MobOCloud: Extending Cloud Computing with Mobile Opportunistic Networks

Shantanu Pal
School of Computer Science
University of St Andrews
St Andrews, Fife KY16 9SX, UK
sp66@st-andrews.ac.uk

Tristan Henderson
School of Computer Science
University of St Andrews
St Andrews, Fife KY16 9SX, UK
tnhh@st-andrews.ac.uk

ABSTRACT

Cloud-computing applications are fast replacing traditional mobile and desktop applications such as e-mail, document editing, or photo storage. Such applications may require ubiquitous network access that is not always available, e.g., in remote areas without infrastructure, but also in areas with infrastructure where the costs of access are too high for users, such as tourists who do not wish to pay high roaming charges. This paper explores the use of opportunistic networking in a "Device to Device to Cloud" architecture that gives tourists access to cloud computing resources via local users' network connections. We use real-world trace-driven simulations and a new tourist mobility model to evaluate two options: storing data at well-situated hubs versus exploiting the mobility of local users, and find that the latter improves message delivery performance.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Store and forward networks; C.2.2 [**Network Protocols**]: Routing protocols

General Terms

Performance

Keywords

Opportunistic Networks, Cloud Computing, Mobile Applications

1. INTRODUCTION

There is now one mobile phone for every two people in the world [5]. As such phones become more powerful and sophisticated, users have started to use their phones as personal information processing tools rather than simply for making phone calls. At the same time, the rise of cloud computing means that resource constraints on a smartphone can be alleviated by offloading computation to the cloud. Mobile cloud computing [9] has thus arisen as a means for improving the capabilities of mobile devices. But one large constraint for the mobile cloud is the requirement for Internet connectivity. This is a concern in challenged environments where there is limited network availability, but also in areas where the cost of accessing the Internet is simply too high.

In such cases the use of an opportunistic network [19] may help to improve the availability and accessibility of information using co-located user's to relay information instead of relying on fixed infrastructure for communication. Nodes in an opportunistic network communicate with each other with the help of other nodes. Human interaction is one of the major parts of such communication as the opportunity of forwarding any message depends on their nature and behaviour of the said interaction [7]. By using opportunistic networks it may be possible to overcome the limitations of high Internet access costs or unavailability of network infrastructure, e.g., for tourists who may wish to gain access to remote cloud applications with the help of local users' mobile phones.

This paper aims to address the following research questions:

1. Is it possible to use an opportunistic network to provide cloud services to tourists that lack access to network infrastructure?
2. Does the mobility and social interactions of local users help to send tourists' information efficiently?
3. Do local users' mobile networks succeed at integrating tourists' and cloud networks successfully to build an integrated opportunistic mobile cloud platform?

The goal of this paper is two-fold. First we investigate how to build an opportunistic mobile cloud infrastructure for tourists with the help of the local users' mobile communications that can enable sending or retrieving of data to or from the cloud. Second we use opportunistic mobile cloud scenario to investigate efficient routing algorithms for tourists to access the remote cloud applications.

The contributions of this paper are as follows:

1. We introduce a "Device to Device to Cloud" architecture, which enables tourists to use cloud applications without any Internet connection;
2. We devise a new tourist-based mobility model according to user behaviour, nature of travel and activities in different places;
3. We evaluate two options for sending tourists' data into the cloud: storing data at well-situated hubs and by exploiting the mobility of local users. Our results show that leverage local users' mobility improves message delivery performance.

The rest of the paper is organised as follows. Section 2 describes related work. Section 3 outlines our proposed architecture. We describe our simulations for evaluating different forwarding techniques in Section 4, followed by the results. We discuss the implications of these results in Section 6, and conclude.

2. BACKGROUND AND RELATED WORK

Our goal is to understand how we can provide mobile cloud computing in the absence of dedicated network infrastructure. Much research has investigated how to best integrate mobile applications

CHANTS'13, September 30, 2013, Miami, Florida, USA.

