

Socio-Technological Testbed for Evaluation of Combined Social and Communication Networks

Zhuo Lu

Department of Computer Science
University of Memphis, TN, 38152
Email: zhuo.lu@memphis.edu

Yalin Sagduyu and Yi Shi

Intelligent Automation Inc.
Rockville MD, 20855
Emails: {ysagduyu, yshi}@i-a-i.com

Abstract—A socio-technological network can be viewed as a hybrid network that combines heterogeneous communication and social links. Although there have been an increasing number of efforts to study the design and performance of socio-technological networks via theoretical analysis or simulations, little attention has been focused on conducting field or testbed experiments due to the heterogeneity and complexity of these interdependent network structures. To fill this gap, we propose and implement a test and evaluation platform to validate the design and evaluate the performance of combined social and communication networks. The testbed consists of programmable WiFi radios that represent users (according to real social network data) communicating over a multi-hop wireless channel emulator. We run a variety of network applications over the testbed, including information dissemination and social-aware routing. Experimental results show that combined design of social and communication networks can substantially improve the performance (e.g., message delay and delivery ratio) in real-world scenarios. Motivated by these results, our testbed platform provides a high fidelity environment for design, implementation, test, and evaluation of combined social and communication networks.

Index Terms—Social networks; communication networks; socio-technological networks; heterogeneous network testbed; emulation.

I. INTRODUCTION

A novel architecture in form of a *socio-technological network* [1]–[3] has emerged to characterize today’s hybrid network structure, in which nodes have both communication links and social connections to each other. A socio-technological network can be viewed as an integration of communication and social networks, where users can send data to each other via either communication or social links. In particular, the communication link is a short-range link for a node to communicate with neighbors; and the social link is not limited within physical proximity, but depends on a node’s relationship to others. Therefore, when nodes move around in a socio-technological network, their communication links change consistently over time; however, their social links remain unchanged over short periods of time.

In a socio-technological network, a social link can be viewed as a high-level communication channel between two

nodes [2], [4]. Information transmitted over social links will eventually go through physical communication medium. For example, a mobile user can use his smart phone to send messages to a friend. The cellular network infrastructure provides a fundamental physical communication medium for the social link between the mobile user and his friend. In addition, the user can also use his phone to communicate with nearby users over Bluetooth or WiFi. Such short-range wireless communication links combined with the cellular infrastructure enabled social links constitute a socio-technological network architecture. In a socio-technological network with military applications, communication links are based on short-range military radios, and social links represent a more generalized relationship between communication parties (e.g., task units and commanders, report and coordination for different missions). The underlying physical channel for such social links can be airborne networks or long-distance (e.g., satellite) military communications. Furthermore, social and communication links may be coupled with each other. For example, socially close friends may be also geographically close.

Existing research efforts in socio-technology networks typically use theoretical modeling or simulations to study information dissemination, routing and their performance evaluation in socio-technological design [1], [2], [4], [5]. Little attention has been focused on conducting field or testbed experiments on the performance of socio-technological networks due to the heterogeneity and complexity of such networks. To fill this gap, in this paper, *we propose and implement a testbed platform to validate the design and evaluate the performance of socio-technological networks*. To the best of our knowledge, the testbed system is the first full-fledged experimental platform for socio-technological network design and evaluation.

In the implemented testbed, network applications run in programmable RouterStation Pros [6] (i.e., embedded radios running over Linux) that represent network nodes. All nodes are equipped with WiFi radios connected to a high-fidelity wireless channel emulator, called RFnest [7], which can emulate arbitrary multi-hop wireless channels by controllable attenuation of realistic radio frequency (RF) signals in a lab environment. This way, the WiFi radios under RFnest’s topology control form all possible communication links in the network. In addition, Ethernet is used to connect all nodes with a social network server to emulate the social links. The

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under Contract FA9550-12-C-0037. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR.

