Efficient Location-aware Message Delivery for Encounter Networks

Igor Zavalyshyn, Nuno O. Duarte, Nuno Santos

INESC-ID / Instituto Superior Técnico, Universidade de Lisboa

Abstract. Location-aware mobile applications have grown in popularity. Existing applications, however, rely mostly on centralized architectures for nearby friend discovery and message forwarding, which forces users to constantly reveal privacy-sensitive location information to service providers. We present SpotNet, a middleware system for Android that provides location-aware message delivery service for opportunistic networks. SpotNet is based on short-range Bluetooth communication to exchange messages while precluding the existence of a trusted centralized service. SpotNet addresses the challenges of efficient message forwarding and resource utilization, and uses location-aware routing to select next hop for message forwarding based on 'closeness' to destination. We implemented SpotNet on the Android platform. Experimental results demonstrate that our approach is practical in real life scenarios.

1 Introduction

Generalized interest is growing in a class of mobile applications for location-aware message delivery. Facebook, Foursquare, and FireChat are examples of such applications, which allow users to search and communicate with friends closely located. Facebook and Foursquare rely on a centralized architecture to enable nearby friend discovery and message relay between users' mobile devices. Friend discovery functionality is provided by maintaining a central database of approximate user locations inferred from a device's WiFi or 3G network interfaces. FireChat adopts a similar centralized architecture, but complements it with a fallback mechanism in case there is no Internet connectivity. In particular, FireChat leverages WiFi Direct or Bluetooth to detect nearby friends and to establish a peer-to-peer (P2P) network for local message exchange.

However, a consequence of such a centralized architecture is that location information collected by users' devices must be surrendered to the service provider. All this information can then be centrally aggregated by a single entity and be exploited or shared with third parties without users' knowledge or consent, e.g., to provide additional services or targeted advertisement [11, 12, 16, 21]. In such situations, users have little or no control of their data once its uploaded to service provider servers [4, 6, 7].

In this paper, we aim to investigate the feasibility of an alternative but practical architecture for providing location-aware message delivery while preserving users' privacy. In particular, we wish to explore a fully decentralized solution in which both friend discovery and message forwarding operations can take place by relying exclusively on the opportunistic interaction between users' devices: no centralized service should be involved in order to provide this functionality. We seek to design a system that is practical (i.e., works with modern unmodified smartphones and mobile devices), energy-efficient, and general enough to allow client applications to take advantage of its services.

We present SpotNet, a middleware system for Android which provides private location-aware message exchange capability between any set of devices in a mobile ad-hoc network without relying on any cloud infrastructure or Internet connectivity. Through a simple API, application developers can easily embed SpotNet's message delivery functions into their own mobile apps. Through SpotNet, users can send messages either directly or through the set of intermediate nodes, so-called 'carriers', that are unaware of messages' content. Messages can be sent generically to all mobile devices located around certain *spots*, i.e., specific geographical points such as coffee shops, streets, or indoor spaces within shopping malls, etc. To enable decentralized message delivery, we use location-based forwarding to opportunistically select suitable 'carriers' based on their previous location track. For energy efficiency reasons, location-based routing is achieved by relying on small low-powered Bluetooth devices called *beacons* which serve as stationary markers deployed across the city in shops and restaurants, and are used for checkpointing the location track of each node in the network. Beacons emit unique IDs which can then be used to refer to specific geographic spots.

Regarding its novelty, SpotNet can clearly distinguish itself from existing encounter-based applications previously developed for Android platforms such as FireChat [8] or Haggle [17]. These applications use Bluetooth or WiFi Direct communication to exchange files or messages between nearby devices. To the best of our knowledge, SpotNet is unique at both: 1) providing a multi-hop message delivery capability without requiring any support from fixed infrastructure, and 2) leveraging a beacon infrastructure for location-based message forwarding. We implemented SpotNet on Android and our preliminary experimental results demonstrate the feasibility of our approach.

2 Related Work

A lot of work, especially theoretical one, has been carried out in the field of mobile ad hoc networks (MANETs). In general, MANETS consist of wireless networks of mobile devices which can form opportunistically. In such networks, adequate routing algorithms are fundamental for network performance. Message routing must constantly adapt to moving network nodes (i.e., mobile devices). Many routing algorithms abound, ranging from epidemic [22] where information propagates by flooding the network with multiple copies of the same message, to more sophisticated algorithms that consider geographic location of the node and its distance from the destination [14, 15] as well as to node mobility patterns [10, 13]. SpotNet leverages some of this work by using a combination of the geographic routing algorithm with the spray-and-wait technique [18, 24]. Messages thus propagate in a specific direction of destination and their number is limited to efficiently use available resources and avoid network congestion.

