Social contact graph-based group communication schemes for delay tolerant networks

Tuan Le & Mario Gerla

To cite this article: Tuan Le & Mario Gerla (2017): Social contact graph-based group communication schemes for delay tolerant networks, International Journal of Parallel, Emergent and Distributed Systems, DOI: 10.1080/17445760.2017.1300802

To link to this article: http://dx.doi.org/10.1080/17445760.2017.1300802

Published online: 15 Mar 2017.
Social contact graph-based group communication schemes for delay tolerant networks

Tuan Le and Mario Gerla

Department of Computer Science, UCLA, Los Angeles, CA, USA

ABSTRACT

Delay-tolerant networks (DTNs) are sparse mobile ad hoc networks in which there is typically no complete path between the source and destination. Multicast and anycast are important group communication paradigms for numerous DTN applications. For example, multicast is used to disseminate real-time traffic information reporting and software patch to multiple devices, and anycast is used for resource discovery and information exchange in emergency or crisis situation. While multicast and anycast have been studied extensively in the context of the Internet and Mobile Ad-Hoc Networks (MANETs), efficient multicast and anycast in DTNs are significantly different and challenging problems due to frequent partitions and intermittent connectivity among nodes. In this paper, we propose single-copy routing strategies for multicast and anycast based on the multi-hop delivery probabilities. Multicast employs a dynamic tree branching technique that allows routing paths to be efficiently shared among multicast destinations. Anycast selects relay nodes based on social distances to anycast group members. It balances the trade-off between a short path to the closest, single group member and a longer path to the area where many other group members reside. That is, it optimises both the efficiency and robustness of message delivery. Through extensive simulation studies using a real-world mobility trace, we show that our schemes achieve a high delivery ratio, low delay, and low (or comparable) transmission cost compared to other group communication strategies.

ARTICLE HISTORY

Received 3 November 2016
Accepted 23 February 2017

KEYWORDS

Delay tolerant networks; anycast; multicast; social contact graph; social distance; delivery probability

CONTACT

Tuan Le  tuanle@cs.ucla.edu

© 2017 Informa UK Limited, trading as Taylor & Francis Group
1. Introduction

Delay-tolerant networks (DTNs) [1] are characterised as sparsely connected, highly partitioned, and intermittently connected ad-hoc networks. In these challenging environments, end-to-end communication paths between node pairs are rarely available. There are many practical applications of DTNs, including wildlife tracking sensor networks [2,3], peoplenet [4], ocean sensor networks [5,6], military networks [7,8], and vehicular ad-hoc networks [9,10]. To handle the sporadic connectivity of mobile nodes in DTNs, the store-carry-and-forward method is used. That is, messages are temporarily stored and carried by a node until an appropriate communication opportunity with the next relay hop arises.

Recently, there has been a growing interest in DTN group communication protocols such as multicast and anycast. Multicast enables the distribution of data to multiple receivers, such as real-time traffic information reporting, diffusion of participatory sensor data or popular content (news, software patch, etc.) over multiple devices. Anycast allows a node to send a message to any one member in a group of nodes. It can be used in emergency response networks to request the help of a doctor, a fireman, or a police without knowing their IDs or accurate locations. Anycast can also be used in urban community networks, in which people can use the network to call for any cab. Although many multicast and anycast routing protocols have been proposed in the Internet and MANETs, they cannot be easily applied to DTNs due to the lack of stable end-to-end paths to a destination group member in DTNs. Furthermore, in traditional DTN unicast routing, the destination of a message is fixed at the time of creation. By contrast, the destination can change dynamically in anycast routing according to the movement of nodes. As a result, anycast routing is a particularly challenging problem.

In this paper, we focus on developing routing schemes for multicast and anycast that are both robust and efficient (e.g. having a high delivery probability and short delay). The schemes are based on the single-copy model in which there is at most one copy of the message in the network. To cope with the highly volatile node mobility, we exploit the stable social network structure for message forwarding. Specifically, we measure the social-tie strength between nodes and then formulate the forwarding metrics based on the multi-hop delivery probability computed over the social contact graph. Multicast uses a dynamic tree branching technique that bundles multiple multicast receivers into a single copy of the data packet and forwards it to an encounter node with higher multi-hop delivery probabilities to multicast receivers. This allows routing paths to be efficiently shared among multiple destinations. Anycast selects relay nodes based on social distances derived from multi-hop delivery probabilities to anycast group members. The forwarding decision considers the trade-off between a short path to the closest, single group member (i.e. short social distance) and a longer path to the area where many other group members reside. While the former can shorten the delivery delay, it is less robust than the latter, especially when the nearest node is socially isolated from other group members, and it may often leave or move to another location in a dynamic network. Note that the robustness of the latter choice comes from the intuition that a node is more likely to encounter a particular group member if it is closer to many group members.