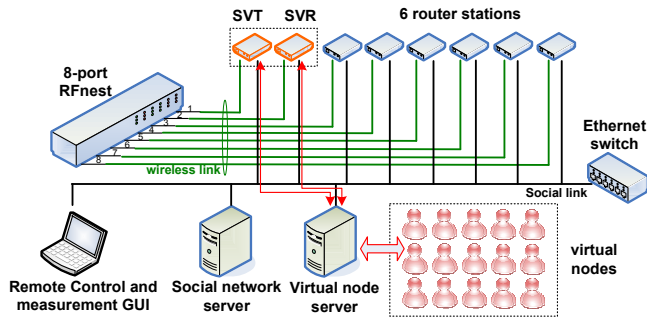


Fig. 1. System architecture of socio-technological testbed.

social network server controls the social relationships between all nodes, and also emulates the delay and failure effects for social links that are potentially coupled with communication link properties.

A variety of network applications were tested over the testbed, including information dissemination and social-aware routing. Experimental results show that the joint design based on social and communication networks can substantially improve the performance in terms of message delay and delivery ratio in real-world scenarios. The testbed supports combined architecture of social and communication links with the use of real-world social network data and emulated wireless channel data in a common platform. In this sense, the capabilities offered by the testbed go beyond simulation studies, and present realistic evaluation with real radio and data configurations.

The remainder of this paper is organized as follows. In Section II, we introduce the design procedure of the testbed. In Sections III and IV, we present the experimental setups and results, respectively. We conclude the paper in Section V.

II. SYSTEM DESIGN

In this section, we present the testbed system architecture and setups, then describe key components in the testbed.

A. System Architecture

The testbed consists of eight RouterStation Pros, an Ethernet switch, a social network server, a virtual node server that supports virtual nodes in the network, and the wireless channel emulator, RFnest. In addition, a graphical user interface (GUI) is designed to send control commands and receive measurements from the testbed.

The architecture of the testbed is shown in Fig. 1. Each RouterStation Pro is a network node with WiFi radio serving as the communication link interface. The WiFi interfaces of all nodes are connected via RF cables to RFnest that emulates realistic multi-hop wireless connectivity by representing different channel characteristics. Besides WiFi connections, an Ethernet switch is used to connect nodes such that they can communicate with each other via Ethernet, which serves as the social link interface. Note that Ethernet is used in the testbed to emulate the social link. In practice, the social link can be based on any type of long-distance network connections, such as cellular connections for commercial users or satellite

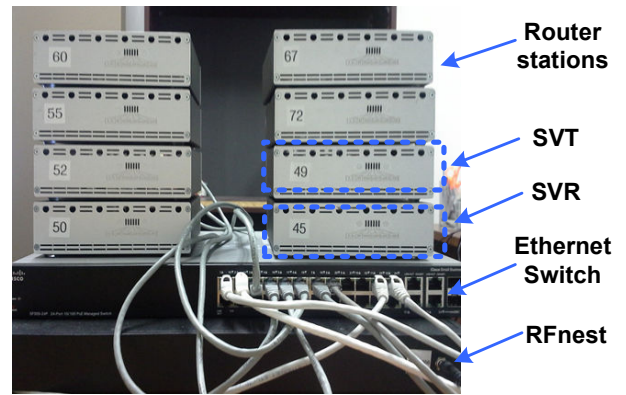


Fig. 2. Picture of the testbed system.

connections for military applications. All social relationships are stored at a social network server that can accommodate and replay any realistic social network data.

In the testbed, there are eight RouterStation Pros that can represent real network nodes. RFnest provides the capability to scale up the experimentation environment by introducing virtual nodes that can seamlessly transmit and receive from actual radios. In order to support experiments with more nodes, we delegate two of the router stations as surrogate virtual transmitter (SVT) and surrogate virtual receiver (SVR) as for virtual nodes running in a virtual node server. As shown in Fig. 1, two router stations are used as SVT and SVR, which transmit and receive packets for virtual nodes in a virtual node server, respectively. This way, real and virtual nodes operate seamlessly without knowing the node type and exchange physical radio signals with each other.

Fig. 2 shows the picture of the entire testbed system, including network nodes, SVT/SVR, the Ethernet switch and RFnest. Each of these key components for the testbed will be described in Section II-B.

B. Key Components

Next, we describe each key component in the testbed as shown in Fig. 2.

