Performance and Energy Consumption Analysis of a Delay-Tolerant Network for Censorship-Resistant Communication

Yue Liu† David R. Bild† David Adrian† Gulshan Singh†
Robert P. Dick† Dan S. Wallach‡ Z. Morley Mao†

†University of Michigan
Ann Arbor, MI, USA
{liuyue,drbild,davadria,gulshan}@umich.edu
{dickrp,zmao}@eecs.umich.edu

‡Rice University
Houston, TX, USA
dwallach@cs.rice.edu

ABSTRACT

Delay Tolerant Networks (DTNs) composed of commodity mobile devices have the potential to support communication applications resistant to blocking and censorship, as well as certain types of surveillance. We analyze the performance and energy consumption of such a network, and consider the impact of random and targeted denial-of-service and censorship attacks. To gather wireless connectivity traces for a DTN composed of human-carried commodity smartphones, we implemented and deployed a prototype DTN-based microblogging application, called 1am, in a college town. We analyzed the system during a time period with 111 users. Although the study provided detailed enough connectivity traces to enable analysis, message posting was too infrequent to draw strong conclusions based on user-initiated messages, alone. We therefore simulated more frequent message initiations and used measured connectivity traces to analyze message propagation. Using a flooding protocol, we found that with an adoption rate of 0.2% of a college town's student and faculty population, the median one-week delivery rate is 85% and the median delivery delay is 13 hours. We also found that the network delivery rate and delay are robust to denial-of-service and censorship attacks eliminating more than half of the participants. Using a measurement-based energy model, we also found that the DTN system would use less than 10.0% of a typical smartphone's battery energy per day in a network of 2,500 users.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks, Store and forward networks*; C.4 [**Performance of Systems**]: Performance attributes, Reliability, availability, and serviceability

MobiHoc'15, June 22–25, 2015, Hangzhou, China.

http://dx.doi.org/10.1145/2746285.2746320.

Keywords

Delay Tolerant Networks, Opportunistic networks, Microblogging, Censorship resistance, Energy overhead

1. INTRODUCTION

Delay Tolerant Networks (DTNs) composed of human carried devices have the potential to support blocking-, censorship-, and surveillance-resistant communication applications. Unlike the Internet, the non-hierarchical, device-to-device structure of DTNs makes it difficult to block, modify, or monitor a large portion of network traffic via attacks on a few devices.

The demand for the functionality offered by infrastructureless communication systems is suggested by the response to intentional widespread service disruption for users of social media applications such as Twitter and Facebook during the Arab Spring [1], ongoing political censorship in Iran and China, and growing awareness of global surveillance of individuals without reasonable cause to believe they have committed crimes. Recently in Hong Kong, in response to the threat of potential denial of service attacks on the Internet-based Instagram, there were 200,000 downloads of a local mesh networking chat program called FireChat between 28 and 30 September 2014 [2]. This application supports text messaging and photo transfer mediated by device-to-device Bluetooth connections [2]. Popular social networking/microblogging services are now commonly censored throughout much of the world. For example, Twitter is coerced to censor its users on a country-by-country basis by the threat of punishment by various governments [3]. In response to this trend, we have evaluated the feasibility, performance, and energy consumption of a DTN system composed of commodity smartphones and tablets when subjected to random and targeted denial-of-service and censorship attacks.

DTNs composed of commodity smartphones and tablets are influenced by constraints on battery energy, computation speed, and network throughput. We evaluate the potential of DTNs to support microblogging using a flooding protocol variant that avoids redundant transmissions. Message delivery rates and latencies, as well as battery energy overheads, are evaluated using contact traces gathered via a deployment study. We developed a prototype microblogging application on Android, named *1am* [4], and made it available to volun-

teers in a university town for installation on their personally owned smartphones.

Over a period of seven months in 2013, 1am had 291 users, with 111 users during a month of high activity on which our analysis focuses. Although users employed the app for microblogging, the limited number of postings made it difficult to draw strong conclusions based on app usage patterns. However, these volunteers' device-to-device wireless connectivity traces enabled findings on feasibility, performance, energy consumption, and resistance to denial-of-service and censorship attacks. The 1am traces are publicly available (see Section 8 for the URL).

- Feasibility and performance: Our analysis indicates that in a college town in which 0.2% of the population install the app, the system has a median delivery rate of 0.85 and a median delivery delay of 13 hours. We pay special attention to the system's performance variations and study their causes. We find that the delivery delays of DTN flooding protocols follow a power law distribution that varies from hours to days (Figure 8), which may be of interest to application designers.
- Robustness to denial-of-service and censorship attacks: We found that attacks that randomly remove half of the network participants only reduce delivery rates to the remaining participants by 3%. Targeted attacks, which remove nodes that appear most frequently on the most rapid message delivery paths, were more harmful, causing the delivery rate to decrease precipitously when half of the participants were removed. However, even when subjected to targeted attacks, the network only suffered a 13% decrease in delivery rate when a third of its participants were removed.
- Energy consumption: We analyzed the impact of participating in the proposed infrastructureless microblogging network on smartphone battery lifespan and found negligible impact on battery energy for participants in our 111-user study. Based on a measurement-based wireless communication aware smartphone power consumption model, our analysis shows that participants in a network of 2,500 active users will each expend 10% of their battery energy during a single day of use.

2. RELATED WORK

The idea of using infrastructureless networks to support censorship-resistant communication is shared by various existing work [5, 6, 7]. Al-Akkad et al. use mobile ad hoc networks (MANETs) to enable continuous tweet dissemination when network infrastructure is disrupted [5], which has fewer applications than DTN-based systems because MANETs require instantaneous end-to-end connections. Hossmann et al. describe a DTN-based Twitter application for disaster communication. They do not provide evaluation results due to the lack of contact traces specific to disaster scenarios [6]. Our goal is providing daily microblogging services in local communities. We therefore use daily human contact traces to study achievable message delivery performance and energy consumption in such systems.

Due to the difficulty of collecting large-scale human contact traces, most existing work on DTNs [8, 9] uses a few existing traces [10, 11] for evaluation. The Dartmouth traces record the association histories of all participants on Dartmouth College's campus WiFi network [10]. Most of the recorded devices are laptops, whose mobility and contact patterns may differ from smartphones. The UCSD traces were collected from participants who used experimental PDAs during the deployment [11]. Their traces show a trend of declining user engagements [11] that is uncommon for normal smartphone users. To gather traces representative of the mobility and contact patterns of modern smartphone users, we installed data collection software on participants' personally owned smartphones.