Several mobile applications such as Haggle [17], AllJoyn [23] or FireChat [8] use device-to-device (D2D) communication to disseminate information in the network. Users can express their interests in the specific topic and receive notifications from nearby users that have similar interests. These applications use Bluetooth or WiFi Direct radio for communication between users. Internet connection is not needed in this case since messages are sent directly from device to device. However, as opposed to SpotNet, in these applications, message delivery is possible only between devices connected in the same wireless network.

EnCore is a social platform [1] that provides an interface for secure and privacy-preserving communication. In EnCore, co-located devices establish a unique encounter ID and a shared key that is used for secure communication between selected users. Similarly to SpotNet, discovery of new devices is done periodically. However, although EnCore provides a secure and reliable D2D communication platform, it does not support message forwarding between devices that are not immediate neighbors of each other. SpotNet on the other hand provides such functionality by default and relies on location-aware routing to deliver messages from sender to receiver in a timely manner.

3 SpotNet Design

We present the design of SpotNet, a middleware system for Android platforms which provides a location-aware message delivery service for opportunistic networks. This middleware consists of a library that is linked with client mobile applications. Its basic message delivery service is provided through a simple API which basically allows for sending and receiving messages. SpotNet API provides simple *send* and *receive* primitives. In addition to the message itself, send takes as input the location of the desired destination nodes (explained in Section 3.2). The receive primitive runs a service that waits for incoming messages.

The underlying infrastructure assumed by SpotNet comprises two types of hardware components. On the one hand, there are *mobile devices* owned by SpotNet users. These devices coincide with the network nodes of SpotNet. In addition to offering end-users an interface for sending and receiving messages, nodes are responsible for relaying messages between users. Messages are staged on the nodes and carried along until other nodes appear within their wireless range that can deliver the message closer to its destination with higher probability. This is the so-called pocket-switched networks [10] which tend to have a structure that is correlated with human movement patterns and the set of places visited on a daily basis. On the other hand, SpotNet also relies on a set of stationary devices named *beacons* whose role is to advertise a fixed location to SpotNet's mobile devices. Beacons can be deployed anywhere in places of interest and emit a unique ID. In order to send a message, users must specify the location of receiver nodes. The location is denoted in terms of "spots", i.e., collections of beacon IDs. SpotNet ensures that all the nodes that can sense the given beacon can receive the messages destined for this location. SpotNet leaves message security up to the application layer and provides best-effort delivery guarantees. Due to the opportunistic operation mode of SpotNet's routing mechanism, message delivery latency may sometimes be too high for certain applications. We leave it up to applications themselves to adopt a complementary transport service in order to speed up message delivery if necessary.

Next, we describe the main design features of SpotNet focusing on: device discovery and basic D2D communication mechanisms, message addressing scheme, location-aware routing, and message forwarding.

3.1 Energy-efficient Device Discovery and Communication

To enable device discovery and communication, SpotNet leverages Bluetooth, which is widely deployed, and is more energy-efficient than comparable technologies, such as WiFi [3]. Current Bluetooth specifications [3] require devices to be paired before any data connection can be established. However, such kind of interaction is cumbersome in SpotNet scenarios. To send or receive a message, users would need to manually pair their devices with the devices of users they encounter on the way. This process not only slows down device discovery and message transfer procedures, but also prevents device interaction in situations where users are not familiar with each other.

To overcome existing limitations of Bluetooth pairing and preclude the involvement of the user in device discovery and message forwarding, we utilize insecure Bluetooth sockets [9]. Bluetooth sockets are supported by most of Android smartphones nowadays and enable connections to be established between devices without pairing. The side-effect of this is that built-in authentication measures are disabled. Nevertheless, authentication and encryption may be performed on higher levels of the application. In SpotNet, the sender initiates a connection by creating an insecure Radio Frequency Communication protocol (RFCOMM) socket connection waiting for a response from the receiver that is constantly listening for incoming connections. Once the socket has been established, the RFCOMM protocol proceeds by creating a channel for data transfer. This channel is later destroyed when the data transfer is successfully completed.

