Has anyone seen my Goose?

- Social Network Services in Developing Regions

Narseo Vallina-Rodriguez*, Pan Hui†, Jon Crowcroft*
nv240@cam.ac.uk, pan.hui@telekom.de, jon.crowcroft@cl.cam.ac.uk
* University of Cambridge
† Deutsche Telekom Laboratories/TU-Berlin

***Abstract*— In developing regions, Internet connectivity is extremely poor, while mobile phone penetration is much higher. The inhabitants of developing conuntries still rely on traditional social mechanisms, such as word-of-mouth, to gather most information that they use in everyday life.**

In this paper, we propose *Goose*, a Social Network Service (SNS) architecture for developing regions, which provides services including friend searching, resource sharing and information seeking. Goose uses both the limited GSM coverage and Delay Tolerant Networking (DTN) technology on mobile phones, to enable social information exchange even without cellular data coverage.

I. Introduction

According to the World Internet Statistics, 76.5% of the world's population still did not have Internet access in year 2008 [1]; this figure mostly comprises inhabitants of developing regions. This population is so large that we must think about the kind of Social Network Service (SNS) that we can provide to them even without Internet connectivity. SNS can improve their standards of living and communication by introducing new services that use the existing social links and interactions within the members of the communities.

Fortunately, the mobile phone has a much higher penetration rate in these regions, compared to the Internet and fixed telephone lines. For example in India, according to a report from the Telecom Regulatory Authority of India, about 5 million new mobile subscribers join the mobile phone market every month, which is far greater than the total PC penetration of 5 million in the whole year of 2005/06 [1]. Mobile phone usage is also reported to be high in African countries. In fact, Marsden *et al.* [20] point out that the deployment of SNS for developing regions should focus on mobile phones.

However, the cellular networks in these regions are usually very poor and expensive, and remain a disproportionately urban phenomenon due to the lack of coverage in many rural areas. On the other hand, satellite networks have coverage in most rural areas but these are extremely expensive. Because of that, it is unrealistic that applications featuring *friend browsing* functionality such as Facebook, Myspace or Orkut would become popular there. Learning from others [19] [18] [17] and from our own experience, the types of SNSs suitable to these regions fall under four main headings: friend searching, resource sharing, information seeking, and product marketing.

[1]Source: `http://www.internetworldststs.com/stats.htm`

In this paper, we identify and summarise the networking characteristics of developing regions, and propose a radical new approach to SNS called *Goose*. Goose is a distributed social network built around the users' social interactions, their inherent mobility, and the presence of local area network interfaces both to forward information and to provide a security scheme. Goose allows sending voice and text messages over Bluetooth, which relies on traditional social networks and human mobility to distribute information in a similar way to how we transmit information by word of mouth.

The rest of this paper is as follows: characterisation of network scenarios (Section II), system architecture (Section III), technical challenges (Section IV), preliminary results and evaluations (Section V), related work (Section VI), and conclusions (Section VII).

II. Characteristics of Developing Regions

Mobile phone penetration in developing regions is much higher than fixed line technologies, as Table I shows. However, despite the rapid deployment of mobile phone technology in both urban and rural areas in developing countries, network coverage is still low, as Figure 1 shows for Sub-Saharan countries [7].

Country	Mobile	Landline
EU	0.95	0.48
USA	0.83	0.53
China	0.41	0.27
India	0.25	0.03
Guatemala	0.76	0.1
Albania	0.63	0.1
Morocco	0.57	0.07
South Africa	0.86	0.09

TABLE I

Fixed and Mobile Phone subscribers per citizen

In countries in central Asia [19] [18]—especially post-Soviet countries such as Kazakhstan, Kyrgyzstan, Tajikistan, and Uzbekistan—the decaying Soviet telecommunications infrastructure limits household access to information. Even in urban areas, the Internet is expensive and inconvenient, and can mostly only be accessed at public access sites such as Internet Cafes. Furthermore, according to our own experience, the data rate for UMTS access is very low, even in urban environments. The standard technology for data is still GPRS, despite many European mobile operators investing in countries

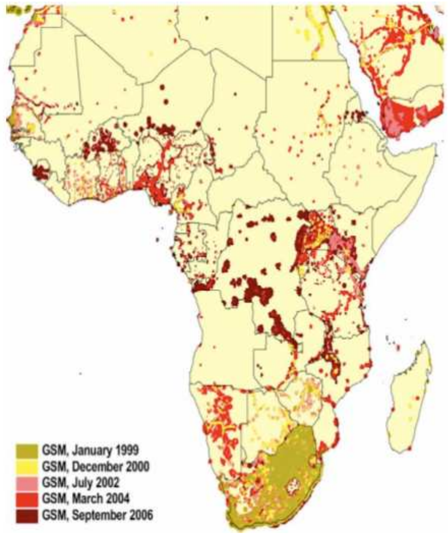

Fig. 1. Mobile Phone Expansion in Sub-Saharan Countries: 1999-2006. Source: The World Bank Development Research Group

such as India, Egypt and Brazil. (For example, it took several minutes to load the popular homepage of British Airways in Bangalore city in early 2009.)