The paper makes the following contributions:

- A multi-hop delivery probability metric for relay selection.
- A dynamic tree branching (DTB) technique for DTN multicast routing.
- An anycast social distance metric (ASDM) for DTN anycast routing.
- An extensive evaluation of the relay selection metrics.

The rest of the paper is organised as follows. Section 2 reviews the related work. Section 3 describes the design of the relay selection in detail. Section 4 presents the experimental results. Section 5 concludes the paper and discusses the future work.

2. Related work

In this section, we review existing works on multicast and anycast routing in DTNs.
2.1. Multicast routing in DTNs

Multicast for DTNs has recently drawn considerable attention. Zhao et al. [11] proposed a set of semantic models to unambiguously describe multicast in the context of DTNs. They incorporated various knowledge oracles such as contact and membership into four classes of DTN routing algorithms: unicast, broadcast, tree, and group. Ye et al. [12] proposed on-demand situation-aware multicast (OS-multicast) in which a node dynamically maintains a multicast tree rooted at itself to all the receivers using local knowledge of the network topology. Xi and Chuah [13] proposed an encounter-based multicast routing scheme (EBMR), which uses the encounter history based on PROPHET DTN unicast routing [14] to disseminate a packet to the neighbours, each of which has the highest delivery predictability (within two hops) to one of the multicast receivers. In [15], the throughput and delay scaling properties of multicast routing in DTNs are discussed, and mobility-assisted routing is used to improve the throughput bound of wireless multicast. In [16], multicast in DTNs is considered from the social network perspective, and the social network concepts such as centrality and social community are exploited to minimize the multicast cost in terms of the number of relays used. In [17], remote communication is used to assist guaranteed multicast delivery in DTNs. The problem of optimising the remote communication cost is formalised as the demand cover problem, which is solved using a graph-indexing-based solution. Compare-split scheme [18] is another DTN multicast routing protocol, which combines the node active level with the contact rate level to determine when and how to split a destination set during a contact. Quota-Based Multicast Routing [19] adopts a quota replication scheme. Messages that are replicated to a more active node are assigned a higher quota value by the source node. In [20], constrained multicasting problems are discussed. Forwarding nodes are selected based on the delivery probability to destinations within the maximum buffered time. Messages with expired buffered time are discarded to reduce the probability of buffer overflow.

Unlike prior works that select relay nodes to multicast receivers based on one or two-hop encounter probability, and thus have a limited local view in forwarder selection, we consider long routing paths (two or more hops) to gain better forwarding opportunities. Furthermore, our scheme bundles multiple multicast receivers into a single copy of the data packet, and thus minimizes the transmission cost.

2.2. Anycast routing in DTNs

Thus far, few works have addressed the DTN anycast routing problem. Nelson et al. [21] proposed to enhance existing unicast protocols to perform anycast communication. To simplify group management, each node carries its own group information throughout the network. Each node also transforms multicast groups into virtual nodes and maintains group-based utility values. When a contact occurs, the node updates the utility for the contact’s group. Messages are forwarded to relay nodes with higher utility values.

Gong et al. [22] proposed a set of semantic models to unambiguously describe anycast in the context of DTNs. They introduced an anycast routing algorithm based on the EMDDA (expected multi-destination delay for anycast) metric. In this algorithm, they assumed that nodes in the network are stationary, and the communication among nodes relies on a few mobile nodes that act as message carriers to deliver messages for the nodes. The algorithm computes the PED (practical expected delay) values from a node to each group member, and then sets EMDDA to be the minimum PED value. A mobile node then carries the message from the current node to the next hop only if the delay to get to the next hop plus the EMDDA of the next hop is smaller than the EMDDA of the current node. This relay process repeats until the message finally reaches any one of the group members.

Xiao et al. [23] proposed an anycast routing scheme based on the MDRA (maximum delivery rate for anycast) metric. MDRA indicates the probability that a message carrier meets a node in the anycast group, and is computed using individual meeting probabilities between a node and each group member. Based on the metric, messages are forwarded from the nodes with low MDRA values to the nodes with high MDRA values until arriving at any one of the destinations.
Another anycast routing technique attempts to utilize genetic algorithms (GAs) for route decisions [24]. The GA is applied to find the appropriate path combination to comply with the delivery needs of a group of anycast sessions simultaneously. However, this work assumes that the mobility of nodes is deterministic and known ahead of time, which is not a valid assumption for most DTNs.

Our work differs from all these studies in two key aspects. First, unlike existing forwarding metrics such as EMDDA and MDRA, which favor a routing path toward an anycast member with the best meeting probability, our proposed social distance based metric ASDM also takes into account the density of group members. More often, ASDM routes the message in the direction where most group members reside to increase the probability of meeting a group member. ASDM may also explore a sparse area with one or a few group members if these nodes have very high reachability probabilities. Thus, ASDM is more suitable for highly unpredictable networks than EMDDA and MDRA. Second, whereas existing works utilize direct encounter probabilities between a node and each group member to compute the forwarding metrics, ASDM is based on multi-hop delivery probabilities, which offer a broader view for forwarder selection.

