Inter-Vehicle Data Dissemination in Sparse Equipped Traffic

Sudarshan S. Chawathe

Department of Computer Science
University of Maine
Orono, ME 04469-5752, USA
chaw@cs.umaine.edu

***Abstract*— We address the following question: How can data be disseminated in an inter-vehicle communications (IVC) network when the traffic of IVC-equipped vehicles is very sparse (e.g., one IVC-equipped vehicle every few minutes), using minimal extra-vehicular infrastructure? We present a data-dissemination scheme that uses wireless dead drops (dead letter boxes) as intermediaries between vehicles. These dead drops are simple transceivers with storage and are not connected to other dead drops, the Internet, or other networks. An important question is the placement of such dead drops. We formulate the corresponding optimization problem and prove that it is NP-hard. We present an optimal greedy approximation algorithm for solving the optimization problem. Our algorithm is based on mapping the dead-drop placement problem to the problem of computing a minimum-weight spanning sub-hypergraph.**

I. Introduction

INTER-VEHICLE COMMUNICATION (IVC) is a term that includes several forms of communication, ranging from purely infrastructure-less communication between vehicles to communication supported by a substantial infrastructure of roadside gateways connected to the Internet [1]. An important design parameter for IVC protocols is the expected density of equipped vehicles. Henceforth, we use the term *traffic density* to mean density of *equipped* vehicles, i.e., vehicles with the proposed IVC capabilities. When traffic is dense, connectivity is relatively easy to establish, typically using a combination of broadcast and multi-hop protocols. There are several additional challenges, such as limiting the used bandwidth and routing messages in a highly dynamic network, which have received considerable attention (e.g., [2], [3]). As traffic becomes less dense, the primary challenge for IVC shifts to connectivity: When very few vehicles are within communication range of each other, the IVC network gets partitioned into clusters and routing messages between these clusters is more challenging. Several recent efforts have addressed connectivity in this situation as well (e.g., [4], [5]). Our focus in this paper is on the case when traffic is extremely sparse, with traffic rates lower than one vehicle per roadway direction (all lanes) every few minutes. The primary question we address is how data can be disseminated in such an environment with a minimal amount of infrastructure.

This work was supported in part by the U.S. National Science Foundation with grants IIS-9984296, IIS-0081860, and CNS-0426683.

Given the rudimentary infrastructure to which we limit ourselves, our goal is providing simple data dissemination. We do not aim to support general-purpose networking primitives, such as routing and discovery, that may be needed by other IVC applications. In particular, our focus is on so-called comfort applications, and not safety applications [1]. Nevertheless, several such comfort applications are both valuable and feasible. They include traveler information services, non-critical roadway information, and latency-tolerant messaging.

When traffic density is very low, vehicles form very small clusters (perhaps single vehicles). Two or more such clusters will not be in communication range of each other, except very rarely. As a result, inter-cluster communication without any infrastructure support (e.g., [4]) is not possible in this situation. Similar observations have been made earlier and methods such as using roadside gateways that are connected to the Internet have been proposed (e.g., FleetNet [6]). However, such gateways represent a substantial investment in infrastructure that may be difficult to justify for roadways with very sparse traffic, such as rural roads.

In light of our goal of disseminating data in this environment with a minimal infrastructure investment, we propose the use of *dead drops*. A dead drop or a dead letter box is, in general, a location where letters are deposited by one party and picked up by another (often for clandestine communication [7]). As we use the term in the context of inter-vehicle data dissemination, a dead drop is a wireless transceiver, placed at an intersection, that stores data transmitted by passing vehicles and delivers them to vehicles that pass later. Dead drops provide a very rudimentary form of infrastructure. Note that the wireless transceivers we envisage as dead drops are completely stand-alone devices that are not connected to the Internet or other networks. In particular, two dead drops cannot communicate with one another except via vehicles that travel between them, receiving data from one and transmitting them to the other. An important advantage of these dead drops is that they are inexpensive and easy to deploy. For example, we can use an open-source firmware on a commodity off-the-shelf 802.11b/g device such as the Linksys WRT54g coupled with a large battery, for a cost below \$100 [8]. Deployment of these dead drops requires nothing more than placing the devices at the desired locations and periodically charging or replacing the batteries. This feature is especially attractive in environments such as rural roadways, national parks, and hazardous or inaccessible areas, as well as for temporary setups.