On the other hand, mobile phone technology has more than 10 times the penetration rate of personal computers. The motivation for using these devices in deploying SNSs is clear: the combination of low data-rate networks, frequent disconnections, and ubquitous mobile handsets with local area network interfaces (such as Bluetooth) implies that handset-based software and services can be deployed to a large percentage of the population even where GSM network coverage is unavailable.

Our idea applies particularly to India and South Africa. In the case of India, there are, on average, 5 million new mobile phone users per month [1]; while in South Africa, around 86% of the inhabitants have a cellular handset despite only 11% of the population earning enough money to be registered for income tax[2].

Figure 2 shows that mobile penetration is quite high in the whole world with a rate of 0.5 mobile phone contracts per citizen in most of the countries. However, the literacy rate in these countries is quite low (as Figure 3 shows) and the population density is generally high (e.g. Bangladesh has a literacy rate of 47% and a population density of 1045 inhabitants per km^2). This is a hinderance for the overall propagation of information using humans as a transport mechanism, and the support of voice messages for the non-literate.

Short Message Service (SMS), is clearly a killer application for mobile services worldwide, with an enormous role on people's social life that grows every year: in 2006, 25 billion SMS were sent globally, compared to 12 billion in 2005 [17]. In [10], the authors mention that text messaging services include SNS features due to their ability to maintain social cohesion and interconnect individuals in a community.

[2]Source: CIA World Fact Book, `https://www.cia.gov/library/publications/the-world-factbook/index.html`

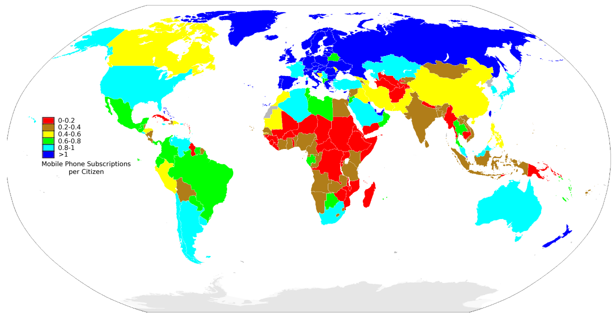

Fig. 2. Mobile Penetration in the world

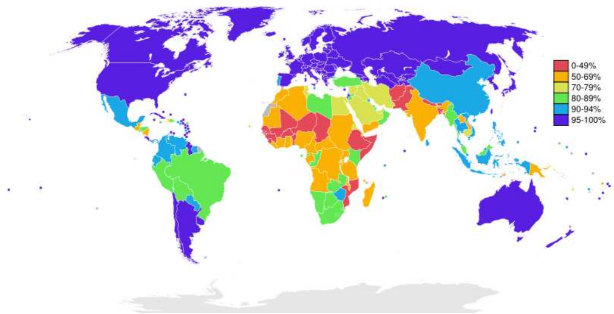

Fig. 3. Literacy Rate in the world

We posit that the SNS should be as simple as SMS in order to be easy for users to adopt. We believe that the above evidence supports the creation of an SMS-style SNS that enables users to send both text and voice messages (for non-literate users), and strengthens the social bonds and communication within the community. The reality of intermittent connectivity, and the lack of GSM coverage in many areas of developing countries, suggest the use of a Delay Tolerant Network (DTN) [13] that exploits human social links and inherent human mobility, by combining Bluetooth and the deployed GSM network in unicast, multicast and broadcast fashions.

A. Application Scenarios

Goose combines features from SNS (such as status and activity reports, alarms, microblogging and resource-availability reports) with the social uses of SMS (such as group coordination [22], leisure purposes [14] and group messaging [9]). Moreover, Goose will extend GSM network coverage by providing connectivity in areas where the network service is limited, non-existent or shut down because of government censorship.

The basic use cases are illustrated in Figure 4. In the unicast case, a user could use Goose to send a message to his wife about the location of their dog. The message can spread over both the GSM network and the social network taking advantage of social encounters and epidemic algorithms[3]. Intermediate nodes buffer the message in persistent storage during connection outages, retransmitting it to other devices until it reaches the destination. In the broadcast example, Vijay sends a message to all the members of the community asking for help since he has lost his goose.

