2017 IEEE 37th International Conference on Distributed Computing Systems

In Vivo Evaluation of the Secure Opportunistic Schemes Middleware using a Delay Tolerant Social Network

Corey E. Baker[1], Allen Starke[2], Tanisha G. Hill-Jarrett[3], Janise McNair[2]
[1]Department of Electrical and Computer Engineering, University of California, San Diego
[2]Department of Electrical and Computer Engineering, University of Florida
[3]Department of Clinical & Health Psychology, University of Florida
Email: cobaker@eng.ucsd.edu, allen1.starke@ufl.edu, thilljarrett@phhp.ufl.edu, mcnair@ece.ufl.edu

***Abstract*—Over the past decade, online social networks (OSNs) such as Twitter and Facebook have thrived and experienced rapid growth to over 1 billion users. A major evolution would be to leverage the characteristics of OSNs to evaluate the effectiveness of the many routing schemes developed by the research community in real-world scenarios. In this demonstration, we showcase the Secure Opportunistic Schemes (SOS) middleware which allows different routing schemes to be easily implemented relieving the burden of security and connection establishment. The feasibility of creating a delay tolerant social network is demonstrated by using SOS to enable AlleyOop Social, a secure delay tolerant networking research platform that serves as a real-life mobile social networking application for iOS devices. AlleyOop Social allows users to interact, publish messages, and discover others that share common interests in an intermittent network using Bluetooth, peer-to-peer WiFi, and infrastructure WiFi.**

I. Introduction

Over the past decade, online social networks (OSNs) such as Twitter and Facebook have thrived and experienced rapid growth to over 1 billion users [1]. A major limitation of OSNs is the dependence on Internet which is often sparse, difficult to maintain, or unavailable in rural areas or developing communities. In developed communities with cellular infrastructure, networks can become overwhelmed by too many users, particularly during emergencies. In natural disaster situations, Internet and cellular communication infrastructures can be severely disrupted, prohibiting users from notifying family, friends, and associates about safety, location, food, water, and other resources. In addition, natural disasters typically damage infrastructure, which increases network traffic demands on any available undamaged infrastructure, causing congestion and delays.

Opportunistic communication can seamlessly supplement Internet connectivity when needed and keep communication channels open even during high-use and extreme situations. Furthermore, opportunistic communication can also serve as a low-cost solution for smart cities, allowing developing and metropolitan areas to route smart city data through mobile and stationary nodes such as pedestrians, vehicles, street lights, public transportation. DTN routing has the ability to deliver data in an intermittent network, but a major challenge for DTN routing is assessing real-world performance [2], [3], [4]. To truly understand the reliability of DTNs and their ability to support social networks, it is imperative that DTN routing schemes are evaluated *in vivo* with use-cases that are replicable, comparable, and available to a variety of researchers.

In this demonstration, we present the Secure Opportunistic Schemes (SOS) middleware, a novel middleware that facilitates secure message delivery in cases where mobile connectivity is limited, unavailable, or non-existent. The SOS middleware supports real-life delay tolerant social networks on mobile devices. This allows mobile devices to leverage SOS to dynamically deliver messages to interested nodes when network infrastructure is not available and improve message delivery when infrastructure is available. Additionally, the AlleyOop Social research platform is leveraged, which serves as an overlay application for SOS to create a delay tolerant social network for Apple iOS devices [5]. AlleyOop Social is named after the basketball play known as an "alley oop". An "alley oop" occurs when one player throws the ball close to the basket, but it is not able to reach the final destination. While the ball is in flight, a teammate that is closer to the basket catches the ball and scores. In the same regard, AlleyOop Social enables wireless mobile users to communicate over longer distances by sending messages that cannot reach the final destination, but are "caught" by intermediate mobile devices, which continue to catch and pass the messages until they are delivered to the final destination.