ACM 978-1-4503-2363-5/13/09 $15.00.
http://dx.doi.org/10.1145/2505494.2505503.

and cloud infrastructure. These various mobile cloud architectures can be characterised as follows:

Mobile-Cloud Computing Device to Cloud (D2C): In this class of architecture, users' mobile applications run in the cloud, where mobile devices are becoming part of a larger cloud environment. Li et al. propose a Mobile Cloud Framework middleware for deploying mobile computation into the cloud [15]. This model delivers a higher rate of efficiency in order to use mobile resources, where computation takes place inside in the cloud, through the local users' mobile devices. The CloneCloud architecture is designed to execute mobile applications over the cloud services by optimising execution time and energy of mobile devices [6]. But these D2C architectures do not work well in the absence of wireless wide area networks (WWAN).

Mobile-Cloud Computing Cloud to Device (C2D): In a C2D architecture, users connect their mobile devices to the cloud servers over the the Internet, where cloud applications are running inside the users' mobile devices. Kim et al. present a C2D approach that provides connectivity between the users' 'Mobile Personal Grid' (MPG) and cloud services [14]. In the MPG several mobile devices are connected with each other to allow access to cloud applications. Another C2D architecture is MobiCloud, which builds a secure cloud framework for mobile computing and communications with the help of mobile devices [10]. This architecture addresses trust management, secure routing as well as risk management issues within the network combining mobile devices and the cloud computing applications. Unlike our work, MobiCloud requires a permanent Internet connection to access the cloud applications needed to build the framework. The mCloud [17] is a C2D architecture where mobile devices themselves become the core computing nodes. This architecture improves the hardware capabilities on mobile devices as well as helping to reduce the bandwidth usage. Verbelen et al.'s Cloudlet system allows users to access remote cloud services by using their mobile phones as a thin client connecting with the cloud server through the Web [21]. But again these C2D architectures are not well suited to the absence of WWANs.

Mobile-Cloud Computing Device to Device (D2D): In a D2D architecture, mobile devices share locally-available network connections to form an environment for sharing information between local users. Pedersen et al. discuss a D2D architecture, where mobile devices create their own *cloud environment* with the help of the various locally available mobile devices [18]. In this approach, mobile devices are able to convey information directly to their nearby mobile devices without any help of the overlay network. Dinh et al. present an architecture composed of 'mobile network services' and the 'cloud controllers', where mobile devices are connected with a network service via various access points, satellite and Base Transceiver Station (BTS) to transmit messages to the nearby mobile devices [8]. But these D2D communications do not enable users to access remote cloud applications into their mobile devices.

All of these systems study a use case where traditional cloud-computing applications are used on mobile devices, or mobile applications are used in a cloud-computing platform, in the presence of available and reliable network infrastructure. Our focus is on providing cloud-computing access to tourists in areas where network infrastructure might be unavailable or too expensive. Canepa et al. present a 'virtual cloud computing provider' for tourists, allowing them to search for information within local mobile devices, thereby avoiding high Internet access charges [5]. Unlike our work, they do not focus on how tourists can access remote cloud-computing services. Hung et al. present a 'smart tourism' system that uses cloud-computing applications [12], but unlike our system, it requires an expensive Internet connection.

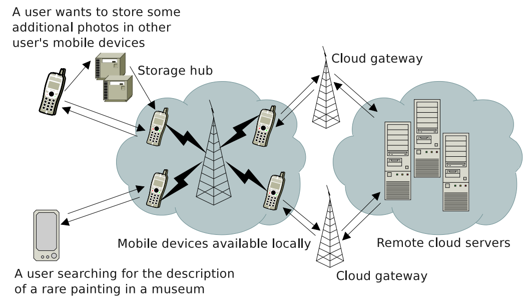

Figure 1: The MobOCloud architecture, is where tourists search for local users who may help to store photographs or perhaps help to understand the meaning of a rare painting. Local users may help tourists by giving storage or information by providing local mobile network infrastructures or with the help of the local users' mobile networks tourists can access cloud applications themselves.