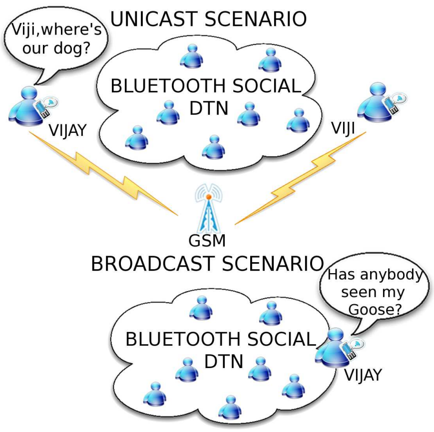

Fig. 4. Goose Application Scenarios.

III. System Architecture

Goose is designed to account for the capabilities of current low-cost handsets and their computational resource constraints. Every Goose device must be able to scan for other devices over Bluetooth, provide Bluetooth I/O, support background processes (so as to receive messages while the user is running other applications), monitor energy consumption and perform low-level memory management. However, mobile phones are characterised by their power constraints, low-power CPU, limited RAM and persistent storage plus their inherent mobility.

The Goose architecture is event-driven, which reduces battery consumption, and enables it to decide which tasks to perform considering the available resources at the moment. The architecture is based on four key elements that will be described in the following subsections: the User Interface, the Contacts Manager, the Forwarding Manager and the Network Manager. All elements communicate with each other through the well-defined APIs as shown in Figure 5.

A. User Interface

The User Interface enables interaction between the user and Goose. It is composed of different views that allow users to create and display voice and text messages, add senders of received messages to the local address book, search for friends' details, exchange contact details, and encrypt messages using a public key.

[3]Note that throughout this article we use the terms epidemic algorithms, rumour spreading and social cascade, which are similiar but refer to different concepts. Rumour spreading is how Goose gets information around between people. Epidemic is a forwarding scheme for getting data to lots of people in the broadcast case, and social cascade is the mechanism used by Goose for software distribution

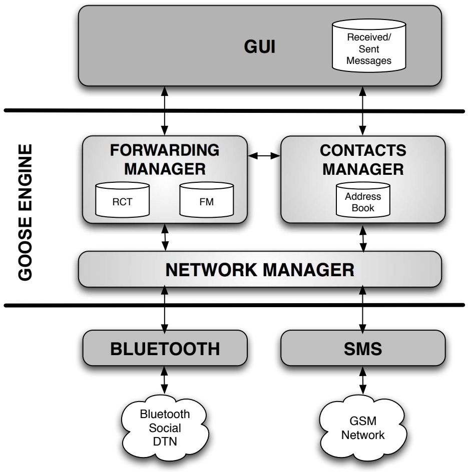

Fig. 5. Goose Architecture

Goose is designed for users that may not be literate and may have difficulties in understanding a complex user interface. We aim to provide a good user experience for every user by designing a highly graphical, accessible and intuitive interface as shown in Figure 6.

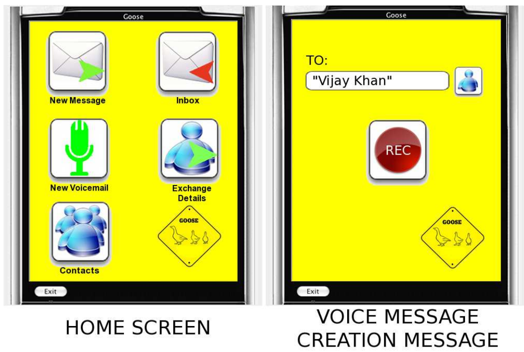

Fig. 6. User Interface for J2ME MIDP 2.0 capable devices. It will support both keyboard-based and touch-screen mobile phones

The only inputs that Goose needs from a user are his/her contact details (i.e. name and phone number). In the case of a non-literate user, this information can be input by a trusted friend or family member. Afterwards, to extend his/her address book, the user will be able to exchange contact details over Bluetooth, or using the user interface when a new message is received. The ability of the non-literate users to associate the characters from the address book with other users may be critical in enabling them to select the recipients of the message they may create in the future.

Figure 6 also shows the interface for sending a voice message. The user must select the recipients of the message by pressing the Buddy icon, followed by the recording button. Since the time is limited to a minute of audio, the mobile phone vibrator will be activated every second for the last 10 seconds before the limit in order to advise the user that his/her time is running out. In the case of a text message, the user interface is similar to a current SMS interface with an additional feature to allow the users to select the priority of the message.

B. Contacts Manager

The Contacts Manager is responsible for storing and managing the users' social network, which is built around her address book. Every address book entry includes the first name, the surname, the phone number, the Bluetooth MAC address, a score that indicates the strength of the link with this particular user, and the last time that there was an interaction or a social encounter with this user.

