limited Conditions

Another application of ONs, according to Pelusi et al. [39], is offering Internet access in limited conditions where no infrastructure exists. Two examples of such projects are the DakNet project [40] and the Saami Network Connectivity (SNC) [20] project. The DakNet project and the similar, but improved, KioskNet system [24], have the purpose of creating a low-cost asynchronous infrastructure that is able to provide connectivity in rural areas where the deployment of standard Internet access is not feasible or cost-effective (such as Indian villages). The two projects propose building kiosks in villages that are equipped with digital storage and wireless communication devices, which interact with mobile base stations periodically. Such stations are mounted on buses, motorcycles and bicycles, and collect the data from the kiosks and deliver it to Internet access points located in the cities (and vice versa). SNC provides a similar solution, but to saami herders in Lapland, in order to protect their heritage and cultural identity, while still helping them integrate into the modern societies from their specific countries (Sweden, Norway, Finland).

2.3.11 Distributed Social Networks

Opportunistic networks can also be used to leverage dissemination of information between the members of a social group, without the need (or the expenses) for a wired static infrastructure. One example of such an implementation is proposed by Thilakarathna et al. [43], which focuses on delay-tolerant applications and services such as content sharing or advertisement propagation between users who are geographically clustered into communities. The authors propose to address important ON issues such as lack of trust, privacy or latency of delivery through combining the advantages of distributed decentralised storage and opportunistic communications. A community-based greedy heuristic algorithm is proposed, which is shown to maximize the content dissemination with limited number of replications. Another distributed social network application that uses ONs is DroidOppPathFinder [2], which generates and shares content about paths for fitness activity in a city, recommending the best paths from specific geographical areas through the analysis of user preference and context information collected by various sensing devices.

2.4 Conclusions

This section has provided the definition of opportunistic networks and has shown the challenges facing the deployment of such networks in real-life. However, we have also shown several use cases where ONs have been successfully deployed, and other areas where interesting and valid propositions have been presented. This leads us to believe that opportunistic networks have a good applicability in real-life, especially if the algorithms and solutions keep evolving, as they have been doing in the past few years.

3 Data Routing and Dissemination

Ever since opportunistic networks were first presented, many data routing and dissemination algorithms have been proposed, ranging from very simple ones to more complex techniques that use prediction or social knowledge when performing routing decisions. There are many ways in which these algorithms may be classified (a taxonomy for routing algorithms is presented in [39], and one for dissemination techniques is proposed in [11]). However, we have split them here into basic, socially-based, and history and prediction-based algorithms, for simplicity. This section presents the most important algorithms in each of these three categories, highlighting their advantages, as well as their issues.

3.1 Basic Algorithms

The basic algorithms presented in this section are among the first algorithms proposed for opportunistic networks, and this is why they are simpler, not taking into consideration many of the aspects that ON routing and dissemination algorithms take for granted nowadays.

3.1.1 Epidemic

The Epidemic algorithm [44] is based on the way a virus spreads: when two potential carriers meet, the one with the virus infects the other one, if it isn't already infected. Thus, when an ON node A encounters a node B, A downloads all the messages from B that it doesn't already contain, and vice versa. The simplest version of this algorithm assumes that a node's data memory is unlimited, so that it can store all the messages that can be at once in the opportunistic network. However, this is unfeasible in real-life, especially as the network grows ever larger, so a modified Epidemic version exists, where the data memory of a node is limited. Thus, when node A's memory is full and it encounters node B, first it has to drop the oldest messages in its memory, in order to make room for the messages that it will download from node B. This also makes the algorithm somewhat inefficient, since some older messages may be important (e.g. they may be addressed to nodes that A is about to encounter) and some new ones totally irrelevant (e.g. their destinations may be nodes that A will never meet).

3.1.2 Spray-and-Wait

Spray-and-Wait [42] is an improvement to Epidemic that attempts to treat the congestion (and consequently the energy consumption) problem by limiting the total number of messages sent in the network, while keeping a high hit rate. As the name states, the algorithm is split into two phases. The Spray phase assumes that, for each message originating at a source node, a predefined number of copies are transferred to the encountered nodes, in the order that they are seen. The nodes that received the message will then do the same with the nodes that they encounter. Secondly, the Wait phase occurs after the end of the Spray phase (i.e. after the message has been transmitted for the given number of times) and it implies that, if a message's destination has not been encountered yet, then the message will only be relayed when (and if) the destination is encountered. Thus, Spray-and-Wait combines the speed of Epidemic routing with the simplicity of direct transmission. However, unlike the basic Epidemic algorithm, Spray-and-Wait doesn't guarantee the maximum hit rate. Moreover, it still doesn't take into account any context information in its routing decisions.

3.2 Socially-Based Algorithms

Since the initial algorithms proposed for routing and data dissemination in ONs proved to be inefficient for large scale networks, new algorithms were required. Opportunistic networks were considered generally to be formed of mobile devices carried by humans, so human mobility and behavior was studied in detail, in order to find patterns. Thus, it has been shown that users tend to interact more with nodes that they have a strong social connection with [9, 13] (i.e. nodes belonging to the same social community). This led to the creation of socially-based algorithms, which base their decision of whether two nodes should exchange a message or not on the communities of the two nodes, as well as the communities of the message's source and destination.

