Message Ferries as Generalized Dominating Sets in Intermittently Connected Mobile Networks

Bahadir K. Polat, Pushkar Sachdeva, Mostafa H. Ammar and Ellen W. Zegura
College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332
{bpolat3, psachdev, ammar, ewz}@cc.gatech.edu

***Abstract*—Message ferrying is a technique for routing data in wireless and mobile networks in which one or more mobile nodes are tasked with storing and carrying data between sources and destinations. To achieve connectivity between all nodes, message ferries may need to relay data to each other. While useful as a routing technique for wireless mobile networks in general, message ferrying is particularly useful in intermittently connected networks where traditional MANET routing protocols are not usable. A wireless and mobile network is said to possess *intrinsic message ferrying capability* if a subset of the nodes can act as message ferries by virtue of their own mobility pattern, without introducing additional nodes or modifying existing node mobility. Our goal in this work is to provide a formalism by which one can characterize intrinsic message ferrying capability. We first observe that the use of message ferries is the mobile generalization of the well-known use of connected dominating set-based routing in wireless networks. We next consider the problem of identifying the set of nodes in a mobile network which can act as message ferries by virtue of their mobility pattern. To this end, we define the concept of a connected message ferry dominating set (CMFDS) in a manner that achieves data delivery within certain performance bounds. We then develop algorithms that can be used to find such a set within a mobile, wireless network. The general CMFDS algorithm is built around a core algorithm that determines whether a single node in the network can act as a ferry. We provide some illustrative examples to show the application of our algorithm to several mobility patterns.**

I. Introduction

Message ferrying [1] is a technique used for routing data in wireless and mobile networks in which one or more (usually) mobile nodes are tasked with delivering data between sources and destinations. Fig. 1 illustrates the operation of a single message ferry as it delivers data in a wireless network. The square box denotes the ferry, which moves counterclockwise along the dashed line; the ferry exchanges data with a non-ferry node whenever the node and ferry are in proximity. Though the figure illustrates a mobile ferry and stationary nodes, the paradigm also applies when the nodes are mobile.

In general, multiple ferries may be used to achieve connectivity [2] as illustrated in Figure 2. This figure depicts three ferries – F1, F2 and F4 – each moving along its own ferry route and regularly coming in proximity with its own subset of nodes. The figure also depicts a ferry, F3, which happens to be stationary. As illustrated, each ferry provides connectivity for a subset of the nodes and communicates with other ferries through ferry-to-ferry contacts. One can see that if the ferries F1 and F2 are regularly in proximity to one another, then the overall network can provide end-to-end paths between all nodes by traversing one or more ferries. We say this network is a "connected message ferry network", understanding that the connectivity is not instantaneous in time, but requires the message ferries to carry messages and wait for the necessary proximity.

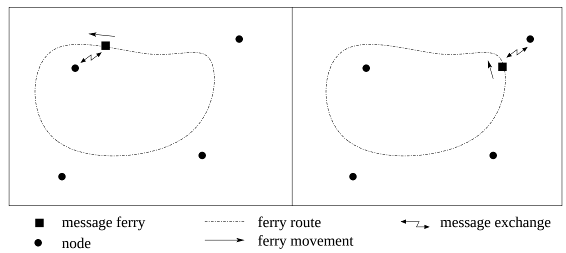


Fig. 1. An example message ferry network

While useful as a routing technique for wireless and mobile networks in general [3], message ferrying is particularly useful in intermittently connected networks such as those illustrated in these figures, where the links on an end-to-end path may not exist contemporaneously and intermediate nodes may need to store data waiting for opportunities to transfer it towards its destination. Message ferrying is among a set of routing techniques that have been developed for such networks (e.g., [4], [5], [6]). Intermittently connected networks are representable by *evolving graphs* which provide so-called *space-time paths* between sources and destinations [7], [8]. Figure 3 shows an example of network state at different time instants and illustrates the concept of space-time paths. There is a space-time path from node S to node D achieved over the time period t_0 to t_4.

Message ferrying is a rich design space. Message fer-

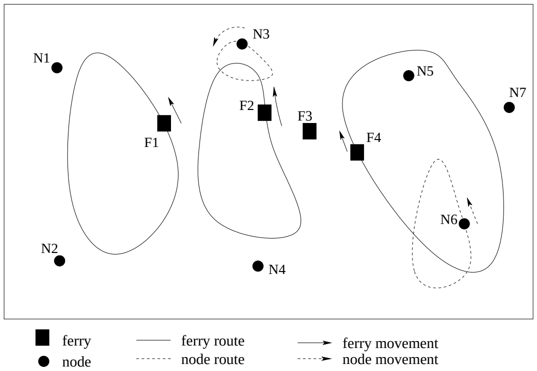


Fig. 2. An example connected message ferry network

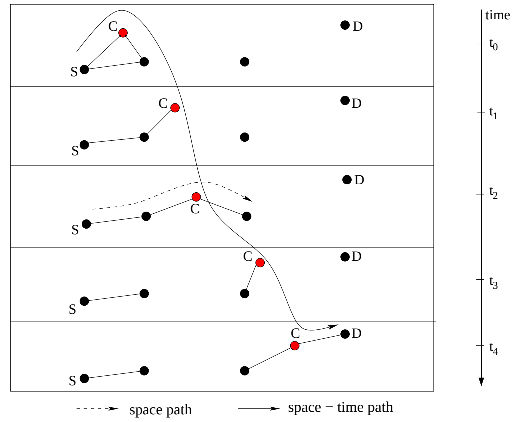


Fig. 3. An example showing evolving graph with space path from S to D at time t_2 and a space-time path from time t_0 to t_4

rying can be performed by nodes already in the network (*intrinsic message ferries*) or by nodes added explicitly for such a task. Additionally, a message ferry's mobility may be controlled to improve its ferrying performance or may be uncontrolled, allowing the message ferrying functions to be performed through the natural mobility of a node. We say that a wireless and mobile network possesses *intrinsic message ferry capability* if a subset of the nodes can act as message ferries by virtue of their own mobility pattern, without introducing additional nodes or modifying existing node mobility. Our work in this paper is concerned with characterizing such intrinsic capability for a given wireless and mobile network. In particular we are interested in determining for a given network's mobility pattern which subset of nodes, if any, can act as intrinsic message ferries, individually or collectively. Answering such a question for a particular network allows the design of a routing strategy for the network using the set of message ferries that has been identified. On a longer time scale, the lack of sufficient intrinsic message ferries could be used to trigger the dispatch of additional nodes to boost connectivity.

Interestingly, it turns out that identifying sufficient intrinsic message ferries is a generalization of the well-known connected dominating set discovery problem [9], [10], [11]. In effect, a set of intrinsic message ferries that can be used to provide end-to-end connectivity in an intermittently connected network is analogous to a connected dominating set (CDS) in a connected (non-time-varying) graph. Since a connected network is a special case of an intermittently connected network, finding a CDS is also a special case of finding an intrinsic message ferry set. Thus our work fits within a broader theme of unifying how researchers think about mobile and non-mobile networks.