We do not envision dead drops being used in exclusion of other IVC methods. Indeed, the presence of alternate IVC methods in nearby areas (within the typical travel range of the traffic being studied) amplifies the benefits of our scheme. For example, information about an accident on a rural road propagates, albeit slower than in alternate IVC methods. When a vehicle with such information arrives at a location with a higher traffic density, the information can be forwarded more rapidly using methods suited to that environment. The key point is that in the absence of the dead drops, the information about the accident would be unlikely to reach a large portion of the overall roadway network, because the few vehicles that pass the accident site may all terminate their journeys before reaching a high-density area. However, when dead drops are used as intermediaries to transfer data between vehicles on rural roads, the information can propagate to a high-density region.

Outline: In the next section, we begin by describing the scheme for data dissemination using dead drops and informally developing a problem definition in Section II-A. The formal definition of our main problem (MCDD) is presented in Section II-B after defining the key terms. In Section III, we develop the connection between the MCDD problem and a problem on hypergraphs: We cover preliminaries in Section III-A, define the min-weight spanning subhypergraph (MSSH) problem in Section III-B, and prove the NP-hardness of MSSH, and thus MCDD, in Section III-C. Section IV presents an optimal greedy approximation algorithm for MCDD. We describe related work in Section V and conclude with Section VI.

II. Data Dissemination using Dead Drops

A. Model of Data Dissemination

The basic mechanism for data dissemination using dead drops is very simple: Whenever a vehicle arrives at an intersection that hosts a dead drop (i.e., the vehicle is in range of the wireless transceiver at the intersection), a bidirectional data-exchange occurs. As the vehicle visits additional dead drops, it accumulates additional data. Similarly, as a dead drop is visited by additional vehicles, it accumulates data. The data transmitted by a vehicle include both data directly sensed or created on board the vehicle and also data received from dead drops encountered earlier. The data transmitted by a dead drop consist mainly of data received from earlier vehicles, but may include location-specific information generated at the site. In general, such data accumulated by both vehicles and dead drops may be filtered, compressed, aggregated, or otherwise processed. The specifics of how the data are processed are not important for the methods presented in this paper. Our methods are designed to enable data to propagate independently of their interpretation. Continual data exchange in this manner in the network of roads can potentially allow data from each site and each vehicle to be disseminated to all other vehicles and sites.

The main question addressed by the rest of this paper is the placement of dead drops. Even though we envision inexpensive devices, placing one at every intersection is not practicable or efficient. We need a method to determine the set of dead-drop locations that provides the *required connectivity* at *minimum cost*. In order to make the optimization problem suggested above more precise, we must explicate the italicized terms.

We quantify the cost of placing dead drops using a cost function that maps each intersection to the cost of a installing dead drop there. We permit nonuniform costs to model, for instance, that placing a dead drop at an intersection with a service station is likely to be much easier than placing one at an intersection at a remote and inaccessible location.

The phrase "required connectivity" may suggest several interpretations. For instance, it may suggest the requirement that data from vehicle A, traveling on an arbitrary route must reach vehicle B, traveling on some other arbitrary route within some time period. Such an interpretation is too rigid and cannot be satisfied in the general case without placing a dead drop at every intersection, thereby solving the problem in a trivial and impractical manner. Therefore, we adopt a more flexible interpretation in which the travel-routes of interest are specified and the placement of dead drops is required to provide connectivity between every pair of routes, in the sense made precise below, where the travel-routes are modeled as *trails*.

We note that the specification of trails (routes of interest) affects only the optimization problem. In a deployed system, vehicles traveling on arbitrary routes that differ from the specified trails interact with dead drops and, through them, with other vehicles in a manner identical to the interactions of vehicles traveling on the specified routes. The only difference is that the latter vehicles enjoy the connectivity guarantees formalized below, while the former do not. The extent of connectivity enjoyed by vehicles that travel on ad hoc routes depends on the details of how a specific problem instance is formulated. In particular, we note that our model does not preclude specifying a very large number of trails, although there is an obvious tradeoff between the required connectivity guarantees and the minimum feasible cost.