3.2 Location-based Message Addressing Scheme

A central goal of SpotNet is to offer a location-based message delivery service. To specify device locations, one possible approach is to leverage GPS interfaces bundled into modern phones to obtain the current device position and use GPS coordinate ranges as addressing scheme. However, keeping the GPS sensor active even for short time periods yields high battery consumption. Due to severe battery drainage, users tend to keep the GPS positioning system turned off most of the time. Moreover, GPS can not be used indoors, thus limiting the scope of possible locations in the user profile. WiFi- or 3G-based location services could alternatively be used. However, not only do they also require access to the energy-inefficient WiFi interface, but can also reveal the device location to a third-party, thereby potentially compromising users' privacy. In fact, a device could be tracked by just following the device MAC and knowing the SSIDs of deployed WiFi APs, for example across a city or inside a shopping mall.

To overcome the hurdles of GPS and WiFi, we propose to use a network of *beacons*, i.e. small low-powered Bluetooth-enabled devices that are distributed across the city. We envision beacons to be deployed at various places, e.g. at main bus stations, business centers, shopping malls, universities, museums, restaurants and cafes. Beacons play the role of location anchors that are constantly advertising their location to nearby devices such that passer-by users can detect this advertisement and store the location ID. The main advantage of this approach is its power efficiency [19]. Beacons use Bluetooth Low Energy (BLE) [3], a low-power wireless technology for communication with smartphones and other devices. Since many smartphone users keep the Bluetooth radio active, the interaction with beacons and other users becomes seamless. Moreover, beacons are passive elements that emit wireless signals only. As a result, beacons cannot keep track of devices' location therefore preserving users' privacy. Another benefit is that beacons can also be adopted for indoor location tracking.

Depending on beacons, however, could raise some concerns about cost and maintenance effort. We address such concerns, firstly, by observing that beacon technology is relatively inexpensive. As of now, the prices for BLE-beacons can start from as cheap as 5 USD a piece with wireless range up to 50 meters depending on the battery level and configured transmission power [2]. Secondly, rather than depending on a single entity to deploy and maintain beacon hardware, which requires significant investment and maintenance effort, SpotNet's beacon infrastructure is fully decentralized. In particular, beacons are deployed independently by unrelated parties (*beacon providers*), e.g., a coffee shop, a parking lot, a shopping mall, etc. A beacon provider only needs to buy and deploy beacons to cover a geographical area of interest. Thus, any shop owner can buy a beacon and deploy it at his place. SpotNet devices can sense the MAC address emitted by each beacon and use it as a location anchor for routing purposes. This list of locations is stored and constantly updated by SpotNet devices. To allow users to easily refer to a given location, beacons' MAC addresses can be named in terms of *spots*, i.e., human-readable aliases that can refer to locations that users can understand (e.g., "McDonalds", "Downtown"). Spots can be used to identify the destination of messages. SpotNet ensures it will be delivered to devices surrounding the beacons covered by that particular spot. Beacon addresses can be mapped to spot names by users themselves or through an intermediate directory service. SpotNet's decentralized architecture allows the beacon infrastructure to grow organically and economically. Note that, although SpotNet uses beacons as its primary location-sensing technology, GPS or WiFi may alternatively be used within the city areas where beacons have not yet been deployed.

3.3 Location-aware Routing

A message is considered to have arrived to its destination if it has been received by the nodes located in the vicinity of destination spot, i.e., to the nodes that can sense at least one of the beacons of the spot's beacon set. The question is then how to route the messages between sender and destination nodes considering that messages must be propagated as SpotNet devices interact opportunistically.

SpotNet exploits the fact that individuals tend to follow certain patterns by visiting the same set of places over and over again: their homes, working places (e.g., office, university), restaurants, etc. Such patterns normally reflect users' daily routines and tend to be fairly stable over time [5]. SpotNet leverages that regularity as a way to predict which spots a given device will likely visit in the future: messages that are destined to one of such spots, can then be piggybacked in that device in order to reach its destination with high probability.

