Title

Opportunistic routing using prefix ordering and self-reported social groups

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2013

Peer reviewed

Opportunistic Routing Using Prefix Ordering and Self-Reported Social Groups

Qian Li and J.J. Garcia-Luna-Aceves
Computer Engineering Department
University of California, Santa Cruz, CA 95064
Email: {liqian, jj}@soe.ucsc.edu

***Abstract*—We present a new approach for opportunistic information dissemination. In contrast to prior approaches to disruption-tolerant networking that focus on physical aspects of connectivity, our approach is based on exploiting the social plane to improve the efficiency with which the physical network infrastructure is used. We integrate the use of social-group information with an approach to routing that eliminates the use of destination-based routing tables. We show that our approach provides correct unicast routing, and compare its performance against that of epidemic routing and an disruption-tolerant, address-based routing scheme. The results from our simulation experiments illustrate that eliminating flooding in connected components of a network and using the social plane to guide information dissemination render substantial performance improvements over traditional methods for routing in disruption-tolerant networks.**

***Keywords*-Social network, Self-reported, ASR, DTN.**

I. Introduction

Many approaches have been proposed and implemented in the recent past to support information dissemination in networks subject to end-to-end connectivity disruption. We have shown [1] that the order capacity of a wireless network can increase if the social groups that determine the flow of information in the network tend to involve nodes within small distances of one another, relative to the network size. However, these capacity gains cannot be approached unless the caching and routing mechanisms used in a wireless network take into account the structure of the social groups operating in the network. As Section II describes, prior proposals for routing in Disruption Tolerant Networks (DTN) have focused mostly on physical-level aspects of network connectivity, even when they have attempted to address the social groups in which nodes participate. Furthermore, the vast majority of routing schemes proposed for DTNs rely on the use of destination-based routing tables and the dissemination of information towards destination addresses.

Section III presents the first approach to information dissemination in DTNs that eliminates per-destination flooding of control packets, allows information dissemination based on the names of destinations or content instead of addresses, and at the same time exploits social-group information to improve the efficacy with which information is sent towards destinations. Many applications (e.g., disaster-relief, or tactical military networks) can be mapped into self-reporting social groups operating over geographically dispersed wireless networks, and supporting self-reported social network is less computationally expensive than detecting them. Accordingly, we focus on nodes self-reporting their social groups. We call our approach, *Automatic Social Routing* (ASR). Each node self-reports its membership in social groups and has a profile denoted with name-value pairs that describes the social groups and node-specific attributes of the node. ASR uses prefix labels and a distributed hash table [2] to eliminate destination-based routing tables and hence the need for flooding of signaling messages. Social Distributed Hash Tables (SDHT) are built to maintain mappings between node profiles and their network locations. If a destination node is unreachable, messages to that node are stored at the current relay, and forwarding paths are selected using social information to forward information opportunistically towards the social groups to which intended destinations are known to belong.

Section IV compares ASR with epidemic routing and an efficient on-demand routing protocol designed for DTNs. The results of our simulated experiments illustrate the benefits of using the social plane to improve the performance of information dissemination in DTNs.

II. Related Work

The vast majority of the work on routing in DTNs reported to date has focused on routing to intended addresses (e.g., [6], [3], [4], [5]). However, past studies on human mobility suggest that exploiting social communities may have a positive impact on the performance of information dissemination schemes [11]. A number of approaches have been proposed in recent years to support content dissemination in DTNs based on the names of information objects or the social contacts established by nodes over time.

DIRECT [7] is an example of content-based information dissemination schemes. Nodes flood their interest in content items denoted by attribute-value pairs; the interest requests establish routes back to the nodes interested in content, and those nodes with replicas of content that match the attribute values in a request are able to answer the request. This scheme is similar to Directed Diffusion [8] and has been shown to have delivery rates close to epidemic routing but with much smaller overhead. However, DIRECT requires the flooding of interest requests, which becomes a performance problem in large DTNs. In ASR, we seek to eliminate flooding altogether.