3. THE MOBOCLOUD ARCHITECTURE

To provide tourists with access to the cloud in the absence of infrastructure, we combine the D2C, C2D and D2D architectures to form a "Device to Device to Cloud" (D2D2C) architecture that leverages local users' mobile network connections. Figure 1 presents two use cases for the MobOCloud architecture.

Suppose Alice is travelling in a rural part of a country where there is no infrastructure to access the Internet. She has taken thousands of photographs and now her phone's memory is full. In this situation she wants to take more photos and is looking for more space to store the photos or show her photos to her friends, by uploading her photos to her usual cloud-computing photo provider. Likewise, suppose Bob is visiting a museum in China and looking at a rare painting of ancient China that is described in an ancient form of native writing. Bob is unable to understand the meaning of the language because he cannot read that particular form of language. He is now looking for someone who might be able to interpret the language or who may be interested in sharing his or her view for the same text. He knows that as a member of CloudLingual[1], a community of professional translators, he will be able to have this language translated into his own via his mobile device, but this requires Internet connectivity to access the remote translators.

The MobOCloud architecture consists of the following components: *tourists' mobile devices*, *local users' mobile networks*, *information storage hubs*, *cloud gateways* and *remote cloud servers*. Tourists' mobile devices cannot directly access cloud services but can communicate with local users; while local users are able to connect their mobile phones with the Internet to reach cloud services. Since tourists may be travelling without easy access to mains power, they may be concerned about their mobile devices' limited battery and storage resources, and wish to store data before the power in their devices run out. Information storage hubs are static

[1] http://www.cloudlingual.com/

devices, situated in tourist spots, where tourists can store their data without an Internet connection. Tourist spots are places and attractions where tourists often visit. When local people visit the hubs, they can then collect data from the hubs and in turn are able to upload the tourists' messages to the remote cloud servers using cloud gateways. Cloud gateways are used to send information to the remote cloud servers. Cloud gateways are situated in places like a library, a university computer science building, etc., where local users are able to access Internet connections for their mobile devices to access cloud applications to upload tourists' messages into the cloud.

In both scenarios, Alice and Bob have difficulties gaining access to the cloud application. MobOCloud provides a solution to allow Alice and Bob the use of the cloud applications with the help of local users' mobile network infrastructure. Alice could then transfer some of her photographs into an information storage hub or a local users' mobile device, which then will allow the local user to send her photographs to their destination using cloud services. Bob may also be able to use the online cloud translator via his mobile device for translating the meaning of the language.

4. EVALUATION

We now evaluate the feasibility of MobOCloud architecture under the different routing scenarios. To this end, we use the Opportunistic Network Environment (ONE) simulator to carry out trace-driven simulations [13].

4.1 Simulation Setup

We explore three different modes of message transferring behaviour for tourists. Note that we assume that all the tourists and local people in the network are trusted, although a reputation framework could easily be incorporated, e.g., [1].

1. Everyone Forwarding (EF): Everyone in the network is willing to store and forward messages for each other. Suppose a tourist sends message to the network, then every other tourist and local people are able to carry and forward this message until it reaches its final destination using the cloud infrastructure.
2. Local Forwarding (LF): In this mode of message transfer, tourists generate messages but unlike in EF mode, tourists do not store and forward messages to their destinations. Only local users store and forward the tourists' messages for them while both are moving within the network.
3. Pickup and Forwarding (PF): As outlined in Section 3, tourists might be concerned about their devices' limited battery and storage. In PF mode, tourists use the information storage hubs located in tourist spots to store their messages. These stored messages will then depend upon the local users to carry and forward them until the messages reach their intended destinations using the cloud infrastructure. Unlike LF mode, in PF mode tourists nodes do not need to interact with local nodes to transfer messages. Instead, interaction takes place via the hubs. An appropriate incentive framework [2] can encourage local users to act accordingly.

