Adaptive Backbone-based Routing in Delay Tolerant Networks

Shuhui Yang
Department of Math, Computer Science, and Statistics
Purdue University Calumet, Hammond, USA
Email: yang246@purdue.edu

Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, USA
Email: jiewu@temple.edu

***Abstract*—In this paper, we develop a localized algorithm for the routing problem in delay tolerant networks (DTNs). We first design a modeling approach to derive a weighted graph from the DTN, taking into consideration the obtained history contact information of the nodes. This modeling provides adaptiveness by accommodating diverse network predication characteristics. Based on the derived weighted graph, we then put forward the concept of a** *delay tolerant network backbone* **for the DTN. When only the nodes in the backbone forward data, the routing in the DTN is achieved with the optimal performance in terms of the expected end-to-end delivery latency. This work is inspired by the widely used virtual backbone-based routing for mobile ad hoc and sensor networks. In DTNs with intermittent connectivity, we explore the meeting frequency between nodes for the construction of the backbone. Accordingly, we develop the** *delay tolerant connected dominating set (DTCDS)* **as an approximation to the delay tolerant network backbone, and further formalize the problem of minimum equally effective DTCDS. A localized heuristic algorithm for constructing an efficient DTCDS is proposed. Performance studies include a theoretical analysis and a comprehensive simulation on the proposed algorithm.**

***Index Terms*—Broadcast, connected dominating set (CDS), delay tolerant networks (DTNs), wireless communication.**

I. Introduction

Recently, there have been many research activities in the area of intermittently connected wireless networks known as delay tolerant networks (DTNs) [28]. Compared with the similar wireless and mobile networks, such as mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs), where a connection is presented for most network parts and for almost all time, DTNs are featured by the intermittent connection. Therefore, the routing protocols for MANETs, such as AVDO and DSR, that assume a contemporaneous end-to-end path between any source and destination pair, do not work in DTNs. Novel routing schemes that are dedicated to DTNs need to be developed. DTN-oriented routing also will serve as an effective complement to traditional centralized wireless communication. For example, the research on mobile cellular traffic offloading explores possible DTN routing, as to cope with huge traffic growth, and provides cost-efficiency [14].

The disconnection in DTNs is largely due to the large-scale movement of wireless nodes. On the other hand, message delivery in such networks is also realized with the help of node movements. A "store-carry-forward" mode is adopted where a node receives a message, stores the message and carries it with itself until it meets with other nodes, i.e. a connection presents, and then forwards the message. The current algorithms for DTN routing can be classified into several categories based on whether the future movement and connection status of the network is known/predictable [28]. Deterministic routing, history or predication-based routing, and epidemic routing, are each dedicated to a different degree of obtained knowledge regarding the prediction of the future movement pattern of nodes. In these approaches, a node, upon the reception of each message, based on available network status information, uses different mechanisms to predict future contacts with other nodes and then decides whether and to whom to forward the message.

For most DTNs, node mobility is not entirely random, especially when the wireless devices are carried by humans, and hence, the social interactions of these carriers will directly affect the communication among the devices and offer some movement patterns for the nodes in the DTNs. One promising way of predicting future contact likeliness is to aggregate contacts in the past to a social graph and then use metrics from complex network analysis (i.e. centrality and similarity) to make forward decisions [2]. There are two potential problems of this method. One is that with a link between two nodes indicating that they have seen each other in the past, the quantity of the meeting frequency cannot be specified. A weighted graph is expected to provide better prediction. The other is that the approach of a node, upon the reception of each message, is to run the algorithm to decide forwarding status for this specific message, which is costly. A more general forwarding/non-forwarding status for each node is expected to provide higher efficiency.

In this paper, we develop an adaptive backbone-based routing approach for DTNs with diverse connection predication characteristics. We first design a modeling approach for DTNs. When the past meeting frequency of two nodes is known, we can assign an edge between these two nodes and use the frequency as the weight of this edge. In this way, a weighted graph can be derived from a DTN with a certain degree of knowledge on node movements. The edge weights are used to predict expected delivery latency. Inspired by the concept of virtual network backbone in MANETs, we propose the *delay tolerant network backbone* for DTNs. The derived graph is more dense (i.e. has a higher average node degree) if more

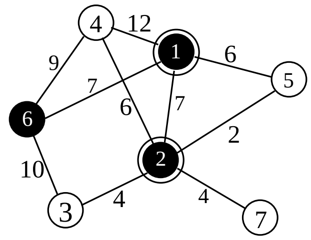


Fig. 1. A sample DTCDS in dark nodes (nodes 1, 2, and 6). Nodes 1 and 2 (in double-circle) form a traditional connected dominating set.

information on node meetings is obtained.

The virtual backbone in MANETs is virtual in that the connections are wireless, while this proposed virtual backbone for DTNs is "virtual" in that, over time, the intermittent connections among nodes form a network backbone and can provide delivery for delay tolerant messages. Routing in DTNs based on the virtual backbone will be adaptive since the construction of the backbone depends on the collected information of nodes' past meeting frequency. With more knowledge, the generated backbone will be more efficient, i.e. relatively smaller in size. In an extreme case when no prediction is available, the backbone will be the entire network, where only epidemic routing or other flooding based methods then may be applied.

Due to the similarity between virtual network backbone for MANETs and the delay tolerant network backbone for DTNs, we further put forward the *delay tolerant connected dominating set (DTCDS)* concept to approximate the delay tolerant network backbone. Taking both efficiency and effectiveness into consideration, we accordingly formalize the *minimum equally effective DTCDS problem*. That is, a DTCDS that is with the minimum nodes (efficiency) and maintains the same expected message delivery latency with that in the original DTN (effectiveness).