In general, a dominating set (DS) of a graph $G = (V, E)$ is a subset $V' \subset V$ such that each node in $V - V'$ is adjacent to some node in V'. A *connected dominating set* (CDS) is a dominating set which also induces a connected subgraph. Das et al. have shown that the presence of a connected dominating set in an ad hoc network can provide simplified *backbone-based routing* [10] and *spine-based routing* [11]. Just as the nodes in a CDS form a connected backbone subgraph that can be used for routing, a set of intrinsic message ferries with sufficient connectivity over time can be used as a mobile routing backbone.

To illustrate the analogy, consider Fig. 2 where nodes $F1$, $F2$, $F3$ and $F4$ together form an intrinsic message ferry set that can provide connectivity among all nodes in the network. Fig. 4 shows the equivalent CDS for this network. As can be seen from Fig. 4, nodes $F1$, $F2$, $F3$ and $F4$ behave just like the *gateway* nodes of a conventional dominating set, with the modification that the ferries are connected to their neighbors over time.

This paper defines the connected message ferry dominating set (CMFDS) problem and develops a heuristic to find a minimum-size CMFDS, given a model for the connectivity between nodes over time. Specifically, Section 2 provides a detailed formalization of message ferrying concepts. We give a formal definition of a message ferry, then we present the concept of a message ferry dominating set (MFDS) — a *space-time* dominating set constituting nodes that behave as intrinsic message ferries. We further define a connected message ferry dominating set (CMFDS), that can be used to provide message ferrying connectivity between all sources and destinations in a mobile ad hoc network. Finding a CMFDS in an intermittently connected network is analogous to finding a CDS in a stationary network, a well known NP-complete problem. There are many existing approximation algorithms to determine the minimum connected dominating set such as by Wu and Li [9] [12], Stojmenovic et al. [13], Alzoubi et al. [14] and Das

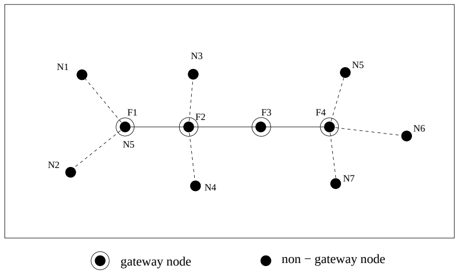


Fig. 4. Equivalent CDS for the network shown in Fig. 2

et al.[10] [11]. In Section 3, we present an algorithm that uses a heuristic approach to determine a CMFDS for the given network. Section 4 shows the application of our algorithm to a stationary network, followed by section 5 showing examples of how network and ferry parameters may impact the message ferrying capabilities of the nodes in the network. Finally, we conclude in Section 6.

II. Connected Message Ferry Dominating Sets

A CMFDS is a set of nodes that can provide message ferrying connectivity between all sources and destinations in a wireless and mobile network. In this section, we develop the formal definition of a CMFDS. In the next section we develop a heuristic to find a CMFDS.

A. Network Model

We consider a wireless and mobile network comprising a set $\mathcal{N}$ of N nodes equipped with wireless interfaces moving within a given space. The network operates for a finite duration starting at time 0 and ending at time T which we call the *network lifetime*. Contacts occur between nodes when they are within radio range of each other. The evolution of the network over time is completely defined by the contact sequences between all pairs of nodes.

A pair of nodes, i and j, experience a sequence of symmetric contacts C_{ij}^k for $k = 1, 2, \ldots, \ell$ where C_{ij}^k represents the time interval of the kth contact: it starts after the $(k-1)$st contact ends and before the kth contact starts. In terms of the data transfer opportunities, the network is fully specified by the sequence of contacts between its nodes.Contact traces can be obtained from mobility models [15].

B. Message Ferry Definition

Informally, a message ferry can provide connectivity to a set of nodes if it meets all nodes in the set on a regular basis. More formally we make the following definitions:

Definition: Ferry Cycle **(FC)** of duration d:

Node $i \in \mathcal{N}$ makes a *ferry cycle* (FC) of duration$d$, on a set of nodes $S \subseteq \mathcal{N}$ if during the cycle duration, d, i makes a set of contacts $\{C_{ij}^1, C_{ij}^2, \ldots, C_{ij}^\ell\}$ with *each* node $j \in S$ and

1) $(\forall C_{ij}^s : 0 < s \leq \ell)$ $(C_{ij}^s \geq \mu)$; that is, for a contact to "count", it needs to be at least μ time units long, and
2) $\sum_{s=1}^{\ell} C_{ij}^s \geq \tau$; that is, the cumulative contact duration with each node is at least τ time units.

The minimum single contact duration in a cycle, μ, is derived from the minimum data unit size that can be transferred in the network. Contacts of less duration are essentially ignored as they are not usable. The minimum cumulative contact duration, τ, provides a lower bound on the throughput between the ferry and the nodes it meets in S during each cycle.

A node is a message ferry if it can make multiple ferry cycles during the network lifetime. This is formalized as follows:

Definition: R-cycle Message Ferry **(MF)**:

The tuple (k, S) represents an R-cycle message ferry, where S is a set of nodes $S \subseteq \mathcal{N}$ and $k \in S$ is the message ferry if k makes a sequence of ferry cycles $\{FC_1, FC_2, ..., FC_R\}$, $R > 0$ on the node set $S - k$, where FC_i starts at time s_i and ends at time e_i and:

1) $e_i > s_i$ for $1 \leq i \leq R$.
2) $s_1 = 0$, that is the first cycle starts at time 0 with the start of the network lifetime.
3) $s_i = e_{i-1}$ for $2 \leq i \leq R$, that is , the start time, s_i, of a cycle FC_i immediately follows the end time, e_{i-1}, of the previous cycle FC_{i-1}.
4) $0 \leq T - e_R < \delta$, that is the last cycle ends with at most δ time left in the network lifetime. Because there are no ferry cycles during this period, we call it *network unusable period,* and it is an indication of how much time at the end of the network lifetime one is willing to "waste".

S is called the *node set of ferry* k and includes k itself. The concepts in the MF definition are shown in Fig. 5.

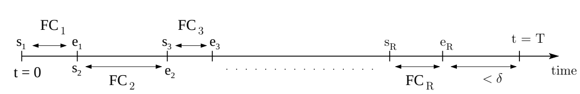


Fig. 5. An example of a message ferry having cycles $FC_1, FC_2, \ldots, FC_R$ during the network lifetime T. It also shows the *network unusable period* δ after the last ferry cycle FC_R.

C. Message Ferry Dominating Set

Informally a message ferry dominating set (MFDS) is a set of ferries as defined above, where the union of their node sets covers all nodes in set $\mathcal{N}$.

Definition: Message Ferry Dominating Set **(MFDS)**:

An MFDS, M, is a set of m MFs,(i, S_i), $i = 1, \dots, m$; such that $\bigcup_{i=1}^{m}(S_i)$ = $\mathcal{N}$.