We use the following commonly-used metrics [11] to evaluate the performance of the MobOCloud architecture for different modes of message transferring behaviour:

1. Delivery Ratio: the proportion of the delivered messages to the total number of messages created in the simulation.
2. Delivery Delay: the total amount of time to send messages from source to destination.
3. Delivery Cost: the total number of medium accesses, normalized by the total number of messages created.

4.2 Mobility Model and Traces

To illustrate the performance of the MobOCloud infrastructure we simulate our own university town, St Andrews. We set up two sets of nodes representing local users and tourists, and generate appropriate points of interests (POIs) for tourists using OpenJUMP.[2]

1. Local Node (LN): To simulate the local users, we use the St Andrews "SASSY" dataset of 27 users' real-world traces over a period of 79 days [3]. Users carried mobile IEEE 802.15.4 sensor motes which could detect each other within an approximate radius of 10 metres.
2. Tourist Node (TN): As we lack traces of tourist activity, we develop a new "TouristActivityBased" movement model to simulate tourists' behaviour and nature of visits in different places. We assume that tourists visit a place, perform an activity, e.g. taking photos and then move onto another place. Using publicly-available tourist data[3] we choose different probabilities for 10 different POIs for tourists. We do the same for all tourists nodes in our simulation. POIs include historical sites (e.g., St Andrews Castle, St Andrews Cathedral, etc), entertainment places (e.g., golf club, cinema, etc) or places that they might visit to collect specific information about the town (e.g., the tourist information centre). Tourists generate one message every 9 to 10 minutes while visiting tourist spots.
3. Cloud Gateway (CG): We situate 6 cloud gateways throughout the town (e.g., Computer Science building, art gallery, student dorms, etc) where local users have Internet connectivity and can forward collected messages to the remote cloud over the Internet.
4. Information Storage Hubs (IS): We situate 7 different information storage hubs in tourist spots throughout the town (e.g., St Andrews Castle, St Andrews Cathedral) where tourists can store their messages without having any Internet connection.

4.3 Routing Protocols

To evaluate the different message transfer modes we consider the Epidemic [20], Prophet [16] and MaxProp [4] routing protocols for our simulation. We use Prophet and MaxProp to examine how real life traces of local users' interaction make significant impacts in our proposed architecture, as these routing protocols are based on encounter history. We use Epidemic as a baseline. We denote these routing protocols as ED, PR and MP respectively.

4.4 Simulation Parameters

Table 1 summarises the simulation parameters. The network consists of 100 nodes (27 LN, 60 TN, 6 CG and 7 IS nodes).

5. RESULTS

We now evaluate the routing protocols to determine the performance impact of the different message forwarding behaviours in MobOCloud architecture.

1. Delivery Ratio: Figure 2 shows how MP routing improves the message delivery performance (median delivery ratio 69.04%) compared with PR and ED routers while everyone in the network is forwarding messages. Figure 3 shows the comparison of delivery ratio of ED, PR and MP routers when tourists' messages are forwarded by local users, using their mobility patterns and by storing tourists' messages into information storage hubs. We find that the delivery ratio of MP, PR and ED routers increases while using local users' mobility patterns compared with

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

[3]http://www.visitscotland.org/pdf/visitorattraction-monitor2009.pdf

Parameter	*Value*
World Size	4500m X 4500m
Simulation Time	7 days
Movement Model	TouristActivityBased Movement Model (See Section 4.2)
Routing Protocol	Epidemic; Prophet; MaxProp
Node Buffer Size	200 MB
Transmission Speed	250 KBps
Transmission Range	10 m
Transmission Medium	Bluetooth Interface
Message TTL	1 day
Node Movement Speed	Min=0.5 km/h Max=1.5 km/h
Generated Message Size	500 KB to 1 MB