A localized solution is even more desired in DTNs than in MANETs, since the information concerning the whole network is hard to collect in an intermittently connected network. We design the *accumulated node coverage condition* for the minimum equally effective DTCDS problem, where each node, after obtaining the meeting frequency with other nodes, decides whether to serve as DTCDS node and help with forwarding or withdraw in a localized manner.

Figure 1 is a sample DTCDS in dark nodes. The weighted graph is derived to represent the DTN where the number over each edge is the past meeting frequency of the two end nodes. Two nodes without an edge are not meeting. With the forwarding by only dark nodes, any message can be delivered end-to-end. The expected end-to-end delay remains the same with that of when all nodes forward messages (see Section III for a detailed calculation on this example).

This paper focuses on using the DTCDS concept to construct a delay tolerant network backbone for DTN routing. The main contributions are

1) *Delay tolerant network modeling.* We design a modeling approach to derive a weighted graph for the DTN, which provides adaptiveness in regard to the different degrees of available knowledge on node movement pattern and the prediction on delivery latency.
2) *The delay tolerant network backbone concept.* We put forward the concept of a delay tolerant virtual network backbone that can be constructed on the derived weighted graph, for effective and efficient routing in DTNs.
3) *The DTCDS problem*. We develop the delay tolerant connected dominating set (DTCDS) concept as an approximation to the delay tolerant network backbone, and formalize the minimum equally effective DTCDS problem with analysis.
4) *Heuristic localized solutions to the minimum equally effective DTCDS problem.* We propose an approach to construct a small and efficient DTCDS for routing.
5) *Performance analysis.* We study the performance of the proposed methods through both theoretical analysis and simulations.

The remainder of the paper is organized as follows: Section II introduces some preliminary works. Section III presents the proposed concept of delay tolerant connected dominating set, and formalizes the problem of minimum equally effective DTCDS. Section IV describes the proposed local heuristic algorithm. A performance study through simulation is presented in Section V. The paper concludes in Section VI.

II. Preliminaries

We review some work on DTN routing and connected dominating set (CDS) construction approaches in MANETs.

A. Routing in DTN

With intermittent connection, any routing approach developed for DTNs implements message delivery in the manner of "store-carry-forward" [4]. Different routing approaches have different ways for the node with a message to decide whether to forward, whom to forward to, and how many times to forward when connections are presented. These decisions are made according to the knowledge obtained on the future movement and connection status of the nodes. The routing approaches can be classified into the following categories based on the different degrees of the network behavior prediction [28].

When no prediction is available, blind flooding is the only way to provide message delivery with optimal latency. In DTNs, the blind flooding idea is called epidemic routing [22]. In epidemic routing, a node with a message will transfer the message to all other nodes that it meets who do not have this message. The delivery ratio is very close to optimal. However, large buffer capacity is needed. In another version of epidemic routing, the source will only delivers to the destination when they meet [6]. In this case, message propagation is eliminated, but huge and usually unacceptable delay is expected. However, the resultant delay does provide a theoretical bound for the purpose of research.

When the network future movement is completely known by some kinds of prediction mechanisms, the design of routing is relatively more deterministic. Such as in [5], the message delivery path is explicitly selected with the help of the knowledge on hosts' motion. A more detailed work was proposed in [11] where different knowledge, such as contacts summary and traffic demand, is analyzed, and different routing approaches were developed based on different knowledge sets. The work in [18] models the network motion with a space-time graph, and uses dynamic programming to find shortest paths for the routes. There is another kind of DTNs where the node movement is under control (or partial control). This also belongs to the previous categories. Only other than design routing based on the predicted movement, node trajectory design is also available to assist routing, as in [13], [29].

One promising way to predict future contact likeliness is to aggregate contacts seen in the past to a social graph and use metrics from complex network analysis (i.e. betweenness centrality and similarity) to decide node forwarding [2], [19]. However, as stated in [8], the "static" social graph has the tradeoff between time-related information lost and predictive capability. In such a graph, a link means that two nodes have met in the past. The degree of the meeting frequency cannot be represented. [27] explores the social feature of each node and uses a hypercube based feature matching process to implement routing. In [15], a cyclic mobility pattern is explored to assist the routing. Most of these methods use the prediction to decide for each node whether to forward one specific message, which is cost-expensive.

The work proposed in this paper models the DTN to a weighted graph with contact aggregation for better prediction, and builds a network "backbone" for the purpose of forwarding, limiting the copies of messages and providing latency guarantee. The proposed method is adaptive in that when no prediction information is known, the backbone is then the entire network, and hence, the backbone-based routing degenerates into the epidemic routing.

B. Connected Dominating Set Construction

In mobile ad hoc networks (MANETs) or wireless sensor networks (WSNs), connectivity is at most time assumed. A very efficient way for message delivery, both unicast and broadcast, is to construct a virtual network backbone for the purpose of message forwarding. The connected dominating set (CDS) is a good approximation for the virtual network backbone. The minimum CDS (MCDS) problem is NP-complete. Global solutions, such as MCDS [3] and the greedy algorithm in [7], are based on global state information, and thus, are expensive. The tree-based CDS approach [23] requires network-wide coordination, which causes slow convergence in large-scale networks.

In local approaches, the status of each node depends on its h-hop information only with a small h, and there is no propagation of status information. Local CDS formation algorithms include Wu and Li's marking process (MP) and self-pruning rule, Rules 1 $\&$ 2 [26], several MP variations [1], CEDAR [21], multipoint relay (MPR) [20], and MPR extensions [16]. In [1], the self-pruning rule, Rule k is proposed, which is a general form of Rules 1 $\&$ 2. In Rule k, a node can be withdrawn from the CDS if all its neighbors are interconnected via k ($k \geq 1$) nodes with higher priorities. The probabilistic approximation ratio of Rule k is $O(1)$. Wu and Dai further propose the coverage condition for self-pruning in [25], which can be viewed as a generic framework for several other existing broadcasting algorithms. Some new techniques such as cooperative communication can also be explored for the construction of more efficient CDS [24].