3. Protocol design

In this section, we first describe the social-tie metric and multi-hop delivery probability. We then present a dynamic tree branching (DTB) technique for multicast routing. Lastly, we introduce two anycast forwarding metrics: anycast direct encounter metric (ADEM) and anycast social distance metric (ASDM), followed by a complete anycast routing strategy.

3.1. Social-tie metric

In sociological terms, social tie describes an interpersonal connection by way of friendship or acquaintance. There are many tie strength indicators: frequency, intimacy/closeness, longevity, reciprocity, recency, multiple social context, and mutual confiding (trust) [25]. Among them, the most widely used heuristics in socially-aware networking applications are the recency and frequency of encounters [26].

Two nodes are said to have a strong tie if they have met frequently in the recent past. We compute the social tie between two nodes using the history of encounter events. How much each encounter event contributes to the social-tie value is determined by a weighing function \( F(x) \), where \( x \) is the time span from the encounter event to the current time. Assume that the system time is represented by an integer, and is based on \( n \) encounter events of node \( i \). Then, the social-tie value of node \( i \)'s relationship with node \( j \) at the current time \( t_{base} \), denoted by \( R_i(j) \), is computed as:

\[
R_i(j) = \sum_{k=1}^{n} F(t_{base} - t_{j_k})
\]

where \( F(x) \) is a weighing function, \( \{t_{j_1}, t_{j_2}, \ldots, t_{j_n}\} \) are the encounter times when node \( i \) met node \( j \), and \( t_{j_1} < t_{j_2} < \cdots < t_{j_n} \leq t_{base} \).

As an example, suppose node \( i \) met node \( j \) at times 1, 3, and 5, and that the current time (\( t_{base} \)) is 10. Then, node \( i \)'s social-tie relationship with node \( j \) at \( t_{base} \), denoted by \( R_i(j) \), is computed as:

\[
R_i(j) = F(10 - 1) + F(10 - 3) + F(10 - 5) = F(9) + F(7) + F(5)
\]

The weighing function \( F(x) \) essentially reflects the influence of the recency and frequency of encounter events. In order to give more weight to more recent encounter events, \( F(x) \) should be a monotonically non-increasing function. A class of functions that satisfy this condition is \( F(x) = (\frac{1}{2})^{\lambda x} \), where \( z \geq 2 \) and \( 0 \leq \lambda \leq 1 \). The control parameter \( \lambda \) allows a trade-off between recency and frequency in contributing to the social-tie value. As \( \lambda \) approaches 0, frequency contributes more than recency.
On the other hand, as $\lambda$ approaches 1, recency has higher weight than frequency. The social-tie value is solely determined by frequency when $\lambda = 0$, and by recency when $\lambda = 1$. In our experiments, the values of $z$ and $\lambda$ are carefully tuned based on the analysis of the network characteristic and are set to 2 and $e^{-4}$, respectively.

### 3.2. Social knowledge formation

In order to make an informed forwarding decision, a node needs to obtain network-wide knowledge of social-tie strength between any node pairs. This knowledge is contributed by both local observation and knowledge exchange.

#### 3.2.1. Local observation

Upon each encounter event, a node records the encounter node ID and the timestamp of the encounter event, and stores it in the encounter table. Periodically, social-tie values between the current node and its direct encounters are re-computed using Equation (1), where the input comes from the history of encounter events stored in the encounter table. In addition, each node maintains a social-tie-table, where each row has the following format:

$$(\text{peerX, peerY, social-tie-value, timestamp})$$

Through local observation, peerX is always the current node ID. PeerY is the encounter node ID. Timestamp is the time at which the social-tie value between peerX and peerY is computed. It is the $t_{\text{base}}$ variable in Equation (1). As we will see next, timestamp plays an important role in knowledge exchange among nodes.

#### 3.2.2. Knowledge exchange

Nodes, especially those that are not socially active, tend to have limited knowledge of the social network through local observation (i.e. through direct contacts with other nodes). To gain knowledge of nodes that have never met, during the encounter period, nodes can exchange and merge their local observations in the form of a social-tie-table. In the event of a merge conflict (i.e. when there are two entries with the same peerIDs), we keep the entry with the latest timestamp. Through this process, a node can learn the social-tie values between different pairs of nodes in the network.