3.2.1 Socio-Aware Overlay

The Socio-Aware Overlay algorithm [45] is a data dissemination technique that creates an overlay for an ON with publish/subscribe communication, composed of nodes having high centrality values that have the best visibility in a community. Each of these nodes, called hubs or brokers, represents a community, and thus leverages the communication between nodes pertaining to its community and other nodes. The Socio-Aware algorithm is based on a publish/subscribe approach, with nodes subscribing to channels that publish data. When two nodes meet, subscriptions and unsubscriptions with the destination of community broker nodes are exchanged, as well as a list of centrality values with a time stamp. When a broker node changes upon calculation of its centrality, the subscription list is transferred to the new broker. When a publication reaches the broker, it is propagated to all other brokers, and then the broker checks its own subscription list. If there are members in its community that must receive the publication, the broker floods the community with the information. Being socially-aware, the algorithm has its own community detection method. This method assumes a community structure that is based on a classification of the nodes in an opportunistic network, from the standpoint of another node. A first type of node is one from the same community, having a high number of contacts of long/stable durations. Another type of node is called a familiar stranger and has a high number of contacts with the current node, but the contact durations are short. There are also stranger nodes, where the contact duration is short and the number of contacts is low, and finally friend nodes, with few contacts, but high contact durations. In order to construct an overlay for publish/subscribe systems, community detection is performed in a decentralized fashion. Thus, each node must detect its own local community. The disadvantage of this method is that broker nodes tend to be congested, given that all data directed to their community must first pass through them. If the members of a community are subscribed to many channels, it would be more suitable to be able to have multiple brokers, in order to increase the efficiency.

3.2.2 BUBBLE Rap

BUBBLE Rap [30] is a routing algorithm for opportunistic networks that uses knowledge about nodes' social communities to deliver messages. It assumes that a mobile device carrier's role in the society is also true in the network. Therefore, the first step performed by BUBBLE Rap is to forward data to more popular nodes than the current node. The second assumption made in BUBBLE Rap is that the communities people form in their social lives are also observed in the network layer, so the second part of the algorithm is to identify the members of the destination community and pass them the message. Thus, a message is bubbled up the hierarchical ranking tree using a global popularity level, until it reaches a node that is in the same community as the destination. Then, the message is bubbled up using a local ranking until it reaches its target. The popularity of a node is given by its betweenness centrality, which is the number of times a node is on the shortest path between two other nodes in the network. Community detection is done using *k*-CLIQUE [31], which dynamically detects the community of a node by analyzing its contacts with other devices. A distributed version of BUBBLE Rap entitled DiBuBB is also proposed by the authors. It uses distributed *k*-CLIQUE for community detection, together with a cumulative or single window algorithm for distributed centrality computation. The single window (S-window) algorithm computes centrality as the number of encounters the current node has had in the last time window (chosen usually to be six hours), while the cumulative window (C-window) algorithm counts the number of individual nodes encountered for each time window and then performs an exponential smoothing on the cumulated values. The same issue that can appear at the Socio-Aware Overlay may be present at BUBBLE Rap: popular nodes tend to get congested, because they have to carry messages for many other nodes.

3.2.3 SRSN

The SRSN algorithm [6] is based on the assumption that ad-hoc detected communities may miss important aspects of the true organization of an opportunistic network, where, for example, a node might have a strong social link to another node that is encountered rarely. In such a situation, a detected social network might omit this tie and thus yield sub-optimal forwarding paths. Therefore, two types of social networks are considered. First of all, there is a detected social network (DSN) as given by a community detection algorithm such as *k*-CLIQUE, an approach similar to the one taken by BUBBLE Rap. Secondly, the authors also propose a self-reported social network (SRSN) as given by social network links (in this case, Facebook relationships). The algorithm follows a few steps: nodes generate data, carry it around the network and, when they encounter another node, they only exchange information if the two nodes are in the same network (either DSN of SRSN). Therefore, there are two versions of this algorithm: one that uses the DSN, and another one that uses SRSN. Through extensive experiments, the authors show that using SRSN information instead of DSN decreases the delivery cost and produces comparable delivery ratio. This happens because the two social networks differ in terms of structural and role equivalence, with the better approximation being obtained through the SRSN. The results presented by the SRSN algorithm are relevant in terms of highlighting the importance of using readily-available information (such as Facebook, Google+, Twitter or LinkedIn social relationships) for approximating a user's social relationships. However, in situations where the social network doesn't correctly approximate the network's behavior, or where social network information is not available, such an algorithm can't be used.

3.2.4 ContentPlace

ContentPlace [7] deals with data dissemination in resource-constrained ONs, by making content available in regions where interested users are present, without overusing available resources. To optimize content availability, it exploits learned information about users' social relationships to decide where to place user data. The design of ContentPlace is based on two assumptions: users can be grouped together logically, according to the type of content they are interested in, and their movement is driven by social relationships. When a node encounters another node, it decides what information seen on the other node should be replicated locally. Thus, ContentPlace defines a utility function by means of which each node can associate a utility value to any data object. When a node encounters another peer, it selects the set of data objects that maximizes the local utility of its cache. Due to performance issues, when two nodes meet, they do not advertise all information about their data objects, but instead they exchange a summary of data objects in their caches. Finally, the data exchange is accomplished when a user receives a data object it is subscribed to when it is found in an encountered node's cache. To have a suitable representation of users' social behavior, an approach that is similar to the caveman model is used, that has a community structure which assumes that users are grouped into home communities, while at the same time having relationships in acquainted communities. The utility is a weighted sum of one component for each community its user has relationships with. Community detection is done using *k*-CLIQUE. By using weights based on the social aspect of opportunistic networking, ContentPlace offers the possibility of defining different policies. There are five policies defined: Most Frequently Visited (MFV), Most Likely Next (MLN), Future (F), Present (P) and Uniform Social (US). These policies allow the network manager to change the behavior of the nodes, according to the configuration of the network.