III. Delay Tolerant Connected Dominating Set (DTCDS)

A. Network Model

In a delay tolerant network with n nodes, each node u maintains a metric called *meeting frequency (MF)* for any other node v in the network, f_{uv}, ($f_{uv} \in [0, +\infty)$), which indicates the frequency that node u and node v have met in the past. $f_{uv} = 0$ when node u and v have no chance of meeting, and $f_{uv} = +\infty$ means they are always within each other's transmission range, i.e. always connected. Therefore, the expected data delivery latency between node u and v can be calculated as

$$E(l_{uv}) = \frac{1}{f_{uv}}. \tag{1}$$

Note that when $f_{uv} = +\infty$, $E(l_{uv}) = 0$. Here, we ignore the time needed for physical transmission of the message. This is because, in a DTN, the data transmission time is insignificant compared with the delay incurred by intermittent connectivity. Note that the meeting frequency could be predetermined (when node movement is known, such as in inter-planet satellite communication networks), or self-learned over time.

This proposed network model is similar to the time-space model in [17]. However, instead of having a four-dimensional graph, we compress the time axis and merge networks (at different time spots) into one graph. Compared with the social network modeling [2], [10], a weight is added to each edge to represent not only that they have met in the past, but also how often they have met, which offers better future contact prediction. A similar DTN modeling approach is proposed in [12] where the encounter history of both meeting frequency and duration time is considered and used to generate weight for each edge between two nodes that have met. In this model, the weight of an edge represents the future contact probability, and hence, is a prediction of forwarding opportunity/delivery ratio. The proposed modeling approach in this work, with only frequency being the weight, more explicitly shows the expected delivery latency between the two nodes. Additionally, although the contact duration is also important, results in [9] show that there are high correlation coefficients of duration and frequency in many traces, and hence, we simply consider only frequency in this paper.

As shown in Figure 2, five nodes move around in a DTN, and have contacts with each other when they are in proximity.

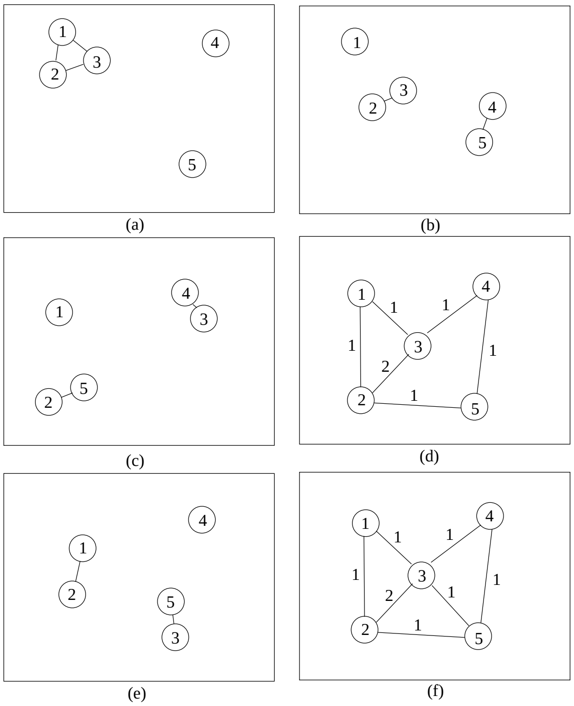


Fig. 2. Weighted graphs derived from a DTN. (a), (b), (c), and (e) are DTN status at time t_1, t_2, t_3, and t_4, (d) and (f) are the derived weighted graphs for this DTN at different times.

(a) $\sim$ (c) and (e) are their contacts records at different time t_1, t_2, t_3, and t_4. An edge in these figures indicates the wireless connection. (d) then is the weighted graph derived to represent this DTN at t_3, and (f) is the updated weighted graph at time t_4 when more contact information is accumulated. Note that an edge in the weighted graphs (d) and (f) indicates that a connection is available during a certain period of time (with some expected delay). The edge weight indicates the meeting frequency during that time. Note that, different from the graph derived from a MANET or a WSN, the weighted graph for a DTN is not a unit disk graph.

Therefore, the adaptiveness of the proposed backbone-based routing method is realized by this network modeling approach, and is delivered in an easy and natural way. Over time, if nodes get more information on meeting frequency, the derived graph will be more dense (i.e. more edges and greater average node degree). The following proposed algorithm does not need to adjust in response to the different degrees of obtained knowledge, and hence, will be less complex.

For a path $R = a_1, a_2, \ldots, a_m$, the expected path delivery latency indicates the delay of messages delivered between end nodes a_1 and a_m via the intermediate nodes along this path, which can be calculated as

$$E(l_R) = \Sigma_{i=1}^{m-1} \frac{1}{f_{i(i+1)}}. \tag{2}$$

B. A Delay Tolerant Network Backbone

With the above definitions and assumptions, we may use an undirected, weighted graph, $G = (V, E, w)$, to represent a DTN, where each mobile device is a node, in set V, and for any two nodes u and v, if $f_{uv} > 0$, there is an edge between u and v in set E. In real application, we may also include an adjustable system parameter f_{thres} as a meeting frequency threshold. Such that only when $f_{uv} > f_{thres}$, the edge between u and v exists. Larger f_{thres} helps to reduce the number of edges, simplify the graph, and hence reduce the running time of the following proposed algorithm. Smaller f_{thres} generates a denser graph, and a smaller forwarding node set is expected to be achieved. The graph G is a weighted graph, with $w(u, v) = f_{uv}$.