II. Related Work

In recent years, a number of social-aware and social-based routing schemes have leveraged social interactions to deliver data using delay tolerant networks (DTNs) [6]. However, related work has primarily evaluated routing protocols in simulation environments, which provide valuable analyses, but are based on synthetic mobility patterns to emulate node movement and tend to use abstract models to imitate the radio response of real commodity wireless technologies[2], [3], [4]. There are a few studies that have taken on the approach of demonstrating DTNs in realistic environments [7], [8],

1063-6927/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDCS.2017.258

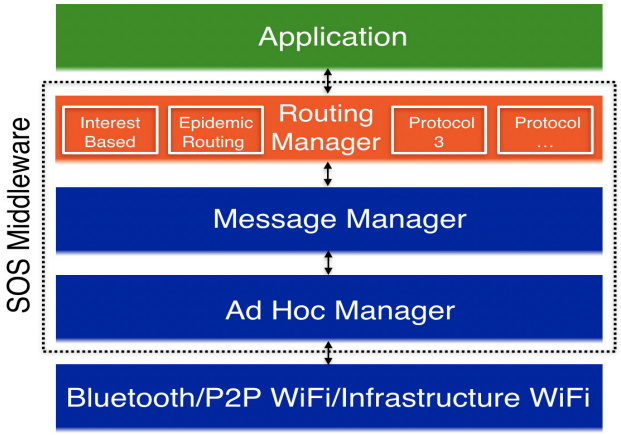


Fig. 1. SOS middleware system stack - green represents mobile applications created by developers, orange represents the modular routing layer consisting of multiple opportunistic schemes created by academic researchers, blue represents foundational layers in the middleware that consist of encryption, authentication, peer discovery, connection management, and data dissemination. Managers marked with the color "blue" cannot be modified by mobile application developers or academic researchers.

[9]. However, these studies do not consider other significant aspects, such as user security and privacy along with the limitation of operating with only the epidemic routing scheme. Various middlewares [10], [11], [12], [13], testbeds [14], [15], [16], [17], [18], [19], [20], [21], and mobile applications have been developed to address providing deployable delay tolerant networking applications which can operate with minimal infrastructure and effectively evaluate DTN routing protocols.

III. Secure Opportunistic Schemes (SOS) Middleware

The SOS middleware is an underlying framework that turns the AlleyOop Social research platform into a delay tolerant mobile social network. The SOS middleware takes a modular approach to abstract away much of the complexity involved in implementing opportunistic routing schemes such as device dicovery, establishing D2D connections, and handling device security and privacy. DTNs are intended to provide an overlay architecture above the existing transport layer and ensure reliable routing during intermittency [22]. Building on the knowledge gained from previous middlewares [23], [24], SOS hides the complexity of the network stack (session and presentation) within the ad hoc manager, message manager, and routing manager, allowing any mobile application to run at the application layer as depicted in Figure 1. Different from other middlewares such as the Haggle Project [23], a separate instance of the SOS middleware is intended to run within each mobile application as opposed to a daemon which often requires devices to be rooted or jailbroken. Designing SOS in this manner allows for the middleware to be integrated within any mobile application in iOS, enabling them to support opportunistic communication without jailbreaking devices along with being compliant with App Store regulations.

A. Application

Mobile applications serve as an overlay to the SOS middleware. Applications can be of any form such as social networking, medical, or any other type of application that would like to share data opportunistically. Mobile applications are responsible for providing a user interface to users and storing data to local or online storage systems. The SOS Middleware provides a number of API's for sending/receiving data, surrounding user notification, routing protocol selection, and security and privacy preferences. Existing mobile applications can simply add the SOS middleware as a framework and start using the aforementioned API's to send and receive data. Applications are responsible providing the data to be sent as well as handling data once it has been received and decrypted.