B. Terminology and Problem Statement

We model a network of roadways in a conventional manner, using a *road-graph* whose vertices represent intersections and whose edges represent road segments. A *trail* of length l, $l \geq 1$ is a sequence of l intersections: $t = (i_1, i_2, \ldots, i_l)$; we say t *visits* each of the intersections $i_1, i_2, \ldots i_l$. Figure 1 suggests a few trails in a small road graph. This example, and its derivatives summarized by Figures 2 and 3 are artificially small for the purpose of concise exposition. A realistic instance is likely to include dozens or hundreds of trails. Our methods are designed to work for such large instances. In fact, a prime reason for preferring a polynomial-time approximation algorithm over an exponential-time exact one is that the latter will not scale to realistic instances.

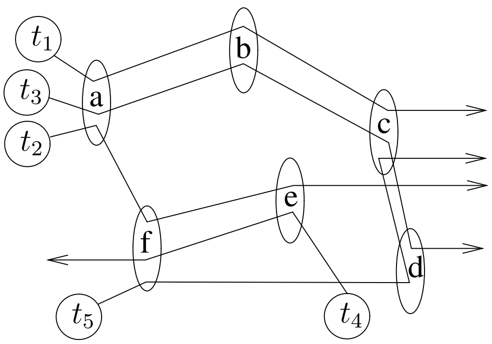

Fig. 1. Trails and intersections: Traffic trails $t_1, \ldots, t_5$ as they pass through six intersections a through f. The ovals denote communication ranges of potential dead drops at intersections. The underlying network of roads has been omitted for clarity.

We say trail t *touches* trail t' at intersection i, and write $\tau_i(t, t')$, if both t and t' visit the intersection i. Given a set $X \subseteq I$ of intersections, we say t *touches* t' *in* X, and write $\tau_X(t, t')$, if $\tau_i(t, t')$ for some intersection $i \in X$. Subscripts on the relation τ may be dropped when they are clear from context. The touches relation is symmetric in t and t': $\tau_i(t, t')$ iff $\tau_i(t', t)$; therefore, we may refer to t and t' touching at an intersection i without regard to order. We note that a pair of trails may touch at several intersections, in general. We use the relation *connects* to refer to the transitive closure of the touches relation with respect to t and t'. Intuitively, two trails are connected if there is a sequence of one or more touching trails leading from one to the other. More precisely, we say trail t connects to trail t' by a set $X \subseteq I$ of intersections, and write $\tau^+_X(t, t')$, if (1) $\tau_X(t, t')$ or (2) there is a sequence one or more trails $t_1, t_2, \ldots, t_k$ such that $\tau_X(t, t_1)$, $\tau_X(t_k, t')$, and $\tau_X(t_i, t_{i+1})$ for all $i \in 1, \ldots, k-1$. The symmetry of τ_X implies the symmetry of τ^+_X. We model the cost of maintaining a dead drop at an intersection using a cost function $c : I \to \mathbb{R}^+$. The cost of a set S of dead drops is obtained by extending c over S in the natural way: $c(S) = \sum_{i \in S} c(i)$. We may now define our problem formally:

> *Min-Cost Connecting Dead Drops (MCDD)*: Given a set T of trails visiting intersections in I, and a cost function $c : I \to \mathbb{R}^+$, find a minimum-cost set of dead drops $S \subseteq I$ such that every pair of trails $t, t' \in T$ is connected by S: $\tau^+_S(t, t')$.

III. Hypergraph Formulation

A. Hypergraphs

For our purposes, a *hypergraph* $\mathcal{H} = (V, H)$ consists of a finite set V of *vertices* and a set $H \subseteq 2^V$ of *hyperedges*, with $\emptyset \notin H$ [9]. Intuitively, we may think of a hypergraph as a generalization of a graph that permits edges that are incident on a variable number (one or more) of vertices, instead of being limited to two vertices per edge. For example, consider the hypergraph $\mathcal{H}_0 = (V_0, H_0)$, where $V_0 = \{t_1, t_2, t_3, t_4, t_5\}$ and $H_0 = \{\{t_1, t_2, t_3\}, \{t_1, t_3\}, \{t_1, t_3, t_5\}, \{t_2, t_4\}, \{t_2, t_4, t_5\}, \{t_3, t_5\}\}$. This hypergraph is depicted in Figure 3, which represents each hyperedge using a closed curve that encloses its incident vertices.