To materialize this idea, each SpotNet device maintains a *location profile* for each user which includes a history of the spots visited by the user. Periodically, each user device runs a Bluetooth discovery process in order to detect nearby beacons. Once a beacon is detected, the node adds the location ID and current timestamp to the user's location profile. The location profile keeps track of the spots' visiting patterns of a given user. If other users know at least some of those frequently visited spots they can efficiently forward their messages towards the destination's location. To do so, they would need to relay the message through the phones of other users, i.e., carriers, that are more likely to go near the destination. To determine potential message carriers for certain spots, in addition to detecting nearby beacons, SpotNet's Bluetooth discovery process also locates nearby user devices and exchanges location profiles with them. Based on the information contained in the received location profile, a node updates a *routing matrix* which indicates for each carrier node C (indicated in the matrix's lines) the probability that C visits a given destination spot S (indicated in the matrix's columns). This probability is calculated by determining the fraction of time (e.g. hours) in which C was able to sense a beacon of spot S considering a predefined time period (e.g., a day or a week). This fraction can be determined trivially through direct inspection of C's location profile. For privacy issues, users may specify which places from their location profile they do not want to be disclosed (e.g. home or office locations). Alternatively, users may decide not to share the full location profile with other users, but only its *probability vector*, i.e., the line of the routing matrix which corresponds to the spot visiting probability of that particular device. To avoid unlimited growth of the routing matrix, it is also possible to limit its maximum number of lines (N_L) and columns (N_C). In that case, SpotNet fills in the routing matrix based on the most recently visited N_L spots and encountered N_C carrier nodes. The routing matrices maintained by SpotNet nodes represent then each node's location pattern and play a central role in the message forwarding algorithm, as explained in the next section.

3.4 Message Forwarding Algorithm

As mentioned above, SpotNet's routing decisions are based upon devices' probability to visit certain spots. The message forwarding algorithm implemented by SpotNet leverages this information to identify which devices are potentially good candidates to speed up message delivery by carrying it to its destination.

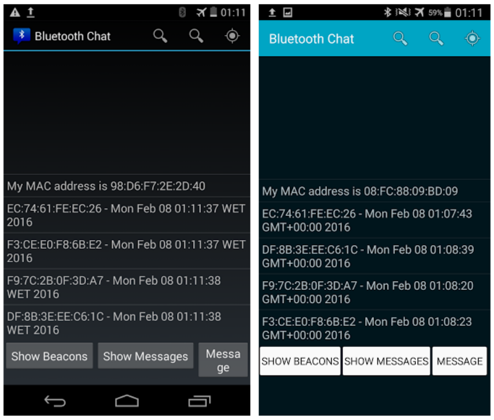
SpotNet's message forwarding algorithm has to analyze the line in the routing matrix that corresponds to the encountered device and compare it with the destination spot specified in the message. The message is relayed if the spots of both the encountered location profile and the destination location are equal—i.e., a match exists—otherwise it is stored until a better candidate is found.

The main challenge, however, is to decide if a match exists. For efficient message forwarding, the number of devices involved in the delivery of each message must be limited but still provide high delivery rates. At the same time, device discovery has to be performed as often as possible to detect beacons and other nearby users while keeping in mind battery consumption. In other words, message forwarding solely based on the location profile results in multiple carriers of the same message. Every message consumes memory and CPU resources of the carrier and limits the amount of messages it can create and store. This kind of approach is similar to epidemic routing [22] where information is disseminated between all the encountered nodes using pair-wise message exchange. Even if the scope of possible message carriers is limited by selecting only those whose probability of reaching the destination spot is non-null, multiple message duplicates will still be produced, thereby increasing network resource usage inefficiency.

To reduce the number of message copies circulating across the network, SpotNet limits the number of times each message is forwarded by leveraging the *spray-and-wait technique* [20]. According to it, during the first phase of the routing process, the sender 'sprays' a message into the network by forwarding it to the number of encountered devices. During the second phase, all the resulting message carriers including the sender itself store the message and 'wait' until reaching the destination device. Once the final recipient appears within the radio range of any of them, the message is finally delivered. We use spray-and-wait technique in combination with location-aware forwarding. Each message has a spray counter in the header which is updated by each message carrier. Once this counter reaches a specific value, the message can not be forwarded further but is instead stored until reaching the destination. By doing so we specifically limit the number of hops in order to reduce the message delivery overhead. Initiating the Bluetooth connection and establishing the RFCOMM sockets with the detected devices is a time- and resource-consuming process. Therefore, devices perform it only if the messages they carry can still be forwarded further according to the spray count, or if they come across the destination device. To avoid repeated connections, every device maintains the list of devices it contacted recently. The connection to a device is only initiated if it does not appear in the list.