1) *Network Nodes*: The RouterStation Pros that represent network nodes in the testbed are products by Ubiquiti Networks [6], featuring a fast 680MHz CPU, 64MB RAM, and 16MB Flash. All nodes run over OpenWrt [8], which is a Linux-based operating system for embedded devices. Each node is equipped with WiFi (IEEE 802.11b/g) radio and Ethernet interfaces.

An RF cable is used to connect each node's WiFi radio output to RFnest such that all nodes operate in a multi-hop wireless environment emulated by RFnest. In addition, an Ethernet cable is used to connect each node's Ethernet interface to a switch. Although all nodes are connected to the same switch, they can only communicate with each other via Ethernet if they have a social connection specified by the social network server. The WiFi and Ethernet connections serve as the communication and social links, respectively.

2) *RFnest*: The network channel emulator, RFnest [7], combines the advantages of low-cost and flexible emulation environment and realistic hardware performance to provide hardware-in-the-loop wireless evaluation capability. RFnest allows real wireless networks to send signals over an emulated channel and scales up the evaluation by seamlessly integrating virtual nodes into the scenario with actual signal-level interactions between the real and virtual nodes [7].

RFnest has two different versions: analog and digital versions. The analog version features multi-hop mobile wireless topology control via signal attenuation on realistic RF signals. In addition, multi-access, interference, and broadcast channel configurations are realistically supported with signal-level interactions. The digital version provides functions such as fading, propagation delay and Doppler effects. The testbed currently uses the analog version of RFnest and can be further adapted to the digital version. The advanced functions in the digital version will empower the testbed with the capability of offering a more versatile emulation of realistic multi-hop wireless connectivity, including propagation delay, multipath and Doppler effects.

3) *Virtual Nodes and SVT/SVR*: The analog RFnest in the testbed has eight RF ports, meaning that the network emulation can only accommodate up to eight real nodes. In order to support more nodes, RFnest enables a *virtual node mechanism* feature (more details can be found in [7]). Under the virtual node mechanism, multiple instances of nodes can be run at the same server. Each instance corresponds to a virtual node. The virtual mechanism is implemented as a transparent packet capturing and scheduling module at the Linux kernel. Therefore, an application running inside a virtual node is not aware of the virtual node mechanism, and assumes the role of a real node to transmit and receive packets.

When a virtual node transmits a packet, the kernel module first captures the packet, then automatically updates the corresponding channel gains in RFnest and sends the packet using the SVT’s WiFi radio. If the receiver is also a virtual node, the packet is sent to the SVR’s WiFi radio, and then delivered to the virtual node at the virtual node server. This represents the way virtual and real nodes communicate via wireless links. Message delivery over social links works in a similar way but using Ethernet instead of WiFi.

4) *Social Network Server*: A social-network server maintains social relationships of all nodes (including real and virtual ones). Social distance is defined in the social-network server. When a node wants to send a packet to the other node, it can send the packet via the social link if the two nodes have a social relationship. In the testbed, a node will not send a packet directly to the destination via a social link, it will always send the packet to the social network server, which will examine and validate the social relationship, then forward the packet to the destination. The process is similar to sending a message in Facebook: a user’s message is always forwarded by the Facebook server to the other user.

A user can send queries to the social network server to add, list, delete friends as well as update social ties. The social

network server is designed to accommodate realistic social network datasets, and also has the capability of the social link delay and failure emulations.

C. Summary

The testbed features an open and scalable architecture. It is a fast, efficient, and high-fidelity testbed for general socio-technological network design and applications. The virtual node mechanism can scale the network size up to 15 virtual nodes (over the analog RFnest) with further extension to over 100 virtual nodes (over the digital RFnest).

III. EXPERIMENTS AND SETUPS

In this section, we demonstrate how the testbed supports deployment and performance evaluation of socio-technological network applications. We first present the experimental setups, then discuss the social network datasets, and finally introduce the performance metrics.

A. Network and Routing Setups

The experimental network in the testbed consists of 21 nodes, in which 6 nodes are real nodes and 15 nodes are virtual nodes (running in a virtual node server). RFnest was connected to the 6 real nodes, one SVT and one SVR to emulate realistic multi-hop wireless channels.