B. Routing manager

The routing manager is responsible for leveraging D2D connections to transform any application into a delay tolerant networking application that delivers messages to out of range nodes in the midst of intermittency. Routing in SOS is designed for modularity, permitting additional DTN routing schemes to be developed on top of the message manager and run seamlessly under the Application layer. Designing SOS in this manner allows for a flexible middleware that enables applications to dynamically change based on user preference without the need of modifying hardware or other layers in the software stack. Currently, the routing manager in SOS has two DTN routing protocols implemented: epidemic routing and interest-based routing. Epidemic routing is a simple routing scheme that achieves effectiveness through gratuitous replication and delivery of messages upon node encounters [25]. The IB routing protocol operates in a similar manner to epidemic routing, except, instead of propagating messages to all users, messages are only propagated to interested users who are subscribed to the publisher of the original message. Due to the modular nature of the SOS middleware, additional routing protocols can be added to the routing manager. APIs are available to all protocols in the routing manager to facilitate communication between the message manager and the application layer. Both the IB and Epidemic routing protocols are written in less than 100 lines of Swift code.

C. Message manager

The message manager notifies the respective protocol used in the routing manager whenever a new peer has been discovered or lost. Additionally, the message manager is responsible for taking action whenever a connection state changes. For example, if the disconnection between two users is lost, the message manager knows what messages were not transferred. Lastly, the message manager translates messages between the routing manager and ad hoc manager in a common format for both layers to interpret.

D. Ad hoc manager

The ad hoc manager manages Apple's multipeer connectivity (MPC) framework, which allows communication between

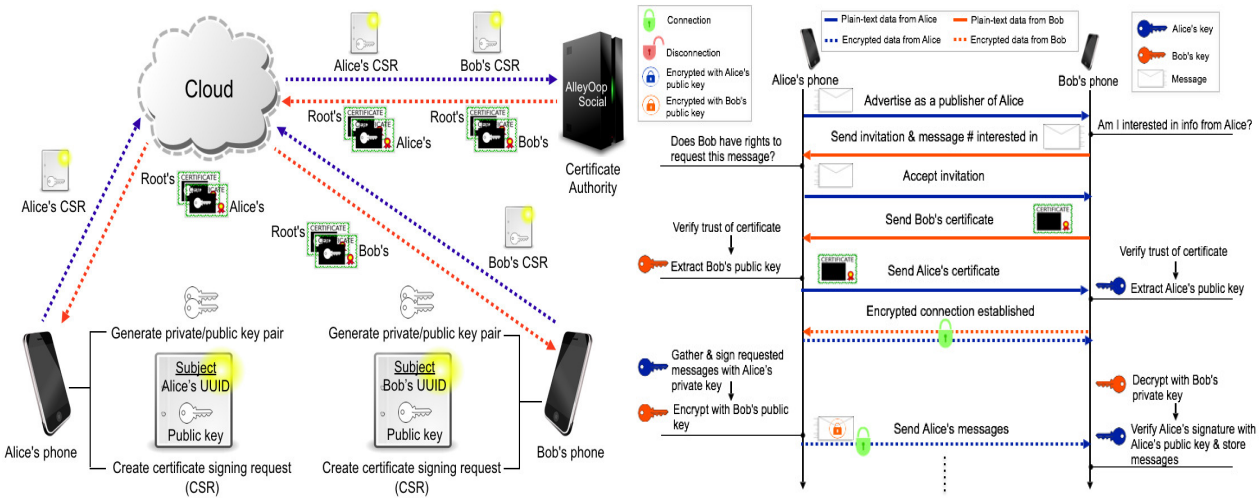


Fig. 2. (a) One-time infrastructure requirement, occurs during account creation to enable DTN security. (b) Decentralized communication between nodes.

iOS, macOS, and tvOS devices using peer-to-peer WiFi, Bluetooth personal area networks, or infrastructure WiFi networks[1]. To the best of our knowledge, SOS is the first middleware to leverage MPC to evaluate multiple delay tolerant routing schemes. The ad hoc manager is responsible for viewing discovered peers, establishing D2D connections, encrypting connections, encrypting data from end-to-end, generating keys, validating certificates, as well as signing and verifying data sent and received data. Apple's documentation on how to use MPC is detailed, but the company does not disclose specific details on how MPC works. For example, specifics about the encryption methods MPC uses are not provided. Details about how the SOS middleware handles security and privacy are elaborated on in Section IV.