A number of approaches based on social-group information have been proposed for information dissemination in DTNs (e.g., [9], [10], [16], [17], [18]). Hsu et al. [10] classify nodes by their mobility profiles using a set of predefined locations to determine profiles and an association matrix denoting the importance of such locations in a mobility profile. Data are sent to all nodes fitting a profile. The approach is better than epidemic dissemination, but it cannot ensure accuracy and mobility profiles cannot fully represent the social ties among nodes.

Daly et al. [16] propose SimBet Routing for information dissemination in DTNs based on betweenness centrality and similarity of nodes in their social networks to determine which neighbor a node should select as a relay. The authors show this approach outperforms PRoPHET [20] when nodes have low connectivity. However, computing node centrality and similarity require exchanging and updating the encounter history from each node, which incurs excessive overhead. Furthermore, constructing paths based on betweenness centrality can cause congestion when the same nodes are selected for too many paths.

BUBBLE [17] combines the knowledge of community structure with the knowledge of node centrality to make forwarding decisions. It assumes each node belongs to at least one community, and each node has a global ranking (centrality) across the whole system, and also a local ranking within its local community. Flooding is used to get the betweenness centrality. Community detection uses centralized algorithms. Li and Cao [18] propose data forwarding schemes based on social network parameters. The authors consider both users willingness to forward and their contact opportunity, which results in a better forwarding strategy than purely contact-based approaches. They assume that the stronger the social tie is, the larger the social willingness is.

III. Automatic Social Routing (ASR)

We assume that the social networks operating over a DTN consist of users with predefined and self-reported social information. Each user belongs to at least one social group. Each node stores a profile containing the node's identity (*Node_ID*), and the names and attributes of the social groups to which the node belongs. To describe ASR in concrete terms, we use a conference scenario as the social context in which ASR is used, and use a conference trace file [12] to evaluate the routing schemes. In this setting, the profile of a participant states the talks that she needs to attend (*Task*), different areas that she visits (*Area*), topics of interest (*Interests*), set of contacts from email or text messages (*Contacts*), and a country of origin (*Country*). Different social groups are defined based on the values assigned to these various attributes.

From the standpoint of the underlying wireless network connectivity, social groups may overlap physically, or be disjoint from each other. On the other hand, nodes may communicate with other nodes in the same or different social groups. Accordingly, there are three types of routing to consider: (a) routing within a social group among nodes that have physical multi-hop connectivity, (b) routing within a social group among nodes that do not have physical multi-hop connectivity, and (c) routing across social groups. In ASR, nodes establish multi-hop signaling with other nodes in the same social groups to which they belong to enable routing to specific individuals in the same social group while avoiding per-destination flooding. To route across social groups, nodes first target the social groups to which the intended destinations belong, followed by routing to individuals once information reaches nodes in the social groups of intended destinations. This is very similar to traditional hierarchical routing, but without establishing any strict clusters or subnets of nodes. Figure 1 illustrates how routing is attained in ASR, which we discuss next.

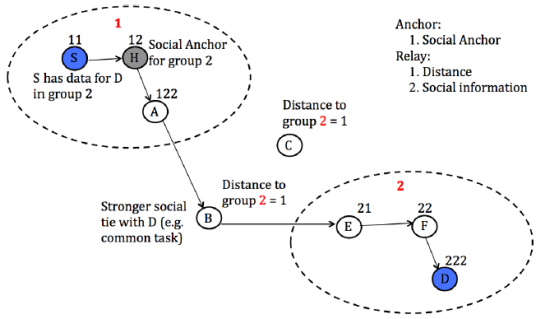

Fig. 1. Example of routing in ASR within and across social groups

A. Routing within Social Groups

Routing from sources to destinations that belong to the same social group and have physical multi-hop connectivity between them is attained by adopting the prefix routing approach first proposed in [2] to operate within the context of social groups. Each social group elects a labeling root node using a distributed algorithm based on neighbor-to-neighbor signaling. Each node transmits a *Hello* message to its neighbors specifying the node profile, and for each social group to which it belongs the Hello states the root identifier of the social group, the label of the node, and the node identifiers and labels assigned to its immediate neighboring nodes. For each social group, if Σ is the finite set of symbols, then the routing label of a node, l, is a string with symbols from Σ such that $|l| \geq 1$. The root node has the smallest label of the social group. Once a node has a routing label, it assigns a unique suffix s_i to each of its children, i. The child then assigns itself the label $l \odot s_i$, where $\odot$ is the concatenation operator.