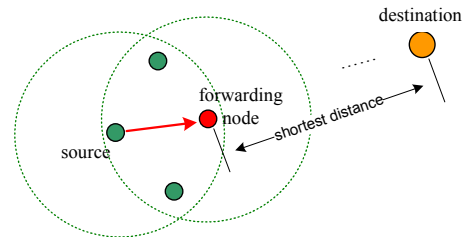


Fig. 3. Greedy routing in socio-technological networks.

The performance of information dissemination in socio-technological networks was evaluated in the testbed under the greedy routing protocol [2], [5] that operates with local information only. The procedure is illustrated in Fig. 3: if the source has a message to send to the destination, it first examines its neighbors via both social and communication links. Then, it chooses the neighbor who is closest (i.e., has the shortest physical distance) to the destination as the forwarding node. It sends the message to the forwarding node via the corresponding social or communication link. In summary, messages are gradually moved closer to the destination.

B. Social Network Setups

The Reality Mining dataset [9] was adopted to build the social network during the experiments. The Reality Mining project was conducted from 2004-2005 at the MIT Media Laboratory. The Reality Mining study followed 94 subjects (included students and faculty in a research institution) using mobile phones pre-installed with a mobile application that recorded the data about call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status. The

data measurements lasted nine months for all subjects in the experiments. The dataset also collected self-report relational data from each individual, where subjects were asked about their proximity to, and friendship (social connection) with, others.

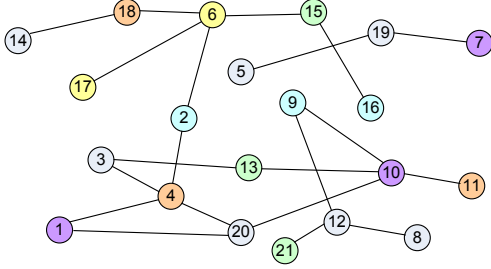


Fig. 4. The social connections of 21 nodes from Reality Mining dataset.

We used the data of 21 out of 94 individuals to build the social network for the testbed. The social connections between the 21 nodes are shown in Fig. 4. The social network server stored all the relationships in Fig. 4 as the social links in our experiments.

C. Control Panel

A comprehensive control GUI was developed to set control parameters to the testbed. As shown in Fig. 5, the GUI features a variety of control functions for the testbed.

- *Source-destination.* The testbed supports an arbitrary number of source-destination pairs (e.g., 5 pairs in Fig. 5).
- *Mobility.* The testbed can adopt any mobility trace (including the Reality Mining dataset) to move the nodes around in the network. When a node moves, its signal attenuation to other nodes will be updated in RFnest. We can also adjust the mobility speed during the emulation.
- *Map size.* The testbed supports random network topology, and can control the map size or switch between random and grid (i.e., nodes are placed with equal space) maps.
- *Social link failure and delay.* Social links are based on Ethernet communication that is reliable in nature. The testbed adopts a failure mechanism that can intentionally drop or delay messages over social links to emulate any type of realistic network channels, such as satellite or cellular links.
- *Routing selection.* Different social-aware routing protocols can be designed and integrated into the testbed.

With all these flexible control functions, it becomes convenient to evaluate the performance and validate the design of socio-technological networks in realistic environments with controllable and repeatable conditions.

D. Performance Metrics

Two performance metrics, success probability and delivery delay, are defined to evaluate the performance of socio-technological networks in our experiments.

- The success probability is the probability that a message can be finally delivered to the destination.