4 Implementation

We implemented the SpotNet functionality as a middleware layer for Android OS. SpotNet code was written in Java. To test our system, we implemented a client application which provides users with a simple interface to send and receive messages to and from other users visiting specific spots. This application allows users to communicate even if they do not know each other in person. It



My MAC address is 98:D6:F7:2E:2D:40

EC:74:61:FE:EC:26 - Mon Feb 08 01:11:37 WET 2016

F3:CE:E0:F8:6B:E2 - Mon Feb 08 01:11:37 WET 2016

F9:7C:2B:0F:3D:A7 - Mon Feb 08 01:11:38 WET 2016

DF:8B:3E:EE:C6:1C - Mon Feb 08 01:11:38 WET 2016

My MAC address is 08:FC:88:09:BD:09

EC:74:61:FE:EC:26 - Mon Feb 08 01:07:43 GMT+00:00 2016

DF:8B:3E:EE:C6:1C - Mon Feb 08 01:08:39 GMT+00:00 2016

F9:7C:2B:0F:3D:A7 - Mon Feb 08 01:08:20 GMT+00:00 2016

F3:CE:E0:F8:6B:E2 - Mon Feb 08 01:08:23 GMT+00:00 2016

Fig. 1: Location profiles of two users: Alice (left) and Bob (right).

could be used, for example, to ask for directions to the nearest metro station, or notify visitors of a specific place of upcoming events.

Once the client application is installed and activated, it requests permissions to turn on Bluetooth radio and make a device discoverable. As mentioned above, the SpotNet middleware runs a device discoverability service to allow the device to participate in the SpotNet network. Every device runs the discovery process every 20 seconds. The list of detected devices is then analyzed and processed accordingly. If a beacon is detected, the SpotNet middleware updates the location profile with the beacon's MAC address and current time. If any other device is detected, it exchanges profiles, refreshes the routing matrix, and implements the message forwarding algorithm described in Section 3.4. In case there are messages to be delivered to the device (i.e., the current device location coincides with the message's destination spot), the SpotNet middleware notifies the application of the incoming message. The application reads the message from an internal buffer of the SpotNet middleware, and displays the message to the local user. (Figure 1 illustrates the location profiles of two users as shown by our client application.)

In order to send a message to a specific user, the sender needs to know the destination spot. In our implementation, the SpotNet middleware provides an API call that allows the client application to associate spot names to beacon's MAC addresses. The application offers a simple visual interface that allows the user to manage their spots and internally propagates that association to the SpotNet middleware through this API call.

Internally, the SpotNet middleware appends multiple parameters to every message. These parameters include: a unique identifier, the number of times it was forwarded (spray count), the MAC addresses of sender and destination spot's beacons, name of sender spot, and name of destination spot. In our current implementation, the list of all the deployed beacons is hard-coded into the application so that the user can just select them from a drop-down list.

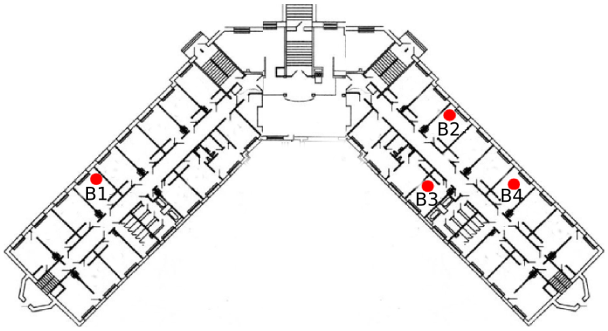


Fig. 2: Location of beacons during the indirect message forwarding experiment

5 Evaluation

We evaluated our SpotNet prototype using a small-sized experimental setup. The main goal of the experiments was to address the following questions:

- How do messages propagate in a SpotNet network, particularly if next hop is selected based on its location profile? (Section 5.1)
- How does the SpotNet client application running on users' devices affect battery consumption? (Section 5.2)

Experimental Setup. We used a small setup consisting of: three LG Nexus 4 and one Samsung Galaxy S4 smartphones all running Android OS, with the SpotNet application installed, and four Bluetooth Low Energy beacons iBKS105. The smartphones were given to 4 users working at different rooms of the same floor at the INESC building. We deployed beacons at each room and integrated their IDs and corresponding locations into the application (see Figure 2). Users could communicate either directly by being at the same room, e.g. during a meeting, or by message passing through other users' devices. To avoid false records in the location profiles, the advertising power of each beacon was configured in a way that it could only be detected within a couple of meters around it.