We are unaware of existing models for the energy consumptions of common DTN routing protocols or applications. Socievole et al. evaluated the energy consumption of five DTN routing protocols via simulation [12]. They modeled Bluetooth communication, while we modeled 802.11 ad hoc communication. In addition, their analysis considered networks with a maximum size of 200 nodes, while we considered networks with 50,000 nodes.

Various existing work on improving the energy overhead of DTN flooding protocols assumes a simple linear relationship between the network's energy consumption and the average number of copies per message [13, 14]. However, our energy model shows that the cost of learning each node's received and not-yet-received messages to avoid redundant transmissions is the leading energy contributor in large networks, as this cost scales quadratically with network size. This result suggests the simple linear model is inadequate, and possibly misleading.

3. 1AM: MICROBLOGGING ON DTNS

This section describes the design of *1am*, an infrastructureless DTN-based microblogging service. This information is provided to give background and context to the subsequent analysis sections of the paper.

3.1 Overview

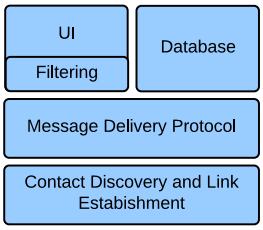

Figure 1: 1am architecture.

1am is designed to functionality similar to Twitter. Posts are public text messages containing up to 514 characters. Users can post new messages, make comments, and retransmit (analogous to retweet) messages and comments from others. Its main user interface (UI) displays a timeline of incoming messages.

We now describe some of 1am's characteristics and features.

Identity Management: 1am has no central registry that might be used to enable blocking, censorship, or surveillance attacks; it uses a fully distributed design. Users are uniquely identified by self-generated public keys, created locally during application installation. In other words, the public key is the identity, eliminating any need of certification authorities for certifying identity-key associations. We allow 1am users to have human readable user names, which may collide. We resolve user name collisions by comparing public keys that are extremely unlikely to collide. The UI is designed to make the implications clear to users when the same user name is associated with different keys.

Message Authenticity: Like all public communication systems, 1am is subject to rumors and misinformation. Conventional solutions relying on centralized filtering or censor-

ship cannot be used, due to 1am's decentralized architecture. To increase the cost of disinformation campaigns, we protect the authenticity of 1am posts with digital signatures, which are individually verified by each receiver. The intention is for credible 1am users to gradually build up positive reputations. In addition, it would be possible to use distributed spam or propaganda detection techniques [15]. There is no requirement for users to associate their actual identities with their 1am identities.

Multimedia Support: Bandwidth is a scarce resource in device-to-device networks: 1am cannot support content of arbitrary size. Our study shows that it is feasible to support images. However, without re-designs of the message delivery protocol or major improvements on wireless transceiver energy efficiency, video sharing cannot be supported in our system (see Section 7).

Figure 1 illustrates the architecture of 1am. From the bottom up, the first layer is in charge of **contact discovery and link establishment**. It detects contact opportunities, establishes an 802.11 ad hoc channel with discovered peers, and informs higher level protocols. Note that this layer is not 1am-specific and should be shared by all DTN applications. The second layer is the **message delivery protocol** that routes 1am messages to intended receivers. Finally, when a 1am message arrives, the **UI** is informed and the message is stored in a **database** for future use, e.g., re-transmission. The UI's displays messages according to given filtering rules, e.g., those from immediate wireless communication neighbors or "followed" identities.

We choose 802.11 ad hoc technology over Bluetooth for device-to-device communication, because the former has longer transmission range. We avoided WiFi Direct due to the inconvenience of its pairing process, which requires human involvement, as well as its lack of broadcast support.

Although modern smartphone hardware and operating systems support 802.11 ad hoc communications, at the time of the study an API was not provided by Android. During our 1am deployment study, to avoid requiring users to "root" their phones, we used an ad hoc network emulation layer to identify direct wireless connectivity, deliver messages, and gather data[1] (See Section 4.1 for more details).

3.2 Message Delivery Protocol

We use a flooding protocol for message delivery for two reasons. First, flooding protocols can be used to support arbitrary filtering rules. At this stage we are not sure which rules are most appropriate for DTN-based microblogging services and therefore avoid any premature optimization tying the system design to specific rules. Second, in non-congested situations, e.g., for applications with small traffic volume, flooding protocols have the best message delivery performance and worst energy consumption among all routing protocols. Therefore, they serve as good reference points for understanding the limitations of the underlying network and provide guidance for further optimization.

To avoid redundant message transmissions, we exchange Bloom filters during each contact to learn about each node's received and not-yet-received messages. Bloom filters are compact data structures for storing elements and testing set membership [16]. They are widely used in DTN routing protocols for their space- and bandwidth-efficiency. Specifically, each node stores all its received messages in a Bloom filter that is exchanged at each contact. The other party could then test which of its own messages have not been received by the node. Classic Bloom filters use 14.4 bits per key for a 10^{-3} set membership test error rate [16]. This is much smaller than the original message it documents.

[1]We subsequently developed an application-level method that supports direct device-to-device WiFi communication on Android platforms.

4. USER TRACE COLLECTION

We developed a prototype of 1am and provided it to volunteers in a college town with a student and faculty population of 50,000. The application was instrumented (with knowledge of the study participants) to gather mobility and wireless connectivity data. This section describes the data collection platform and provides some statistics on the gathered traces. It gives an overview of participants' motion and contact patterns, and provides background for understanding the performance numbers in Section 5.

4.1 Deployment Platform

Commodity Android devices do not support appropriate 802.11 ad hoc communication [17]. Enabling it would require rooting participants' phones, which we consider an unacceptable barrier to study participation. Instead, we applied a widely used technique in the DTN research community to estimate 802.11 contact opportunities. To simplify, two devices with simultaneous visibility of common WiFi access points are considered to be capable of direct communication [9, 18].

We use a data collection platform we designed for assisting DTN protocol deployments, called MANES [19]. It gathers time-varying location and wireless environment measurements. It also estimates contact opportunities based on WiFi scan results and provides applications with an emulated 802.11 ad hoc communication interface. MANES replaces the "contact discovery and link establishment" layer in Figure 1.

MANES has two components, an Android client that gathers GPS and WiFi scan measurements, and a backend server that aggregates and stores user traces. Location data are uploaded as soon as they become available to support up-to-date topology estimation, as the server uses the estimated topology to relay packets between 802.11 neighbors.

To save energy for the client, automated sampling frequency adjustment is used when taking both WiFi and GPS measurements. If there are no significant changes since the last measurement, further WiFi scans are postponed for 120 seconds and further GPS scans are postponed for five minutes. Otherwise, the WiFi sampling frequency is 30 seconds and the GPS sampling frequency is one minute. Because humans are usually stationary, the average WiFi measurement period in the collected traces is 114.5 seconds.