### 3.3. Multi-hop delivery probability

The delivery probability $P(i,j)$ represents the likelihood that a message buffered at node $i$ will be delivered to node $j$, either through direct contact or through a sequence of two or more relays. We propose to compute the delivery probability based on the social contact graph constructed from the local social-tie table. In the social-tie table, each unique peerID represents a graph node, and each pair of peerIDs (or row) represents an undirected edge between two graph nodes. Assume there are $n$ entries in the social-tie table. Then, the edge weight $w_k(i,j)$ of the $k$th entry is defined as the meeting probability between two nodes $i$ and $j$ relative to other pairs of nodes in the social-tie table, and is computed as:

$$w_k(i,j) = \frac{\text{social-tie-value}_{\text{row-}k}}{\sum_{k=1}^{n} \text{social-tie-value}_{\text{row-}k}}$$  \hspace{1cm} (2)$$

where $i$ and $j$ are unique peerIDs, and $\sum_{k=1}^{n} w_k = 1$. Note that we normalise the social-tie values between 0 and 1 by dividing each social-tie value by the summation of all the values in the table. The normalised social-tie values represent the edge weights in the social contact graph. As an example, Figure 1 shows the social-tie table of node S after meeting and merging node A’s social-tie table, and the resulting social contact graph with the edge weights properly computed using Equation (2). For simplicity, the fourth column for the timestamp is not shown, and the social-tie values are in the form of integers.
In a graph, two nodes can be connected by many different paths. However, as described in the routing strategy subsections, our multicast and anycast scheme are based on a single-copy routing strategy to reduce the replication cost. Therefore, only the path with the highest delivery probability will be selected. Thus, we are interested in computing the delivery probability through the most probable path. Given a \( \text{PATH}_k(i,j) \) between two nodes \( i \) and \( j \), the delivery probability over the \( k \)th path can be computed as:

\[
P_k(i,j) = \prod_e w(e), \quad \forall e \in \text{PATH}_k(i,j)
\]  

One way to compute the delivery probability over the most probable path is to find all the paths between \( i \) and \( j \), compute the delivery probability through each path, and then select the maximum value. Suppose there are \( n \) paths between \( i \) and \( j \). Then, the delivery probability through the most probable path \( Q(i,j) \) can be computed as:

\[
Q(i,j) = \max \{ P_k(i,j), 1 \leq k \leq n \}
\]  

However, this approach is computationally infeasible as finding all the paths between two nodes on an undirected graph is NP-hard. This can be proven as follows:

It is shown in [27] that finding the longest path between two graph nodes in an undirected graph is NP-hard. Suppose that we could find all the paths between two nodes in polynomial time. Then, by sorting the results in polynomial time, we could find the longest path, also in polynomial time. This contradiction shows that finding all the paths between two graph nodes is NP-hard.

Alternatively, we propose to transform the problem of finding a path where the product of edge weights is maximised, into the problem of finding a path where the sum of edge weights is minimised. Note that the two problems are equivalent as shown below:

\[
\arg \max_{\text{PATH}_k} P_k(i,j) \equiv \arg \max_{\text{PATH}_k} \log(P_k(i,j))
\]

\[
= \arg \min_{\text{PATH}_k} -\log \left( \prod_e w(e) \right), \quad \forall e \in \text{PATH}_k
\]

\[
= \arg \min_{\text{PATH}_k} \sum_e -\log(w(e)), \quad \forall e \in \text{PATH}_k
\]

A polynomial-time algorithm such as Dijkstra’s algorithm can then be used to find the least-cost path (which is the most probable path) and the corresponding delivery probability over that path. Note that the edge weights need to be transformed by negating the log values of the current edge weights.

As an example, consider again the contact graph in Figure 1. Suppose that S’s objective is to deliver a message to E. Thus, upon meeting A, S is interested in computing the delivery probability from A to E. S, in turn, runs Dijkstra’s algorithm using the log-transformed edge weights (not shown on the graph). The resulting least-cost path is \( \text{PATH}_{A\rightarrow D\rightarrow E} \) with the cost (summation of logs) \( = (-\log 2/15) + (-\log 3/15) = 1.574 \). Note that the cost of \( \text{PATH}_{A\rightarrow B\rightarrow D\rightarrow E} \) is \( (-\log 1/15) + (-\log 4/15) + (-\log 3/15) = 1.637 \).

![Figure 1](image-url)
2.449. The delivery probability is the product of non-transformed edge weights on \( PATH_{A \rightarrow D \rightarrow E} \), which is \( \frac{2}{15} \times \frac{3}{15} = 0.0267 \). For comparison, the product of non-transformed edge weights on \( PATH_{A \rightarrow B \rightarrow D \rightarrow E} \) is \( \frac{1}{15} \times \frac{4}{15} \times \frac{3}{15} = 0.0036 < 0.0267 \). This confirms that our approach correctly identifies the most probable path and computes the delivery probability over that path.