5.1 Message Propagation

To evaluate SpotNet's message propagation mechanism, we tested our client application under two scenarios. In the first scenario, a message is sent directly to another device located in the same room. In the second scenario, a message is sent to a remote device indirectly using intermediate devices as message relays.

1. Direct message transmission. We start from the first application scenario in which two users Alice and Bob are sitting in the same room within the range of beacon B1 (see Figure 2). Alice creates a message which is addressed to Bob, writes a message text, specifies his location and clicks the Send button. The message is then stored in the message buffer. Since both devices are within the wireless range of each other they are detected during the discovery process and all the available messages are sent directly without requesting the location profile information. The message was delivered in 17 seconds. The delay is caused by the discovery process since it takes 12 seconds on average to complete. Additional time is needed to initiate a connection and open the socket. (This process may be accelerated by using the latest 4.1 version of Bluetooth.) Although a message was delivered directly to destination, spray count was still incremented by one.

2. Indirect message transmission. In the second application scenario, a message is relayed from the sender to the receiver by intermediate users. In this experiment, Alice sends a message which arrives to Bob's location relayed by Charlie. Initially, Alice, Bob, and Charlie are placed apart from each other located close to beacons B1, B2, and B3, respectively. Since Bob and Charlie work in continuous rooms, Charlie's device is also able to sense beacon B2. The experiment starts when Alice creates a message for Bob using the SpotNet application. As destination spot of the message, Alice uses a spot ID that comprises beacon B3, which is placed in the meeting room of Bob's department. Since there is no direct connection between Alice and Bob's devices, the message is then stored in the buffer while the application is looking for devices to deliver it to the recipient. We then recreate a scenario in which Alice and Charlie meet each other in the kitchen area during the coffee break. This opportunity allows devices to become directly reachable and exchange location profiles. Since Charlie and Bob work in the same department, Charlie's device probability of reaching B2 is greater than zero and SpotNet forwards Alice's message to Charlie's device. Eventually, Charlie returns to his office and encounters Bob. Once SpotNet detects Bob's device, the message is automatically sent to his device through direct communication and the message arrives to its final destination. In total, this experiment took approximately 1 minute, from which 34 seconds were taken up by SpotNet's point-to-point protocols performed between devices.

Summary. The experiment results showed that the forwarding mechanism works as intended in case of direct and indirect communication. For direct communication the message delivery delay mainly depends on the device discovery duration. For indirect communication next hop is selected based on location profile information. In case of multiple devices detected with the same location profile, the algorithm selects the device that is supposed to be at the location of destination earlier. Location-aware routing based on beacon deployments provides a simple and energy-efficient alternative to GPS-based routing protocols.

5.2 Battery Consumption

As mentioned in Section 3, device discovery and message forwarding are costly procedures in terms of energy consumption. To measure the battery consumption of SpotNet, we carry out an experiment in which two SpotNet clients are

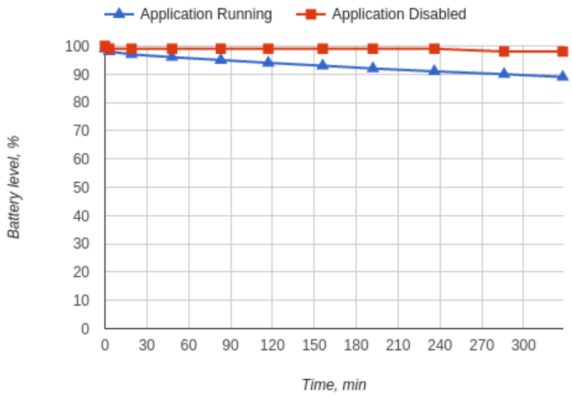


Fig. 3: Battery consumption of SpotNet application

constantly exchanging messages between each other every 20 seconds during a 6 hour period. The spray count update function is intentionally disabled so that each client repeatedly discovers nearby devices, requests their location profile and forwards the message. Batteries are initially fully charged on both devices and the battery level is measured every minute while the experiment is running. All other applications are disabled on both devices and flight mode is activated. We consider the case when neither SpotNet nor other applications are running as a baseline for the experiment.