Both the score and the time stamp are used by the Forwarding Manager to make forwarding decisions. The score will be increased every time a message is sent to this user or received from this user. In the same way, when a Bluetooth scan is performed, if an entry from the address book is discovered, the score of this particular user will be increased as a proxy for social interaction based on proximity with this particular user. On the other hand, the score will be decreased in time if there is no further interaction with this user or if a potential attack from them is detected.

C. Forwarding Manager

The Forwarding Manager must be aware that performing networking tasks can affect the battery life of the handset. It must reduce unnecessary communication and Bluetooth scanning to increase the battery life when running Goose. Therefore, we cannot rely on forwarding algorithms that exchange excessive information about the network state over Bluetooth. In the same way, due to human mobility, forwarding algorithms based on maintaining routing tables may also be extremely inefficient in highly dynamic environments.

Goose only performs Bluetooth device scanning, by invoking the Network Manager, when the Forwarding Manager is going to send data over Bluetooth. This guarantees that the information about nearby devices is updated at the time of sending information. In fact, this approach improves the network performance by increasing the probability of establishing connections with other devices and also of being discovered by other devices. Further Bluetooth issues will be explained in Section III-D.

We have defined an unique message format based on XML that includes information about the sender, a set of destination phone numbers, information with forwarding hints, information for replica control and security, metadata about the message and, finally, the content that can be a text message of 140 bytes maximum or an audio file limited to a minute[4]. Figure 7 details the different pieces of information that a message contains.

The Forwarding Manager uses the Replica Control Table (RPC), which contains the message GUID (Sender Bluetooth MAC Address and the timestamp of the sender device when

[4]The audio file is a WAV file encoded at 11.025 KHz using 8 bit PCM at a bitrate of 18kbps. A minute of audio thus encoded requires 646 KB

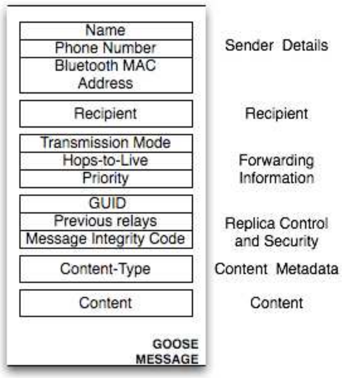

Fig. 7. Goose Message Format.

the message was created) and a list of MAC address that already received this message, in order to provide a replica control mechanism. Every node in a multi-hop transmission will include its Bluetooth MAC address in the message's Previous Relays list to notify future nodes that they already have this message. Relays also store MAC addresses of the nodes to which they have forwarded the message in the RPC table, to avoid retransmission of the messages to the same nodes.

The Forwarding Manager will take into account the message priority and available device resources to decide whether to transmit information or not. As we have previously mentioned, Goose supports unicast, multicast and broadcast modes and also friend searching; however, we need different forwarding strategies for each case.

Broadcast messages are sent among any devices in the network in a best-effort, epidemic fashion. This provide indirect evidence that DTNs can play an important role forwarding information within communities and enabling services such as SNSs on top of them. The forwarding algorithm considers message priority, and forwards while the Hop-to-Live (HTL) field has not expired. There is also a second table which stores the messages to forward in the future. If it becomes full, it will evict messages based on their timestamps, their priorities and the number of times they have been forwarded to other devices. There are several studies about the efficiency of epidemic algorithms using human mobility on DTNs such as [8], and [15]. These show empirically that the delivery ratio can be acceptable; however the algorithms imply a lot of connection between nodes, which quickly drains the battery of the device.

Multicast and unicast transmissions are usually sent to a well-known set of users stored in the address book. In other words, this means that using nodes that are in our address book in combination with unknown nearby devices, increase the chance of reaching the destination through Friends of Friends because of the well-known short diameter of social networks. Moreover, only unicast text messages with a high priority are sent straight over the GSM network if the device is under network coverage.

The forwarding algorithm chooses those well-known nearby devices that have a higher score [15] or, in case that there is no known nearby device, we forward the information randomly to a limited number of nearby devices following an controlled epidemic approach, as in the broadcast message case. In fact, this reflects how we transmit information by word-of-mouth in our real life: we give information to our friends, who deliver that information to other friends we may have in common.

Finally, in the case of *search friends* queries, despite being a particular case of broadcast messages, the Forwarding Manager will only transmit them to nearby devices that are included in the address book in a similar fashion to a multicast message. Once again, it takes inspiration from the fact that someone's details can be easily be obtained from a mutual contact. The main consequence of this is that we do not transmit redundant information. This feature can be found on the Facebook "People you might know" application and on the LinkedIn social network, which shows the number of hops between two users.

D. Network Manager

The Network Manager encapsulates both Bluetooth and SMS interfaces. It is responsible for sending and receiving data, and discovering nearby Bluetooth devices when requested by the Forwarding Manager or the Contacts Manager.