3.4. Multicast routing strategy

We consider a single-copy model in which, at any point in time, there is at most one copy of the data packet per multicast destination in the network. That is, the number of copies of a packet can be at most equal to the number of multicast destinations since each destination needs to have its own copy of the packet. Furthermore, copies that are intended for different destinations can be scattered at different nodes. Suppose that there are \( D \) multicast receivers. Our key idea for multicast routing is to have the source node \( S \) delegate a subset \( Q \subseteq D \) to an encounter node \( E \) subject to the following forwarding constraint:

\[
\forall x \in Q, P(E, x) > (1 + \beta) \cdot P(S, x)
\]

where \( P(i, j) \) is the multi-hop delivery probability from node \( i \) to node \( j \), and \( \beta > 0 \) (set in our experiments as 0.3) is used to avoid replicating to an encounter node with a too similar delivery probability. Subsequently, each intermediate node follows the same strategy on a smaller subset, until the multicast data is delivered to all multicast members. For example, in Figure 2, at time \( t_1 \), \( S \) encounters two nodes \( n_1 \) and \( n_3 \). After exchanging and merging social-tie tables, \( S \) computes the delivery probabilities from \( n_1 \) and \( n_3 \) to the multicast members. \( S \) then finds that \( n_1 \) has a higher delivery probability to \( D_1 \) than itself, and \( n_3 \) has a higher delivery probability to \( D_2, D_3, \) and \( D_4 \) than itself. Thus, \( S \) creates two copies of the packet. One copy is sent to \( n_1 \) with a header that includes \( D_1 \) in the final destination set. The other copy is sent to \( n_3 \) with a header that includes \( D_2, D_3, \) and \( D_4 \) in the final destination set. To obey the single-copy model, \( S \) removes \( D_1, D_2, D_3, \) and \( D_4 \) out of its destination set. Since the destination set at \( S \) becomes empty, \( S \) removes the data from its caching buffer. At time \( t_2 \), \( n_1 \) meets \( n_2 \), and \( n_3 \) meets \( D_2 \) and \( n_4 \). Since \( n_2 \) has a higher delivery probability to \( D_1 \) than \( n_1, n_1 \) forwards its only copy to \( n_2 \). Similarly, \( n_3 \) duplicates two copies, one with a header that includes \( D_2 \) and \( D_3 \) to be sent to \( D_2 \), and the other with a header that includes \( D_4 \) to be sent to \( n_4 \). At time \( t_3 \), direct transmissions are performed upon meeting multicast members, ignoring the delivery probability comparison step.

3.5. Anycast delivery probability metric

In this subsection, we introduce two relay selection metrics for anycast routing based on the direct one-hop and multi-hop delivery probability. We consider an anycast group \( D \) of size \( n \), in which \( D = \{d_1, d_2, \ldots, d_n\} \).
3.5.1. Anycast direct encounter metric
ADEM is defined as the probability of directly encountering at least one node in the anycast group. We compute this metric by first normalising the social-tie values between 0 and 1. Let $M(i, j)$ denote the meeting probability between two nodes $i$ and $j$. Then, based on our earlier analysis from Section 3.3, $M(i, j)$ is the normalised social-tie value between $i$ and $j$. That is, $M(i, j) = w(i, j)$, where $w(i, j)$ is computed as in Equation (2). The probability that a node $x$ meets any node in set $D$ can be computed as:

$$ADEM(x, D) = 1 - \prod_{d \in D} (1 - M(x, d))$$ (6)

where $\prod_{d \in D} (1 - M(x, d))$ is the probability that $x$ does not meet all members of the group. A metric similar to ADEM has been proposed in [23]. Yet, ADEM differs in terms of how $M(x, d)$ is computed.

In this paper, we introduce ADEM primarily for comparison purposes against anycast social distance metric, which we describe next.

3.5.2. Anycast social distance metric
ASDM is defined as the probability of successfully delivering a packet to any members of an anycast group based on social distances to members of the group. The social distance $SD(x, d_i)$ from a node $x$ to a member $d_i \in D$ is formulated as:

$$SD(x, d_i) = 1 - P(x, d_i)$$ (7)

where $P(x, d_i)$ is the multi-hop delivery probability over the most probable path from $x$ to $d_i$ (see Section 3.3). This formulation favours an encounter node $x$ with a high multi-hop delivery probability to $d_i$ (i.e. a small social distance toward the destination). Intuitively, in order to increase the chance of reaching any group member in an unpredictable network, we should favour a relay node that is ‘socially’ close to the network area where more group members reside. Inspired from [28], we model ASDM based on the individual social distance to each group member as shown below:

$$ASDM(x, D) \propto \sum_{d \in D} \frac{1}{SD(x, d)^\alpha}$$ (8)

where $0 \leq SD(x, d) \leq 1$ and $\alpha > 0$. The control parameter $\alpha$ determines the balance between forwarding in the direction where most group members reside and forwarding toward a few close members. While a small value of $\alpha$ favours the former direction, a larger value of $\alpha$ prefers the latter direction. Depending on the network characteristic, $\alpha$ should be tuned carefully so that an anycast packet can be forwarded in the direction that has a high chance to meet a group member with a short delay. In our experiments, $\alpha$ is set to 1.5. The value of $ASDM(x, D)$ ranges from 0 to $\infty$, and it is $\infty$ when $x$ is an anycast group member.