4.2 Participant Recruitment

1am traces are collected from fully informed users with their consent. We released 1am to the university community and advertised via posters, gifts of promotional water bottle, and word of mouth. We had 291 users, each of whom used the application at least three days from March to October in 2013. Most were university students. The median trace duration per user is 24 days. We use the first 31 days of traces, the month with the maximum number of participants, for the analysis in the rest of this paper. We refer to this period as the *study period*. During the study period, we had

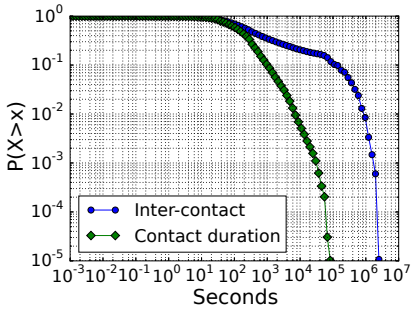

Figure 2: Contact duration and inter-contact time distributions, each following (truncated) power law distributions.

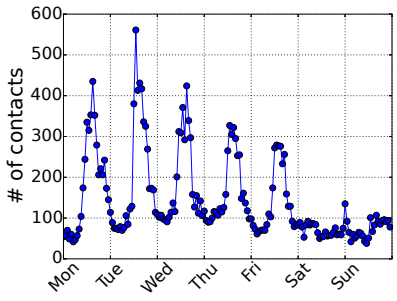

Figure 3: Average # of contacts by weekday, showing a strong weekday/weekend pattern.

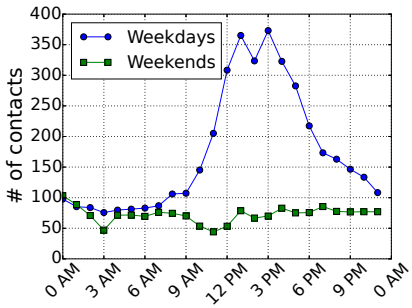

Figure 4: Average # of contacts by hours, showing a strong diurnal pattern with its peak in the early afternoon.

Table 1: Comparison of Contact Traces

	UCSD	Dartmouth	1am
Year	2002	2003/2004	2013
Device	PDA	Laptop/PDA	Smartphone
Duration (days)	77	114	31
Granularity (seconds)	120	300	114.5
Devices participating	273	6,648	111
Number of contacts	195,364	4,058,284	98,072
# contacts/pair/hour	0.0028	0.000067	0.033

111 active study participants, with an average of 85 active participants each day.

4.3 Contact Statistics

During the study period, 98,072 inter-participant wireless communication contacts were made, with an average of 20 contacts, six of which were unique, per user, per day. 10% of users made no contacts. The average contact rate is 0.0329 contacts per pair per hour, which is much higher than the rate of other WiFi-based contact traces widely used in the DTN community (see the last row in Table 1). We suspect there have been changes in normal smartphone users' behaviors since those traces were gathered. In our study, users were carrying their own modern, compact, highly functional smartphones and in the UCSD [11] and Dartmouth traces [10], they were carrying either limited-functionality PDAs or laptops; these devices were carried less frequently than modern smartphones, resulting in decreased contact frequency. It is also possible that 1am users tend to recruit friends and acquaintances with whom they have frequent contact. Previous research shows correlation between physical proximity and social relationships [20].

Figure 2 shows the distributions of contact duration and inter-contact time. The contact duration has a median of 149 seconds. The inter-contact time has a (truncated) power law distribution in the 30 s–1 day range, with a coefficient of 0.3. This is consistent with prior research findings [18].

4.4 Diurnal Patterns

Figure 3 and Figure 4 illustrate a weekly and diurnal pattern of contact numbers, which is expected for a population that is working or studying during weekdays. Note that a pair can meet and have contacts multiple times in a day. On an average weekday (Figure 4, "Weekdays"), the contact number starts to rapidly increase at 9 a.m. and is at its highest from 1 p.m. to 3 p.m. It then gradually decreases, reaching a minimum at around 3 a.m. On an average weekend (Figure 4, "Weekends") the contact number stays almost constant at the minimum level of the average weekday. This difference would be expected for a population in which most contacts result from co-location during weekday work or study.

4.5 Geographic Patterns

To better understand the study population, we examined the geographic locations of contacts. Not all contacts are associated with GPS readings (only 27.6% are), but all are associated with WiFi scan results. This is expected, as humans typically spend most of their time indoors and GPS readings are often not available indoors. We used the following algorithm to estimate GPS locations based on WiFi scan results. We first estimate the GPS locations of WiFi access points using location samples containing both GPS readings and WiFi scan results; the location of one access point is the average of the GPS locations for all measurements with WiFi scans containing that access point. We then estimate the location of each contact event associated with only WiFi scans as the estimated location of the access point with the highest signal strength in these scans. This algorithm allows us to estimate the locations of 99.4% of the contact events.

To simplify explanation and analysis, we divide the entire area into 100 m×100 m discrete locations that we associate contact events with. There are 239 discrete locations with at least one contact, each. Contacts are distributed unevenly. If the locations are ordered by frequency of contacts, we find that 90% of the contacts occur in the first 25% of locations. Figure 5 shows the map of the first 100 locations, with the shade indicating number of contacts. These locations are visually clustered into two components, clearly outlining the two campuses of the university that where the study was done. The Northeast Campus, where the study was most heavily promoted, has more contacts.

To further understand the context of the contacts on each campus, we consider their temporal patterns. Figure 6 shows the average number of contacts in the two campuses by weekday. The Northeast Campus curve ("Location1") has clear weekly and diurnal patterns, while the Southwest Campus curve ("Location2") does not. This suggests that most 1am users work in the Northeast Campus (in fact in a few very concentrated areas of that campus, i.e., the darkest few grid elements), instead of the Southwest Campus.

4.6 Discussion

The above statistics show that 1am users in these traces are a community of people whose contacts are generally a result of work- or study-related co-location. Most of their

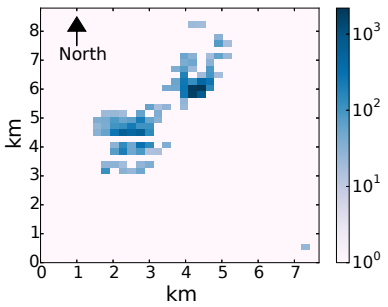

Figure 5: Geographic distribution of the contacts, illustrating the two university campuses. The majority of contacts occured in the Northeast Campus.