When a node is initialized, it determines the smallest routing label and root it can attain for the social group it belongs using the information it receives in the Hellos from its neighbors. If a node does not obtain a routing label for a social group to which it belongs within a local labeling timeout period, it assigns itself as the root node and sends a Hello stating that. The Hello assigns labels to those neighbors in the social group. If a labeled node receives a Hello with a root identifier lower than its own root for a social group, the node accepts

the lower root and receives a routing label. Eventually, for each social group, the node with the lowest identifier in the group is elected as the root and all the other nodes in the same connected component of the network are ordered with respect to that root node. The root node is elected such that: (i) each node is assigned a label denoting the relative location of the node with respect to the root; (ii) the labels of a source and a destination define one or multiple valid routes between two nodes; and (iii) node mobility and link or node failures and additions have limited impact on the labels already assigned to other nodes. Nodes sends their Hellos periodically to help refresh their labels to cope with node mobility, failures, and additions.

To route to a given destination in the same social group, a source must find the prefix label of the destination and the destination must publish its prefix label, and this is accomplished by means of *social anchors*. A social anchor is a node whose own label is the closest match within its two-hop neighborhood to the hashing of the name of a social group κ_{group} and a set of zero or more interests κ_{ints} within that group. To publish its presence, a node hashes its name, identifier or set of attributes that describe it, using a common hashing function over the terms selected for publishing. The node then unicasts a publish request towards the prefix label resulting from the hashing, and the requests states the node profile and its prefix label. To communicate with a destination, a source hashes the name, identifier or attributes describing the destination and obtains the prefix label of a social anchor; it then unicasts a subscription request towards the social anchor. This publish-subscribe signaling takes place among nodes in the same social groups.

For fault-tolerance purposes, multiple social anchors are used to maintain the information of a social group associated with specific interests. More specifically, the neighbors of a social anchor also become social anchors for the same group of individual. Therefore, if one social anchor moves away or dies, there are other social anchors around to keep the publish-subscribe process working. A consistent hash function is used to avoid remapping of nodes. The hash function takes node profiles as input and returns a prefix label of the associated social anchor. Note that, because social groups and individuals may be characterized by a variable number of attributes, all nodes must agree beforehand on the subset of attributes to be used for hashing the description of a given social group or individuals into prefix labels.

In the example shown in Figure 1, if A were the intended destination of node S, then S would hash A's name to obtain A's social anchor's prefix label, then obtain A's prefix (122) from social anchor, after which it could route to A based on its own prefix (11) and A's prefix.

When the routes to a destination are broken, a route error message is sent to the source of data by the relay that is unable to forward data to the destination. As a result of the route error, the source deletes cached routes to the destination and requests and contacts an anchor for a new route. If a destination node is unreachable, a packet intended for that destination is stored at the current node (source or relay), and a forwarding path is selected using the routing strategy explained in the following subsection.

B. Routing across Social Groups

To route to a destination in a different social group, the node sends a subscription request to a social anchor selected among the nodes in the social groups to which the source belongs. The source uses the name or attributes describing the social group of the destination, rather than the destination itself. We assume that the source is aware of the destination's social group.

To enable nodes to operate as social anchors for social groups to which they do not belong, nodes that come into physical contact with nodes in other social groups publish their encounters with social anchors in their own social groups. ASR combines physical distance information with social network information for routing across social groups. If a node has either a physical or social connection with nodes in another social group, it publishes the mapping between the connection and the associated social anchor's prefix label with the corresponding social anchor in its own group. If the publishing request is meant to inform the social anchor of a status change in the social connection status, it consists of the encountered node's name and its social group, number of common tasks, friendship between each other, number of common friends, and encounter duration in time t. If the publishing request is generated because of a change in physical connection status, in addition to above information, it also contains the physical distance (network hops) to the node with which the encounter happened. Node sets the distance value in the publishing request to be infinite when the physical connection breaks (node moves away or dies); otherwise, it is a finite integer. The node with a prefix label that is the closest match in its two-hop neighborhood to the prefix label stated in a publishing request becomes the designated social anchor for the mapping, then stores it and builds the SDHT. The SDHT consists of two tables storing published attributes, one lists the nodes currently having physical connection with nodes in the social group this social anchor representing for. a node with infinite physical distance value stated in a publishing request is removed from the list by social anchor. The other table lists the nodes having social connections but no physical connections with nodes in that social group. This SDHT building policy guarantees that there is no overlapping of nodes between the two tables. The neighbors of the designated social anchor also get the publishing request and store it into SDHT structure. The frequency to update node's connection status to social anchors is controlled by the change of node's physical or social connection with another social group.