As an example, consider an anycast group $D = \{d_1, d_2, d_3, d_4\}$. Suppose that the current node with the anycast packet meets two relay nodes $u$ and $v$ that have the multi-hop delivery probabilities to the four group members as follows: $P(u, d_i) = [0.2, 0.4, 0.3, 0.1]$ and $P(v, d_i) = [0.1, 0.05, 0.1, 0.5]$. The corresponding social distances are $SD(u, d_i) = [0.8, 0.6, 0.7, 0.9]$ and $SD(v, d_i) = [0.9, 0.95, 0.9, 0.5]$. As we can see, $u$ is socially closer to $d_1$, $d_2$, and $d_3$ than $v$, whereas $v$ is closer to $d_4$ than $u$. Also, note that the shortest distance between a relay to group $D$ is $SD(v, d_4) = 0.5$. Since $SD(v, d_4) = 0.5$ is not significantly shorter than $\min \{SD(u, d_i), 1 \leq i \leq 4\} = 0.6$ (their difference is only 0.1), ASDM metric (with $\alpha = 1.5$) will prefer $u$ (density) over $v$ (proximity) as a relay node ($ASDM(u, D) = 6.43 > ASDM(v, D) = 6.25$). That is, ASDM will select the direction toward an area where most group members reside. Note that if $SD(v, d_4) \leq 0.4$ (i.e. when $P(v, d_4) \geq 0.6$), ASDM will instead favour $v$ (proximity) over $u$ (density) as a relay node. This decision is justifiable since the value of $SD(v, d_4)$ is now in a safe range, in which we have a certain confidence in reaching the closest anycast member $d_4$ despite that it may be socially isolated from other group members.
Compared to ADEM metric, ASDM is more conservative. That is, ASDM is less attracted toward the closest, single group member than ADEM. Rather, ASDM attempts to balance between moving toward an area with many group members (which is more robust) and moving toward a few closer members (which is more efficient). Furthermore, by considering long routing paths with multi-hop delivery probabilities, ASDM has a broader view for forwarder selection than ADEM, which only considers one-hop routing paths through direct meeting between a relay node and a group member.

To see the benefit of using multi-hop delivery probabilities in the formulation of an anycast routing metric, considering the following example. Suppose that the source node $s$ meets node $x$, and it aims to deliver a packet to an anycast group $D = \{d_1, d_2\}$. Figure 3 shows the contact graph constructed by $s$. Node $x$ has no edges to anycast members since it has not encountered any group members in the past. Thus, $ADEM(x, D) = 0$. Consequently, $s$ mistakenly identifies $x$ as a bad relay node, even though $x$ may often meet node $v$, which has strong connections to anycast members. In contrast, ASDM takes into account a broader view of the network. ASDM uses multi-hop delivery probabilities from $x$ to $d_1$ and $d_2$ in computing social distances, which results in $ASDM(x, D) > 0$. As a result, $s$ correctly identifies $x$ as closer to the anycast group, and thus selects $x$ as a next relay node, which is a desired behaviour.

### 3.6. Anycast routing strategy

We use a single-copy model in which, at any point in time, there is at most one copy of the message in the network. Consider an anycast group $D$. When a message carrier node $u$ encounters a node $v$, they exchange and merge their social-tie tables. If node $v$ is an anycast member, node $u$ will perform a direct delivery of the message to $v$ and remove the message from its caching buffer. Otherwise, node $u$ needs to determine if $v$ is a good relay node. It does that by first computing the delivery probability from $v$ to the anycast group using one of the metrics introduced in Section 3.5. Node $u$ then forwards the message to $v$ if $v$ has a larger delivery probability metric than $u$, i.e. $ADEM(v, D) > ADEM(u, D)$ or $ASDM(v, D) > ASDM(u, D)$ depending on the forwarding metric used. If the message is forwarded to $v$, $u$ will remove the message from its caching buffer to comply with the single-copy model. Otherwise, $u$ continues to hold the message until the next meeting opportunity arises.