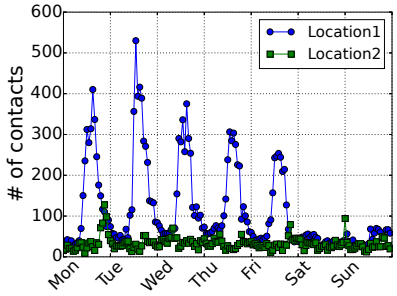

Figure 6: Total # of contacts in the two campuses in an average week. "Location1" (the Northeast Campus) shows weekly and diurnal patterns. "Location2" (the Southwest Campus) does not have obvious patterns.

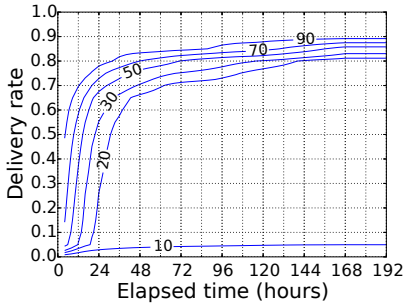

Figure 7: 1am's message delivery rate over time, showing two stages with significantly different propagation speeds. A point on the X-percentile line represents the X-percentile delivery rate at the considered time of the flooding processes.

contacts occur within a few work/study areas. Their motion patterns are less concentrated and correlated during nights and weekends. They continue to have contacts at these times, but with lower frequency. Our conclusions in the rest of the paper will be most relevant to similar deployment scenarios.

5. MESSAGE DELIVERY PERFORMANCE

In this section we evaluate 1am's message delivery performance using two metrics, **message delivery rate**, the percentage of intended receivers that have successfully received a message, and **message delivery delay**, the elapsed time from message initiation to its successful delivery. We also discuss the robustness of the system when subjected to a variety of potential attacks.

5.1 Methodology

Because 1am users initiated too few posts during the study period (on average, around one post per user per month), we cannot draw strong conclusions based on user-initiated messages. We instead simulated more frequent message initiations. In order to fully explore the dynamics of the underlying network, we cause every device in the trace to initiate a message every six minutes. We do not initiate messages from a particular smartphone when it is inactive, i.e., in sleep mode or powered off.

Since 1am uses a flooding protocol to distribute messages, we use timed shortest path analysis to obtain message delivery rates and delays. While this method overlooks performance degradation in congested situations when some messages are forced to follow circuitous routes, it produces an upper bound on message delivery performance that is fairly tight for applications with small traffic volume.

We calculate the shortest-path message distribution tree for each initiated message. The following analysis is based on the aggregated shortest-path trees associated with all message initiations.

5.2 Overall Performance

Figure 7 illustrates the progress of the simulated flooding processes over time. The X-axis shows the elapsed time in hours since message initiation. The Y-axis shows the delivery rate: the ratio of active nodes that have received the message. If a message has not reached a participant within a week of initiation, we consider the delivery failed. As there are too many flooding processes to illustrate individually, we use percentile lines to show the distributions of their delivery rates. For example, at 24 h, the 50th percentile line has a delivery rate of 0.68, meaning 24 h after message initiation the median delivery rate of all the flooding processes is 0.68.

The 50th percentile line shows that it takes a median of 14 hours to reach 60% of participants, 24 hours to reach 70%, and a week to reach 85%. The fastest-spreading 10% of messages (the 90th percentile line) take 6 hours to reach 60% of participants, 12 hours to reach 70%, and a week to reach 90%. The slowest 10% of messages (the 10th percentile line) reached almost no other participants, because 10% of participants had no contacts during the study period.

The message propagation speed (the slope of the percentile lines) has two stages. To reach the first 60% of participants it takes 3 hours for every 10% of progress, but it becomes increasingly difficult to reach the rest of the participants. For example, the median case (the 50th percentile line) takes only 18 hours to progress from 0% to 60%, but 50 hours to progress from 60% to 80%. The power-law distribution of delivery delays in Figure 8 also illustrates this phenomenon.

To explain this change in propagation speed and understand why it takes so long to reach the last 40% of participants, we examine the difference in contact numbers for receivers reached in different stages of the flooding processes. Different flooding processes have different propagation speeds. Therefore, we did not use absolute elapsed time to quantify a receiver's stage. For example, 9 h is late in a flooding process that finishes in 10 hours, but is early in another one that finishes in 100 hours. To define a metric independent of propagation speed, we order the receivers in each flooding process by their reception times and set the stage of a receiver based on its comparative order, i.e., the delivery rate upon that receiver's message reception. We aggregate same-stage receivers of all the flooding processes and show their contact number distributions in Figure 9. The X-axis shows the stages of receivers using delivery rates. The Y-axis is a node's total number of contacts during the study period. As there are multiple nodes at the same stage, we use percentile lines to show their contact number distributions.

There is a clear change in trend at the stage associated with 60% delivery rate. Below that, the contact number distributions are almost constant. Above that, the number of contacts per node decreases dramatically. This is consistent

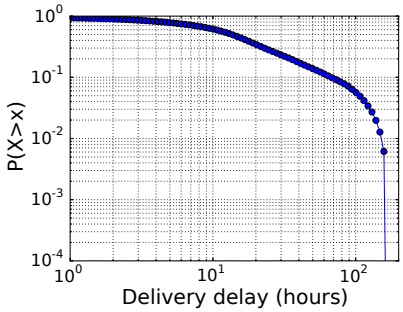

Figure 8: Message delivery delay distribution, which follows a (truncated) power law.

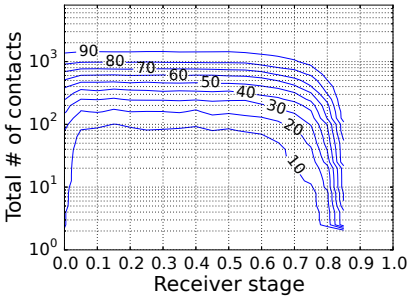

Figure 9: Contact number distributions of different-stage receivers, showing a transition at 0.6 delivery rate. A point on the X-percentile line represents the X-percentile contact number of receivers reached at the considered stage.

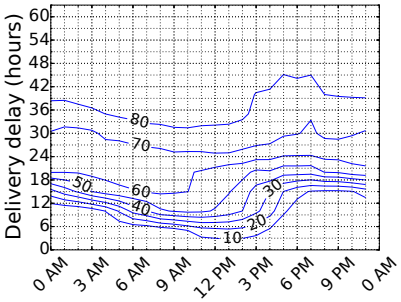

Figure 10: Delivery delays (to reach 60% of the network) of messages with different initiation hours, showing significant variations. A point on the X-percentile line represents the X-percentile delivery delay of the considered initiation hour.

with Figure 7, which shows a constant propagation speed before reaching 60% delivery rate, and a decreasing speed after that. This result suggests that diminishing contact opportunities prolong delivery delays for the last 40% nodes.