In such a graph, edges between nodes indicate that these two nodes will meet within a certain period of time, instead of directly connecting, as in other wired or wireless networks. Here, the virtual network backbone concept can be interpreted as a four-dimensional backbone.

A delay tolerant network backbone is then a subset of the nodes in the DTN, such that each node in the network can send messages to any other node in the network with the forwarding by only the backbone nodes. For non-backbone nodes, they are involved in the transmission only if they are the source. Note that, the forwarding here is not broadcasting since a backbone node needs to forward to its neighbor, one by one, upon encounters. Therefore, the forwarding node needs to keep record of to which neighbor it has forwarded the message and for a same message, and only forwards to each neighbor once.

The design goal here is two-fold.

1) *Efficiency*. The selected backbone should be small, such that limited copies of the message are propagated in the network.
2) *Effectiveness*. Although instead of blind flooding, only a partial node set forwards the message, the expected end-to-end delivery latency should not be affected.

C. The Delay Tolerant Connected Dominating Set

The connected dominating set is widely used to construct an efficient virtual network backbone in MANETs. Inspired by this, we define a delay tolerant connected dominating set (DTCDS), to approximate the delay tolerant network backbone of a DTN.

The main idea is that, given the weighted graph derived using the above method, a subset of nodes are selected such that they are connected among themselves, and every non-selected node has at least one selected node as a neighbor.

Definition 1: **(DTCDS)** Given a DTN, a weighted graph $G = (V, E, w)$ is derived, with V being the mobile node set, E being edges between any two nodes when their meeting frequency is greater than 0, and $w(u, v) = f_{uv}$ being the weight function. Consider a subset of nodes $V' \subseteq V$, and $E' = \{(u, v)|u, v \in V', (u, v) \in E\}$, such that

1) (V', E') is connected, and

2) $\forall v, v \notin V', \exists u, u \in V', f_{uv} > 0$.

Note that the definition of DTCDS is the same with that of the traditional CDS when the graph is given. It is special on how the graph is derived from a DTN. Also, the weights of edges do not affect the construction of a DTCDS, and will only be considered when a better DTCDS is desired. As described above, we would like to have a DTCDS with a smaller number of nodes to limit the message propagation, and hence, to provide an efficient routing.

Meanwhile, we also try not to enlarge the delay of delivery. That is, the expected latency of messages delivered between any two nodes in the network through forwarding by DTCDS should be the same with that of the delivery through the entire network, i.e. blind flooding.

Definition 2: **(The Minimum Equally Effective DTCDS)** The minimum equally effective DTCDS of a given weighted graph is the one that

1) has the smallest number of selected nodes $|V'|$, and
2) $\forall v, u$, the expected delivery latency between them via $G' = (V', E')$ is the same with that via $G = (V, E)$.

Theorem 1: The minimum equally effective DTCDS contains all intermediate nodes on shortest paths of the graph.

Proof: Since the DTCDS concept is the same with the CDS after a graph is derived from the DTN, the minimum DTCDS, i.e. the minimum equally effective DTCDS with only the first requirement, is the minimum CDS of the graph.

As for the second requirement, in the original graph G with the blind flooding message delivery method, the optimal latency between any two nodes, u and v, is via a shortest path connecting them in the weighted graph. Therefore, in order to guarantee that the backbone has the same performance in terms of delay latency, if any node w is on a shortest path of any other two nodes in the graph, then w needs to be included in the resultant backbone.

Next, we prove that the set of all intermediate nodes on shortest paths is a CDS.

1) *Coverage.* Assume there is a node u in the graph without any neighbor being in the backbone. If we select any other node, v, in the graph and find the shortest path connecting them, there must be one neighbor of u on that path. Therefore, for any node in the graph, there is at least one of its neighbors in the backbone.
2) *Connectivity.* The selected nodes are all those that are on shortest paths. Therefore, they must be connected within themselves.

Since the set of all intermediate nodes on shortest paths is a CDS for the graph, and also cannot be further reduced, it also satisfies the first requirement. Therefore, the set of all intermediate nodes on shortest paths is the minimum equally effective DTCDS of the graph. ■

From the above analysis, we know that finding the minimum equally effective DTCDS is equivalent to finding the minimum set of all nodes that are on shortest paths connecting any two nodes in the graph. Given a weighted graph, this problem can be solved by finding all-pair shortest paths. However, in DTNs, only localized solutions are practical, where the global information is not available, and each node makes a decision on whether to help with forwarding based on only very limited local information. In the following, we will design an localized algorithm for the minimum equally effective DTCDS problem.

IV. LOCALIZED HEURISTIC SOLUTION

Note that in DTNs, global information is expensive and impractical. In this section, we propose a heuristic localized approach to find the minimum equally effective DTCDS in a weighted graph. A localized approach relies only on local information, i.e., the properties of nodes in its vicinity. In addition, unlike the traditional distributed approach, there is no sequential propagation of any partial computation result in the localized approach. The status and decision of each node depends on its h-hop topology, usually only for a small constant h. This local information is collected by h rounds of "Hello" message exchanges among neighbors. In DTNs, it means that the partial graph is built via h rounds of meeting with each neighbor, and is completed with time. A typical h value is 2 or 3. Also, no location information is needed in the proposed algorithm.

A. The Accumulated Node Coverage Condition

In [25], Wu and Dai proposed the coverage condition for CDS construction for undirected graphs, where a node v is unmarked if, for any two neighbors, u and w of v, a replacement path exists connecting u and w such that each intermediate node on the path has a higher priority than v. The coverage condition generates a CDS since, for each withdrawn node, there must exist a replacement path for each pair of its neighbors to maintain the connectivity.