In our routing strategy, we take a group-based view at each intermediate relay node to make a forwarding decision. That is, we always consider the movement behaviour of the entire anycast group at each intermediate routing step. This approach is suitable in highly unpredictable network environments. In contrast, another approach to anycast routing is to have the source node simply picks the ‘best’ group member according to some metric, and then use unicast techniques to reach it. We name a family of this approach unicast-based anycasting (UBA). UBA may not perform well since the best group member is likely to change over time, and unicasting denies intermediate relay nodes the opportunity to react to this change. In Section 4, we will evaluate our proposed...
Table 1. Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(x)$ parameter $z$</td>
<td>2</td>
</tr>
<tr>
<td>$F(x)$ parameter $\lambda$</td>
<td>$e^{-4}$</td>
</tr>
<tr>
<td>ASDM parameter $\alpha$</td>
<td>1.5</td>
</tr>
<tr>
<td>RxNoiseFigure</td>
<td>7</td>
</tr>
<tr>
<td>TxPowerLevels</td>
<td>1</td>
</tr>
<tr>
<td>TxPowerStart/TxPowerEnd</td>
<td>12.5 dBm</td>
</tr>
<tr>
<td>$m_{\text{channelStartingFrequency}}$</td>
<td>2407 MHz</td>
</tr>
<tr>
<td>TxGain/RxGain</td>
<td>1.0</td>
</tr>
<tr>
<td>EnergyDetectionThreshold</td>
<td>$-74.5$ dBm</td>
</tr>
<tr>
<td>CcaModelThreshold</td>
<td>$-77.5$ dBm</td>
</tr>
<tr>
<td>RTSThreshold</td>
<td>0 B</td>
</tr>
<tr>
<td>CWMin</td>
<td>15</td>
</tr>
<tr>
<td>CWMax</td>
<td>1023</td>
</tr>
<tr>
<td>ShortEntryLimit</td>
<td>7</td>
</tr>
<tr>
<td>LongEntryLimit</td>
<td>7</td>
</tr>
<tr>
<td>SlotTime</td>
<td>$20 \mu$s</td>
</tr>
<tr>
<td>SIFS</td>
<td>$20 \mu$s</td>
</tr>
</tbody>
</table>

Table 2. Characteristics of the Cabspotting and ZebraNet traces.

<table>
<thead>
<tr>
<th>Trace</th>
<th>Number of nodes</th>
<th>Duration (s)</th>
<th>Area (m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cabspotting</td>
<td>116</td>
<td>3600</td>
<td>$5700 \times 6600$</td>
</tr>
<tr>
<td>ZebraNet</td>
<td>40</td>
<td>5000</td>
<td>$1500 \times 1500$</td>
</tr>
</tbody>
</table>

group-based anycast routing scheme against UBA. We implement UBA by computing the multi-hop delivery probabilities $P(s, d)$ from the source node $s$ to each anycast group member $d \in D$. Node $s$ then selects $d_{\text{best}} = \max \{P(s, d), d \in D\}$ as the best group member, and unicasts the message to $d_{\text{best}}$ by forwarding to an encounter node $e$ that has $P(e, d_{\text{best}}) > P(s, d_{\text{best}})$. Each intermediate node keeps the final destination $d_{\text{best}}$ unchanged, and follows the same relay strategy until the message is delivered to $d_{\text{best}}$.

4. Performance evaluation

In this section, we evaluate the performance of the proposed DTB-based multicast routing and ASDM-based anycast routing scheme in a packet-level simulation, using a real-world mobility trace. We first describe the simulation setup, followed by the metrics used and the results.

4.1. Simulation setup

We implement the proposed routing protocols using the NS-3.19 network simulator. We adopt the IEEE 802.11g wireless channel model and the PHY/MAC parameters as listed in Table 1. To obtain meaningful results, we use two real-life mobility traces: the Cabspotting trace [29] and traces from the ZebraNet wildlife tracking experiment [30]. Cabspotting consists of global positioning system (GPS) coordinates of 483 cabs, collected over a period of three consecutive weeks. For our studies, we select an NS-3 compatible trace file from downtown San Francisco (area dimensions: $5700 \times 6600$ m) with 116 cabs, tracked over a period of 1 h [31]. ZebraNet dataset contains animal position information in UTM format collected from seven sensor nodes on seven different zebras at Sweetwaters Game Reserve near Nanyuki, Kenya. We use the ZebraNet trace generator written by Yong Wang [30] to generate a ZebraNet mobility trace with 40 nodes in an area of $1500 \times 1500$ m. Table 2 summarizes the statistics of the two traces.
We evaluate DTB-based multicast routing against the following schemes:

- **EBMR [13]** is a single-copy multicast routing protocol. It is based on PROPHET DTN unicast routing [32], which considers the delivery probability to multicast receivers within two hops.
- **Epidemic routing [33]** is a flooding-based multiple-copy routing algorithm. The multicast implementation of Epidemic routing creates a copy of the data, bundles all multicast destinations into the copy, and forwards the copy to each encounter node. Epidemic routing typically has the highest delivery ratio and lowest delay, but also has the highest delivery cost.