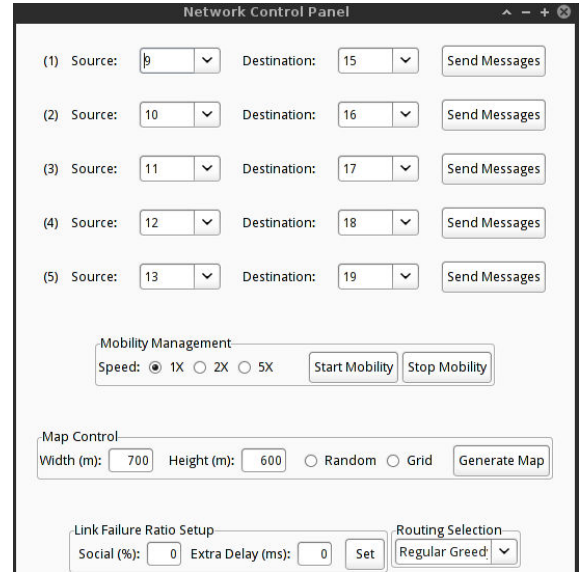


Fig. 5. The control panel for the testbed.

- The delivery delay is the end-to-end delay of a message that travels from the source to the destination.

For each experiment, both success probability and average delivery delay are measured in the network.

IV. EXPERIMENTAL RESULTS

Next, we describe the results and observations from socio-technological experiments on the testbed. We uniformly distributed the 21 nodes over a 700m by 600m map. Each node was set to transmit to a randomly selected destination at the rate of 10 messages per second under greedy routing.

To evaluate the network performance under different conditions, during the experiments, we manually changed the social link failure probability in the control panel shown in Fig. 5. We also adjusted the WiFi transmission power to measure the communication link failure probability.

A. Information Dissemination

We first measure the performance of information dissemination in a social-technological network under greedy routing.

Fig. 6 illustrates how the 21 nodes with IDs n3-n23 (n1 and n2 are SVT and SVR respectively) forward messages in the socio-technological network. As shown in Fig. 6, nodes choose either social or communication link to forward a message. The communication link is based on WiFi radio, therefore is only used for short-range forwarding. The social link is used for both short-range and long-range forwarding because it can be based on cellular or satellite backbone in practice. For example, in Fig. 6, n6 is sending messages to n18 by first forwarding messages to n5 via a social link, then to n18 via a communication link.

Then, we measure the success probability and delivery delay. We are interested in the performance in terms of success probability and delivery delay as a function of the distance between a source and a destination. We use the



Fig. 6. How nodes use social and communication links to send messages.

metric of hop distance as the distance metric between two nodes [2]. In particular, we define the hop distance $h(i, j)$ between source i and destination j as the ceil of their physical distance $d(i, j)$ divided by the WiFi transmission range r ; i.e., $h(i, j) = \lceil d(i, j)/r \rceil$. Thus, it is evident that the larger the hop distance $h(i, j)$, the farther node i is from node j .

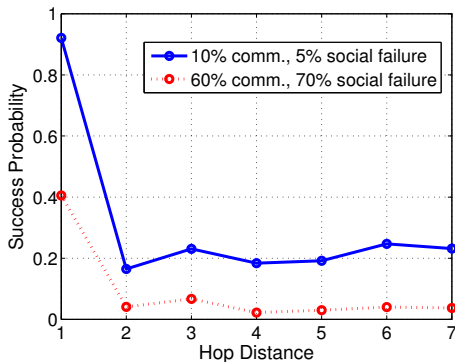


Fig. 7. Success probability vs. hop distance.

Fig. 7 shows the delivery success probability versus the hop distance for different communication and social link failure ratios. We can see from Fig. 7 that as the hop distance increases, the success probability first sharply decreases then stabilizes. In addition, if the social link is more reliable with lower failure ratio, the success probability is also improved. For example, when the social link failure ratio is 5% with 10% communication link failure as shown in Fig. 7, the success probability remains at around 20% regardless of the physical distance between the source and the destination. This also indicates that socio-technological networks have advantages for information dissemination to distant nodes over conventional multi-hop wireless ad-hoc networks.