Here, in order to satisfy both efficiency and effectiveness requirements, we extend the node coverage condition and introduce the accumulated delivery latency concept. Inspired by the flow network concept, the meeting frequency of two nodes via different paths can be added, and hence, latency can be decreased. For example, there are two paths connecting nodes u and v, R_1 and R_2. Their expected path delivery latencies are $E(l_{R_1})$ and $E(l_{R_2})$, respectively. Since the expected message delivery latency between u and v via path R_1 is $E(l_{R_1})$, the "meeting frequency" of nodes u and v, which actually indicate how often these two nodes exchange data, is therefore $\frac{1}{E(l_{R_1})}$. Also, we have the "meeting frequency" of nodes u and v via path R_2 as $\frac{1}{E(l_{R_2})}$. Then, we can see that nodes u and v have a way to exchange messages with a frequency of $\frac{1}{E(l_{R_1})}$ and have another way to exchange messages with a frequency of $\frac{1}{E(l_{R_2})}$. Therefore, the accumulated effect is that node u and v can exchange messages with the frequency of $(\frac{1}{E(l_{R_1})} + \frac{1}{E(l_{R_2})})$ in total. Accordingly, the message delivery latency between these two nodes, considering the accumulated frequency, is $1/(\frac{1}{E(l_{R_1})} + \frac{1}{E(l_{R_2})})$.

Accumulated Delivery Latency. *Assume that there are d node disjoint paths connecting node u and v, R_i, with expected* *path delivery latencies of $E(l_{R_1}), E(l_{R_2}), \ldots, E(l_{R_d})$. Then the accumulated delivery latency between u and v via these paths is*

$$E(L_{uv}) = 1/\Sigma_{i=1}^{d} \frac{1}{E(l_{R_i})}. \qquad (3)$$

Accumulated Node Coverage Condition. *Node v is unmarked if, for any two neighbors of v, u and w, a group of replacement paths, $R_1, R_2, \ldots, R_t$, exists connecting u to w such that*

1) *each intermediate node (if there is any) on any replacement path $R_i, (i = 1, \ldots, t)$ has a higher priority than v, and*
2) *the accumulated delivery latency of the group of replacement paths is smaller than or equal to the delivery latency of path u, v, w. That is*

$$E(L_{uw}) \leq E(l_R) = \frac{1}{f_{uv}} + \frac{1}{f_{vw}}. \qquad (4)$$

Based on the accumulated node coverage condition (ANCC), we design the following localized ANCC algorithm for the minimum equally effective DTCDS (in Algorithm 1).

Algorithm 1 ANCC algorithm

[1.] Each node sets up the meeting frequency for each neighbor by recording its having met with them, and exchanges this information with neighbors.
[2.] Each node determines its status (marked/unmarked) using the accumulated node coverage condition.

Note that Algorithm 1 is dynamic. When each node applies this algorithm, step 1 is first conducted and should be performed during the entire lifetime of this node. On the other hand, only when enough new/updated information is collected from step 1, step 2 will be triggered. In another words, if the movement pattern of this node is relatively stable, its status (being marked or not, serving as backbone node or not) will also be stable, and step 2 will not be performed very often.

In the example in Figure 1, node priority is its ID, and a smaller ID indicates higher priority. Based on the accumulated node coverage condition, node 3 withdraws. There are two node disjoint paths, $2 \to 1 \to 6$, and $2 \to 4 \to 6$, connecting the two neighbors (nodes 2 and 6) of node 3. The accumulated delivery latency of these two paths is $1/(\frac{1}{(1/7)+(1/7)} + \frac{1}{(1/6)+(1/9)}) = 0.19$. The delivery delay of path $2 \to 3 \to 6$ is $\frac{1}{4} + \frac{1}{10} = 0.35$, which is greater than that of the replacement paths. With the nodes on those replacement paths having higher priority than that of node 3, node 3 is safe to withdraw. Node 4, 5, and 7 all withdraw according to the accumulated node coverage condition. Node 6 fails to withdraw because, for its neighbor pair nodes 1 and 3, although there exists a replacement path $1 \to 2 \to 3$, the delivery latency is 0.39, and is greater than that of path $1 \to 6 \to 3$ (which is 0.27). Note that the coverage condition [25] for CDS will generate a virtual network backbone of nodes 1 and 2 only, as shown in the figure with the double-circle nodes.

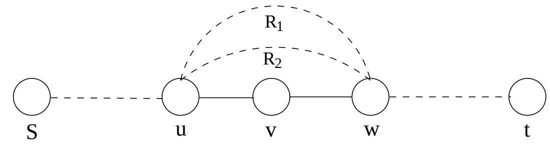


Fig. 3. Accumulated Node Coverage Condition.

Theorem 2: Given a weighted graph $G = (V, E, w)$, V' generated by ANCC algorithm constructs a DTCDS.

Proof: The DTCDS for a weighted graph derived from a DTN is the same concept with CDS in a MANET. Therefore, if a CDS is generated by the ANCC algorithm, then a DTCDS is achieved as well.

Note that the first requirement in the proposed accumulated node coverage condition is the node coverage condition in [25] for the CDS problem. That is, a marked node set V'' by requirement 1 is a CDS. Since marked node set V' by both requirements in accumulated node coverage condition has the property of $V'' \subseteq V'$, V' is a CDS for the graph, and hence, it is also a DTCDS for the original DTN. ■

B. Property

We have shown the correctness of the proposed localized algorithm. We now prove its effectiveness.

Theorem 3: Routing with only marked nodes by the ANCC algorithm forwarding has the same performance as routing with all nodes forwarding in terms of message delay.