Each ferry in an MFDS can move data among nodes in its own set. Complete end-to-end connectivity is not assured, however, unless the ferries themselves are connected over space time paths. We make the following definition for ferry communication:

Definition: Two ferries (i, S_i) and (j, S_j) are *directly linked* (D-linked) if i and j are both elements of $S_i \cap S_j$. (Note that because links are symmetric, either i and j are both in $S_i \cap S_j$ or neither is.)

If two ferries are in each other's node set, then those two ferries can communicate directly on a regular basis. We call this type of link *a D-link*. D-links form the basis for communicating message ferries to provide overall network connectivity.

We make the following definitions:

Definition: Message Ferry Graph **(MFG)**:

The message ferry graph of an MFDS M, MFG(M), is a graph G = (V, E) whose vertices are the message ferries in the MFDS, i.e., $V = M$ and edges, E, are all of the D-links that exist between the message ferries of the MFDS M. An *MFG-path* between two vertices (ferries) consists of one or more edges (D-links) taken to reach from one vertex to the other.

The MFG is a method used to represent the ferries and their links and provides a mechanism to determine whether a set of ferries representing an MFDS can provide connectivity among all network nodes, thus forming a CMFDS. This is formalized in the following definition.

Definition: Connected Message Ferry Dominating Set **(CMFDS)**:

A CMFDS is a ferry set P, such that P is an MFDS and MFG(P) is connected. Note that a connected MFG(P) means there exists at least one potentially multi-hop space-time path between any two ferries (vertices) in MFG(P). Also note that, not all MFDS M may yield a CMFDS P.

III. Finding a Connected Message Ferry Dominating Set

In this section we will present our algorithm that finds a connected message ferry dominating set for a particular network.

A. Problem Statement

The problem is to find a connected message ferry dominating set (CMFDS) for a network with lifetime T and with a set $\mathcal{N}$ of N nodes that is described with a given contact trace t_r. The CMFDS is constrained with the following ferry parameters (as described in the previous section).

- μ, the minimum contact duration
- τ, the minimum accumulated contact duration within a ferry cycle
- δ, the maximum allowed unusable network time

Further we are interested in achieving the following objectives:

- Achieving some target cycle length criteria. For example, we may desire a bound on the average cycle length (T_{avg}), or on the maximum cycle length (T_{max}). Note that, end-to-end delivery delay has positive correlation with the ferry cycle length [1].
- Minimizing the number of nodes in the CMFDS.

As mentioned earlier, the CMFDS problem is a generalization of the NP-hard connected dominating set problem. Our efforts are, therefore, aimed at developing heuristics for this problem.

Our algorithm incorporates the following steps:

1) Determine the *ferry capability* of each node present in the network. The *ferry capability* of a node will be high if it provides ferry service to a large number of nodes i.e. the larger the size of the ferry node set S, the better the *ferry capability* of the node.
2) Use a greedy algorithm, that picks nodes with high ferry capability, to find an MFDS M.
3) Construct a connected MFG(P) for CMFDS P. The vertices (ferries) of the connected MFG(P) together constitutes a CMFDS P. However, if a connected MFG(P) cannot be obtained then we declare that CMFDS for the network does not exist.

The detailed description of these steps is provided in the rest of this section. We begin by describing a basic subroutine in our heuristic that can test whether a single node can act as a ferry for a set of nodes according to a given set of ferry parameters.

B. Finding Individual Message Ferries

In this section, we present the algorithm `Find_Ferry`$(i, \mathcal{N})$ that determines the node set $S \subseteq \mathcal{N}$, for which node i behaves as a message ferry. The algorithm takes as input the trace file t_r with contacts, network node set $\mathcal{N}$, network lifetime T, node id $i \in \mathcal{N}$ and the ferry parameters: minimum contact time μ, cumulative contact time τ, network unusable period δ and average cycle length T_{avg}.

The *Find_Ferry(i, $\mathcal{N}$)* algorithm is shown in Algorithm 1. It starts with the maximum possible node set, $\mathcal{N}$, and calculates if node i can act as a ferry for the entire network. If not, the *least-interacting* node is removed from the node set and then the node i is tested for this reduced set of network nodes. The least interacting node is the one with the shortest total contact duration with

Algorithm 1: Find_Ferry(i, $\mathcal{N}$)

Input: network parameters - t_r, $\mathcal{N}$ and T; ferry parameters - μ, τ, δ and T_{avg}; node i where $i \in \mathcal{N}$
Output: finds node set S for which i is a ferry and returns tuple (i, S)

$S = \mathcal{N}$
is_ferry = false
repeat
    is_ferry = Ferry_Test(i , S)
    if is_ferry *is false* **then**
        /* remove the least-interacting node k from S */
        $S = S - \{k\}$
until is_ferry *is true* **or** $S = \{i\}$
return (i, S)

node i throughout the network lifetime. The process is repeated until either node i comes out to be a ferry for some node set $S \subseteq \mathcal{N}$ or the set S is left with only one node, the node i itself.

To determine whether node i is a ferry for node set S, the algorithm, described above, uses subroutine Ferry_Test(i, S), which returns true if (i, S) represents a ferry; otherwise it returns false. This algorithm is described in Algorithm 2. For node i to provide ferry service to any node j, it must meet node j for at least τ time units cumulatively in each cycle. We read the contact times (considering only contacts that last longer than or equal to μ) between ferry i and node j from the trace file and sum them until the total contact time between them becomes τ. The total time taken in order to achieve a cumulative ferry-node interaction of length τ time units defines the minimum cycle length, c_j, required for i to act as a ferry for node j. Similarly, we calculate the minimum cycle length for every other node $j \in S$, except node i itself, and then select the cycle length with the maximum value. This way we are assured that for this cycle length, $cycle\_legnth$, node i can behave as a ferry for every node $j \in S$. We repeat this process to find other subsequent ferry cycles until no more cycles can be found (e.g. the end of the network is reached). Finally, we check if the network unusable period is less than or equal to δ and the average cycle length is less than or equal to T_{avg}. If both constraints are satisfied then the subroutine returns true otherwise it returns false. Note that, we have used the average cycle length, T_{avg}, to achieve the target cycle length criteria, but one can also use other parameters like maximum cycle length, T_{max}, in order to impose stricter bounds on the ferry cycle length.