In summary, due to the participants' differences in contact opportunities, the 1am communication system has two stages in its delivery performance. It has a constant propagation speed (3 hours per 10% progress) for the first 60% of the participants, who are well connected with each other. It is increasingly difficult to reach the remaining nodes, which have fewer and fewer contacts. Overall, with a message lifetime of one week, the system has a median delivery rate of 0.85 and a median delivery delay of 13 hours.

5.3 Initiation Time

We also look into the influence of message initiation time on delivery performance. There is a significant difference between weekday and weekend initiations. Weekday initiations have a significantly faster delivery; it takes a median of 12 hours to reach 60% of the network on weekdays, compared to 30 hours on weekends. This is consistent with Figure 4 showing that there are significantly more contacts on weekdays than weekends.

There are also significant performance variations by hour-of-day, as illustrated in Figure 10. The X-axis shows the initiation hour. The Y-axis shows the delivery delay to reach 60% of the network. For example, in the median case (the 50th percentile line), a message sent at 11 a.m. only takes 10 hours to reach 60% of the network, while another sent at 8 p.m. takes 21 hours.

In general, the closer before noon a message is initiated, the shorter its delay. The later after noon a message is initiated, the longer the delay. The shortest delay occurs at around 12 p.m., and the longest delay occurs at around 7 p.m. This trend is consistent with 1am's daily contact dynamics. The majority of the contacts occur between 10 a.m. and 6 p.m. (Figure 4). Messages initiated before this period have the advantage of using all these contact opportunities. Those initiated after the start of this period have fewer contact opportunities and longer delays.

6. ROBUSTNESS

As a natural consequence of their distributed, non-hierarchical structures, DTNs are robust to blocking and censorship accomplished by removing or controlling network resources. Evaluating a network's robustness to resource removal can be used to determine its robustness to both attacks. The relationship between resource removal and blocking (denial-of-service) attacks is straightforward; an attacker might eliminate smartphones, perhaps via malware or by jamming their wireless signals at specific locations. The relationship with censorship is slightly more complex; censorship can be seen as a selective form of resource removal. Because censorship requires control over network devices, or their communication channels, a network structure robust to resource removal is also robust to censorship.

Resistance to attacks in which many network participants simultaneously and rapidly inject messages requires additional consideration. Non-hierarchical networks remain vulnerable to such denial-of-service attacks. However, resisting them is possible. Recall that 1am messages are signed using the private keys of the sending identities. It is practical for receiving mobile devices to detect when particular identities are sending messages at an undesirable rate and then cease forwarding them. An attacker might conceivably adopt a new identity and use it to launch an attack. That strategy can be defeated by assigning higher priorities to messages from nodes that have previously sent messages of interest.

Increasing the cost of surveillance is a secondary goal. 1am messages are public by design. However, there remains value in increasing the cost of reprisal-oriented network surveillance, e.g., associating messages with senders or receivers, or associating identities with individuals. The use of a non-hierarchical device-to-device network has the potential to increase the cost of reprisal-oriented surveillance because it requires a physical presence throughout a large portion of the network to observe enough traffic to associate senders and receivers. Associating identities with messages is also expensive because attackers would require a widespread physical presence, perhaps capable of wireless signal triangulation, to determine the location of the mobile device originating a particular message.

6.1 Attacks

We now explain the evaluation of 1am when subjected to two classes of resource removal attacks: the removal of participating devices and the removal of communication channels. Device removal attacks prevent specific network nodes from transmitting or relaying specific (censorship) or

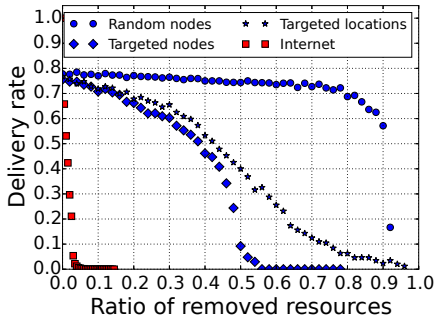

Figure 11: Degradation of message delivery rates under resource removal attacks. "Random node removal" has almost no effect while "targeted node removal" is the most effective. Under "targeted node removal", the Internet is fragmented much faster than the 1am network.

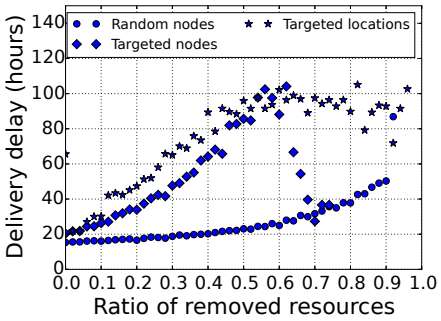

Figure 12: Degradation of message delivery delay under resource removal attacks. Again, "Random node removal" has almost no effect. Both targeted removal attacks are more effective, with "targeted location removal" resulting in faster delay increase.

all (blocking) messages, e.g., through malware deployment, physical attacks, or wireless signal jamming. Two specific strategies are considered. "Random device removal" disables devices at random, and "targeted device removal" disables devices in order of their importance. More *important devices* occur more frequently on the shortest message delivery paths.

The other class of attack disrupts physically local communications, e.g., via radio jamming on 802.11 frequencies [21]. We call this "location removal attacks" because it disables all communication within a local area, or location. We defined a *location* to be a 100 m×100 m grid element, i.e., a region with an area similar to that covered by an 802.11b transmitter. Only "targeted location removal" is discussed because "random location removal" is less effective and we believe intelligent attackers would be capable of identifying *important locations*, i.e., ones with most contact occurrences.

We simulated the effects of both classes of attacks by recalculating the message delivery performance when subject to them. In the case of device removal attacks, all contacts involving the affected devices are removed. In the case of location removal attacks, all contacts occurring within the affected locations are removed.

6.2 Attack Performance

Figures 11 and 12 illustrate how attacks affect message delivery rates and delays. The X-axis shows the ratio of removed (controlled) resources. The Y-axis in Figure 11 is the median delivery rate after a week, and Y-axis in Figure 12 is the median delivery delay.

Figure 11 shows that random device removal has almost no effect on delivery rates until more than 80% of devices are removed. Targeted removal attacks more quickly impact delivery rates. Targeted device removal and targeted location removal have similar impact with the first being slightly more effective. Targeted device removal causes delivery rate to deteriorate rapidly after 30%–40% of the most important devices are removed and the network collapses when 50% are removed. Targeted location removal does not have the same cutoff effect at the 50% removal ratio.