Table 1: Simulation parameters

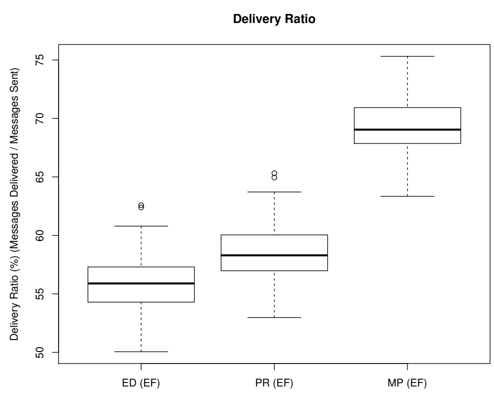

Figure 2: In EF, ED routing generates large amounts of messages in the network, although due to the nodes' buffer size and TTL most messages were dropped before they reached their destination. MP and PR routing sends messages by keeping the previous history of delivered messages. This results in a higher delivery ratio for MP(EF) and PR(EF) than ED(EF).

the storing of data into hubs. We see that in the first case the message delivery ratio of MP router is more than doubled (median delivery ratio 37.84%) than in the second case (median delivery ratio 15.83%). We believe the reason for this is that the probability of meeting a tourist and local node is higher when both of them are moving in the network. On the contrary, the delivery ratio of ED router decreases when everyone in the network is forwarding messages (Figure 2), because a large amount of messages stored by all nodes is dropped, as buffer constraint is a pivotal factor of this routing performance. We find that exploring local users' mobility patterns, D2D2C infrastructure gives a better message delivery performance (μ = 31.69, σ = 5.05) than storing data into hubs (μ = 14.29, σ = 2.23). But as expected, in the baseline EF, the message delivery performance is higher (μ = 61.26, σ = 6.22); a two-way analysis of variance (ANOVA) shows significant differences in the overall rate of the message delivery ratio [F (2,891) = 0.00, $p < 0.05$].

2. Delivery Cost: Figure 4 shows that the delivery cost of ED routing is higher than the delivery cost of MP and PR routers. We believe the reason is that by ED routing, a node will generate more copies of a message so that its buffer saves many copies of different messages, and buffers thus fill up. We find that message delivery cost, by using LF (μ = 57.66, σ = 10.43) is lower than message delivery cost of PF (μ = 71.21, σ = 13.12) (Figure 5). As expected message delivery cost is higher when choosing baseline EF (μ = 104.99, σ = 14.28), a two-way ANOVA shows significant differences in the overall rate of message delivery cost [F (2,891) = 0.00, $p < 0.05$].
3. Delivery Delay: Figure 6 shows that the delay in ED routing is higher than MP and PR routers when everyone forwards messages in the network. Figure 7 shows that when tourists store their messages into hubs, the delay of ED, MP and PR routers are largely higher than that of forwarding data using local users mobility patterns. This is reasonable because the copies of messages reaching their destinations are quicker in the later case. The reason for the increase of the delay is that by storing data into hubs, each copy of a message has to wait for a longer period of time to reach its destination, as this depends on the encounters between hubs and local nodes. We find that PF has a

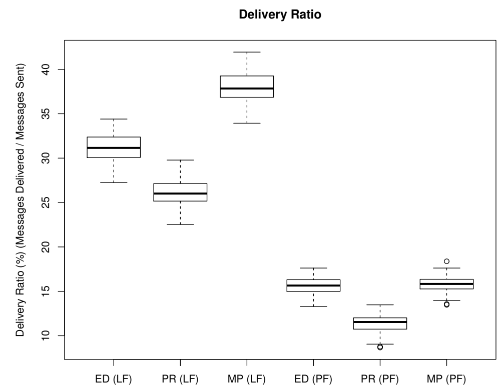

Figure 3: In LF, only tourist nodes generate messages and local nodes forward them when ever nodes encounter each other. In PF, tourist nodes deposit messages to the static storage hubs then messages were forward depending on the interactions between storage hubs and local users. This result is a higher delivery ratio for LF than other PF modes.