Algorithm 2: Ferry_Test (i, S)

Input: network parameters - t_r, $\mathcal{N}$ and T; ferry parameters - μ, τ, δ and T_{avg}; node i where $i \in \mathcal{N}$; node set S such that $S \subseteq \mathcal{N}$
Output: true if tuple (i, S) represent a message ferry else returns false

num_cycles = 0
$total\_ferry\_duration$ = 0
$search\_more\_cycles$ = true
repeat
    $cycle\_start$ = $total\_ferry\_duration$
    foreach *node* $j \in S - \{i\}$ **do**
        Read(t_r) to find contacts between j and i after time $cycle\_start$, such that each contact $>= \mu$
        Add these contacts until $total\_contact\_time$ = τ or network end time T is reached
        if $total\_contact\_time = \tau$ **then**
            /* set minimum cycle length c_j for node j when node i is ferry */
            Mark time when $total\_contact\_time$ = τ is reached as $cycle\_end$
            c_j = $cycle\_end$ - $cycle\_start$;
        else
            /* no more ferry cycles possible */
            $search\_more\_cycles$ = false
            break
    if $search\_more\_cycles$ *is true* **then**
        /* select max cycle length so that node i is ferry for all $k \in S$ */
        $cycle\_length = max\{c_1, \ldots, c_k\}$: $k \in S$, $k \neq i$
        $total\_ferry\_duration$ += $cycle\_length$
        $num\_cycles = num\_cycles + 1$
until $search\_more\_cycles$ *is false*
/* check constraints */
if $(T - total\_ferry\_duration) < \delta$ **and** $(total\_ferry\_duration/num\_cycles) \leq T_{avg}$ **then**
    return *true*
else
    return *false*

C. Finding an MFDS

Once the individual message ferries and their node sets have been determined, this stage of the algorithm picks a set of message ferries to construct an MFDS M, such that the union of their node sets covers all the nodes in the network. Mathematically, MFDS M represents a set of m message ferries (i, S_i) where $m > 0$ and $i =$

Algorithm 3: Find_MFDS(K)

Input: set K of all message ferries (i, S_i) where $i \in \mathcal{N}$ and node set S_i of ferry i such that $S_i \subseteq \mathcal{N}$

Output: an MFDS M of m message ferries (k, S_k) where $k = 1, \ldots, m$; such that $\bigcup_{k=1}^{m}(S_k) = \mathcal{N}$

```
C_v = {}
M = {}
/* repeat until all nodes in N are covered by ferry
   node sets in MFDS M */
repeat
    select ferry i from K for which (N − C_v) ∩ S_i
    is maximum
    insert (i, S_i) into M
    remove (i, S_i) from K
    C_v = C_v ∪ S_i
until C_v = N
return M
```

$1, 2, \ldots, m$ such that $\bigcup_{i=1}^{m}(S_i) = \mathcal{N}$. We have used a heuristic approach that attempts to greedily minimize the size of the MFDS.

The algorithm `Find_MFDS`(K) shown in Algorithm 3 describes our approach to find MFDS. The algorithm finds an approximation to minimum MFDS, which is essentially the well studied Set Cover Problem. The MFDS M is initially empty. The greedy algorithm iteratively adds a message ferry i into MFDS M such that the node set S_i of ferry i has the maximum number of *uncovered* nodes. A node j is said to be *uncovered* if it is not present in the node sets of any of the ferries in the MFDS i.e. $j \notin C_v$ (refer to Algorithm 3). The process terminates when M becomes a dominating set.

D. Finding a CMFDS

This stage of the algorithm produces a CMFDS P, given the MFDS M of the network. The algorithm `Find_CMFDS`(M), shown in Algorithm 4, first initializes the CMFDS P to the ferry set in MFDS M and then constructs the corresponding MFG(P). If the generated MFG(P) is not connected, then in that case, our algorithm adds more ferries into the current set P. While adding new ferries into P, we prefer those ferries whose node set contains comparatively greater number of existing ferries (that are already present in the CMFDS P). Mathematically, we select ferry (i, S_i) to be added to CMFDS P if MFG(P) is disconnected and $i \notin P$ and S_i contains comparatively greater number of nodes j such that $j \in P$. Note that, if two or more ferries have the same number of existing ferries (those already present in P) in their node sets then we pick the one with greater size node set. This process is repeated until the MFG(P)

Algorithm 4: Find_CMFDS(M)

constructs **Input**: MFDS M

Output: a CMFDS P of p message ferries (k, S_k) where $k = 1, \ldots, p$; such that $\bigcup_{k=1}^{p}(S_k) = \mathcal{N}$ and MFG(P) is connected

```
P = M
Connect all nodes in P by all possible D-Links to
form an MFG(P)
if MFG(P) is not connected then
    /* Add more ferries in P to make MFG(P)
       connected */
    repeat
        select ferry (i, S_i) such that i ∉ P, but
        connects to maximum number of ferries j
        where j ∈ P
        insert (i, S_i) into P
        Connect all nodes in P by all possible
        D-Links to form an MFG(P)
    until MFG(P) is connected or no more ferry
    can be added
if MFG(P) is connected then
    return P
else
    return null /* CMFDS does not exist */
```

becomes connected or when no more ferries are left in the network to be added to P.

If eventually we get a connected MFG(P), then the set P becomes a CMFDS. However, if the MFG(P) remains disconnected and no more ferries can be added to make it connected, then the subroutine returns null, stating that a CMFDS for such a network does not exist.

E. Illustrative Example

In this section, we present a simple example that illustrates the working of our algorithm. Consider the network topology shown in Fig. 6. Nodes $N4$, $N5$, $N7$, $N8$, $N9$ and $N10$ are mobile nodes, where dashed lines show their route and arrows indicate the direction of their movement, while all the rest are stationary nodes. Let us assume that each node-node interaction lasts for at least τ time units. First, we run the `Find_Ferry`($i, \mathcal{N}$) algorithm for each node present in the network. The node set S_i for each ferry (i, S_i), where $i \in \mathcal{N}$ and $S_i \subseteq \mathcal{N}$, determined by this subroutine are as follows:

- $S_1\{1, 4\}$; $S_2\{2, 4\})$; $S_3\{3, 4\}$; $S_4\{1, 2, 3, 4, 5\}$
- $S_5\{4, 5, 6\}$, $S_6\{5, 6, 8\}$; $S_7\{7, 8\}$; $S_8\{6, 7, 8, 9\}$
- $S_9\{8, 9, 10, 11\}$; $S_{10}\{9, 10\}$; $S_{11}\{9, 11\}$

After determining the ferry capabilities of each node, this set of tuples, K, is served as an input to the procedure `Find_MFDS`(K), which selects message ferries greedily (the one with larger node set is preferred)

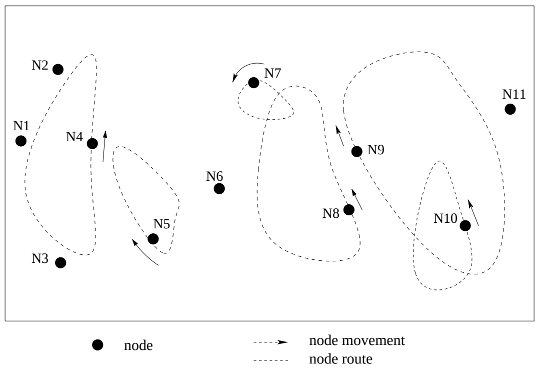


Fig. 6. An example showing a sparsely connected mobile ad hoc network

till a dominating set is found. For this network, it first picks node $N4$ ($S_4 = \{1, 2, 3, 4, 5\}$) , then node $N8$ ($S_8 = \{6, 7, 8, 9\}$) and finally adds node $N9$ ($S_9 = \{8, 9, 10, 11\}$) . The union of their node sets ($S_4 \cup S_9 \cup S_8 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$) contain each and every node of the network. Hence, these three ferries form a message ferry dominating set and the procedure `Find_MFDS`(K) ends here. Now, the subroutine `Find_CMFDS`(M) constructs an MFG(P) for the set $P = \{4, 8, 9\}$. This is shown in Fig. 7.