Figure 3 shows our results. SpotNet continuously running on the device and exchanging messages every 20 seconds consumes only 10% of the battery in 6 hours. The battery level changes linearly and steadily. These results show that SpotNet can be efficiently used on modern smartphones without significantly degrading their autonomy. Combined with automatic sleep mode in user-defined locations, it can seamlessly run in the background during the whole day. This factor is essential for network growth and overall user satisfaction.

6 Conclusion

We presented SpotNet a privacy-aware decentralised message delivery middleware that allows users to communicate without relying on Internet connectivity and cloud infrastructure. Location-aware message forwarding based on beacons may be efficiently used in a city environment where they are deployed in various public places. Interaction with these devices is seamless to the user and does not require any user attention. We implemented the SpotNet logic in the form of middleware for Android OS. Preliminary evaluation demonstrates the feasibility of this approach in real life scenario and display modest energy-consumption.

Acknowledgments: This work was partially supported by the EC through project H2020-645342 (reTHINK), and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID).

References

1. Aditya, P., Erdélyi, V., Lentz, M., Shi, E., Bhattacharjee, B., Druschel, P.: EnCore: Private, Context-based Communication for Mobile Social Apps. In: Proc. of MobiSys (2014)
2. Aislelabs: The Hitchhikers Guide to iBeacon Hardware: a Comprehensive Report by Aislelabs (2015) (2015), `http://www.aislelabs.com/reports/beacon-guide`
3. Bluetooth, S.: Bluetooth Specification Version 4.2. Bluetooth, SIG (2014)
4. Boggs, B.C., Edwards, M.L.: Does What Happens on Facebook Stay on Facebook? Discovery, Admissibility, Ethics, and Social Media. ILL. BJ (2010)
5. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless communications and mobile computing (2002)
6. Cheng, J.: Are Deleted Photos Really Gone from Facebook? Not Always. Ars Technica (2009)
7. Cheng, J.: 'Deleted' Facebook Photos Still Not Deleted. Ars Technica (2010)
8. Cohen, N.: Hong Kong Protests Propel FireChat Phone-to-Phone App. The New York Times pp. 26–28 (2014)
9. Google: Android API documentation (2016), `http://developer.android.com`
10. Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., Diot, C.: Pocket Switched Networks and Human Mobility in Conference Environments. In: Proc. of the WDTN (2005)
11. Kincaid, J.: Senators Call Out Facebook on Instant Personalization, Other Privacy Issues. TechCrunch, April (2010)
12. Kravets, D.: Judge Approves $9.5 Million Facebook 'Beacon' Accord. The Wired, March 5, 125–126 (2010)
13. Lindgren, A., Doria, A., Schelén, O.: Probabilistic Routing in Intermittently Connected Networks. ACM SIGMOBILE 7(3) (2003)
14. Mauve, M., Widmer, J., Hartenstein, H.: A Survey on Position-based Routing in Mobile Ad Hoc Networks. Network, IEEE 15(6), 30–39 (2001)
15. Perkins, Charles and Belding-Royer, Elizabeth and Das, Samir: Ad Hoc Ondemand Distance Vector (AODV) Routing. Tech. rep. (1999)
16. Sandberg, S.: The Role of Advertising on Facebook. The Facebook blog, July (2010)
17. Scott, J., Crowcroft, J., Hui, P., Diot, C.: Haggle: a Networking Architecture Designed Around Mobile Users. In: Proc. of WONS (2006)
18. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data Mules: Modeling and Analysis of a Three-tier Architecture for Sparse Sensor Networks. Ad Hoc Networks (2003)
19. Siekkinen, M., Hiienkari, M., Nurminen, J.K., Nieminen, J.: How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4. In: Proc. of WCNCW (2012)
20. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks. In: Proc. of WDTN (2005)
21. Steel, E., Fowler, G.: Facebook in Privacy Breach. The Wall Street Journal 18, 21–22 (2010)
22. Vahdat, A., Becker, D., et al.: Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University (2000)
23. Wang, Y., Wei, L., Jin, Q., Ma, J.: AllJoyn Based Direct Proximity Service Development: Overview and Prototype. In: Proc. of CSE (2014)
24. Zhao, W., Ammar, M., Zegura, E.: A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In: Proc. of MobiHoc (2004)