3.3 History and Prediction-Based Algorithms

There are some situations where social information is not present and algorithms that are able to detect communities are too costly in terms of computing power. Because of such cases (and because social information doesn't always lead to good approximations of contact behavior), a new type of algorithms has appeared. These algorithms base their routing decisions on the history of contacts between nodes. If a node A has encountered a node B many times in the recent past, it is assumed that it will encounter it again in the near future, because the two nodes have similar paths. Moreover, based on the shapes of past contacts, various types of distributions and approximations have been employed to predict the future behavior of ON nodes and perform optimal routing decisions.

3.3.1 PROPHET

PROPHET [36] is a prediction-based routing algorithm for ONs which performs probabilistic routing by establishing a metric called delivery predictability (P) at every node A for a known destination B. This probability signifies A's chance to successfully deliver a message to B. When two PROPHET nodes meet, they exchange summary vectors which (among other information) contain the delivery predictability P. When a node receives this information, it updates its internal delivery predictability vector (whose size is equal to the total number of nodes in the ON), and then decides which messages to request from the other node based on the forwarding strategy used. There are three steps performed when computing delivery predictability values. Firstly, whenever a node is encountered, the local value of the metric is updated, which leads to a higher P for nodes that are encountered more often. Secondly, since nodes that are not in contact for long periods of time are not very likely to be good forwarders towards each other, the delivery predictability must age, thus being reduced with the passage of time. The aging process is based on an aging constant and a given unit of time. Finally, the delivery predictability also has a transitive property, based on the fact that, if two nodes A and B meet each other often, and node A also has many encounters with a node C, then C is also a good forwarding node for A. Based on a scaling constant, the transitivity property also impacts the computation of the delivery predictability. The forwarding strategy chosen by the authors is a simple one, and implies that, when two nodes A and B meet, if the delivery probability of the destination of a message at B is higher for A, then the message is transferred (and the other way around as well). The main caveat of this algorithm is that, since nodes are not being split into communities, there exists the risk of flooding a popular node (such as a professor in an academic environment, which interacts with students from various study years). This can be avoided by redirecting a part of the messages destined for a such a node to other less popular nodes.

3.3.2 RANK

Hui and Crowcroft [29] study the impact of predictable human interactions on forwarding in PSNs. By applying vertex similarity on a dataset extracted from mobility traces, they observe that adaptive forwarding algorithms can be built by using the history of past encounters. Furthermore, the authors design a distributed forwarding algorithm based on node centrality and show that it is efficient in terms of hit rate and delivery latency. This greedy algorithm is entitled RANK, and (similarly to BUBBLE Rap) it uses popular nodes to disseminate data. The popularity of a node is quantified by the Freeman betweenness centrality, which is defined as the number of times a node falls on the shortest path of other nodes. The authors assume that each node knows its own centrality and the centrality of the nodes it encounters, but not of the other nodes in the network, so it can't know the highest centrality in the system. Therefore, the greedy algorithm pushes traffic on all paths to nodes that have a higher centrality than the current node, until the destination is reached or the messages expire. Since knowing the individual centrality for each node at any point in time is complicated, the authors propose analyzing the past activity of a node to see if it was a good carrier in the past, and then use this information for future forwarding. Therefore, they analyze how well the past centrality can predict the future centrality for a given node, and for this reason they extract three consecutive 3-week sessions for a mobility trace and run a set of greedy RANK emulations on the last two data sessions, using centrality values from the first session. The test results show that human mobility is predictable to a certain degree and that past contact information can successfully be used to approximate the future behavior of a node in the ON. However, one of the main limitations of the RANK algorithm is that it only focuses on a day-to-day analysis, whereas a finer-grained predictability may prove to be more useful for messages that have a lower tolerance for delays. Another important caveat of the algorithm is that, although it uses prediction of future node behavior, it doesn't consider the ON nodes as belonging to communities, which may lead to congestion at the nodes that are most popular in terms of centrality.

3.3.3 dLife

dLife [38] is an opportunistic network routing algorithm that is able to capture the dynamic represented by time-evolving social ties between pairs of nodes. The authors highlight the fact that user behavior is dynamic, the network itself evolves, meaning that network ties are created and broken constantly. This is why dLife focuses on the different behavior users have in different daily periods of time, instead of estimating their behavior per day. The dynamics of social structures are represented as a weighted contact graph, where the weights are used to express how long a pair of nodes is in contact over different periods of time. There are two complementary utility functions employed by dLife: the Time-Evolving Contact Duration (TECD), which is the evolution of social interaction among pairs of users in the same daily interval over consecutive days, and the TECD Importance (TECD$_i$), which is the evolution of a user's importance, based on its node degree and social strength towards its neighbors, in different periods of time. TECD is used to forward messages to nodes that have a stronger social relationship with the destination than the current carrier. Each node computes the average of its contact duration with other nodes during the same set of daily time periods over consecutive days. If the carrier and the encountered node have no social information towards the destination, forwarding is done based on TECD$_i$, where the encountered node gets a message if it has a greater importance than the carrier. The authors also propose a community-based version of dLife, entitled dLifeComm, where the social communities are computed similarly to BUBBLE Rap (i.e. using *k*-CLIQUE), but the decision whether to forward to a node is done based on TECD and TECD$_i$, thus changing over time. dLife thus offers the advantage of using both the history of contacts, as well as social information, in performing routing decisions.