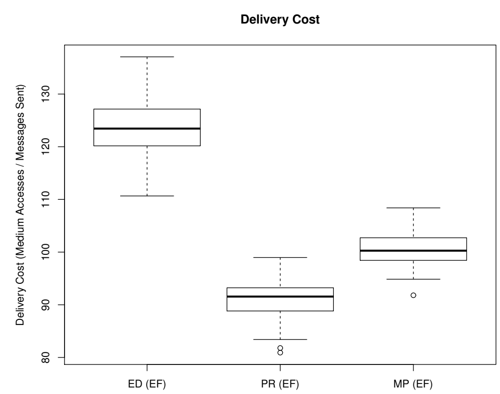

Figure 4: MP and PR routers delivere higher amounts of messages to their destinations by using encounter histories. This results in lower delivery costs for MP and PR routers over the ED router in EF mode.

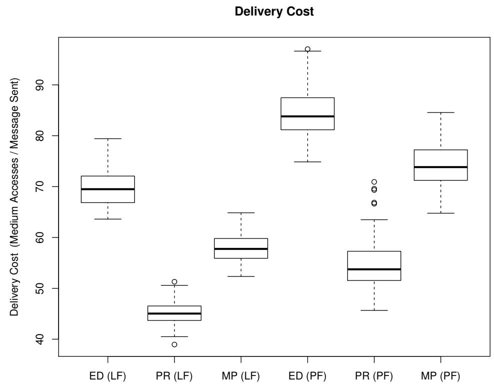

Figure 5: Local users' mobility patterns in LF mode helped a higher number of tourists' messages to reach their destinations than in PF mode. This resulted in a lower delivery cost for LF mode than PF mode.

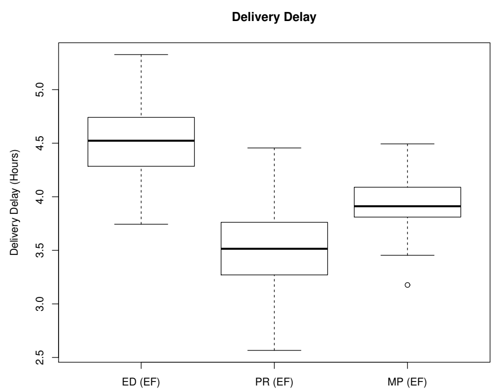

Figure 6: In EF, MP and PR routers send messages by keeping an encounter history of delivered messages, whereas the ED router drops a larger amount of messages depending on the nodes' buffer size and TTL. This results in higher delivery delays for ED(EF) than PR(EF) and MP(EF).

higher message delivery delay (μ = 3.47, σ = 0.10) than LF (μ = 3.06, σ = 0.10). As expected message delivery delay is higher when choosing the baseline EF (μ = 3.99, σ = 0.52), a two-way ANOVA shows significant differences in the overall rate of message delivery delay [F (2,891) = 0.00, $p < 0.05$].

6. DISCUSSION

We summarise our results as follows:

- As a baseline, EF routing provides the best message delivery performance in the D2D2C architecture, but at the same time it increases the message delivery cost and delays.
- By storing data into hubs, PF mode of message transferring behaviour may lead to dramatically lower routing performance for our proposed D2D2C architecture.
- Exploring local users' mobility patterns, LF mode provides the optimal message delivery performance when considering message delivery cost and delays.