SMS is a cheap service that works over the GSM network with very low battery consumption. A single SMS can contain up to 255 segments of 140 octets; however, the operator usually limits it to 8 segments and they charge the customer per segment sent. As a consequence, the length of a message depends on the encoding used and it can therefore be used to send any kind of data format.

In the case of Bluetooth, despite being a highly deployed and power efficient low-range radio interface[5], it presents the drawback of being a half-duplex system that has an exclusive and time consuming inquiry procedure: a Bluetooth device requires from 2.5 to 10 seconds on average for discovering nearby devices and the inquiring device cannot be used for anything else for the whole duration. In fact, if a peer tries to connect to a device performing a inquiry, the connection will fail; and if two devices are performing an inquiry at the same time, they will not be discoverable until they finish the process. Because of all these reasons, Bluetooth usage in mobile scenarios can be extremely challenging and we should use the interface only when it is absolutely necessary in order to optimise the number of opportunistic connections.

E. Security and Privacy

Security and privacy in both social networks and DTNs is a crucial issue that the research community is trying to tackle. Unfortunately, not much progress has been made in this field because of its fully distributed nature, the dynamics of the network and the difficulty of relying on a central server. Because of that, our approach to security is provided by both the Contacts Manager and the Forwarding Manager, and takes advantage of the social links between users.

The Contacts Manager includes a cryptographic scheme to encipher message content on unicast Bluetooth transfers. This scheme relies on real live social bounds between people to exchange public keys between each other on demand at the same time they exchange their contact details over Bluetooth. After that, all the future content exchanged between this pair of devices will be encrypted for intermediate nodes.

Moreover,the Forwarding Manager makes routing decisions based on existing personal contacts between users too. The next hops in the communication are chosen considering the contact's rank and their scores, which are equivalent to the reputation or the trust level with that device. As a result, it is preferable that the information is forwarded through a set of trusted nodes to reduce potential threats such as spoofing, in which an intermediate node can capture a packet and spoof the address of the sender.

Finally, security against eavesdropping attacks is not provided; in which an intermediate node listens to the communication between two nodes and identifies the sender and the recipient of the messages. A potential solution would be using similar mechanisms as onion routing in the Internet, but this may require to negotiate a virtual circuit through the network, which is an assumption that cannot be made in such a dynamic environment.

IV. Technical Challenges

Developing mobile applications remains extremely challenging. Mobile phones are constrained by battery life, processing power and memory; however, the main issue that mobile developers need to face is the inter-operability caused by the large number and range of mobile platforms. In this section we will talk about how we deal with the *mobile fragmentation* and also how to deal with software distribution in scenarios as the ones already described in Section II.

A. Mobile Fragmentation and J2ME

Developing mobile applications is a radically different scenario than the PC world where Windows is clearly the dominant OS with a 90% share of the market. As we can see in the Table II, even when Symbian is the dominant platform, the mobile operating system market is much more equitable with a considerable number of mobile operating systems in the market. In fact, new operative systems such as Google's Android in 2008, are being released every year. That phenomenon is known as "mobile fragmentation" and, while competition and diversity may benefit consumers, it's becoming a serious problem for the developer community which is more interested in developing mass market applications.

Mobile web applications and *Rich Internet Applications (RIA)* using Microsoft Silverlight and FlashLite are good solutions for some applications; however, they do not provide access to some operating system services such as Bluetooth services. On the other hand, Python and Java Micro Edition

[5]Bluetooth radio range is usually 10m in standard mobile phones

Operative System	Market Share
Symbian	57.1%
RIM (Blackberry)	17.4%
Windows Mobile	12%
Linux	7.3%
MAC OS X	2.8%
Others	3.4%

TABLE II
SMARTPHONES OPERATIVE SYSTEM MARKET SHARE. SOURCE: GARTNER RESEARCH

(J2ME) seem to be the best solutions to deploy cross-platform applications.

However, while Python is supported by smartphone platforms such as Windows Mobile and Nokia S60 devices, the Mobile Information Device Profile (MIDP) and the Connected Limited Device Configuration (CLDC) provide a standard Java runtime environment and a core application functionality required by mobile applications that is supported by today's most popular mobile devices [6].

However, there exist some inconsistency issues due to handset features (e.g. how they treat audio/video and the display size) which force the developer to test the application on every platform before release; thus proving that Java's motto "write once, run everywhere", that Sun tried to deliver, fails in the mobile ecosystem. Nevertheless, the main features required by Goose (e.g. Bluetooth API extension known as JSR-82) are supported by a broad range of mobile handsets[7].

B. Battery Charging Issues

In the same way that network coverage may be limited in rural areas, the access to the electrical grid could also be challenging. Fortunately, there are some commercial emergency chargers that exploit solar and human power in the way of solar panels and hand cranked mechanisms respectively.