3.4 Conclusions

We have presented in this section several ON routing and dissemination algorithms. For each of them, we have shown that they have both strengths, as well as weaknesses. Some of these algorithms are suitable for a certain type of situation, other are better for different scenarios. There is no single best algorithm, and this is because opportunistic networks are so varied and can range from large and dense networks with thousands of participants, to small and sparse networks that must make the most of any contacts between nodes. This is why the research area of ONs is so vast and keeps evolving constantly, with the proposed solutions becoming better and better.

4 Potential Solutions

In this section we present a couple of alternatives to the existing solutions shown in Section 3 and highlight the improvements they bring.

4.1 SPRINT

SPRINT (Social PRedIction-based routing in opportunistic NeTworks) [14] is a novel ON point-to-point routing algorithm that takes advantage of both social knowledge, as well as contact prediction when making decisions. It uses social information about the nodes from contact history and from existing self-reported social networks. Moreover, it includes a Poisson-based prediction of a node's future behavior. Through extensive experiments, it has been shown that SPRINT performs better that existing socially-aware opportunistic routing solutions in terms of hit rate, latency, delivery cost[6] and hop count[7]. This section presents the motivations behind SPRINT and the functionality of the algorithm.

4.1.1 Social Knowledge in Opportunistic Networks

The addition of social information to SPRINT is motivated by the results presented in [13] and [15]. There, an academic environment is analyzed in terms of contact and inter-contact times, as well as the relationships between the social connection strengths and number and duration of contacts between two nodes. Firstly, it is shown that the self-reported social network information (i.e. Facebook friend data gathered from the participants in the network) is a better approximation of the contact behavior of a node than the data obtained by a community-detection algorithm such as *k*-CLIQUE. By taking this information into account, four modified BUBBLE Rap versions are proposed, which use social network information instead of *k*-CLIQUE data.

The first such BUBBLE Rap version is called Social, and it performs the decision of whether to use the local or the global community (as shown in Section 3.2.2) based on social network information, instead of *k*-CLIQUE data. Thus, the communities assumed by BUBBLE Rap are formed through self-reported social networks knowledge, instead of using community detection algorithms. The second method (entitled Max) computes a node's centrality (and thus its importance in its own community) as the maximum between the centrality as reported by the C-window algorithm used by BUBBLE Rap, and a node's popularity in terms of Facebook friends. The other two BUBBLE Rap improvements (Popularity and Popularity Squared) use a weighted sum between the C-window centrality and a node's popularity to compute the final centrality value (Popularity uses a regular sum, whereas Popularity Squared employs a squared sum between the two values).

These four proposed BUBBLE Rap enhancements are tested both on an academic trace, where social connections are expected to exist between students attending the same classes, as well as on a different trace taken in and around the town of St. Andrews [6]. It is shown that all four socially-based BUBBLE Rap versions

[6] The ratio between the total number of messages exchanged in the network and the number of generated messages.

[7] The number of nodes that carried a message until its destination on the shortest path.

outperform the base implementation for both traces in terms of hit rate. Moreover, the best results are obtained by the Popularity version.

4.1.2 Predicting Opportunistic Nodes' Behavior

As stated in Section 2.2, an important challenge in mobile networks is knowing when and to which node should a message be passed, in order for it to reach its destination (and do it as fast as possible). Therefore, it would be important if we were able to predict the future behavior of a node in such a network, in regard to its encounters and contact durations. Such a method is proposed in [12], by approximating the time series of a node's contacts as a Poisson distribution.

ON nodes, as previously stated in this chapter, are generally mobile devices that belong to humans, and if there's one thing certain about people, it is that they are creatures of habit. They follow similar daily patterns, from home to work or school, where they spend a generally fixed amount of time, after which they return home. Similarly, in weekends they tend to go to the same places and visit the same locations. This is also valid on a smaller scale, as shown by the analysis in [12], namely an academic scenario where the nodes are the students and professors from a faculty. They have a fixed daily schedule and interact with each other at fixed times in a day, namely when they attend classes. Through analysis of a mobility trace taken in such an academic environment, it has been shown that a node's behavior in an opportunistic network in terms of number of contacts per time unit can be approximated as a Poisson distribution. The shape of the distribution is proven to apply to the mobility trace analyzed by performing a chi-squared test, with only 2.49% of all the Poisson hypotheses rejected. Moreover, the paper shows that, by removing the final week from the trace and attempting to predict each node's number of contacts using a Poisson distribution, a percentage of 98.24% correct predictions is obtained. Similar results are also achieved for the St. Andrews trace mentioned above. We refer the reader to [12] for the complete set of tests and results.

4.1.3 SPRINT Algorithm

The SPRINT algorithm [14] combines socially-aware routing (both learned and offline social information) with node behavior prediction, in order to improve the performance of ON routing.