Proof: Without loss of generality, we assume that a message is sent from node s to node t in the DTN. Using the method of blind flooding, the first arrived copy of the message is transferred through the network via a "shortest path", given the weight of each edge being the message delivery latency between these two nodes.

As shown in Figure 3, this shortest path is $s, \ldots, u, v, w, \ldots, t$. We can also assume that v is withdrawn based on the accumulated node coverage condition. That is, v will not help to forward this earliest copy of the message.

According to the accumulated node coverage condition, v is withdrawn because there exists a group of node disjoint paths with the accumulated message delivery latency smaller than or equal to that of path u, v, w. Therefore, although v, which is on the shortest path from s to t, is not forwarding, the detour the message (and its copies) takes does not delay the delivery any longer. ■

Figure 4 is a sample delay tolerant network backbone generated by ANCC. There are 50 nodes in the DTN. There are 28 dark circle nodes who are selected as the delay tolerant network backbone; 24 double-circle nodes are selected as a CDS by the node coverage condition [25].

C. Discussion

The proposed ANCC algorithm is adaptive in that the amount of selected forwarding nodes depends on the degree of obtained knowledge on the history of nodes' contacts. In the initial state before any information is collected, the derived graph only contains nodes with no edge. No node withdraws, and the routing is the extreme blind flooding case. When nodes accumulate meeting frequency with other nodes, each of them runs ANCC to decide whether to withdraw. Therefore, the expected delivery latency is always guaranteed to be optimal. With more node movement pattern learned, a smaller forwarding node set is expected to be generated.

Additionally, other existing routing methods for DTNs can also be integrated into ANCC. For example, in order to further reduce message copies, the forwarding nodes can apply probabilistic routing approaches to decide whether to relay a specific message. In this case, the ANCC is applied as a preprocessing to prune message copies. The resultant delivery latency will not be worse than when directly applying those probabilistic routings in the original DTNs. But the number of message copies is further reduced in this way.

In implementation, the accumulated node coverage condition can be relaxed so that the accumulated delivery latency of the group of replacement paths is smaller than or equal to k times of the delivery latency of path u, v, w. That is

$$E(L_{uw}) \leq k \times E(l_R) = k \times (\frac{1}{f_{uv}} + \frac{1}{f_{vw}}). \qquad (5)$$

k is a small number that is greater than 1. In this case, the property of ANCC in which it has the same delivery latency with blind flooding does not hold any more. However, the size of network backbone will decrease significantly. This performance tradeoff will be studied in the following simulation.

As mentioned in Section III, the work in [12] also involves deriving a weighted graph from the DTN, but with the weight (p_{uv}) of each edge being the contact probability of the two end nodes u and v. Note that if we replace this modeling method with our proposed one, we can still run ANCC to generate a forwarding node set on the weighted graph. However, in this case, not the expected end-to-end delivery, but the expected end-to-end delivery ratio, is guaranteed using our proposed ANCC algorithm.

V. Simulation

We evaluate the proposed algorithm ANCC via experiments on a custom simulator. We analyze the performance tradeoff of the proposed algorithm with different system settings. Since we model the DTN with a weighted undirected graph, and construct a delay tolerant network backbone on the derived graph, we directly generate weighted graphs in the simulation to implement the ANCC algorithm. The performance of the node coverage approach [25] for the traditional connected dominating set (NC-CDS) is also shown for comparison.

A. Simulation Environment

To generate a random graph, n nodes are randomly placed in a restricted 100×100 area. Graphs that are not connected are discarded (i.e., some messages will never reach their destinations even with epidemic routing). The tunable parameters in the simulation are as follows. 1) The number of nodes n. We vary the number of deployed nodes from 20 to 160 to check the scalability of the algorithms. 2) The average node degree d. d represents the density of the derived graph, and hence, the amount of accumulated knowledge regarding the meeting frequency. We use 6, 18, and 30 as the values of d to generate sparse, median, and dense networks. 3) The number of hops h. In implementation, 2, 3, or 4 hops of local information is collected for the localized algorithms. 4) The range of meeting frequency for weight assignment, r. We use 10, 50, 100 as r's value to study its effect on the performance.

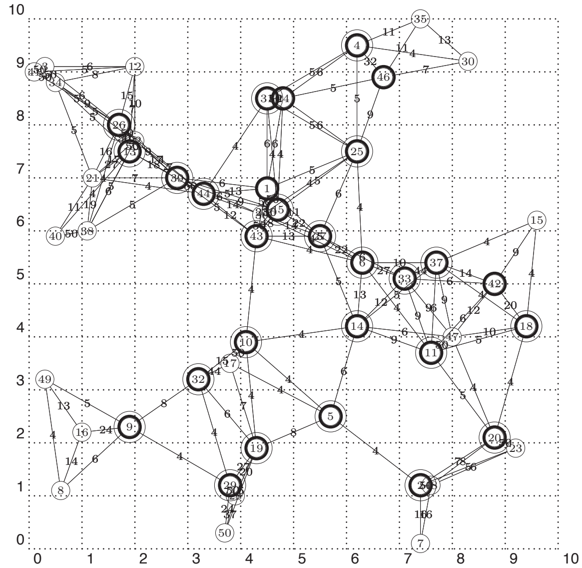


Fig. 4. A sample of delay tolerant network (50 nodes). A DTCDS in dark nodes (28 nodes). A CDS in double circle nodes (24 nodes).