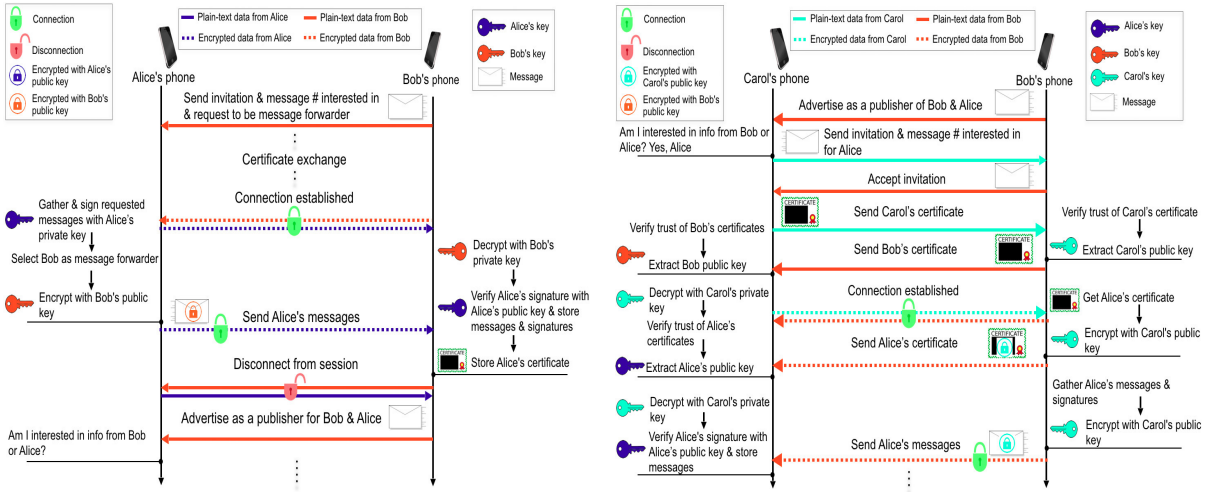
IV. Privacy and Security

In regard to network security there is no "one-size-fits-all" approach [26]. Security concerns may become exacerbated in delay tolerant and ad hoc applications where nodes are vulnerable to attacks such as eavesdropping, denial of service, and compromised devices. Providing secure communication that prevents an adversary from accessing and/or modifying data is a fundamental requirement of any DTN application [27]. Previous research discusses security in opportunistic applications conceptually and makes no claims that the implementations are secure [23]. The intent of the section is to provide a novel, but simple concept and implementation of an initial layer of security for DTN protocols and enable the overlaying mobile application to detect the identity of its users, send encrypted information, verify the originating source of the information being forwarded, and ensure that data have not been modified — all with minimal dependence on centralized infrastructures.

[1]Apple Inc., Multipeer connectivity framework reference, https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework/

Additional security can be added to AlleyOop Social by incorporating mechanisms such as distributing CA functionality amongst nodes [28], or integrating trust measurements within the routing schemes [29] available in the routing manager discussed in Section III. To enable the initial layer of security in the SOS middleware, AlleyOop Social leverages conventional public-key infrastructure (PKI) techniques to create a one-time PKI requirement that occurs during initial download and user-signup for the application. AlleyOop Social assumes that users will have Internet connectivity during the initial download and installation of the mobile app. After the one-time infrastructure requirement, Internet connectivity is no longer needed for privacy, security, and message dissemination. The process of generating keys and receiving X.509 certificates in AlleyOop Social's one-time infrastructure requirement is depicted in Figure 2a.