The following four node characteristics are used to inform routing decisions based on information regarding social connections when physical distance information is not available:

Similarity: Two users are similar if they have common task (in general, they have common attribute values in their profiles). The similarity $Sim(x, y)$ between user x and y is

calculated by: $Sim(x,y) = |N(x) \cap N(y)|$, where $N(x)$ and $N(y)$ are the set of tasks of user x and y respectively. If the similarity index is larger than a threshold, we call user x and y are socially similar.

Social Affiliation: Two users are affiliated or socially close if they are friends in their profile friend list. If social affiliation is put into a $n \times n$ symmetric matrix, where n is the number of users in the network, the affiliation matrix has elements:

$$A_{xy} = \begin{cases} 1 & \text{if } x, y \text{ are friends} \\ 0 & \text{otherwise} \end{cases}$$

We consider contacts to be bidirectional, so that a contact that exists between x and y necessarily implies that a contact exists between y and x. This Affiliation value will add one point into users social tie strength calculation if it is positive; otherwise, we ignore this part.

Spatial Closeness: Two users are spatially close if they have more common friends in their profile friend list or contact list. The spatial closeness $S(x,y)$ between user x and y is calculated by: $S(x,y) = |F(x) \cap F(y)|$, where $F(x)$ and $F(y)$ are the set of friends of user x and y respectively. The value of $S(x,y)$ determines how spatially close user x and y are.

Temporal Closeness: Two users are temporally close if they meet or communicate often in a certain time, and is was calculated by: $T_t(x,y) = \frac{C(x,y)}{t}$, where $C(x,y)$ is the time of communication happening between x, y in a given period of time, and t is the period of time we calculate in. So $T_t(x,y)$ is the communication frequency between x and y.

These are then composed into a single value, as

$$C_{social} = \omega_{sim} Sim(x,y) + \omega_a A(x,y) + \omega_s S(x,y) + \omega_t T_t(x,y) \quad (1)$$

which represents how socially close two nodes are and implies social distance between nodes in reversal. The weights ω denote the relative importance of each attribute. Their value depends on the application scenario. Due to space limitations, we briefly apply those values on a single scenario which we used to show the performance also demonstrate some crucial properties of our approach. The default values of the weights ω_{sim} = 0.45, ω_a = 0.2, ω_s = 0.05, and ω_t = 0.3 are those providing the best performance in terms of delivery ratio in our simulations.

If there is more than one node having a strong social tie with either an individual destination or other nodes in the social group of the intended destination, the social closeness values of opportunistic nodes are ranked from highest to lowest. The first k nodes are selected as opportunistic contacts to forward message to an intended destination.

The following rules are used to route data across social groups:

a) **distance(k):** forwarding path $u \rightarrow v$ is allowed if v is within distance k to d in the current network topology, and satisfies distance $D(v) < D(u)$.

b) **neighbor(k):** by comparing node characteristics, forwarding path $u \rightarrow v$ is allowed if v and d are socially close within distance k (or has the highest ranking among all opportunistic nodes) in social graph with social distance $S(v) < S(u)$.

c) **non-increasing-social-distance:** with the same distance(k) to d, forwarding path $u \rightarrow v$ is allowed if social distance from v to d is less than the one from u to d, $S(v|D) < S(u|D)$.

According to the **non-increasing-social-distance**, when nodes have both social distance and physical distance information available, the physical distance is given higher priority. Equation (1) then is adjusted to be:

$$C = \omega_{sim} Sim(x,y) + \omega_a A(x,y) + \omega_s S(x,y) + \omega_t T_t(x,y) + \omega_d / D(x,y) \quad (2)$$

where ω_d = 1000, so that physical distance dominates the selection. Equation (2) also consists with **non-increasing-social-distance** forwarding rule that with same physical distance, social distance is the criteria to select a relay node.