Fig. 8 shows the average delivery delay as a function of hop distance with different communication and social link failure ratios. It is observed from Fig. 8 that the delivery delay does not increase proportionally with hop distance increasing. In particular, as hop distance increases, the delay first increases,

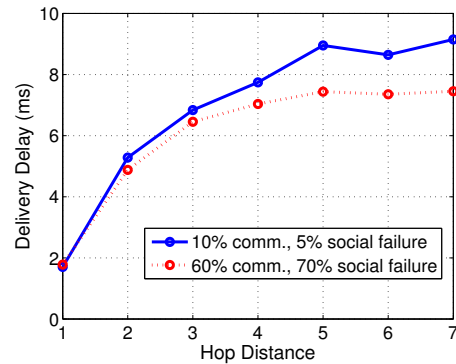


Fig. 8. Delivery delay vs. hop distance.

but starts to converge when hop distance becomes 5. This is because in socio-technological networks, the social link is capable of forwarding a message from a node to the other with physical distance larger than the range of the point-to-point wireless link. Therefore, a node can always have a chance to find a social link (i.e., the small world property [5]) for message forwarding that reduces the hop distance larger than 1, leading to a bounded average delivery delay shown in Fig. 8.

B. Persistent Transmission for Reliability

Social and network link failures can significantly reduce the success probability in a socio-technological network. Persistent transmission is one effective way to improve the link success ratio. In the persistent transmission strategy, a node keeps transmitting a message to its neighbor until the message transmission is successful or after a number of pre-defined transmissions (like the behavior of TCP). We implemented the persistent transmission strategy at each node to see how it improves the success probability during the experiments.

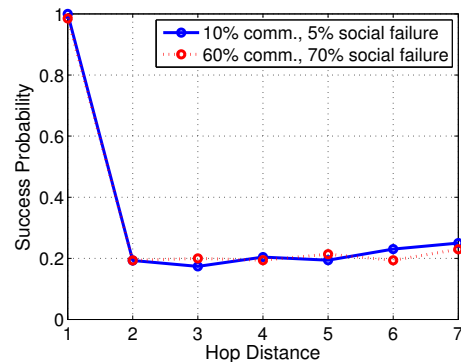


Fig. 9. Success probability vs. hop distance under persistent transmission.

Fig. 9 shows the success probability versus the hop distance for different communication and social link failure ratios under persistent transmission. It is noted from Fig. 9 that such a persistent transmission scheme can eliminate the effect of both communication and social link failures. For the case of 60% communication and 70% social link failures, the success probability is substantially improved compared with that shown

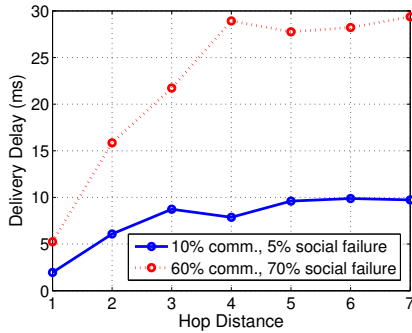


Fig. 10. Delivery delay vs. hop distance under persistent transmission.

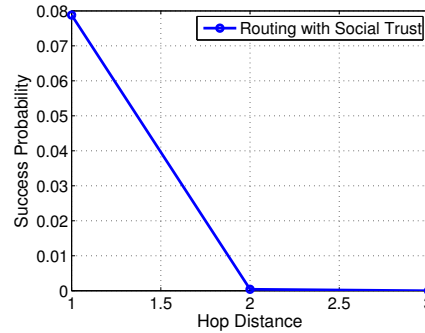


Fig. 11. Success probability vs. hop distance with social trust.

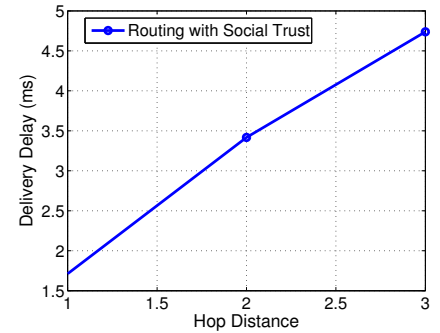


Fig. 12. Delivery delay vs. hop distance with social trust.

in Fig. 7. We note that although persistent transmission can eliminate the link failure effect and reliably deliver a message from one node to the other, there is still a large chance (around 80% as indicated in Fig. 9) that a message is not successfully delivered to its destination because of the low node density (21 nodes on a 700m by 600m region) in our experiments. In other words, a node cannot always find a forwarding node in the experiments.