The behavior of SPRINT when two nodes running it get in contact is very similar to ContentPlace's behavior, as shown in Section 3.2.4: each node computes a utility value for both its messages, as well as the ones belonging to the encountered node, and then attempts to maximize its data cache by selecting the messages with the highest utility values. The novel part of SPRINT is its utility function and the way it uses both social, as well as prediction information, to compute the importance of a message. The formula used by a node A to compute the utility of a message

M is shown below. w_1 and w_2 are weight values which follow the conditions that $w_1 + w_2 = 1$ and $w_1 > w_2$. U_1 and U_2 are individual utility components.

$$u(M,A) = w_1 * U_1(M,A) + w_2 * U_2(M,A)$$

$$U_1(M,A) = freshness(M) + p(M,A) * (1 - \frac{enc(M,A)}{24})$$

$$U_2(M,A) = c_e(M,A) * \frac{s_n(M) + hop(M) + pop(A) + t(M,A)}{4}$$

The $freshness(M)$ component of U_1 favors new messages, being positive if the message has been created less than a day ago, and 0 otherwise. $p(M,A)$ is the probability of node A being able to deliver a message M closer to its destination, and is based on predicting a node's behavior, combined with the idea that a node has a higher chance of interacting with nodes it is socially connected with and/or has encountered before. It is computed based on the knowledge that the node contacts follow a Poisson distribution. The first step is to count how many times node A encountered each of the other nodes in the network. If a node has been previously met in the same day of the week or in the same two-hour interval as the current time, the total encounters value is increased by 1. For the nodes encountered in the past that are in the same social community as node A, the total number of contacts is doubled. Then, the probabilities of encountering nodes based on past contacts are computed by performing a ratio between the number of encounters per node and the total number of encounters. The next step consists of computing the number of encounters N that node A will have for each of the next 24 hours by using the Poisson distribution probabilities and choosing the value with the highest probability as N. The first N nodes are then picked as potential future contacts for each of the next 24 hours (sorted by probability), and for the rest of them $p(M,A)$ is set to 0. U_1 also uses $enc(M,A)$, which is the time (in hours) until the destination of message M will be met by A according to the probabilities previously computed. If the destination will never be encountered, then $enc(M,A)$ is set to 24 (so the product is 0).

The second component of the utility function is U_2. $c_e(M,A)$ is set to 1 if node A is in the same community as the destination of message M or if it will encounter a node that has a social relationship with M, and 0 otherwise. The prediction information computed for U_1 is used to analyze the potential future encounters of a node. The $s_n(M)$ component is set to 1 (and 0 otherwise) if the source and destination of M do not have a social connection. $hop(M)$ represents the normalized number of nodes that M has visited, $pop(A)$ is the popularity value of A according to its social network information (i.e. number of Facebook friends in the opportunistic network), and finally $t(M,A)$ is the total time spent by node A in contact with M's destination.

SPRINT is compared to BUBBLE Rap and it is shown that it performs better in terms of hit rate, delivery latency, hop count and delivery cost for three mobility traces and one synthetic mobility model simulation. The complete results, along with their analysis, are presented in detail in [14]. The algorithm's main advantage over other solutions is that it doesn't rely solely on one method for deciding the next hop. It combines social information, from both offline and online sources, with ad-hoc prediction mechanisms (which can be switched on-the-fly, according to the characteristics of the ON), to offer a more complete view of a node's behavior.

4.2 SENSE

SENSE [16] is a collaborative selfish node detection and incentive mechanism for opportunistic networks that is not only able to detect the selfish nodes in an ON, but also has the possibility of improving the network's performance by incentivising the participating nodes into carrying data for other nodes. Altruism is an important component of ONs, since nodes must rely on each other for a successful transmission of their intended messages. Thus, nodes refusing to participate in the routing process are punished by the algorithm, and therefore have no way to get their messages to be delivered, unless they accept to help other nodes route their data as well.

4.2.1 Selfishness and Altruism in Opportunistic Networks

Most routing and dissemination algorithms proposed so far generally assume that the nodes in an opportunistic network are willing to participate in the routing process at all times. However, in real-life scenarios this isn't necessarily true, since a node may be selfish towards a subset of nodes, and unselfish for the rest. Several reasons for this selfishness exist, such as a node being low on resources (battery life, memory, CPU, network, bandwidth, etc.) and trying to save them for future use, fear of malicious data from unknown users, or even lack of interest in helping the nodes from a different social community. The existence of selfish nodes in an opportunistic network might lead to messages having high delays or never being delivered at all. Thus, these nodes should be detected and avoided when routing. Furthermore, incentive mechanisms should reward nodes when they actively take part in the network and punish them when they do not.

There are several altruism models that specify how selfish nodes are spread in the network and how they behave towards the nodes they encounter. SENSE uses the community-biased model, which assumes that people in a community have greater incentives to carry messages for other members of the same community. In this case, altruism is modelled using an intra and an inter-community altruism level. This is one of the most realistic altruism models available, since it is a good approximation of an opportunistic network where the nodes are mobile devices carried by humans that interact based on social relationships. However, altruism values should also be distributed inside a community (i.e. not all nodes in the same community should have the same altruistic values towards each other), using a uniform or normal distribution.

4.2.2 SENSE Algorithm

Each SENSE [16] node has a four-section data memory. Firstly, there is the list of messages that the node has generated in the course of time G. Secondly, each node has a list of messages that it stores, carries and forwards for other nodes C. Additionally, each node has another two sections of data memory that contain information regarding past transfers: a list of past forwards O and a list of past receives I. O contains information regarding past message forward operations performed either by the current node, or by other nodes. The list of past receives I contains information regarding past message receive operations.