To route to the destination in another social group, the source uses the name describing the social group of the destination as the hash input to get the social anchor's prefix label. The source unicasts a subscription request towards the social anchor through prefix routing. The designated social anchor stores the SDHT for nodes who have social distance or physical distance information with nodes in destination's social group. Social anchor selects first from the table storing nodes with physical distance information to the destination's group, which provides a direct forwarding path to destination's group. If that table is empty, social anchor selects nodes from the other table with social distance information to the destination's group by ranking the nodes according to Equation (1). The relay node selection follows the above forwarding rules. Social anchor sends back a reply to source containing the selected relay node for continuous communication between source and destination. Once the packet is forwarded to a node belonging to destination's group, routing to destination is attained by the prefix routing approach we described in Section III-A.

Figure 1 illustrates how routing takes place across social groups. The source S is in social group 1 with label '11', and it has packet for node D, which the source knows to be in social group 2 but has no cached route to it. The first criteria to select a relay node is to find a node that has distance information to the destination group. The source asks the node serving as social anchor for group 2 for a relay node. In this example, node H is the social anchor for group 2, it selects first k relay nodes that have distance information or strong social ties with the destination. If no distance information is available in group 1, relay nodes are chosen by their social connection with the destination. There are two nodes between the two groups with direct connection to group 2, and both of them are at one-hop distance to node A. Accordingly, node A is selected as one relay node by social anchor H. Both node

B and C has direct connection to group 2, but node B has a stronger social tie with destination D after node A compares their social distance to group 2. In this case, node A selects node B to be next hop and sends the packet to it. Node B forwards the packet to node E which is in group 2. After the packet has been sent to group 2, prefix routing is carried out to the specific destination.

C. Routing between Physically Disconnected Components

End-to-end paths need not be always available. If routing must happen between physically disconnected components within or across social groups, social information is used to select relaying nodes. In the current two-hop neighborhood, the source or a relaying node calculates neighbors' social distance to destination or destination's group using four nodes characteristics introduced in previous subsection. The node who has a shorter social distance or who has an immediate neighbor with a shorter social distance is selected as the next hop of the packet. A copy of packet is then sent to the selected relaying node. To achieve fast packet delivery, if a network disconnection happens between two social groups, the relaying node sends a copy of the packet to every member from the destination group during the first encounter, until reaching the maximum copy a node can create. In addition, when the destination is unreachable in the current two-hop neighborhood, the relaying node must keep the packet and perform periodically next hop calculation until the packet lifetime expires, or it meets the destination or nodes having routes to the destination.

IV. Evaluation And Comparison

We compare ASR with two other data dissemination approaches. We use an Epidemic scheme [13] in which there is no user grouping information and all nodes can be relays. According to [13], we set a message hop limit, and a buffer size limit in the implementation. A message life timer is assigned to each message. Each node periodically deletes timeout messages from its message buffer. This is also the most aggressive forwarding strategy in DTN. We also implemented an efficient disruption-tolerant, address-oriented routing protocol based on that reported in [6]. We denote this protocol by DAR and use it for comparison because it is very efficient for DTN routing towards destination addresses without taking into account social groups. Each node has a message buffer to store route unknown messages till their timeout. These messages are kept until current node encounters a node having route to destination or destination directly.

A. Experimental Dataset

In this paper, we use a dataset collected from the seventh HOPE (Hackers On Planet Earth) conference held on July 18-20, 2008. Conference attendees received RFID badges that uniquely identified and tracked them across the conference space. The dataset was collected from the three days of the conference, and the content included participants' location, interest, profile, friend list, and event details.

We selected the 83 most active participants from the dataset, i.e., the most interests listed in their profiles and the most talks attended. We use common interests and social affiliation (e.g., contact with email, text message, their countries and other attributes) to divide them into 4 social groups. The method to classify participants into groups based on their attributes is supported in statistical computing and graphics language R [14]. The hierarchy is shown in Figure 2.