The weight function that is used in the simulation is

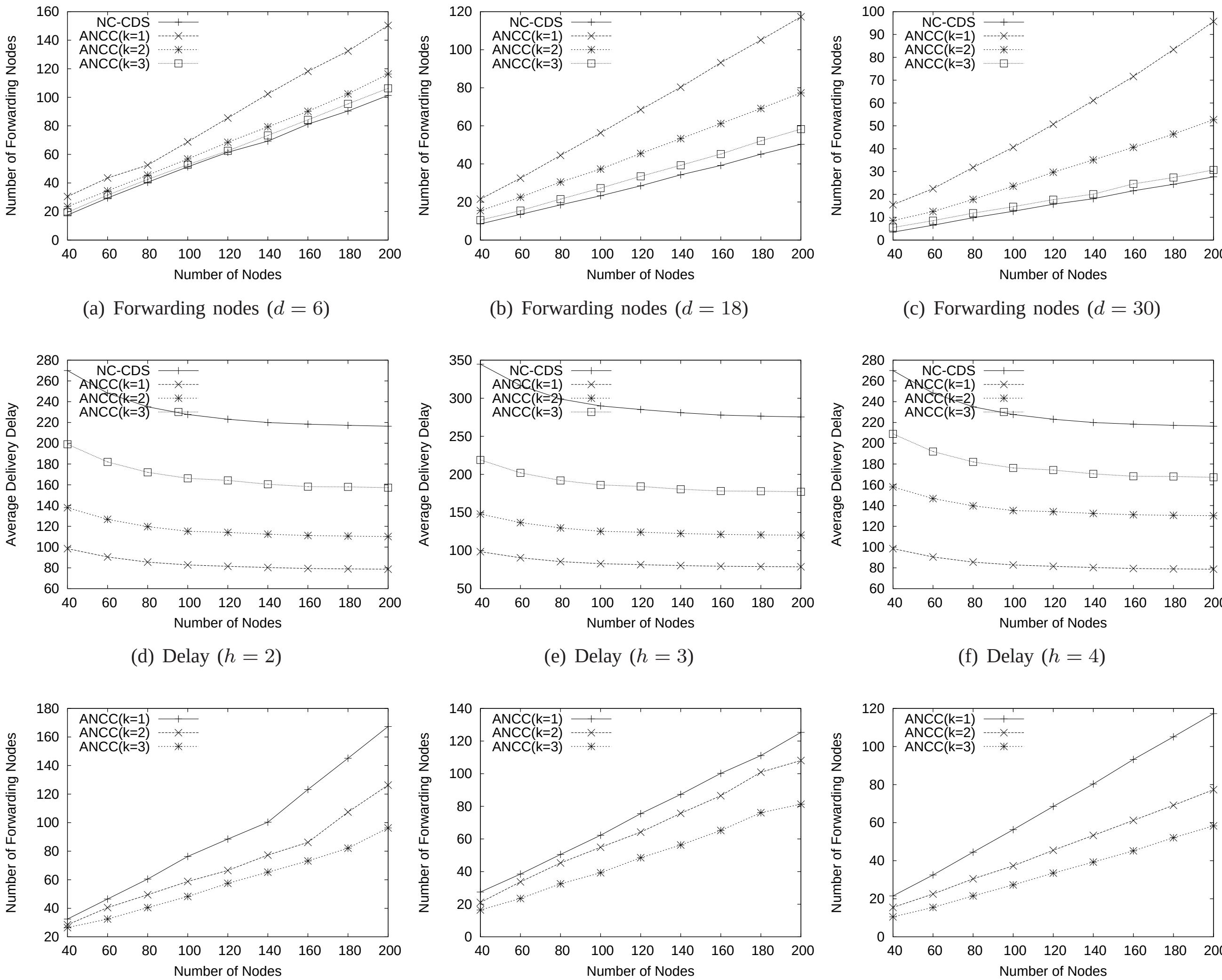
$$w(u, v) = \frac{r}{d} \qquad (6)$$

if there is an edge between nodes u and v. If we use $d(u, v)$ to denote the distance between the two nodes u and v in the graph, then

$$d = \begin{cases} 1 & \text{for } d(u,v) < 1 \\ d(u,v) & \text{for } d(u,v) \geq 1 \end{cases}$$

Assigning meeting frequency approximately inversely proportional to the distance between two nodes is reasonable since, the closer the two nodes are, the more likely they are to meet more frequently. Additionally, this assignment allows us to adjust the range of meeting frequency, i.e., all the generated weights are in the range of $(0, r]$.

The following performance metrics are compared: (1) The number of forwarding nodes selected by ANCC and NC-CDS with different system parameters. (2) The expected average message delivery latency in the network. In the custom simulator, for each tunable parameter, the simulation is repeated 100 times, or until the confidence interval is sufficiently small ($\pm 1\%$, for the confidence level of 90%).



(a) Forwarding nodes ($d = 6$) (b) Forwarding nodes ($d = 18$) (c) Forwarding nodes ($d = 30$)

(d) Delay ($h = 2$) (e) Delay ($h = 3$) (f) Delay ($h = 4$)

(g) Range ($r = 10$) (h) Range ($r = 50$) (i) Range ($r = 100$)

Fig. 5. Comparisons of ANCC and NC-CDS with different parameters (k, h, d, and r).

B. Simulation Results

Figure 5 is the comparison of the proposed ANCC and NC-CDS with different parameters. (a), (b), and (c) are the comparisons in terms of the number of forwarding nodes when the average node degree in the graph is 6, 18, and 30, with $h = 2, r = 100$. We can see that NC-CDS has the smallest forwarding node set. For ANCC, with a larger k, the forwarding node set is smaller. When $k = 3$, the size of the forwarding node set of ANCC is only slightly greater than that of NC-CDS. When the graph is denser (greater d), which means more of nodes' meeting information is obtained, the forwarding node set gets smaller for both NC-CDS and ANCC. We can see that ANCC reduces the number of forwarding nodes significantly compared with epidemic routing where all nodes are forwarding nodes.