We evaluate ASDM-based anycast routing against the following schemes:

- **ADEM** was introduced in Section 3.5.1. It is a single-copy anycast routing protocol, which forwards the message based on the probability of directly encountering any one group member.
- **UBA** was introduced in Section 3.6. It extends the unicast routing protocol, and routes the message to a fixed destination, to which the source node has the highest multi-hop delivery probability at the time of message creation.
- **Epidemic routing [33]** is a flooding-based multiple-copy routing algorithm, which has a delivery ratio and delay that approach the theoretical maximum, but also has the highest delivery cost.

### 4.2. Evaluation metrics

We use the following metrics for evaluation:

- **Delivery ratio**: for multicast routing, it is the proportion of destinations that receive the data item out of the total number of intended destinations. For anycast routing, it is the proportion of data items that are received by an anycast group out of the total number of unique data items generated.
- **Average delay**: the average interval of time required for a multicast destination (or an anycast group) to receive the data item.
- **Average cost**: the average number of relays required for a multicast destination (or an anycast group) to receive the data item.
4.3. Multicast routing performance

Figure 4(a) and 5(a) compare the delivery ratio among the multicast routing schemes. As expected, Epidemic has the highest delivery ratio of up to 94%. By using a flooding method, Epidemic has a high chance to successfully deliver a data item to a multicast group, even when the group is comprised of hard-to-reach members. DTB and EBMR deliver up to 78% and 63% of the messages, respectively. Figure 4(b) and 5(b) depict the average delay. Again, Epidemic has the best delivery delay, followed by DTB. DTB successfully delivers messages by up to 22% less time than EBMR. Note that DTB considers long routing paths that may generate faster routes to multicast destinations. Lastly, average cost is compared in Figure 4(c) and 5(c). Epidemic has the highest cost as it floods the message to every network node. DTB has a slightly higher cost than EBMR because DTB may use longer paths for faster delivery.

4.4. Anycast routing performance

Figure 6(a) and 7(a) show the delivery ratio of the compared anycast routing schemes. As we increase the simulation time, the delivery ratio of all schemes is improved. Epidemic has the best delivery ratio of up to 88% due to its flooding strategy. ASDM outperforms ADEM and UBA by up to 15 and 22%, respectively. The improvement of ASDM over ADEM is a result of two factors. First, the multi-hop delivery probability generates more path choices to reach a group member than the direct (one-hop) delivery probability. Second, from this pool of available paths, the function of social distances to group members allows the selection of the most probable path to reach at least one group member. Lastly, UBA performs the worst because the best group member at the time of message creation is likely to change over time, and thus, unicasting to this member is not guaranteed to be successful within the simulation time.
Figure 6(b) and 7(b) compare the average delay. ASDM delivers messages by up to 11% and 17% less time than ADEM and UBA, respective. This is because ASDM considers multi-hop forwarding opportunities, which enable a packet to travel through a fast route to an anycast group member. Furthermore, a well-tuned parameter $\alpha$ in ASDM function, while favouring density of group members over proximity, can drive a packet to the nearest group member if it possesses a high successful delivery probability. This has the effect of reducing the routing delay.

Lastly, average cost is depicted in Figure 6(c) and 7(c). The cost of ASDM is lower than ADEM, UBA, and Epidemic by up to 11%, 23%, and 39%, respectively. UBA has a high cost because it is vulnerable to the movement of the best node that is selected at the time of message creation. ADEM has a lower cost than UBA because ADEM takes a group-based view for anycast routing, and thus is not vulnerable to the movement of a particular group member. Finally, ASDM has the lowest cost. Compared to ADEM, ASDM is least susceptible to the movement of the entire anycast group due to its consideration of a broad network view based on multi-hop delivery probabilities and the density of group members. Although ADEM always prefers proximity, it may incur a higher cost than ASDM since the chosen path can be highly unstable due to its limited network view (using one-hop delivery probability). Furthermore, the closest anycast member on the volatile, short path may often move away from the current message carrier node, causing the carrier to recompute the path and perform more forwarding than necessary.

5. Conclusion and future work

In this paper, we proposed single-copy routing strategies for multicast and anycast. Multicast routing employs a dynamic tree branching technique that allows routing paths to be efficiently shared among multicast destinations, thus reducing the transmission cost. Anycast routing selects relays based on social distances to group members. It optimises the efficiency and robustness of message delivery by balancing the trade-off between a short path to the closest, single group member and a longer path over which many other group members are accessible. Furthermore, the relay selection metrics are derived from multi-hop delivery probabilities, which offer a broader view for forwarder selection, and thus making our schemes less vulnerable to the volatile movement of group members. Experimental results show that our schemes achieve a high delivery ratio, low delay, and low (or comparable) transmission cost compared to other group communication strategies.

In future work, we plan to investigate other classes of monotonically non-increasing weighing functions that are used to evaluate the social-tie metric. We also plan to relax the assumptions of infinite storage, and evaluate the schemes in conjunction with buffer management policies in other DTN networks such as ocean sensor networks and military networks.
Disclosure statement

No potential conflict of interest was reported by the authors.

References