One Laptop Per Child Project [2] proposes several alternatives to provide energy to charge laptop's battery that can be easily ported to mobile phones. They analysed and evaluated several power alternatives that are designed taking into account the available local resources such as:

- Animal power by using a system of belts and pulleys moved by animals to get a high rpm at the alternator.
- Human power alternatives. Solutions include kinetic energy recovery reusing human energy that has been already wasted in cases such as walking [12], hand cranked or pedalling systems that exploit the continuous stamina from legs (e.g. dynamo on bicycles).
- Chemical power alternatives using wood gas generator and fuel cells that generate electricity from hydrogen, methane, methanol and even from organic matter by using bio-reactors.
- Solar and heat power using solar panels, stirling engines or thermoelectric generators.
- Wind and hydroelectric power.

[6]Almost three quarters of current handsets support MIDP 2.0 API

[7]It is necessary to emphasise that not all the Java enabled phones that have Bluetooth hardware built in, support the JSR-82 Bluetooth API

Those alternative energy sources cannot be used in the same scenarios and some of them require an expensive infrastructure. In some cases it may be necessary to provide solutions oriented for a single user such as small solar panels that can be attached to the device while in other cases it may be interesting to build big permanent solutions such as animal power systems that can provide energy for a whole community. Nevertheless, the impact of Bluetooth on battery life is not severe even under extreme traffic, as we can see in the experiment results that are explained in Section V showing that Goose can run for several hours in mobile devices without draining the battery.

C. Software Distribution

Software distribution is critical for services such as Goose, which will be used in areas where the Internet penetration may be low. However, we perceive two complementary ways of distributing Goose.

The first approach relies on the mobile operators. Some may perceive this as a potential threat to the business model of the operator since they are providing their most profitable service for free to their customers. Nevertheless, mobile operators could attract new customers by shipping Goose software onto their mobile phones and offering their customers to send free SMS and voice messages over Bluetooth using social links. Moreover, Goose will allow the operators to provide data services in remote or isolated areas where deploying a GSM network would be difficult or expensive, thus reducing the traffic load in their networks.

The second distribution channel relies completely on the users to distribute by social cascade. If some users have Internet access, then they can download the application from the Internet to their mobile phones and then share it with other people over Bluetooth. In this second case, the software distribution may follow a similar pattern as gossips and rumours distributions in society.

V. PRELIMINARY AND EVALUATIONS

As we have previously mentioned, the type of handsets we initially want to target are MIDP 2.0 capable devices. Table III, summarises the list of APIs required to build Goose. In fact, those requirements are met by a lot of mobile phone models that are currently being sold in developing countries.

Sony-Ericsson K320i with 15MB of user memory (with an additional Memory Stick slot that is usually shipped with the handset) and JSR-82 API is an example of such a device. K320i costs around Rs.3000 in India (55 USD aprox.) but they are generally subsidised by the operators for a lower price to the customer. In fact, Goose does not need any additional hardware such as GPS, WiFi and photo camera.

We also analised the possible Goose's footprint in the device. The length of the Bluetooth message may vary a lot depending on the number of carriers that have already forwarded the information and also the number of devices that make up the recipient. In the case of a new unicast message, the size of the message header may be between 140 and 180 bytes. Then, if the body is a text message of 140 bytes, the

API Name	JSR	Description
MIDP	JSR-37, JSR-118	Allows the developers to write downloadable applications and services for network-connectable mobile devices. When combined with the CLDC, it is the JRE for today's most popular compact mobile devices. It provides background processing and access to battery level information.
CLDC	JSR-30, JSR-139	Defines the base set of application programming interfaces and a virtual machine for resource-constrained devices. It also provides two variants of the Thread class, both subsets of the J2SE version 1.3
WMA	JSR-120, JSR-205	Optional package that provides platform-independent access to wireless resources like SMS.
Bluetooth	JSR-82	Standard Bluetooth library.
MMAPI	JSR-135	Optional package that provides a Media API. Necessary for audio encoding[8].

TABLE III
JAVA APIS REQUIRED BY GOOSE

total size of a new created message is around 300 KB; whereas in the case of a minute audio message, the size of the message can be around 800 KB. It shows that Goose must provide a good mechanism to decide which messages must be stored in the store-and-forward approach to make the most of the limited persistent storage.

While sending data over SMS implies monetary cost; store-and-forward mechanisms may have an impact both on the available memory and the battery. In fact, battery consumption is a key issue in Goose as in many mobile applications. Figure 8 shows the impact of scanning, sending, and receiving data. We have run an experiment with two HTC Touch[9] devices with Li-Ion batteries separated by 2 meters. One of the devices is discovering and sending messages of 1MB every 35 seconds approximately (there's a small delay introduced by the Bluetooth scanning task) to a second handset that only receives the messages. In total, 1527MB were exchanged in more that 14 hours.