We separate the dataset into three parts. 50% data is used for training, 10% for tuning, and 40% for simulating. We believe this dataset is large and accurate enough to simulate our scheme.

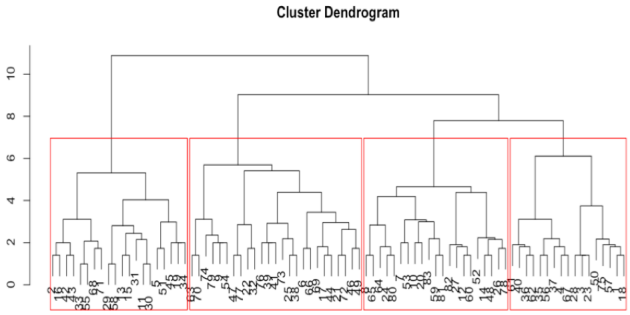

Fig. 2. Node Classification

B. Simulation Setup

We use a hi-fidelity event-driven packet level network simulator, Qualnet-v.4.0 [15]. We first import the dataset into Qualnet, and then we set a node position file according to the conference trace file, which means at each second in the simulation, network topology represents the real conference scenario. The trace file was collected using RFID tracking data, and in order to adapt it to 802.11b transmission ranges we increased the coordinate distances in the trace file. Therefore, the coordinate value in the trace file represents 802.11b transmission ranges in our simulation. We collect statistics from 1 hour up to 10 hours, which only contains daytime node activity. We simulate scenarios with different numbers of concurrent data flows to see the performance of three schemes under different network load. Simulations were instrumented in networks of nodes deployed in a terrain of dimensions 1600m X 1600m. PHY-Model in the nodes was PHY802.11b with transmission range of 300m.

Data sources are generators that produce a constant bit rate (CBR). In the trace file, some nodes are moving fast, in order to fully explore the use of social network information in the fast mobile scenario, we set the data rate of 2 packets per minute and each source was allowed to transmit up to 1200 packets. Data pairs are randomly selected. All the experiments are run multiple times with 10 different seeds to avoid any artifact of pseudo random number generators.

We set same key parameters for three schemes. Hello message interval is set to 3 sec, there is the maximum 100 messages buffer size, all list and buffer flush timers are set