Persistent transmission does come with a cost, which is the delay performance degradation due to the persistent strategy of retransmitting the same message. Fig. 10 demonstrates the average delivery delay versus the hop distance with different communication and social link failure ratios under persistent transmission. From Fig. 10, we observe that the delay increases with more failure ratio. For example, in the 60% communication and 70% social link failure case, the delivery delay at the hop distance of 7 increases from 7.37 ms (as shown in Fig. 8) to 29.52 ms (as shown in Fig. 10) under persistent transmission. Therefore, there exists a trade-off between the success probability improvement and the delay degradation in the persistent transmission scheme.

Figs. 9 and 10 illustrate that persistent transmission for socio-technological networks can be flexibly used to balance a tradeoff between delivery reliability and delay performance.

C. Routing via Communication Link with Social Trust

We then evaluate the performance of routing via the communication link with social trust. Social trust based routing has been proposed in many protocols [10]–[12]. In social trust based routing, a node will forward a message to another node via the communication link only if the two nodes have a social relationship (i.e., they have built a social trust) in the network. Therefore, nodes can use only communication links to forward messages. Social links are used for routing decision making.

Fig. 11 shows the experimental results of success probability versus hop distance under greedy routing with social trust. We see from Fig. 11 that the success probability is 0.0012 when the hop distance is 2, and becomes almost 0 when the hop distance is larger than 2. This is because it is unlikely for a message to be routed in a hop-by-hop manner with social trust because of the low node density in the experiments.

Fig. 12 shows the experimental results of delivery delay versus hop distance under greedy routing with social trust. It is easy to validate that the delay increases linearly when the hop distance increases because of hop-by-hop forwarding.

V. CONCLUSION

In this paper, we designed and implemented a socio-technological network testbed for performance evaluation and testing of the combined social and communication network architecture in realistic environments. Experimental results show that joint network design based on social and communication links can substantially improve the message delay and delivery ratio performance in real-world scenarios. The designed testbed platform emerges as a high fidelity environment for design, implementation, test, and evaluation of hybrid and heterogeneous networks in both commercial and military applications. Our future work includes design optimization and large-scale experiments with larger network sizes.

REFERENCES

- [1] C. Fuchs, “The internet as a self-organizing socio-technological system,” *Cybernetics & Human Knowing*, vol. 12, pp. 37–81, 2005.
- [2] K. Neema, Y. E. Sagduyu, and Y. Shi, “Search delay and success in combined social and communication networks,” in *Proc. of IEEE Globecom*, 2013.
- [3] F. W. Geels, “From sectoral systems of innovation to socio-technical systems: Insights about dynamics and change from sociology and institutional theory,” *Research Policy*, vol. 33, Sept. 2004.
- [4] K. K. Dhara, V. Krishnaswamy, and T. Singh, “Reconsidering social networks for enterprise communication services,” in *Proc. of IEEE Globecom*, 2010.
- [5] H. Inaltekin, M. Chiang, and H. Poor, “Delay of social search on small-world random geometric graphs,” *Journal of Mathematical Sociology*, 2012.
- [6] Router Station Pro, <http://www.ubnt.com/routerstation>.
- [7] J. Yackoski, B. Azimi-Sadjadi, A. Namazi, and J. Li, “RFnest: Radio frequency network emulator simulator tool,” in *Proc. of IEEE MILCOM*, 2011.
- [8] OpenWrt, <https://openwrt.org/>.
- [9] RealityMining, <http://realitycommons.media.mit.edu/realitymining.html>.
- [10] Q. Li, S. Zhu, and G. Cao, “Routing in socially selfish delay tolerant networks,” in *Proc. of IEEE INFOCOM*, Mar. 2010.
- [11] S. Trifunovic, F. Legendre, and C. Anastasiades, “Social trust in opportunistic networks,” in *Proc. of IEEE INFOCOM*, Mar. 2010.
- [12] I.-R. Chen, F. Bao, M. Chang, and J.-H. Cho, “Trust management for encounter-based routing in delay tolerant networks,” in *Proc. of IEEE Globecom*, 2010.