Using the one-time infrastructure requirement in Figure 2a is not without limitations. The obvious shortfall is the "one-time" requirement. A fair assumption is that the AlleyOop Social application along with others using the SOS middleware will acquire their mobile applications from the Apple App Store, which currently requires an Internet connection. Assuming users sign up shortly after acquiring the application addresses some concerns with the "one-time" requirement. Additionally, if a connection between a device and the cloud is somehow compromised, or a malicious device attempts to provide someone else's unique user-identifier during user sign-up, a certificate with the wrong credentials could be generated by the CA. To circumvent this issue, the cloud can ask the CA to compare and validate the unique user-identifier provided in the certificate with the unique user-identifier affiliated with the logged in user. Other limitations are also prevalent with the current security scheme such as an Internet connection is required to revoke specific user certificates, update CA root certificates, replenish expired certificates, and notify users of known malicious devices.



(a) AlleyOop Social message forwarder selection

(b) AlleyOop Social message forwarder dissemination

Fig. 3. AlleyOop Social message forwarder

V. Message Dissemination

After sign-up is complete and a mobile device receives its respective certificate and AlleyOop Social CA root certificate, the user can disseminate messages to other AlleyOop Social users using any DTN routing protocol discussed in Section III-B. Whenever a user creates a message or performs an action such as follow/unfollow of a user, AlleyOop Social performs the following two operations: 1) saves the action to the local database on the mobile device and 2) synchronizes the action with the cloud when the Internet becomes available. Once an action is saved to the local database of the device it can be disseminated using a DTN routing protocol to interested AlleyOop Social users without the use of Internet. The following sections expound upon how messages are disseminated after being created by a user in the AlleyOop Social application layer and passed to the routing manager.

A. Advertisements and node discovery

Mobile devices roam freely advertising and browsing for basic information in plain-text to assist other AlleyOop Social enabled devices with making the decision of whether or not to request a connection. For example, the epidemic and IB routing protocols discussed in Section III-B advertises a plain-text key/value dictionary consisting of *UserID/MessageNumber*. The key field in the dictionary is a 10 byte unique user identification string. The value field of the dictionary is the latest *MessageNumber* that the advertising device has for the particular *UserID*. A browsing node is now able to quickly decide whether it is interested in the *MessageNumber* for the respective *UserID* string and whether it should request a connection from the advertising node. Figure 2b depicts a typical scenario in AlleyOop Social where Bob's device is interested in messages from Alice's device.

B. Forwarder Selection & Dissemination

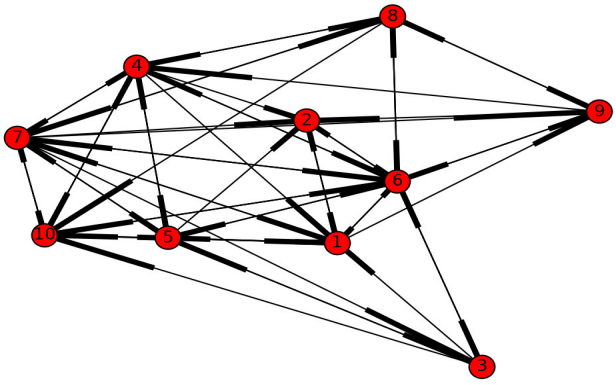
Depending on the DTN routing protocol being used in the Routing layer of the mobile device, prospective nodes can become message forwarders for other users. For example, in epidemic and IB routing, a node becomes a message forwarder for a particular user-identifier whenever a new message is requested and received. When a node becomes a message forwarder, it follows a similar process to the one outlined in Figure 2b, with particular differences that are shown in Figures 3a. Figure 3b shows the interaction between Bob and Carol when Carol is interested in Alice's message that Bob is forwarding. The process is similar to message dissemination in Figure 2b, except Bob sends his certificate to Carol to establish an encrypted connection and in addition, forwards Alice's certificate.