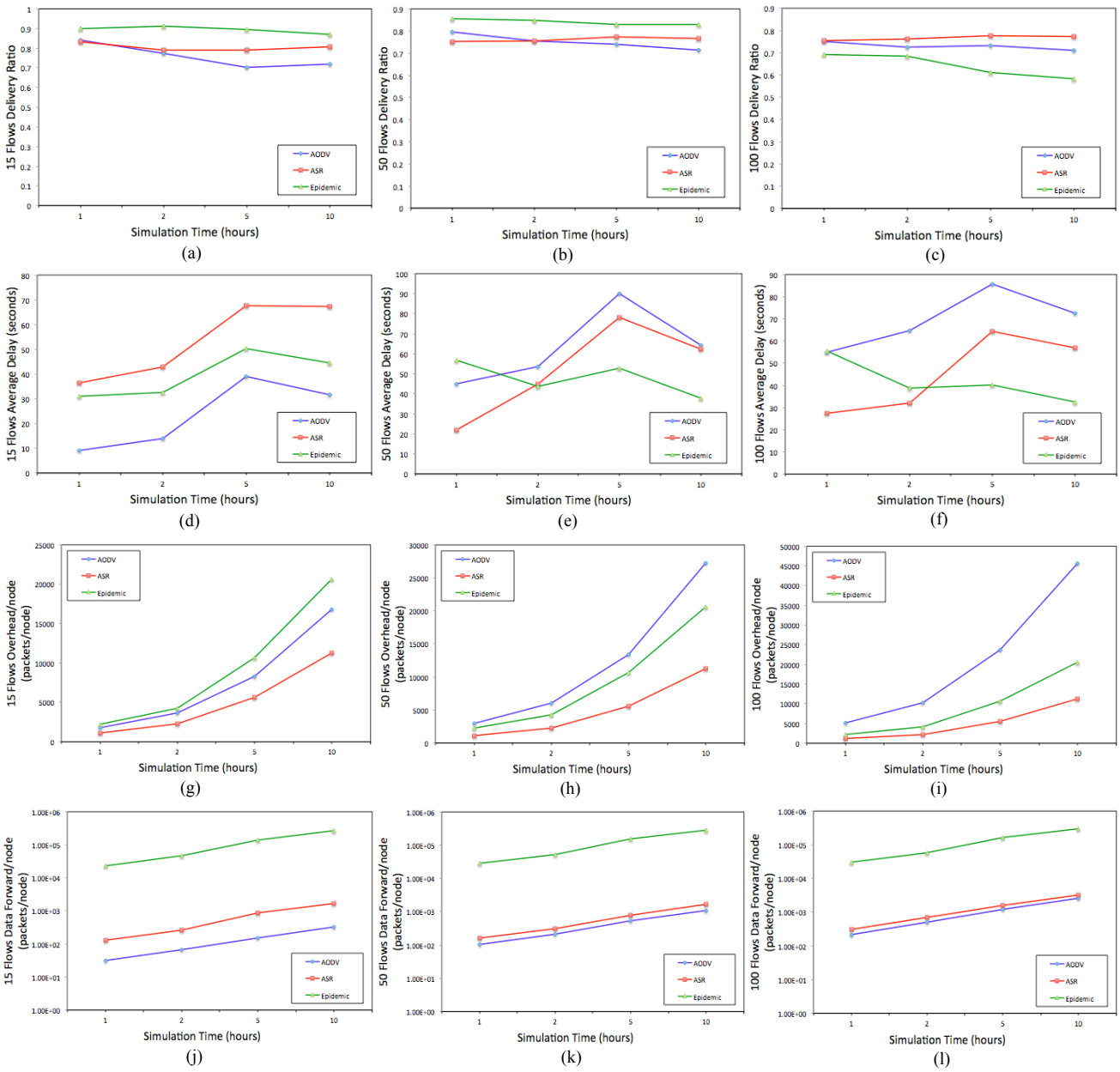

Fig. 3. (a)-(c) delivery ratio with different data flows, (d)-(f) average delay with different data flows, (g)-(i) signaling overhead per node with different data flows, (j)-(l) data forwarding per node with different data flows

to 1 hour, and we set three times hello message interval for neighborhood status check.

C. Simulation Results

We use the following four metrics for comparison:

Overhead: The number of signaling packets sent per node.

Average Delay: The average time between data generation time and data receiving time at destination. It contains message-holding time in each node's buffer, which represents DTN network character.

Delivery Ratio (goodput): The number of data packet received divided by number of data packet sent.

Data Forwarding: The number of data packets sent including initiating and forwarding at each node.

Figure 3 summarizes the simulation results for the three schemes under different scenarios. In Figures 3 (a) - (c), all three schemes have lower delivery ratios as the number of data flows increase. Epidemic routing has the highest delivery ratio for the 15- and 50-flows scenarios, but it has lower delivery ratio than ASR and DAR for the 100-flows scenario. One reason for the degradation in performance in epidemic routing with 100 data flows is the large number of transmissions needed to support many data flows, which causes too many collisions. Other contributing factor is that we set the limit of

data buffer size, and eight hops is the maximum hop limit to avoid looping. ASR always achieves higher delivery ratio than DAR under different numbers of data flows.

Figures 3 (d) - (f) show the average delay of three schemes. With a small number of data flows, DAR attains lower delays than ASR, because very few routes are needed in the network. In our approach, sources need to send requests to social anchors first, which increases end-to-end delays. With large data traffic, epidemic routing attains the smallest delays for those packets that are delivered because of the characteristic of flooding; however, fare fewer packets get delivered. Under high data load, ASR has lower end-to-end delay than DAR. DAR performs the worst under high data load. When the number of data flows increases from 15 flows to 100 flows, the end-to-end delay of DAR increases 4 times, and is 33% higher than in ASR.

Figures 3 (g) - (i) show the average overhead under different data flows. With low data load, epidemic routing has the highest overhead and this is mainly because of the cost to exchange summary vectors to determine which messages have not been seen by each other and the cost to request copies of those messages. When the number of data flows increases, DAR generates too many control packets, the typical one is route request messages. Its overhead is 4.5 times higher than ASR and 2.25 times higher than epidemic routing with 100 data flows. ASR has the lowest overhead under different data load. In other words, our approach provides similar delivery ratio but much lower overhead than DAR.