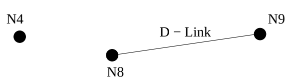


Fig. 7. MFG(P) for $P = \{4, 8, 9\}$

Since, the MFG(P) is disconnected, it adds node $N5$ (or node $N6$) into set P (size of S_5 and S_6 is greater than the node sets of other competing nodes). Suppose, we selected node $N5$, then $P = \{4, 5, 8, 9\}$. The MFG(P) would now appear as in Fig.8.

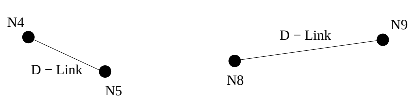


Fig. 8. MFG(P) for $P = \{4, 5, 8, 9\}$

The resultant MFG(P) is still disconnected, so the algorithm adds node $N6$ into set P (since node set S_6 contains two already existing ferries, namely, $N5$ and $N8$ whereas the other competing nodes have only one). Finally, the MFG(P) becomes connected (Fig. 9). The vertices(ferries) of the MFG(P) together constitute a CMFDS $P = \{4, 5, 6, 8, 9\}$.

Since, the connected MFG(P) does not have any loops, so the spanning tree of MFG(P) would be same as MFG(P). This could be used as a backbone to perform routing in the network shown in Fig. 6.

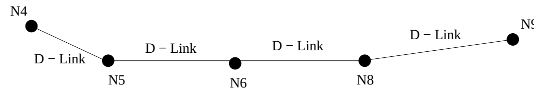


Fig. 9. MFG(P) for CMFDS $P = \{4, 5, 6, 8, 9\}$

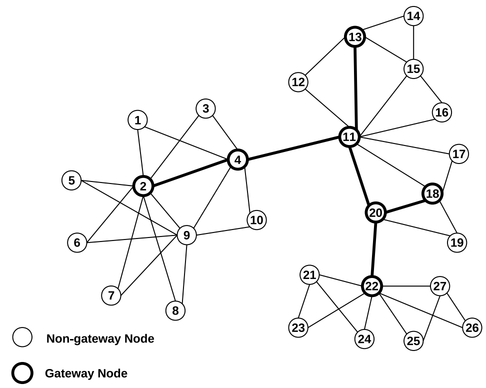


Fig. 10. Connected Dominating Set in a stationary network detected by the CMFDS algorithm

IV. CMFDS IN A STATIONARY NETWORK

From a message ferrying perspective, a stationary network is a special case of an intermittently connected network where the contacts last throughout the network duration due to low or zero node mobility. In order to illustrate that CMFDS problem is indeed a generalization of the CDS problem, we present here an example that shows the application of our algorithm to a stationary connected network. It shows that the CMFDS approach can be used to find a CDS in a stationary network.

In this example, we applied our algorithm to the network shown in Fig. 10. We used the ferry parameters shown in Table I. Note, we have used these values as default for all our simulations unless specified otherwise.

TABLE I
DEFAULT VALUES OF FERRY PARAMETERS

minimum contact duration (μ)	1 second
cumulative contact duration (τ)	3 seconds
average ferry cycle length (T_{avg})	5000 seconds
network unusable period (δ)	7500 seconds

The nodes marked as gateways are detected to be the ”message ferries” constituting the CMFDS i.e. $P = \{2, 4, 11, 13, 18, 20, 22\}$. Note that, the CMFDS, here, is actually a CDS, where the nodes in P form a connected backbone and all nodes not in P are just one-hop away from at least one node in P.

V. CMFDS IN MOBILE NETWORKS

In this section, we show examples of the application of our algorithm to two mobility models: the Random Way point (RWP) [15] and Shortest Path Map Based

(SPMB) [16] mobility model. We , first, lay down the basic settings that we have used for each mobility model, followed by a discussion of the results of the application of our algorithm in intermittently connected networks. Our results demonstrate the changes in the CMFDS (size as well as the constituting nodes) and how it correlates to its performance in terms of average message delivery delay as the network and ferry parameters change. Our goal is to show examples of how network parameters and message ferry parameters may impact the intrinsic message ferrying capabilities of the nodes in the network.

A. Basic simulation settings

We have used the Opportunistic Network Environment (ONE) simulator [17] to create the contact traces of our example networks. We chose two mobility models: Random Way Point (RWP) [15] and Shortest Path Map Based (SPMB) [16]. The SPMB model is a derivative of the RWP model, where nodes move on a map using the streets and roads defined on that map. A node randomly picks a speed and a destination on the map and moves there taking the shortest path over the defined roads or paths. The SPMB model also allows one to define *points of interest* on a map, such as restaurants, movie theaters. The nodes are then assigned certain points of interest with a certain probability of visiting them. In our simulations, we have used 18 points of interest with 0.6 visit probability.

We have used the default network settings shown in Table II in all our simulations, unless specified otherwise.

TABLE II
DEFAULT NETWORK SETTINGS FOR MOBILITY MODELS

	RWP Model	SPMB Model
Network Area	3km x 3km	4.5km x 3.4km
Network Duration	24 hours	24 hours
Wireless Range	250 m	250 m
Node Speed	Specified in the experiment	buses: $7-10$ m/s cars: $7-15$ m/s pedestrians: $1-2$ m/s
Node Pause Time	$0-10$ secs	buses: $5-10$ secs cars: $0-10$ secs pedestrians: $0-10$ secs

As a further step, we process the contact traces obtained from the mobility models described above to produce *non-overlapping* contacts, that is, a node can be in contact with only one other node at a time. While this is an optional step, we choose to use it since we use the wireless link for point-to-point communication only. Translating any contact trace into one that consists of only one-to-one contacts requires scheduling of multi point contacts. This can be done in several ways. We propose the approach explained in the Appendix.

B. Impact of Network Parameters

In this section, we show examples of how the network parameters, speed and density, may affect the existence and size of a CMFDS in a network. We used constant ferry parameters, specified in Table I, in our experiments.

1) Density: We first evaluate the impact of the number of nodes on the size of the CMFDS in a network.