VI. Real World Evaluation

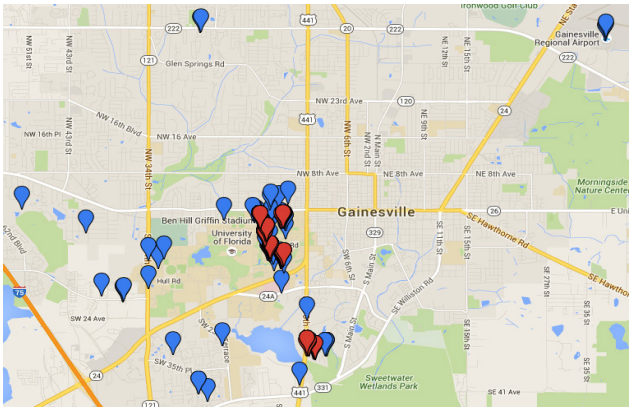
The AlleyOop Social application was available for beta testing in Apple TestFlight app for 7 days. AlleyOop Social had thirty one (31) testers who downloaded the application around the United States. Due to limited amount of users and a critical mass of users who are socially related to each other, this section will constrain the results to users who passed at least one (1) D2D message using the IB routing protocol in Gainesville, FL. Ten devices used AlleyOop Social in a $\sim$11km x 8km area depicted in Figure 4b, resulting in users posting 259 unique messages.

A. Social relationships

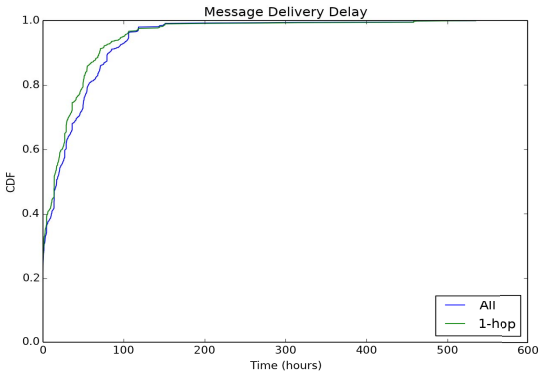
Many of the students were friends before the field study and typically interacted during the school week. Individual users were given the freedom to choose other users to subscribe to; therefore, all users did not follow each other. The digraph $\mathcal{G}(V, E)$ formed by the total nodes who participated $n = |V(\mathcal{G})| = 10$ is depicted in Figure 4a. A social



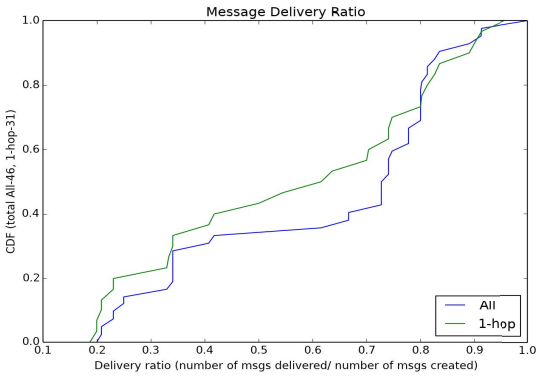
(a) Social relationship directed graph for the ten (10) active users in Gainesville, FL



(b) AlleyOop Social map of "active user" message generation (blue) and message dissemination (red) in a ∼11km x 8km area



(c) Delay



(d) Delivery

Fig. 4. AlleyOop Social real world evaluation results

relationship between a node pair $i, j \in E$ is an edge $e_{i,j}$, meaning that user i follows user j. The edge $e_{i,j}$ does not necessarily mean the edge $e_{j,i}$ exists because some users did not follow each other back as in the case for node 1 and node 3 in Figure 4a. The density of the social relationships is 0.64, meaning that the majority of the possible social relationships were formed naturally by the participating nodes. The compactness of $\mathcal{G}$ can be determined by calculating the average shortest path length between all node relationship pairs $\sum_{i \geq j} l(i,j) / \frac{n(n-1)}{2} = 1.3$, along with the maximum shortest path length, otherwise known as the diameter d between any two nodes $d(\mathcal{G}) = \max_{i,j \in V} \; l(i,j) = 2$. The compactness of the social relationship graph reveals that even if a user does not follow another user directly, there is still an indirect follower that is two degrees away. The center nodes (6 and 7) of the social relationship graph has a radius of 1 which reflects the nodes with the smallest eccentricity$_{i,j} = \max_{i,j \in V} \; l(i,j)$.