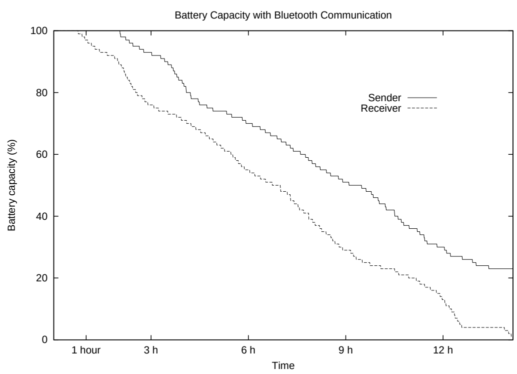

Fig. 8. Bluetooth activity impact on a Li-Ion battery

As we can see, receiving data has a lower impact on the battery life compared with sending intense traffic and discovering bluetooth devices. The impact of Bluetooth is not very severe since the sender was able to continue under intense traffic for more than 14 hours. Goose handsets will seldom be under such traffic intensity even in dense social situations in which messages will be forwarded between devices in a store-and-forward manner to increase the message delivery ratio. Usually, an average SMS user in Europe and North America sends 350 SMSs per month and the average tweet frequency for a twitter user is 2.5 messages per day while only 5% of the users tweet more than 6 times in a single day [3].

[9]The devices are almost two years old

We also ran a second experiment to study the nearby devices availability in order to understand the influence of other Bluetooth devices' activity. A total of 6 Bluetooth devices were placed in range within each other and all of them they were discovering each other and sending data every minute with a small random variation to avoid device synchronisation. As we can see in Figure 9, the device under study may not have all the devices available at the same time to transmit information but it will still continue having at least 2 out of 5 devices available in the worst case to establish an opportunistic connection with them.

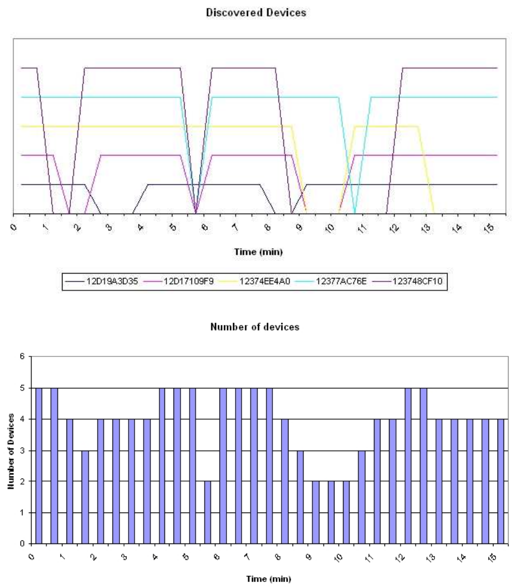

Fig. 9. Bluetooth Neighbours availability

Finally, in terms of usability and users' experience, we conducted a survey with a total of 63 participants from members of the Computer Lab of the University of Cambridge and Microsoft Research Cambridge in 2007. The purpose of the survey was to understand how people feel about using DTN technologies.

The survey results showed that 60% of the participants send at least 34% of their email and messages with recipients in the city they live in and usually to people they meet almost every day. The participants said that they can tolerate certain delay in applications such as file sharing, advertising or email and messaging among many others.

In addition, 45% of the people surveyed are willing to share their resources (e.g. battery, memory and CPU) with other users to forward their information if they can get benefits from this collaboration while 33% will do it even in an altruistic manner. However, in order to carry data for forwarding to others, the decision may be also affected by who is the sender and what is the current state of the handset resources. Most people would like to carry data for a friend if they have plenty of resources while only a 33% of the people would also do it for strangers.This provide indirect evidence that DTNs can play an important role forwarding information within communities and enabling services such as SNSs on top of them.

VI. Related Work

In recent years, many mobile SNSs uses Bluetooth to discover nearby contacts. Some examples are Aka-Aki [4], BluetoothFlirt [6] and Imity [16]. However, those social networks cannot be extended to developing regions and isolated areas since the user to user connections are through a web service, requiring a 3G or GPRS connection.

There are many projects such as OLPC that aim to provide technology and services for developing areas. Goose can easily run on top of these ad-hoc networks both over WiFi or Bluetooth. While there is not any work focused on providing social mobile networks in developing countries, there are several projects about providing services and connectivity in developing areas by using DTNs.

TIER Research group at the University of California Berkeley, are investigating the design and deployment of new technologies and services for emerging regions [11]. In addition to this, other services such as Kaash [5] gather rural health care data relying on physical device transport to overcome the lack of connectivity as DarkNet [21] does by copying data to a USB drive and physically carry the drive over vehicles.