Figure 12 shows that random device removal has almost no effect on delivery latency until more than 60% of nodes are removed. On the other hand, both targeted removal attacks have immediate influence on delay. For example, the targeted location removal incurs around 20 hours increase in delay per 10% of removed devices. It is interesting that contrary to its impact on delivery rate, targeted location removal has a larger impact on message delivery latency than does targeted device removal. The following explanation is consistent with our observations. When a location is removed, it is likely that devices will meet in other locations to complete the delivery process with higher delay. In contrast, when a node is removed, this can terminate further distribution of messages, resulting in reduced in delivery rates, without reducing the delivery delays of the remaining messages.

6.3 Comparison to the Internet

We are interested in comparing the performance degradation of the 1am network and the Internet when subject to resource removal attacks. Albert, Jeong, and Barabasi studied the fragmentation of the Internet when its most critical routers are removed [22]. They quantity network fragmentation using the relative size of the largest cluster, which is an upper bound of the network's average delivery rate. We therefore plotted Internet fragmentation as a function of the percent of removed routers from their paper, in direct comparison with 1am's delivery rate curve in Figure 11. Figure 11 shows that when the same ratio of resources are removed, the Internet becomes much more fragmented than the 1am network. This is due to its hierarchical structure; its normal function relies on a few backbone servers. We do not claim that the 1am network is more robust than the Internet under attacks of the same scale, because it may be more difficult to eliminate or compromise 5% of resources on the Internet than on the 1am network.

6.4 Conclusion and Discussion

Our analysis shows that it is ineffective to block or censor 1am communications by randomly removing network resources, unless the majority (more than 80%) of the resources are removed. It is more effective to target the most important resources, i.e., devices with numerous contacts and locations where many contacts occur. Even using this strategy, it is necessary to eliminate a large percentage of the 1am network, relative to a hierarchical network. The Internet is fragmented when less than 5% of its most critical routers are removed, while the 1am network does not degrade rapidly until more than 30% of its devices or locations are removed. Eliminating 30% of network devices, or radio jamming 30%

Table 2: Power Model Parameters

	Meaning	Value
N	Network size	Deployment dependent
E_f	Energy cost of send+receive one 802.11b frame	26.4 mJ
f_m	Message initiation frequency	Message type dependent (see Table 5)
F_m	# of 802.11b frames per message	Message type dependent (see Table 5)
B_{frame}	# of data bits per frame	4095/2
$R_{contact}$	Avg. # of contacts per pair per hour	0.0329 (1am trace)

Table 3: Operation/State Dependent WiFi Power

P_{idle} (idle state)	210 mW
P_{high} (send/receive state)	341 mW
E_{sw} (one power state switch from low-high-low)	242 mJ
E_{ping} (send+receive one 802.11b frame (ping))	2.16 mJ
E_{send} (send one 802.11b frame (UDP))	19.6 mJ
E_{recv} (receive one 802.11b frame (UDP))	6.8 mJ

of the public areas in a town or campus is likely to be an expensive undertaking for an attacker.

7. ENERGY MODEL

This section describes an energy model for 1am running on commodity smartphones and describes the relationship between battery depletion and network scale. We first develop an energy model for the WiFi component in 802.11 ad hoc mode based on measurements, then elaborate on the three contributors to energy consumption for 1am: contact discovery, communication, and computation. For the convenience of the readers, Table 2 lists a few parameters frequently used throughout this section.

We use a Monsoon power monitor [23] to measure power consumption. The devices used for power modeling are rooted Samsung Galaxy Nexus smartphones running CyanogenMod 10.2.1-toro (Android 4.3.1). The wireless chip of these devices is Broadcom BCM4330. The CPU is a Dual-Core 1.2 GHz ARM Cortex-A9.

7.1 WiFi Component in 802.11 Ad hoc Mode

We measure the smartphone's power consumption for different packet transmission frequencies. The experimental setup follows. Two devices are put into 802.11b ad hoc mode and a 2 Mbps communication channel is established. One device uses the "ping" program to send periodic ICMP packets, which are padded into maximum 802.11b frames. We measure the power consumption of the other device, which is placed in sleep mode with only the WiFi component awake. The ping frequency (f_{ping}) is varied from 0.1 Hz to 100 Hz.

We observe clear changes in wireless interface power management states as f_{ping} changes. When $f_{ping} < 0.5$ Hz, the WiFi component switches on and off for each frame and consumes significant energy, as the energy cost for one power state switch is much larger than that for one send/receive operation. When $f_{ping} > 1$ Hz, the WiFi component stays in the high power state. From 0.5 Hz to 1 Hz, the frequency of power state switches gradually decreases, and we see a decreasing power consumption despite the increase in ping frequency. Based on this result, we use a conservative two-stage model for the WiFi component. Send/receive operations separated by more than one second cause power state switches. In other words, the device stays in the high power state until there is a gap between operations longer than one second.

Table 3 summarizes the measured energy numbers for different power states/operations. Note that we also measured the energy cost of sending/receiving one (maximum) 802.11b frame through an application program operating UDP sockets (E_{send} and E_{recv}), as these numbers are more relevant for application energy modeling.

7.2 Ongoing Contact Discovery

In order to avoid relying on centralized control and scheduling, we used Zheng's and Hou's asynchronous activation algorithm [24] to check for contact opportunities periodically. The activation pattern within each period is deliberately designed so that two unsynchronized devices have overlapping activation times. The authors showed that it is possible to achieve a 9/73 duty cycle [24]. Therefore, the hourly energy cost for contact discovery is $E_{base} = P_{idle} \cdot 3,600\,\text{seconds} \cdot \frac{9}{73}$, where P_{idle} is the idle power consumption for staying awake in 802.11 ad hoc mode, the minimal power state in which devices could discover each other.

7.3 Communication

Communication happens upon each contact, causing the following sequence of operations. The two devices first switch into the high power state, then exchange Bloom filters to learn about each other's received and not-yet-received messages, then deliver the not-yet-received messages, and finally switch back into the low-power state. The communication energy consumption is therefore the sum of the Bloom filter exchange (we also call it metadata exchange) and message delivery energy consumptions.

7.3.1 Message Delivery Cost

All messages are flooded to the whole network, so the number of messages received per node per hour is the total number of messages generated by the network per hour, Nf_mF_m. f_m is the message initiation frequency (in 1/hour) and N is the number of nodes in the network. F_m is the number of 802.11b frames each message consumes (see Table 5 for typical message sizes). Therefore, the hourly energy cost for message delivery is $E_{msg} = E_f(Nf_mF_m)$. E_f is the energy cost of sending and receiving one maximum size 802.11b frame, $E_f = E_{send} + E_{recv}$.