Additional features can be determined by translating Figure 4a to a undirected graph. This means that if a two-way relationship did not already exist, it will exist in the undirectional graph making $e_{i,j} = e_{j,i}$ for all $i, j \in E$. Now the network transitivity is computed to be $T(\mathcal{G}) = 3 *$ number of triangles / number of connected triads $= 0.80$ which measures the extent that a friend k of a friend j is also a friend of i.

B. Message dissemination

The social relationship graph in Figure 4a provides an overall understanding of nodes' interests in messages along with providing insight into how nodes may cluster due to who they follow. Figure 4a does not provide any insight on physical node locations or mobility during the evaluation. Figure 4b assists with understanding node mobility by showing where users created messages (blue) and passed messages (red) in Gainesville, FL. A total of 967 messages were disseminated from user-to-user using IB routing in AlleyOop Social. The total amount of subscriptions made by the ten (10) active users was 46. Figure 4c provide the delay results for messages disseminated via "1-hop" and "All" hops. In regard to "All" messages, Figure 4c shows that 0.43 of the messages delivered had a delay of 24 hours or less, while 0.90 of the messages had a delay of 94 hours or less. In regard to "1-hop" delay, that 0.44 of the messages delivered had a delay of 24 hours or less, while 0.92 of the messages had a delay of 94 hours or less for "1-hop" messages.

In regard to message delivery, Figure 4d shows that 0.30 of the subscriptions had a delivery ratio greater than 0.80 for "All" messages. 0.50 of the subscriptions had a delivery ratio greater than 0.70 for all messages. 0.25 of the subscriptions had a delivery ratio of 0.80 for "1-hop" messages. Users delivered 0.826 of the 967 messages via 1-hop. The additional 0.174 were delivered using 2-hops or more and is depicted in "All". The compactness of the social relationships between the nodes discussed in Section VI-A partially explains why the majority of the messages were delivered within "1-hop". Note the low density due to real people being able to operate freely in a large city area (88km^2), which resulted 0.93 of the messages being delivered within in 94 hours of creation. DTN simulations typically model 50 to 100 nodes in a constrained simulation space ranging between 0.25km^2 - 4km^2. In addition, node mobility tends to become stationary, for at least 5-8 hours a day due to the human requirement to sleep, thus limiting possible interactions between nodes. The results at such a low density provide promising insight into delay tolerant social networks and suggest further investigations at higher densities are needed.

VII. Demonstration

During the demonstration attendees will be able to download AlleyOop Social on their iOS devices via Apple TestFlight. Users can follow friends, post new messages, as well as toggle between DTN routing schemes inside the application. We will demonstrate both the online and offline modes by disconnecting mobile devices from cellular and WiFi networks.

References

[1] M. Faloutsos, T. Karagiannis, and S.-H. Moon, "Online social networks," *Network, IEEE*, vol. 24, no. 5, pp. 4–5, 2010.

[2] P. Hui and A. Lindgren, "Phase transitions of opportunistic communication," in *Proceedings of the third ACM workshop on Challenged networks*. ACM, 2008, pp. 73–80.

[3] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.

[4] C. Baker, J. Almodovar-Faria, P. S. Juste, and J. McNair, "Low energy socially cognizant routing for delay tolerant mobile networks," in *Military Communications Conference, MILCOM 2013-2013 IEEE*. IEEE, 2013, pp. 299–304.

[5] C. E. Baker, A. Starke, S. Xing, and J. McNair, "A research platform for real-world evaluation of routing schemes in delay tolerant social networks," *arXiv preprint arXiv:1702.05654*, 2017.