Figures 3 (j) - (l) present the data packets forwarded per node. As expected, epidemic routing has the highest data forwarding statistic. ASR has slightly higher forwarding load than DAR. This is because a node carrying data and encountering a member from the destination group must forward the data to that node. This incurs additional data forwarding overhead than DAR. However, the difference between our approach and DAR becomes very small as the number of data flows increases.

V. Conclusion

We proposed a novel approach to routing in DTNs that takes advantage of social-group information and eliminates the vast majority of the signaling overhead present in traditional routing schemes for DTNs. We used a social network model of social affiliations of group members, and developed an integrated approach to routing within and across social groups that operates efficiently even when the underlying network is disconnected by eliminating flooding by means of social clues maintained in a DHT. From the simulated experiments based on a real-world trace file, we find out that ASR has similar delivery ratio but far lower end-to-end delay and overhead than DAR, especially under high data load. It is also far more efficient than epidemic routing, and yet very resilient. If the network is temporally disconnected or labeling is not up to date, data packets are stored at relaying nodes, and routing resumes once forwarding opportunities occur, which saves considerable signaling overhead.

Acknowledgment

This research was sponsored in part by the Baskin Chair of Computer Engineering and by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053.

References

[1] B. Azimdoost, H. Sadjadpour, and J.J. Garcia-Luna-Aceves, "The Impact of Social Groups on The Capacity of Wireless Networks," *First IEEE NSW 2011*, West Point, New York, June 22–24, 2011.

[2] J.J. Garcia-Luna-Aceves and D. Sampath, "Scalable Integrated Routing Using Prefix Labels and Distributed Hash Tables for MANETs," *Proc. IEEE MASS 09*, 2009.

[3] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *Proc. ACM MobiHoc 04*, May 2004.

[4] B. Burns, O. Brook, and B. Levine, "MV routing and capacity building in disruption tolerant networks," *Proc. IEEE Infocom 2005*, March 2005.

[5] W. Zhao, Y. Chen, M. Ammar, M. D. Corner, B. N. Levine, and E. Zegura, "Capacity Enhancement using Throwboxes in DTNs," *Proc. IEEE MASS 06*, October 2006.

[6] J. Boice, J.J.Garcia-Luna-Aceves, and K.Obraczka, "On-demand routing in disruptive environments," *Proc. IFIP Networking 2007*, Atlanta, GA, May 14–18, 2007.

[7] I. Solis and J.J. Garcia-Luna-Aceves, "Robust content dissemination in disrupted environments," *Proc. CHANTS 08: ACM MobiCom 2008 Workshop on Challenged Networks*, Sept. 2008.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Proc. ACM MobiCom 2000*, 2000.

[9] A. Mtibaa et al., "Are you moved by your social network application?" *Proc. ACM SIGCOMM WOSN 08*, Seattle, WA, USA, Aug. 2008.

[10] W.-J. Hsu et al., "Prole-cast: Behavior-aware mobile networking," *Proc. IEEE WCNC 2008*, Las Vegas, NV, USA, Mar. 2008.

[11] P. Hui et al., "Pocket switched networks and human mobility in conference environments," *Proc. 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, August 2005.

[12] Aestetix and C. Petro, "CRAWDAD data set hope/amd," Aug. 2008.

[13] A. Vahdat, D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks", Duke Tech Report CS-2000-06, 2000.

[14] "The R Project for Statistical Computing", Available: http://www.r-project.org/index.html.

[15] S. N. Technologies. Qualnet. Available: http://www.scalable-networks.com/.

[16] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," *Proc. ACM MobiHoc 07*, 2007.

[17] P. Hui et al., "Bubble rap: social-based forwarding in delay tolerant networks," *Proc. ACM MobiHoc 08*, pages 241250, 2008.

[18] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," *Proc. IEEE INFOCOM 2010*, 2010.

[19] J. Su et al., "Haggle: seamless networking for mobile applications," *Proc. UbiComp '07*, September 2007.

[20] A. Lindgren, A. Doria, O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 7, No. 3, July 2003.