When two nodes A and B running SENSE meet, they each compute an altruism value towards the other node and, based on that value, decide if they will help the other node. If the two nodes decide that they are unselfish towards one another, they exchange I and O and update them with the new information. This way, a node can have a more informed view of the behavior of various nodes in the network, through gossiping.

After two nodes decide to be altruistic towards one another and they finish exchanging knowledge about past encounters, each of them advertises its own specific information, such as battery level and metadata about the messages it carries. Based on the lists of past encounters I and O, each node computes a perceived altruism value for the other node with regard to the messages stored in its own data memory (in other words, it computes how willing the encountered node is to forward a certain type of message). If this value is within certain thresholds, the communication continues and the desired opportunistic routing or dissemination algorithm is applied. If (for example) node B's computed altruism is not within the given limits for any of A's stored messages, then it is considered selfish by node A, so A doesn't send it messages for routing and doesn't accept messages from B, either. Node A then notifies B that it considers it selfish, so B won't end up considering node A selfish. This also functions as an incentive mechanism, because if a node wants its messages to be routed by other nodes, it shouldn't be selfish towards them. Therefore, every time a node is notified that it is selfish in regard to a certain message, it increases its altruism value. If there is a social connection between the selfish node and the source of the message, the inter-community altruism is increased. Otherwise, the intra-community altruism value grows.

The formula for computing altruism values for a node N and a message m based on the list of past forwards O and on the list of past receives I is the following:

$$altruism(N,m) = \sum_{o \in O, i \in I, o.m = i.m}^{N.id = o.d, N.id = i.s} type(m, o.m) * thr(o.b)$$

A past encounter x has a field $x.m$ which specifies the message that was sent or received, $x.s$ is the source of the transfer, $x.d$ is the destination and $x.b$ is the battery level of the source. $type$ is a function that returns 1 if the types of the two messages received as parameters are the same (in terms of communities, priorities, etc.), and 0 otherwise, while thr returns 1 if the value received as parameter is higher than a preset threshold, and 0 if it's not the case. Thus, the function counts how many messages of the same type as *m* have been forwarded with the help of node *N*, when *N*'s battery was at an acceptable level.

Test results show that SENSE can help improve opportunistic network performance (with metrics such as hit rate, delivery latency, hop count and delivery cost) when selfish nodes exist. It is demonstrated that SENSE outperforms a scenario where selfish nodes are present, but no selfishness detection and incentive mechanism is available. Moreover, it even performs better than existing similar algorithms, such as IRONMAN [5]. It is also shown that SENSE can successfully differentiate between a node being selfish on purpose, and a node not being able to deliver messages due to low battery power. For the full set of tests and results, we refer to reader to [16]. The main advantage of SENSE is not only that it can detect and avoid selfish nodes, but also that it can limit the total number of messages sent in the opportunistic network by carefully selecting a message's destination, based on the social connection and history of routing.

5 Future Trends

One of the main limitations of research in this area so far is that it has mostly focused on point-to-point communication. However, we believe that the future of opportunistic networks is heading towards data dissemination, where communication is done based on a publish/subscribe paradigm. This is why we are expecting a focus on data dissemination instead of point-to-point routing in the near future. This includes moving towards information-centric networks (ICNs) and the Internet of Things (IoT).

An ICN is a novel method of making the Internet more data-oriented and content-centric [21] and is basically a global-scale version of the publish/subscribe paradigm. The focus changes from referring to data by its location (and an IP address) to requesting Named Data Objects (NDOs) instead. When an ICN network element receives a request for content, it can respond with the content directly if it has the data cached, or it can request it from its peers otherwise. This way, an end-user is not concerned with the location of an object, only with its actual name, thus being able to receive it from any number of hosts. Mobile devices play an important role in ICNs, since they may be used to cache data as closely as possible to interested users, based on context information. Therefore, efficient opportunistic routing and dissemination algorithms have to employed in order to move the data accordingly and replicate it as needed.

The Internet of Things [23] aims to improve social connectivity in physical communities by leveraging information detected by mobile devices. It assumes a number of such devices being able to communicate between each other to gather context data, which is then used to make automated decisions. The deployment of IoT generally has three steps. The first one is getting more devices onto the network, the second step is making them rely on each other, coordinating their actions for simple tasks without human intervention, and the final step is to understand these devices as a single system that needs to be programmed. The more devices will connected, the more important will the role of the routing and dissemination protocols be.

It is estimated that IoT will have to accommodate over 50,000 billion objects of very diverse types by 2020 [32]. Standardization and interoperability will thus be absolute necessities for interfacing them with the Internet. New media access techniques, communication protocols and sustainable standards will need to be developed to make Things communicate with each other and with people. One approach would be the encapsulation of smart wireless identifiable devices and embedded devices in Web services. We can also consider the importance of enhancing the quality of service aspects like response time, resource consumption, throughput, availability, and reliability. The discovery and use of knowledge about services availability and of publish/subscribe/notify mechanisms would also contribute to enhancing the management of complex Thing structures.