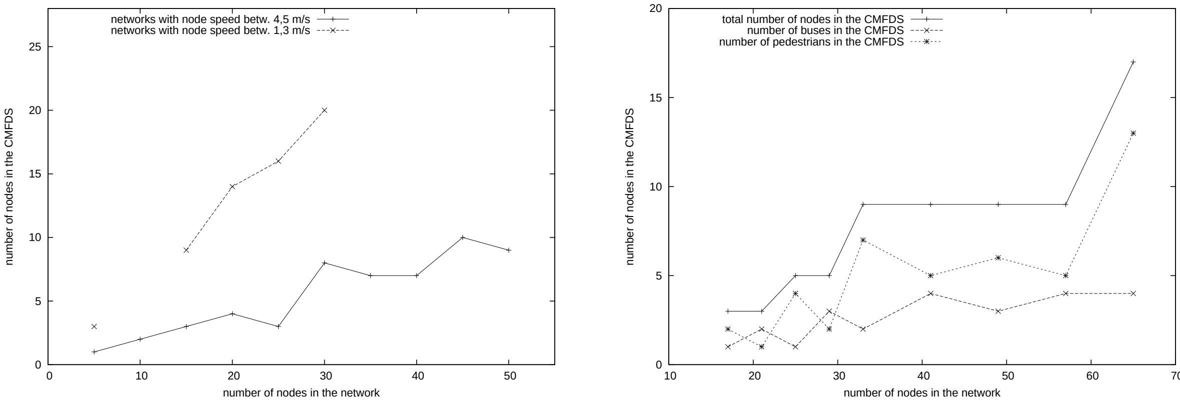
For the RWP mobility model, we conducted experiments with two sets of networks in Fig. 11(a). As we increase the number of nodes in the network, the size of the CMFDS tends to increase. As the density of the network increases, greater number of message ferries are required to form a CMFDS. It is important to note that the graph in Fig 11(a) does not demonstrate a strictly increasing trend. Sometimes a more crowded network has an equal or a slightly smaller size CMFDS than a less crowded one. This might happen when due to random node mobility message ferries in a crowded network end up having bigger node sets.

For SPMB mobility model, we identified nodes as buses and pedestrians. We kept a constant number of 5 buses in the network while increasing the number of pedestrians and observed the changes in the CMFDS. Fig. 11(b) demonstrates the number of nodes in the CMFDS of the network as the number of pedestrians in the network increase. A similar increasing trend in the CMFDS size is observed. The topmost curve in in Fig. 11(b) reflects the total number of nodes in the CMFDS while the bottom two curves show that how many of them are buses and pedestrians. Although the buses have higher mobility and more regular routes, pedestrians also become message ferries. Especially, after when most of the buses, 3 or 4 out of 5, are used as message ferries, a further increase in the number of pedestrians results in more pedestrians becoming message ferries. Note that, not all buses are included in the CMFDS because some of them may end up having many common nodes in their node sets.

2) Speed: To evaluate the impact of node mobility, we changed the speed of nodes in the network and observed its impact on the CMFDS size.

Figure 12(a) shows the results for three sets of networks using RWP mobility model, each set having different node densities. We have used the ferry parameters specified in Table I. We observe that with increase in node mobility, the size of the CMFDS decreases. This is an expected behavior since nodes with higher mobility are able to contact more nodes, for the same average cycle length, thus have bigger node sets. With bigger node sets, fewer ferries are needed to cover the entire network.

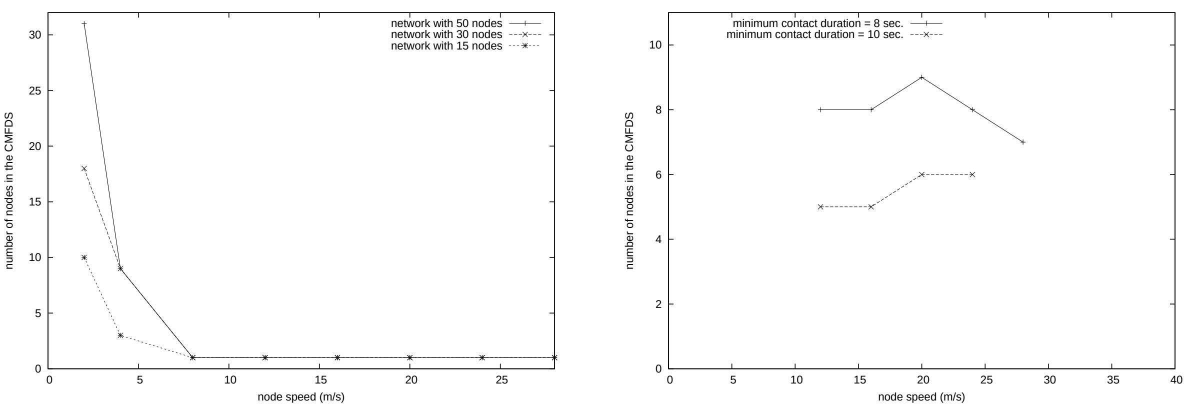
However, when the mobility becomes so high that most of the individual contacts lasts shorter than μ then the message ferries in the network start to disappear.



(a) The change in the size of the CMFDS as the number of nodes in a network increases (RWP model)

(b) The change in the size of the CMFDS as the number of nodes in a network increases (SPMB model)

Fig. 11. Impact of network density on CMFDS (used ferry parameters as in Table I)



(a) The change in the size of the CMFDS as the speed of nodes in a network increases (RWP model)

(b) The change in the size of the CMFDS as the speed of nodes in a network increases (RWP model)

Fig. 12. Impact of network mobility on CMFDS (used ferry parameters as in Table I)

Figure 12(b) shows this behavior where the network has 15 nodes and uses the ferry parameters in Table I except for μ. At lower nodes speeds no CMFDS is detected, but as the node speed increases message ferries start to emerge and a CMFDS is found. However, after a certain node speed is reached (approximately 25 m/s), the individual contacts become shorter than the minimum contact duration (μ). Since these contacts are not accounted for message ferry detection calculations, the ferries start to disappear. We conducted the experiment for different values of μ, 8 and 10 seconds, and similar behavior was observed.

C. *Impact of Ferry Parameters*

In this section, we show examples of how message ferry parameters, cumulative contact duration (τ) and average ferry cycle length (T_{avg}), affect the existence and the size of CMFDS in a network. We also show examples of how changes in these values impact the performance (in terms of message delivery delay) of the CMFDS when used for routing.

1) Cumulative Contact Duration (τ):: We applied the CMFDS algorithm to the same network trace multiple times, each time using a different value of τ. We repeated the experiment for both mobility models. We have used the ferry parameters given in Table I, except for the value of T_{avg}. Here, we chose a larger T_{avg} value to demonstrate a wider range of τ value. We performed the experiments on networks having different node speeds as shown in Figure 13(a). We observe that as τ increases, the size of CMFDS P also increases. This is an expected behavior because increasing τ, with constant T_{avg}, results in smaller node sets of ferries. This means more message ferries are needed to build a CMFDS. After a certain point, as τ increases the node sets of ferries become too small to form a CMFDS. As seen in Fig 13(a), networks with high mobility maintain a CMFDS for larger values of τ.

For the SPMB model, we performed the experiments on three different networks as shown in Fig. 13(b). The behavior is similar to what we observed for networks with RWP mobility model (Fig. 13(a)).

2) Average Cycle Length (T_{avg}): We applied the CMFDS detection algorithm to the same network trace multiple times, each time changing the value of T_{avg}, and observed the changes in the resulting CMFDS.