(d), (e), and (f) are the comparisons in terms of the expected average message delivery delay (i.e., the average length of shortest paths via network backbone) when the hop number h is 2, 3, and 4, with $d = 18, r = 100$. We can see that NC-CDS has the largest message delay, and since more local information contributes to more nodes' withdrawal, a larger h leads to a larger delay for NC-CDS. For ANCC, the amount of local information will not affect the delay performance due to the "equal effectiveness" feature of the selected backbone. ANCC generates the optimal performance in terms of delivery latency, the same with that of the epidemic routing. When k is larger than 1, more local information contributes to more withdrawn nodes, and hence, greater average latency.

(g), (h), and (i) are the comparisons in terms of the number of forwarding nodes when the weight range is 10, 50, and 100, with $h = 2, d = 18$. NC-CDS is not plotted since the weight range difference does not affect the result from NC-CDS. We can see from the comparison of these three figures that a greater r corresponds to a smaller forwarding node set.

However, the sizes of forwarding node sets with r being 50 and 100 are very close to each other. r does not affect the relative performance of ANCC with other different parameters.

From the above results, we can see that ANCC reduces the forwarding node number compared with epidemic routing, and hence, improves the energy efficiency in both broadcast and unicast routing, while maintaining the same performance in terms of the expected end-to-end delivery latency. With a small but greater than 1 value of k, such as 2 or 3, the number of forwarding nodes can be further reduced, and the delivery latency will be enlarged but still acceptable. Both ANCC and NC-CDS have good performance in terms of scalability: the performance does not degenerate with an increasing number of nodes in the network.

C. Summary

Simulation results in this section can be summarized as follows:

1) ANCC reduces the size of the forwarding node set by 50% to 80%, depending on the different graph density.
2) The performance in terms of forwarding node set size of ANCC when k is 3 is very close to that of NC-CDS.
3) The performance in terms of expected average delivery latency of ANCC when k is 2 is 1.5 times of that of the optimal solution, and around 2 times when k is 3.
4) More local information helps to reduce the size of the backbone by ANCC with the fixed optimal end-to-end delay.
5) More information on nodes' encounters (and hence a more dense graph) helps to reduce the size of the forwarding node set; a small forwarding set is expected when the range of the weights is relatively large.

VI. Conclusions

In this paper, we design a modeling method that derives a weighted graph from the DTN, and put forth the concept of a delay tolerant network backbone for DTNs based on the knowledge on meeting frequency. We also develop the concept of the delay tolerant connected dominating set (DTCDS) to approximate the delay tolerant backbone in DTNs. A heuristic localized algorithm (ANCC) for DTCDS construction is proposed. The proposed ANCC generates a small network backbone that maintains the optimal performance in terms of the expected end-to-end delay. With the tuning of system parameters, the tradeoff between the two performance metrics, backbone size and expected end-to-end delay, can be achieved.

Acknowledgment

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transaction on Parallel and Distributed Systems*, 15(10):908–920, 2004.

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

[3] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad-hoc networks using a spine. In *Proc. of IC3N*, 1997.

[4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *ACM MobiHoc*, 2008.

[5] A. Ferreia. Building a reference combinatorial model for manets. *IEEE Network*, 8(1), 2004.

[6] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc networks. In *Proc. of INFOCOM*, 2001.

[7] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[8] T. Hossmann, T. Spyropoulos, and F. Legendre. From contacts to graphs: Pitfalls in using complex network analysis for dtn routing. In *First IEEE International Workshop on Network Science For Communication Networks*, 2009.

[9] X. Hossmann, T. Spyropoulos, and F. Legendre. Putting contacts into context: Mobility modeling beyond inter-contact times. In *ACM MobiHoc*, 2011.

[10] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *ACM MobiHoc*, 2008.

[11] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant networks. In *Proc. of ACM SIGCOM*, 2004.

[12] F. Li and J. Wu. MOPS: Providing content-based service in disruption tolerant networks. In *29th International Conference on Distributed Computing Systems (ICDCS)*, 2009.

[13] Q. Li and D. Rus. Communication in disconnected ad hoc networks using message relay. *Parallel Distributed Computing Journal*, 63(24-37):75–86, 2003.

[14] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng. Multiple mobile data offloading through delay tolerant networks. In *6th ACM International Workshop on Challenged Networks*, 2011.

[15] C. Liu and J. Wu. Practical routing in a cyclic mobispace. *Transactions on Networking*, 19(2):369–382, 2011.

[16] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2):111–122, 2002.

[17] A. McDonald and T. Znati. A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE JASC*, 1999.

[18] S. Merugu, M. Ammar, and E. Zegura. Routing in space and time in networks with predictable mobility. Git-cc-04-7, Georgia Institute of Technology, 2004.

[19] A. Pietilainen and C. Diot. Dissemination in opportunistic social networks: the role of temporal communities. In *ACM MobiHoc*, 2012.

[20] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. In *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, 2002.

[21] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks*, 17(8):1454–1465, Aug. 1999.

[22] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Cs-200006, Duke University, 2000.

[23] P. J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *Proc. of IEEE INFOCOM'2002*, 2002.

[24] J. Wu, M. Cardei, F. Dai, and S. Yang. Extended dominating set in ad hoc networks using cooperative communication. In *IFIP Networking*, 2005.

[25] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Transactions on Computers*, 53(10):1343–1354, 2004.

[26] J. Wu and H. Li. Domination and its applications in ad hoc wireless networks with unidirectional links. In *Proc. of International Conference on Parallel Processing*, 2000.

[27] J. Wu and Y. Wang. Social feature-based multi-path routing in delay tolerant networks. In *31st IEEE International Conference on Computer Communications*, 2012.

[28] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Survey & Tutorials*, 2006.

[29] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proc. of INFOCOM*, 2005.