7.3.2 Metadata Exchange Cost

The metadata of each node's received messages within the preceeding seven days are recorded in Bloom filters—recall that 1am's message lifetime is one week. We now derive the size of each node's Bloom filter. Each node receives in total $Nf_m \cdot 12\,\text{hours} \cdot 7\,\text{days} \cdot F_m$ messages per week, as we assume 12 hours of active time per day for initiating messages. Each message requires 14.4 bits in the Bloom filter (recall Section 3.2), so the size of each node's Bloom filter (measured in 802.11b frames) is $\frac{14.4\,\text{b} \cdot (Nf_m \cdot 12\,\text{h} \cdot 7\,\text{d} \cdot F_m)}{B_{frame}}$, where B_{frame} is the maximum number of data bits carried by one 802.11b frame.

As each node's Bloom filter is exchanged upon each contact, the energy cost per contact is $E_f \frac{14.4\,\text{b} \cdot (Nf_m \cdot 84\,h \cdot F_m)}{B_{frame}} + E_{sw}$. The extra term E_{sw} is the energy cost of switching on and off the WiFi component to enable communication (see Table 3). On the other hand, each node's average contacts per hour is $NR_{contact}$, as $R_{contact}$ is the average number of contacts per pair per hour. Therefore, each node's hourly cost for

Table 4: Energy Cost of 1am Computations

	Execution time for 5,000 runs (ms)	Energy cost per run (mJ)
Signature (OpenSSL)	741.9 [25]	0.1
Verification (OpenSSL)	3819.7 [25]	0.6
Verification (SpongeCastle)	1,572,347	239
Others	65,776	10.0

metadata exchange is

$$E_{md} = NR_{contact}\left(E_f \frac{14.4\,\mathrm{b}\cdot(Nf_m\cdot 84\,\mathrm{h}\cdot F_m)}{B_{frame}} + E_{sw}\right).$$

Overall, the hourly communication energy cost per node is $E_{comm} = E_{msg} + E_{md}$, which scales with N^2 as E_{md} scales with N^2 and E_{msg} scales with N.

7.4 Computation

Each node's hourly energy cost for computation is $E_{comp} = Nf_m E_{mcomp}$, where Nf_m is the number of 1am messages received per node per hour, and E_{mcomp} is the energy cost of processing one message, which includes the cost of signing, verification, and all other operations, e.g., serialization/deserialization and database queries and insertions[2].

Table 4 lists the computation time and energy cost for various message handling operations[3]. For digital signature and verification, 1am uses ECDSA with nistp256 for their space-friendly public keys [26]. Unfortunately Android's default cryptography library, SpongeCastle [27], is much less energy efficient (2,000× less) than an unsupported native library, OpenSSL [28] (see Table 4). Because it is non-trivial to port OpenSSL to Android and measure its performance, we used a previous paper's reported computation time for the considered algorithm parameters, CPU model, and OS [25]. In summary, $E_{mcomp} = 0.1\,\mathrm{mJ} + 0.6\,\mathrm{mJ} + F_m \cdot 10.0\,\mathrm{mJ}$, for signing, verification, and other operations, respectively. The cost for all operations except signing and verification is proportional to the message size, as reflected in the last term of E_{mcomp}.

Summing the cost of contact discovery, communication, and computation yields 1am's total hourly energy cost

$$E = E_{base} + E_{comm} + E_{comp}, \quad (1)$$

which scales in $\mathcal{O}(N^2)$, as E_{base} is constant, E_{comp} is $\mathcal{O}(N)$, and E_{comm} is $\mathcal{O}(N^2)$. Note that this model assumes an uniform pairwise contact rate ($R_{contact}$) independent of the network size (N). In reality, $R_{contact}$ may decrease as N increases, offsetting the $\mathcal{O}(N^2)$ trend, because an increasing number of nodes may never meet as the network becomes larger and more geographically distributed. Nevertheless, this model produces an upper bound on energy consumption that is fairly tight for small local communities with sufficient mixing, e.g., a university campus.

7.5 Energy Cost for Multimedia Support

Using the energy model in Equation 1, we study 1am's energy overhead for various network sizes (N) and message types (text, image, and video). N is varied from 100 to 50,000,

[2]In fact signing is only conducted once by the message initiator while every receiver verifies. We simplified the model by assuming that signing and verification occur the same number of times, leading to an upper bound cost estimation.

[3]The energy cost is the product of the computation time and P_{CPU}, the smartphone's power level with CPU being awake, which is 760 mW according to our measurement.

Table 5: Message Sizes (F_m) and Initiation Frequencies (f_m)

Type	Message size	F_m (frames/msg.)	f_m (msg./hour)
Text	514 characters	1	0.16667
Image	100 KB	44	0.02536
Video clip	1 Mbps × 240 seconds	12000	0.00054

Table 6: Hourly Energy Consumptions for Different Network Sizes and Message Types

N (network size)	Text		Image		Video	
	J	%	J	%	J	%
100	95	0.41	99	0.42	122	0.52
400	101	0.43	124	0.53	257	1.10
700	108	0.46	162	0.70	467	2.00
1,500	138	0.59	327	1.40	1K	5.96
2,000	164	0.70	476	2.04	2K	9.59
5,000	430	1.84	2K	9.08	12K	50.0
50,000	27K	117.6	181K	775.2	1048K	4491.6

Energy percentages are for a 6.48 watt-hour battery.

which correspond to 0.2%–100% adoption rates among students and faculty on a university campus. Table 5 shows the sizes (measured in 802.11b frames) and message initiation frequencies (f_m) of different multimedia messages. Each 1am text message contains at most 514 characters, so one frame per message is sufficient. The text initiation frequency is based on Twitter's average rate, around 2 per user per day [29]. We consider images of sizes typical for microblogging applications, 100 KB. The image initiation frequency is based on Facebook users' photo posting statistics [30]. We consider 352×288 video encoded at 1 Mbps, i.e., high-quality MPEG, of 4 minutes duration, the average video length on the Internet [31]. The video initiation frequency is based on YouTube's statistics [32].

Table 6 shows the hourly energy consumption of different scenarios. We report both in Joules and the percent of energy in a 6.48 Wh battery, a commodity lithium-ion battery widely used in smartphones. We consider the maximum acceptable daily energy consumption to be 25% of the battery, i.e., ∼2.1% per hour assuming 12 hours of active time per day. The shaded cells in Table 6 shows the cases we believe would impose unacceptable battery charging burdens on users.

Table 6 shows that it is practical to flood text and image messages in networks with up to 5,000 and 2,000 nodes, respectively. However, video messages can only be supported in networks with fewer than 700 nodes, despite a much smaller initiation frequency. In an extreme case when video messages are initiated as frequently as text messages, the battery will be depleted in less than 3 hours even in a 100-node network. If both text and image messages are supported, it is practical to have networks with 1,500 nodes. If all three types are supported, the network is limited to 400 nodes.