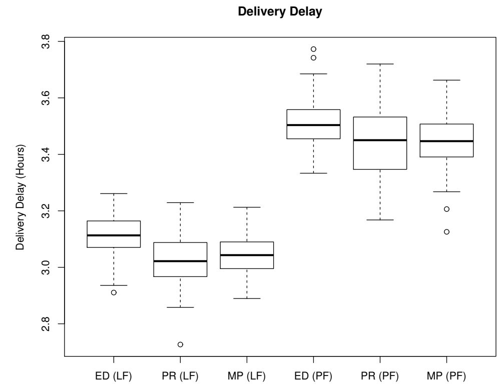

Figure 7: In LF, tourists' messages reach their destinations using local users' mobility patterns. In PF, tourists nodes deposit messages into the storage hubs and those messages are forwarded depending on the interactions between the storage hubs and the local users. This resulted in higher delivery delays for PF over LF.

It is probable that tourists nodes may be unwilling to share their phone memories with other nodes due to the lack of battery power or memory size, e.g., in PF mode they are storing their messages into well situated information storage hubs and in LF mode they depend upon local users' mobility patterns to deliver their messages. Our results indicate that the best possible solution is for a tourist to share their messages with any available local node while they are both moving within the network. Designing a protocol that can provide appropriate incentives is therefore needed [2].

Returning to our MobOCloud scenarios, message delivery performance will be higher for Alice in an EF case. Bob however, is interested in getting the information as soon as possible. As we see that the message delivery delay is minimum in LF mode that would be the best choice for him. Further refinement is needed to find a protocol that can meet the requirements of both applications, or perhaps QoS-like mechanisms for declaring appropriate forwarding strategies are required.

7. CONCLUSIONS AND FUTURE WORK

We have introduced MobOCloud, a new opportunistic mobile cloud architecture. We address two questions. First, whether it is possible to use remote cloud applications in a situation where there is no Internet connection and by avoiding the high costs of such infrastructure when available. Second, by using a "Device to Device to Cloud" architecture, how tourists can efficiently send messages to a cloud destination with the help of the local users' mobile communication network. We evaluated two potential options for transferring tourists' messages to the cloud services, by storing data at well-situated hubs and exploiting the mobility of local users.

We find that the potential for using local users' mobility techniques improves the message delivery performance rather storing data into hubs. We observed however, that minimizing delivery costs and delay may not necessarily be an indicator that a routing protocol will perform better than a protocol with a higher delivery cost and delay.

We evaluated protocol performance with three different proposed modes of message transferring techniques and connectivity patterns in three different routing protocols, Epidemic, Prophet and Max-

Prop. MaxProp routing protocol consistently perform well across the simulations. Even then, it appears that a system designer's choice of routing protocol may depend on the nature of the system's users.

Our future work will investigate many areas, e.g.,

- While sharing information, tourists and local users may not wish to share some of their personal information with each other. How, by using MobOCloud will it possible to mitigate users' privacy concerns while maintaining the same routing performance? In general, opportunistic networks involve reliance on intermediate and unknown nodes, who may attempt to surveil or modify data being sent through the network. What are the privacy requirements for tourist information?
- Limited battery power for small devices (e.g., cellphones, PDAs, etc.) can curtail communications, which in turn affect the data integrity. How will this affect the message delivery performance in MobOCloud infrastructure, when exploring the local users' mobility patterns?

Our current experiments have only looked at one-way communication, where tourists are offloading messages to their destinations with the help of local users. But in the future we would like to investigate where tourists can also download information using the cloud infrastructure. We also plan for more comprehensive experiments with further data, including better traces of tourists, which will in turn help to understand and will make improvements in performance of tourists' message forwarding techniques with better performance.

Acknowledgements

This research work is supported through a PhD studentship awarded by the Scottish Informatics and Computer Science Alliance (SICSA) and the University of St Andrews.

8. REFERENCES

[1] G. Bigwood and T. Henderson. IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. In *Proc. SocialCom*, pages 65–72, Boston, MA, USA, Oct. 2011. doi:10.1109/passat/socialcom.2011.60.

[2] G. Bigwood and T. Henderson. Incentive-Aware Opportunistic Network Routing. In I. Woungang, S. K. Dhurandher, A. Anpalagan, and A. V. Vasilakos, editors, *Routing in Opportunistic Networks*, chapter 14, pages 383–401. Springer New York, 2013. doi:10.1007/978-1-4614-3514-3_14.