Figure 14 shows our results for the SPMB model. We performed the experiment on networks with different bus, car and pedestrian combinations as shown in Fig. 14. Instead of keeping the network unusable period (δ) at a constant value, we preferred to adapt it to the changing T_{avg} values in order to avoid short cycle length with very long network unusable periods. We used a constant ratio is 1.5 between T_{avg} and δ . As can be seen in Fig. 14, increasing the value of T_{avg} results in smaller size CMFDS for a network. This is because by increasing the T_{avg}, keeping τ constant, we are in fact relaxing the time constraint on the ferry cycles. Allowing the message ferries a longer time to finish their cycles enables them to visit more nodes, thus making their node sets bigger. With bigger node sets, fewer message ferries are needed to cover the network, thus the resulting CMFDS becomes smaller in size. Our results for RWP model suggests the same conclusions, the graphs are omitted due to space limitations.

3) Average Cycle Length (T_{avg}) and Average Message Delivery Delay: We applied the CMFDS algorithm to the same network trace multiple times, each time changing the T_{avg} value, and observed the changes in the resulting CMFDS P. Then, we built a spanning tree in MFG(P), that can serve as a backbone to perform routing. We routed messages over this *backbone* and observed the relation between T_{avg} and average delay for the messages routed over the CMFDS.

In our routing scheme, the source nodes relay their messages only to the ferries in the CMFDS. The ferries then relay messages to each other and/or deliver the messages directly to the destination nodes when in proximity. We have considered a low traffic scenario to minimize the queuing delays and avoid packet losses due to congestion. The messages have a Poisson arrival and infinite TTL. Each node has infinite buffer size.

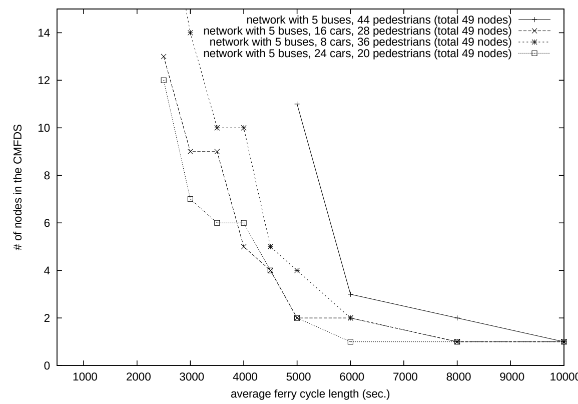


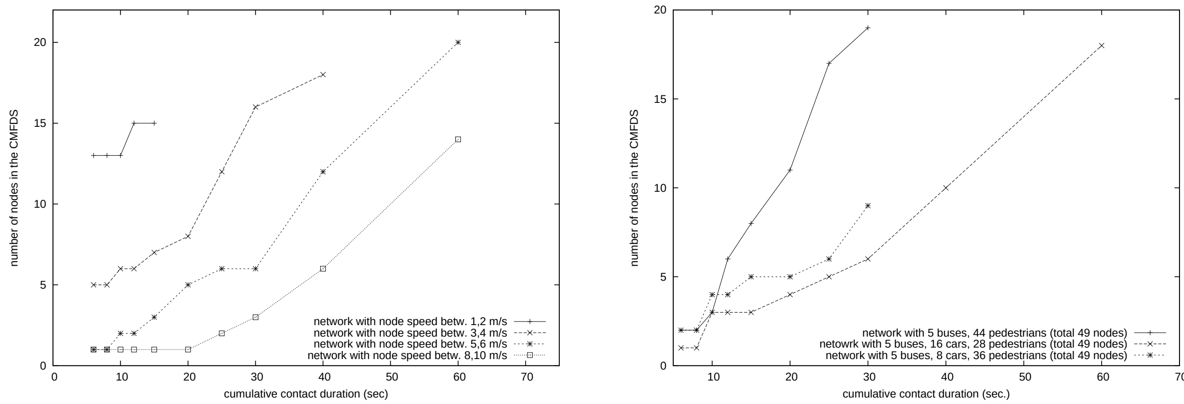
Fig. 14. The change in the size of the CMFDS as the ferry parameter, average cycle length (T_{avg}) increases (SPMB model). δ to T_{avg}-ratio = 1.5. Other ferry parameters are constant: $\mu = 1$ sec, $\tau = 3$ sec

There are 30 nodes in the network whose speeds are uniformly distributed between 5 and 6 m/s. We use $\mu = 1$ second, $\tau = 3$ seconds and maintain a constant ratio of 1.5 between δ and T_{avg}.

Figure 15 shows our results for the RWP model. The two curves in the graph show how the CMFDS size and the average delay changes as the average ferry cycle length changes. We observe that as T_{avg} increases, the average delay increases. The more we relax the ferry cycles by increasing their average length, the bigger the node sets of the message ferries get, thus resulting in a smaller CMFDS. When the T_{avg} is small, the ferries in the CMFDS have small node sets through which the ferries can cycle quickly. But as the average cycle length gets bigger, the CMFDS includes fewer ferries with large node sets through which the ferries take longer to cycle. This in turn leads to longer message delivery delays. We observed similar behavior for the SPMB model. The results are omitted due to space constraints.

We make the following observations regarding CMFDS detection in network with RWP mobility.

- First of all, it may be surprising to some that one is able to identify intrinsic message ferries in networks with RWP mobility in the first place. On closer examination, one can see for certain RWP parameter settings nodes tend to cover the area under consideration well which gives nodes certain message ferrying capability.
- Applying our algorithms to an RWP trace results in identifying specific nodes as making up the CMFDS. Because the nodes in the model are homogeneous, it is expected that any subset of the same size can also act as a CMFDS. In essence for the RWP model, what matters is the size of the CMFDS and not its exact constitution. This is, in general, not true for other mobility patterns.



(a) The change in the size of the CMFDS as the ferry parameter, cumulative contact duration (τ) increases (RWP model)

(b) The change in the size of the CMFDS as the ferry parameter, cumulative contact duration (τ) increases (SPMB model)

Fig. 13. Cumulative Contact Duration (Ferry parameters used - $\mu = 1$ sec, $T_{avg} = 7000$ sec. $\delta = 10500$ sec.)