Because of the fast increase of mobile data traffic volume being generated by bandwidth-hungry smartphone applications, cellular operators are forced to explore various possibilities to offload data traffic away from their core networks. 3G cellular networks are already overloaded with data traffic generated by smartphone applications (e.g. mobile TV). With the advent of IoT, the potentially huge number of Things will not be easily incorporated by today's communication protocols and/or Internet architecture. Mobile data offloading may relieve the problem, by using complementary communication technologies (considering the increasing capacity of WiFi), to deliver traffic originally planned for transmission over cellular networks. Here, opportunistic networks can find quick benefits.

Again related to IoT, new services shall be available for persistent distributed knowledge storing and sharing, and new computational resources shall be used for the execution of complicated tasks. Actual forecasts indicate that in 2015 more than 220 Exabytes of data will be stored [32]. At the same time, optimal distribution of tasks between smart objects with high capabilities and the IoT infrastructure shall be found. New mechanisms and protocols will be needed for privacy and security issues at all IoT levels including the infrastructure. Solutions for stronger security could be based on models employing the context-aware capability of Things. New methods are required for energy saving and energy-efficient and self-sustainable systems. Researchers will look for new power-efficient platforms and technologies and will explore the ability of smart objects to harvest energy from their surroundings.

The large variety of technologies and designs used in the production of Things is a main concern when considering the interoperability. One solution is the adoption of standards for Things intercommunication. Adding self-configuration and self-management properties could be necessary to allow Things to interoperate and, in addition, integrate within the surrounding operational environment. This approach is superior to the centralized management, which can't respond to difficulties induced by the dimensions, dynamicity and complexity of the Internet of Things. The autonomic behavior is important at the operational level as well. Letting autonomic Things react to events generated by context changes facilitates the construction and structuring of large environments that support the Internet of Things.

Special requirements come from the scarcity of Things' resources, and are concerned with power consumption. New methods of efficient management of power consumption are needed and could apply at different levels, from the architecture level of Things to the level of the network routing. They could substantially contribute to lowering the cost of Things, which is essential for the rapid expansion of the Internet of Things.

Some issues come from the distributed nature of the environment in which different operations and decisions are based on the collaboration of Things. One issue is how Things converge on a solution and how the quality of the solution can be evaluated. Another issue is how to protect against faulty Things, including those exhibiting malicious behavior. Finally, the way Things can cope with security issues to preserve confidentiality, privacy, integrity and availability are of high interest. For all these, examples of mechanisms designed to cope with such problems by actively using any communication opportunity were presented throughout the chapter.

6 Conclusions

In this chapter, we have presented opportunistic networks in detail, providing requirements for implementing such networks in real-life and showing the challenges in designing efficient routing and dissemination algorithms. We have also described several scenarios where ONs have been successfully deployed in recent years. Moreover, we analyzed the most important opportunistic routing and dissemination algorithms and techniques, showing their potential issues and how they might be fixed by leveraging social networks, node behavior prediction and selfish node detection and incentive mechanisms. We concluded our presentation by showing that the future trends in the area of mobile networking are veering towards data dissemination, through information-centric networks and the Internet of Things.

Acknowledgements This work was partially supported by the project "ERRIC - Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1", ID: 264207. The work has been cofounded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

References

1. Ian F. Akyildiz, Özgür B. Akan, Chao Chen, Jian Fang, and Weilian Su. Interplanetary internet: state-of-the-art and research challenges. *Comput. Netw.*, 43(2):75–112, October 2003.
2. Valerio Arnaboldi, Marco Conti, Franca Delmastro, Giovanni Minutiello, and Laura Ricci. DroidOppPathFinder: A context and social-aware path recommender system based on opportunistic sensing. In *WOWMOM*, pages 1–3, 2013.

3. Dionysis Athanasopoulos, Apostolos V. Zarras, Valerie Issarny, Evaggelia Pitoura, and Panos Vassiliadis. CoWSAMI: Interface-aware context gathering in ambient intelligence environments. *Pervasive Mob. Comput.*, 4(3):360–389, June 2008.
4. Archana Bharathidasan, Vijay An, and Sai Ponduru. Sensor networks: An overview. Technical report, Department of Computer Science, University of California, Davis, 2002.
5. Greg Bigwood and Tristan Henderson. IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 65–72, 2011.
6. Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, and Saleem Bhatti. Exploiting self-reported social networks for routing in ubiquitous computing environments. In *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, WIMOB '08, pages 484–489, Washington, DC, USA, 2008. IEEE Computer Society.
7. Chiara Boldrini, Marco Conti, and Andrea Passarella. Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. *Pervasive Mob. Comput.*, 4:633–657, 2008.
8. Raffaele Bruno, Marco Conti, and Andrea Passarella. Opportunistic networking overlays for ICT services in crisis management. In *International Conference on Information Systems for Crisis Response and Management. Washington, DC, USA*, 2008.
9. Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge Computer Lab, 2005.
10. Hafedh Chourabi, Taewoo Nam, Shawn Walker, J. Ramon Gil-Garcia, Sehl Mellouli, Karine Nahon, Theresa A. Pardo, and Hans Jochen Scholl. Understanding smart cities: An integrative framework. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 2289–2297, 2012.
11. Radu Ciobanu and Ciprian Dobre. Data dissemination in opportunistic networks. *CoRR*, abs/1202.2571, 2012.
12. Radu Ioan Ciobanu and Ciprian Dobre. Predicting encounters in opportunistic networks. In *Proceedings of the 1st ACM workshop on High performance mobile opportunistic systems*, HP-MOSys '12, pages 9–14, New York, NY, USA, 2012. ACM.
13. Radu Ioan Ciobanu, Ciprian Dobre, and Valentin Cristea. Social aspects to support opportunistic networks in an academic environment. In *Proceedings of the 11th international conference on Ad-hoc, Mobile, and Wireless Networks*, ADHOC-NOW'12, pages 69–82, Berlin, Heidelberg, 2012. Springer-Verlag.
14. Radu Ioan Ciobanu, Ciprian Dobre, and Valentin Cristea. Sprint: Social prediction-based opportunistic routing. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–7, 2013.
15. Radu-Ioan Ciobanu, Ciprian Dobre, Valentin Cristea, and Dhiya Al-Jumeily. Social aspects for opportunistic communication. In *ISPDC*, pages 251–258, 2012.
16. Radu-Ioan Ciobanu, Ciprian Dobre, Mihai Dascalu, Stefan Trausan-Matu, and Valtentin Cristea. Collaborative selfish node detection with an incentive mechanism for opportunistic networks. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1161–1166, 2013.
17. Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. From opportunistic networks to opportunistic computing. *Comm. Mag.*, 48(9):126–139, September 2010.
18. Michael Solomon Desta, Esa Hyytia, Jorg Ott, and Jussi Kangasharju. Characterizing content sharing properties for mobile users in open city squares. In *Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on*, pages 147–154, 2013.
19. Ciprian Dobre, Flavius Manea, and Valentin Cristea. CAPIM: A context-aware platform using integrated mobile services. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, pages 533–540, 2011.
20. Avri Doria. Providing connectivity to the saami nomadic community. In *Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

21. Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, HotNets-X, pages 1:1–1:6, New York, NY, USA, 2011. ACM.
22. Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *J. Netw. Comput. Appl.*, 28(1):1–18, January 2005.
23. Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. Opportunistic IoT: Exploring the social side of the Internet of Things. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 925–929, 2012.
24. S. Guo, M. Derakhshani, M. H. Falaki, U. Ismail, R. Luk, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav. Design and implementation of the KioskNet system. *Comput. Netw.*, 55(1):264–281, January 2011.
25. Hamed Haddadi, Pan Hui, Tristan Henderson, and Ian Brown. Targeted advertising on the handset: Privacy and security challenges. In Hans Jörg Müller, Florian Alt, and Daniel Michelis, editors, *Pervasive Advertising*, Human-Computer Interaction Series, pages 119–137. Springer, 2011.
26. Andreas Heinemann and Tobias Straub. Opportunistic networks as an enabling technology for mobile word-of-mouth advertising. In Key Pousttchi and Dietmar G. Wiedmann, editors, *Handbook of Research on Mobile Marketing Management*, PA: Business Scrience Reference, pages 236–254. Hershey, 2010.
27. Karen Henricksen and Ricky Robinson. A survey of middleware for sensor networks: state-of-the-art and future directions. In *Proceedings of the international workshop on Middleware for sensor networks*, MidSens '06, pages 60–65, New York, NY, USA, 2006. ACM.
28. Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.
29. Pan Hui and Jon Crowcroft. Predictability of human mobility and its impact on forwarding. In *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pages 543–547, 2008.
30. Pan Hui, Jon Crowcroft, and Eiko Yoneki. BUBBLE Rap: social-based forwarding in delay tolerant networks. In *Proc. of the 9th ACM int. symp. on Mobile ad hoc networking and computing*, MobiHoc '08, pages 241–250, New York, USA, 2008. ACM.
31. Pan Hui, Eiko Yoneki, Shu Yan Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, MobiArch '07, pages 7:1–7:8, New York, NY, USA, 2007. ACM.
32. Working Group RFID of the ETP EPOSS INFSO D.4 Networked Enterprise & RFID. Internet of Things in 2020. Roadmap for the future. http://bit.ly/cJGfaq, 2009. [Accessed September 15th, 2013].
33. Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, October 2002.
34. Viet-Duc Le, Hans Scholten, and Paul Havinga. Unified routing for data dissemination in smart city networks. In Viet-Duc Le, editor, *3rd International Conference on the Internet of Things, IOT 2012*, pages 175–182, USA, 2012. IEEE Press.
35. Leszek Lilien, Ajay Gupta, and Zijiang Yang. Opportunistic networks for emergency applications and their standard implementation framework. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa*, pages 588–593, 2007.
36. Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.
37. Abraham Martín-Campillo, Ramon Martí, Eiko Yoneki, and Jon Crowcroft. Electronic triage tag and opportunistic networks in disasters. In *Proceedings of the Special Workshop on Internet and Disasters*, SWID '11, pages 6:1–6:10, New York, NY, USA, 2011. ACM.

38. Waldir Moreira, Paulo Mendes, and Susana Sargento. Opportunistic routing based on daily routines. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–6, 2012.
39. Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.
40. Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, January 2004.
41. Tara Small and Zygmunt J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 233–244, New York, NY, USA, 2003. ACM.
42. Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 252–259, New York, NY, USA, 2005. ACM.
43. Kanchana Thilakarathna, Aline Carneiro Viana, Aruna Seneviratne, and Henrik Petander. The power of hood friendship for opportunistic content dissemination in mobile social networks. Technical report, INRIA, Saclay, France, 2012.
44. Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.
45. Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *Proc. of the 10th ACM Symp. on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 225–234, New York, NY, USA, 2007. ACM.