[6] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 556–578, 2014.

[7] A. Doria, M. Uden, and D. Pandey, "Providing connectivity to the saami nomadic community," in *In Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

[8] A. S. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[9] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "Ibr-dtn: an efficient implementation for embedded systems," in *Proceedings of the third ACM workshop on Challenged networks*. ACM, 2008, pp. 117–120.

[10] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware for mobile social networking," in *Proceedings of the 2nd ACM workshop on Online social networks*. ACM, 2009, pp. 49–54.

[11] Ó. R. Helgason, E. A. Yavuz, S. T. Kouyoumdjieva, L. Pajevic, and G. Karlsson, "A mobile peer-to-peer system for opportunistic content-centric networking," in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*. ACM, 2010, pp. 21–26.

[12] M. Skjegstad, F. T. Johnsen, T. H. Bloebaum, and T. Maseng, "Mist: A reliable and delay-tolerant publish/subscribe solution for dynamic networks," in *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*. IEEE, 2012, pp. 1–8.

[13] D. J. Dubois, Y. Bando, K. Watanabe, A. Miyamoto, M. Sato, W. Papper, and V. M. Bove, "Supporting heterogeneous networks and pervasive storage in mobile content-sharing middleware," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 841–847.

[14] O. Mukhtar and J. Ott, "Backup and bypass: introducing dtn-based ad-hoc networking to mobile phones," in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*. ACM, 2006, pp. 107–109.

[15] J. Morgenroth, S. Schildt, and L. Wolf, "A bundle protocol implementation for android devices," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 443–446.

[16] P. S. Juste, K. Jeong, H. Eom, C. Baker, and R. J. Figueiredo, "Tincan: User-defined p2p virtual network overlays for ad-hoc collaboration." *EAI Endorsed Trans. Collaborative Computing*, vol. 2, p. e4, 2014.

[17] A. Moraleda-Soler, B. Coll-Perales, and J. Gozalvez, "Link-aware opportunistic d2d communications: Open source test-bed and experimental insights into their energy, capacity and qos benefits," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*. IEEE, 2014, pp. 606–610.

[18] F. Ben Abdesslem and A. Lindgren, "Demo: mobile opportunistic system for experience sharing (moses) in indoor exhibitions," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 267–270.

[19] P. S. Paul, S. Nandi, S. K. Dey, K. De, P. Pramanik, and S. Saha, "Challenges in designing testbed for evaluating delay-tolerant hybrid networks," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, 2015, pp. 280–283.

[20] S. Siby, A. Galati, T. Bourchas, M. Olivares, T. R. Gross, and S. Mangold, "Method: A framework for the emulation of a delay tolerant network scenario for media-content distribution in under-served regions," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*. IEEE, 2015, pp. 1–9.

[21] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," 2016.

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

[23] J. Su, J. Scott, P. Hui, J. Crowcroft, E. De Lara, C. Diot, A. Goel, M. H. Lim, and E. Upton, *Haggle: Seamless networking for mobile applications*. Springer, 2007.

[24] M. Caporuscio, P.-G. Raverdy, and V. Issarny, "ubisoap: A service-oriented middleware for ubiquitous networking," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 86–98, 2012.

[25] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[26] R. Perlman, "An overview of pki trust models," *Network, IEEE*, vol. 13, no. 6, pp. 38–43, 1999.

[27] R. Cabaniss, V. Kumar, and S. Madria, "Multi-party encryption (mpe): secure communications in delay tolerant networks," *Wireless Networks*, vol. 21, no. 4, pp. 1243–1258, 2015.

[28] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad hoc networks," in *icnp*. IEEE, 2001, p. 0251.

[29] U. Kumar, G. Thakur, and A. Helmy, "Protect: proximity-based trust-advisor using encounters for mobile societies," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*. ACM, 2010, pp. 636–645.