VII. Conclusions and Future Work

In this paper, we have identified the application scenario of SNS in developing regions and proposed a system architecture, Goose, to achieve this. We have also identified the technical challenges for such a system and provide some preliminary evaluations.

At launch, most SNS provide a reduced set of the features that they have the potential to provide, adding more as their userbase expands.We would like to follow this approach in Goose since, as we mentioned in Section II-A , the potential uses of Goose could be vast also when combined with more traditional mobile web SNS. In fact, by extending the messaging service, Goose can be easily adapted to provide richer content on the social network such as images or videos in a fashion similar to MMS services. Additionally, Goose can also provide a more extended support to the co-existent network infrastructure and other local area wireless interfaces such as WiFi.

Mobile context offers many opportunities to enable interaction between users and we would like to explore and support them on Goose. Community detection can open a new set of potential services on top of Goose at the same time it can offer new forwarding hints. In the same way, user awareness (e.g. location and nearby contacts) and user profile information exchange can support new features such as contacts browsing. However, due to Goose's distributed architecture, it will be challenging to deal with privacy and security policies. Actually, there is still a lot of work to do in terms of privacy and security and it will be interesting closely follow the steps that the research community takes in privacy and security for DTN and social network services.

VIII. Acknowledgements

This work is partly funded by the EC IST SOCIALNETS - Grant agreement number 217141.

The authors would like to thank Derek Murray (University of Cambridge) and Andy Ridge (University of Bath) for their very insightful comments.

References

[1] Financial analysis of telecom industry of china and india (new delhi, 27 june, 2006).
[2] Olpc, battery and power wiki. `http://wiki.laptop.org/go/Battery_and_power`.
[3] Twitter facts blogspot. `http://twitterfacts.blogspot.com/2008/01/whats-cost-of-twitter.html`.
[4] Aka-aki. `http://www.aka-aki.com`.
[5] V. Anantraman, T. Mikkelsen, V. Kumar, and L. Ohno. Handheld computers for rural healthcare, experiences in a large scale implementation, http://kaash.sourceforge.net/doc/dyd02.pdf.
[6] Bluetoothflirt. `http://www.btflirt.com`.
[7] P. Buys, S. Dasgupta, T. Thomas, and D. Wheeler. Determinants of a digital divide in sub-saharan africa: A spatial econometric analysis of cell phone coverage, Feb. 2008.
[8] A. Chaintreau, P. Hui, et al. Impact of human mobility on the design of opportunistic forwarding algorithms. In *Proc. INFOCOM*, April 2006.
[9] S. Counts. Group-based mobile messaging in support of the social side of leisure. In *Computer Supported Cooperative Work*, pages 75–97, 2007.
[10] S. Counts and K. E. Fisher. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet. In *Proceedings of the 41st Hawaii International Conference on System Sciences*, Feb. 2008.
[11] M. Demmer, B. Du, and E. Brewer. Tierstore: A distributed storage system for challenged networks in developing regions. In *Proc. USENIX Conference on File and Storage Technlogies (FAST)*, Feb. 2008.
[12] J. Donelan, Q. Li, V. Naing, J. Hoffer, D. Weber, and A. Kuo. Biomechanical energy harvesting: generating electricity during walking with minimal user effort. *Science*, 319:807–810, 2008.
[13] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proc. SIGCOMM*, 2003.
[14] R. Grinter and M. Eldridge. Wan2tlk?: Everyday text messaging. In *Proceedings of Mobile HCI 2003*, pages 441–448, 2003.
[15] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *MobiHoc '03: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking & computing*, May 2008.
[16] Imity. `http://www.imity.com`.
[17] D. Joshi and V. Avasthi. Mobile internet ux for developing countries. In *Proceedings of Mobile HCI 2007*, 2007.
[18] B. E. Kolko, E. J. Johnson, and E. J. Rose. Mobile social software for the developing world. In *Proceedings of HCI International 2007*, 2007.
[19] B. E. Kolko, E. J. Rose, and E. J. Johnson. Communication as information-seeking: the case for mobile social software for developing regions. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 863–872, New York, NY, USA, 2007. ACM.
[20] G. Mardsen, A. Maunder, and M. Parker. People are people, but technology is not technology. *Royal Society Philosophical Transactions*, October 2008.
[21] A. Pentland, R. Fletcher, and A. Hasson. Darknet: Rethinking connectivity in developing nations. *IEEE Computer*, 37(1):78–83, Jan. 2004.
[22] L. R and B. Yttri. Nobody sits at home and waits for the telephone to ring: Micro and hyper-coordination through the use of the mobile telefone. In *In J. Katz and M. Aakhus Perpetual Contact*, pages 139–169, 2002.