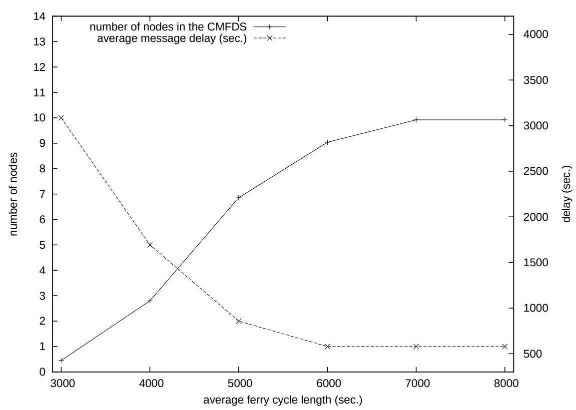


Fig. 15. The change in the size of the CMFDS and average message delay as the ferry parameter, average cycle length increases (RWP model). Ferry parameters are constant: $\mu = 1$ sec, $\tau = 3$ sec, δ to T_{avg}-ratio = 1.5

VI. Concluding Remarks

In this paper we consider the issue of characterizing the message ferrying capability of a wireless and mobile network. We define the concept of a connected message ferry dominating set (CMFDS) as a set of intrinsic message ferries that collectively are able to provide space-time connectivity among network nodes. An important feature of a CMFDS is that the nodes are capable of providing the message ferrying capability by virtue of their own mobility patterns. Once found, a CMFDS can form the basis of a routing strategy within an intermittently connected network [2]. We make the interesting observation that finding a CMFDS in an intermittently connected network is analogous to the well-known problem of finding a connected dominating set in a graph. Using this insight we provide a formalism for defining a message ferry and a CMFDS. We then develop a greedy heuristic that is capable of determining a CMFDS in a given network and with specified ferry parameters. We demonstrate the use of the heuristic in several illustrative examples that also allow us to investigate message ferrying properties of some mobility models.

This work lays the foundation for further efforts that aim to 1) exploit a given CMFDS for effective routing and 2) examine the question of how to determine a CMFDS in a distributed manner and in situations where the entire contact trace may not be known in advance. We plan to examine these are related issue sin our future work.

References

[1] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2004, pp. 187–198.

[2] W. Zhao, M. Ammar, and E. Zegure, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *in IEEE INFOCOM*, 2005, pp. 1407–1418.

[3] H. Jun, W. Zhao, M. H. Ammar, E. W. Zegura, and C. Lee, "Trading latency for energy in wireless ad hoc networks using message ferrying," in *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 220–225.

[4] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2004, pp. 145–158.

[5] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000.

[6] A. Lindgren, A. Doria, and O. S. En, “Probabilistic routing in intermittently connected networks,” in *SIGMOBILE Mobile Computing and Communication Review*, 2004, p. 2003.
[7] S. Merugu, M. Ammar, and E. Zegura, “Routing in space and time in networks with predictable mobility,” Tech. Rep., 2004.
[8] V. Borrel, M. H. Ammar, and E. W. Zegura, “Understanding the wireless and mobile network space: a routing-centered classification,” in *CHANTS ’07: Proceedings of the second ACM workshop on Challenged networks*. New York, NY, USA: ACM, 2007, pp. 11–18.
[9] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks,” in *DIALM ’99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. New York, NY, USA: ACM, 1999, pp. 7–14.
[10] B. Das and V. Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” 1997, pp. 376–380.
[11] R. Sivakumar, B. Das, and V. Bharghavan, “Spine routing in ad hoc networks,” *ACM/Baltzer Cluster Computing Journal (special issue on Mobile Computing*, vol. 1, pp. 237–248, 1998.
[12] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, “On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks,” *IEEE/KICS Journal of Communications and Networks*, vol. 4, pp. 59–70, 2002.
[13] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 14–25, 2002.
[14] K. Alzoubi, P.-J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless ad hoc networks,” in *HICSS ’02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)-Volume 9*. Washington, DC, USA: IEEE Computer Society, 2002, p. 297.
[15] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002. [Online]. Available: http://citeseer.ist.psu.edu/camp02survey.html
[16] F. Ekman, A. Keränen, J. Karvo, and J. Ott, “Working day movement model,” in *MobilityModels ’08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*. New York, NY, USA: ACM, 2008, pp. 33–40.
[17] A. Keränen and J. Ott, “Increasing reality for dtn protocol simulations,” Helsinki University of Technology, Tech. Rep., July 2007.

APPENDIX

In this section, we explain our contact scheduling scheme. A wireless node contacting multiple nodes concurrently produces overlapping contacts in a network. An example of this is demonstrated in Fig. 16. Figure 16(a) shows two contacts that node $n0$ makes with nodes $n1$ and $n2$. Node $n0$ contacts $n1$ between time t_0 - t_1 and node $n2$ between t_1 - t_3. Node $n0$ contacts both nodes $n1$ and $n2$ between time interval t_1 and t_2. This overlapping contact period can only be used for one point-to-point communication.

One way of scheduling contacts is on a first come first serve basis i.e. using the earliest starting contact until it ends, then switching to the contact that has the earliest start time among the remaining contacts and use what is left from it. The weakness of this scheme is that a very long contact prevents the use of any other contacts until it ends so we end up using a single unnecessarily long contact in our ferry calculations and ignore many other shorter contacts that happen at the same time. So, we first break the long contacts into smaller chunks and place those smaller contacts randomly between the start and the end of the long contact. This way we end up having multiple shorter contacts randomly placed within the same contact window.

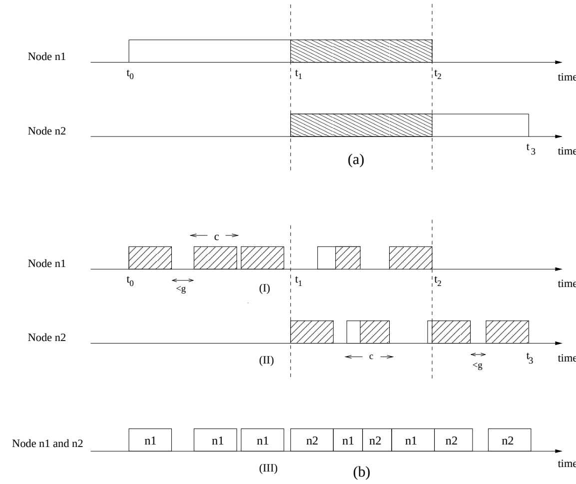


Fig. 16. Scheduling overlapping contacts

Figure 16(b)(I) and 16(b)(II) shows how the contacts of node $n0$ with $n1$ and $n2$ are broken into chunks. The contact with $n1$ is broken into 5 chunks of length c seconds. These smaller contacts are then placed between t_0 and t_2 with random gaps between them. The same is done for $n2$ with 4 chunks. While breaking long contacts causes us to lose some of the contact duration (the gaps between the chunks), it also allows us to switch to other contacts. The loss can be reduced by selecting big chunks with small gaps. Following is the summary of our contact scheduling scheme: Contacts longer than a specified threshold in the contact trace, are chopped into fixed size chunks and placed in the contact trace with gaps. The gap lengths are uniformly distributed between 0 and g (refer Fig. 16). All the contacts longer than the threshold, regardless of whether they are overlapping with any other contacts are chopped to chunks. After this step, we apply the rule of picking the earliest starting contact, described above. Figure 16(b)(III) shows the resulting scheduled contacts. In Fig. 16(b)(II), the white segments of the contact chunks are not used due to overlap with an earlier starting contact chunk. It is important to note that the parameters in the scheduling scheme, such as length threshold, chunk size, gap length may affect the CMFDS result in the network. Breaking long contacts into chunks that are too short for a minimum sized message to be transmitted (namely shorter than μ) prevents those long contacts from being included in the CMFDS calculation.