[3] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. CRAWDAD data set st_andrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.org/st_andrews/sassy, May 2013.

[4] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006. doi:10.1109/infocom.2006.228.

[5] G. H. Canepa and D. Lee. A virtual cloud computing provider for mobile devices. In *Proc. ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, San Francisco, CA, USA, 2010. doi:10.1145/1810931.1810937.

[6] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. CloneCloud: elastic execution between mobile device and cloud. In *Proc. EuroSys*, pages 301–314, Austria, 2011. doi:10.1145/1966445.1966473.

[7] M. Conti and M. Kumar. Opportunities in Opportunistic Computing. *Computer*, 43(1):42–50, Jan. 2010. doi:10.1109/mc.2010.19.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.*, Oct. 2011. doi:10.1002/wcm.1203.

[9] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the cloud: enabling mobile phones as interfaces to cloud applications. In *Proc. Middleware*, pages 83–102, Urbana, IL, USA, 2009. doi:10.1007/978-3-642-10445-9_5.

[10] D. Huang, X. Zhang, M. Kang, and J. Luo. MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication. In *5th IEEE Int'l Symposium on Service Oriented System Engineering*, pages 27–34, Nanjing, China, June 2010. doi:10.1109/sose.2010.20.

[11] P. Hui and J. Crowcroft. How Small Labels Create Big Improvements. In *Proc. PerComW*, pages 65–70, White Plains, NY, USA, Mar. 2007. doi:10.1109/percomw.2007.55.

[12] J. C. Hung, V. Hsu, and Y.-B. Wang. A Smart-Travel System Based on Social Network Service for Cloud Environment. In *Proc. 3rd Int'l Conference on Intelligent Networking and Collaborative Systems*, pages 514–519, Fukuoka, Japan, Nov. 2011. doi:10.1109/incos.2011.38.

[13] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proc. 2nd Int'l Conference on Simulation Tools and Techniques*, Rome, Italy, 2009. doi:10.4108/icst.simutools2009.5674.

[14] K. H. Kim, S. J. Lee, and P. Congdon. On cloud-centric network architecture for multi-dimensional mobility. In *Proc. workshop on Mobile cloud computing*, pages 1–6, Helsinki, Finland, 2012. doi:10.1145/2342509.2342511.

[15] X. Li, H. Zhang, and Y. Zhang. Deploying Mobile Computation in Cloud Service Cloud Computing. In *Proc. Int'l Conference on Cloud Computing*, pages 301–311, Beijing, China, 2009. doi:10.1007/978-3-642-10665-1_27.

[16] A. Lindgren, A. Doria, and O. Schelén. Probabilistic Routing in Intermittently Connected Networks. In *Proc. SAPIR*, pages 239–254, Fortaleza, Brazil, Aug. 2004. doi:10.1007/b99076.

[17] E. Miluzzo, R. Cáceres, and Y. F. Chen. Vision: mClouds - computing on clouds of mobile devices. In *Proc. 3rd ACM workshop on Mobile cloud computing and services*, pages 9–14, Low Wood Bay, Lake District, UK, 2012. doi:10.1145/2307849.2307854.

[18] M. V. Pedersen and F. H. P. Fitzek. Mobile Clouds: The New Content Distribution Platform. *Proc. of the IEEE*, 100(Special Centennial Issue):1400–1403, May 2012. doi:10.1109/jproc.2012.2189806.

[19] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, Nov. 2006. doi:10.1109/mcom.2006.248176.

[20] A. Vahdat and D. Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, Apr. 2000.

[21] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt. Cloudlets: bringing the cloud to the mobile user. In *Proc. 3rd ACM workshop on Mobile cloud computing and services*, pages 29–36, UK, 2012. doi:10.1145/2307849.2307858.