Metadata exchange ($E_{md} \propto N^2$) is the main cause of energy consumption in large networks. This result suggests that simple energy models that ignore metadata exchange and only consider data transmission [13, 14] are inaccurate and possibly misleading. We note that the key to designing more energy-efficient DTN message distribution protocols is to offset the $\mathcal{O}(N^2)$ trend in energy consumption without greatly degrading performance.

8. CONCLUSIONS

Thanks to their structure, DTNs composed of commodity smartphones have the potential to resist blocking and

censorship, and make surveillance expensive. Such networks are capable of supporting text microblogging applications without much reducing participant smartphones' battery lifespan for towns with populations of 50,000 and adoption rates below 5%. Supporting frequent image or video distribution without new dissemation ideas is impractical at similar network scales. The network was able to deliver messages to most participants within a day with only a 0.2% adoption rate in a college town with a population of 50,000. It degrades more gracefully than a hierarchical infrastructure network when subjected to attacks that eliminate participants or geographic locations, suggesting robustness to blocking and censorship. The 1am traces are publicly available at http://ziyang.eecs.umich.edu/projects/whisper/traces.html.

There is evidence of substantial demand for communication applications that are robust to blocking, censorship, and surveillance but developing such applications requires an understanding of how design decisions influence performance, security, and energy consumption properties. This work takes a first step toward this understanding.

9. ACKNOWLEDGMENTS

The authors would like to thank Rongrong Tao and Junzhe Zhang for helping implementing prototypes and performing experiments, and Paul Myers for proofreading the paper. This work was supported in part by the US National Science Foundation (NSF) under Award TC-0964545 and in part by the University of Michigan.

10. REFERENCES

[1] P. N. Howard, A. Duffy, D. Freelon, M. Hussain, W. Mari, and M. Mazaid, "Opening closed regimes: What was the role of social media during the Arab Spring," Project on Information Technology & Political Islam, Jan. 2011.

[2] J. Cook, "Hong Kong protesters are communicating via a mobile app that doesn't actually use the Internet," *businessinsider.com*, Sept. 2014.

[3] B. Ries, "Twitter blocks pro-Ukrainian political account for Russian users," *Mashable.com*, May 2014.

[4] "1am: Censorship-resistant microblogging," 2013, http://1am-networks.org.

[5] A. Al-Akkad, C. Raffelsberger, A. Boden, L. Ramirez, and A. Zimmermann, "Tweeting when online is off? Opportunistically creating mobile ad-hoc networks in response to disrupted infrastructure," in *Proc. Int. Conf. Information Systems for Crisis Response and Management*, May 2014, pp. 662–671.

[6] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, and C. Rohner, "Twitter in disaster mode: Opportunistic communication and distribution of sensor data in emergencies," in *Proc. Extreme Conf. on Communication: The Amazon Expedition*, Sept. 2011, pp. 1:1–1:6.

[7] G. Fanti, Y. Ben David, S. Benthall, E. Brewer, and S. Shenker, "Rangzen: Circumventing government-imposed communication blackouts," University of California, Berkeley, Tech. Rep. UCB/EECS-2013-128, July 2013.

[8] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay tolerant networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[9] L. Song and D. Kotz, "Evaluating opportunistic routing protocols with large realistic contact traces," in *Proc. Wkshp. Challenged Networks*, Sept. 2007, pp. 35–42.

[10] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," *Computer Networks*, vol. 52, no. 14, pp. 2690–2712, 2008.

[11] M. McNett and G. M. Voelker, "Access and mobility of wireless PDA users," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 40–55, 2005.

[12] A. Socievole and S. Marano, "Evaluating the impact of energy consumption on routing performance in delay tolerant networks," in *Proc. Int. Conf. Wireless Communications and Mobile Computing*, Aug. 2012, pp. 481–486.

[13] Y. Li, Y. Jiang, D. Jin, L. Su, L. Zeng, and D. O. Wu, "Energy-efficient optimal opportunistic forwarding for delay-tolerant networks," *IEEE Trans. Vehicular Technology*, vol. 59, no. 9, pp. 4500–4512, 2010.

[14] X. Lu and P. Hui, "An energy-efficient n-epidemic routing protocol for delay tolerant networks," in *Proc. Int. Conf. Networking, Architecture and Storage*, July 2010, pp. 341–347.

[15] D. R. Bild, Y. Liu, R. P. Dick, Z. M. Mao, and D. S. Wallach, "Aggregate characterization of user behavior in Twitter and analysis of the retweet graph," *ACM Trans. Internet Technologies*, accepted, in press.

[16] F. Putze, P. Sanders, and J. Singler, "Cache-, hash-, and space-efficient Bloom filters," *J. of Experimental Algorithms*, no. 4, pp. 108–121, 2007.

[17] S. Trifunovic, B. Distl, D. Schatzmann, and F. Legendre, "WiFi-Opp: ad-hoc-less opportunistic networking," in *Proc. Wkshp. Challenged Networks*, Sept. 2011, pp. 37–42.

[18] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.

[19] "Manes: a mobile ad hoc network emulation system," 2013, http://ziyang.eecs.umich.edu/projects/whisper/manes/.

[20] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proc. Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2011, pp. 1100–1108.

[21] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa, "Performance of IEEE 802.11 under jamming," *Mobile Networks and Applications*, vol. 18, no. 5, pp. 678–696, 2013.

[22] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[23] Monsoon, "Monsoon power monitor," 2008, http://www.msoon.com/LabEquipment/PowerMonitor/.

[24] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. Int. Symp. Mobile Ad Hoc Networking and Computing*, June 2003, pp. 35–45.

[25] A. M. Braga and E. N. Nascimento, "Portability evaluation of cryptographic libraries on Android smartphones," in *Proc. Int. Symp. Cyberspace Safety and Security*, Dec. 2012, pp. 459–469.

[26] J. Lopéz and R. Dahab, "An overview of elliptic curve cryptography," Institute of Computing, State University of Campinas, Tech. Rep. IC–00-10, 2000.

[27] "Sponge Castle," 2014, http://rtyley.github.io/spongycastle/.

[28] "OpenSSL," 2014, https://www.openssl.org/.

[29] "Twitter usage," 2015, https://about.twitter.com/company.

[30] "A focus on efficiency, a whitepaper from Facebook, Ericsson and Qualcomm," internet.org, Sept. 2013.

[31] ComScore, "ComScore releases December 2013 U.S. online video rankings," *conscore.com*, Jan. 2014.

[32] "YouTube statistics," 2014, https://www.youtube.com/yt/press/statistics.html.