A hypergraph $\mathcal{H}' = (V', H')$ is said to be a *sub-hypergraph* of $\mathcal{H} = (V, E)$ if (1) $V' \subseteq V$ and (2) for each hyperedge $h' \in H'$ there is a hyperedge $h \in H$ such that $h' = h \cap V'$. Intuitively, every vertex of $\mathcal{H}'$ is also a vertex of $\mathcal{H}$ and each hyperedge of $\mathcal{H}'$ is the restriction to V' of some hyperedge of $\mathcal{H}$. For example, if $V_1 = \{t_1, t_2, t_4, t_5\}$ then the hypergraph $\mathcal{H}_1 = (V_1, \{\{t_1, t_2\}, \{t_1, t_5\}\})$ is a sub-hypergraph of the hypergraph $\mathcal{H}_0$ depicted in Figure 3 because $\{t_1, t_2\} = \{t_1, t_2, t_3\} \cap V_1$ where $\{t_1, t_2, t_3\} \in H_0$ and, similarly, $\{t_1, t_5\} = \{t_1, t_3, t_5\} \cap V_1$ where $\{t_1, t_3, t_5\} \in H_0$. In contrast, $\mathcal{H}'_1 = (V_1, \{\{t_1, t_2\}, \{t_1, t_2, t_5\}\})$ is not a sub-hypergraph of $\mathcal{H}_0$ because there is no hyperedge in H_0 that is a superset of $\{t_1, t_2, t_5\}$. Similarly, $\mathcal{H}'_1 = (V_1, \{\{t_1, t_2\}, \{t_2, t_5\}\})$ is also not a sub-hypergraph of $\mathcal{H}_0$ because, even though there is an hyperedge $h = \{t_2, t_4, t_5\}$ in H_0 that is a superset of $g = \{t_2, t_5\}$, restricting h to V_1 does not yield g: $h \cap V_1 \neq g$.

A *chain* is an alternating sequence of vertices and hyperedges such that each hyperedge in the sequence is incident on both the vertex preceding it and the one following it. Chains are the hypergraph analog of paths in graphs. More precisely, given a hypergraph $\mathcal{H} = (V, H)$ and an integer $k > 0$, we define a *chain of length* k from a vertex $v_0 \in V$ to a vertex $v_k \in V$ to be a sequence of the form $v_0, h_1, v_1, h_2, v_2, \ldots, h_k, v_k$, where (1) $v_i \in V$, (2) $h_i \in H$, (3) no two vertices in the sequence, with the possible exception of (v_0, v_k), are identical, (4) no two hyperedges in the sequence are identical, and (5) $\{v_{i-1}, v_i\} \subseteq h_i$ for all $i \in 1, 2, \ldots, k$. A chain with $v_0 = v_k$ is called a *cycle* of length k. In the hypergraph $\mathcal{H}_0$ of Figure 3, the sequence t_2, a, t_1, c, t_5 is a chain of length 2 from t_2 to t_5, while $t_2, a, t_1, c, t_5, f, t_2$ is a cycle of length 3. As the figure suggests, the hyperedges a, c, and f refer to $\{t_1, t_2, t_3\}$, $\{t_1, t_3, t_5\}$, and $\{t_2, t_4, t_5\}$, respectively.

We may define a connected graph as one that contains a path between every pair of vertices. Analogously, we say a hypergraph is *connected* if, for every pair of vertices $v, v' \in V$, there is a chain from v to v'. We may verify that the hypergraph $\mathcal{H}_1$ described above is connected. However, using $V = \{t_1, t_2, t_3, t_4, t_5\}$, the hypergraph $\mathcal{H}_2 = (V, \{\{t_1, t_3\}, \{t_2, t_4\}, \{t_2, t_4, t_5\}\})$ is not connected because there is no chain from t_1 to t_5, for instance.

B. Trail Hypergraphs and Duals

Given a set of trails we may construct a hypergraph with one hyperedge for each trail and one vertex for each intersection. The hyperedge corresponding to a trail is incident on the vertices corresponding to the intersections visited by the trail. We refer to such a hypergraph as the *trail hypergraph* for the set of trails. Figure 2 depicts the trail hypergraph for the trails suggested by Figure 1.

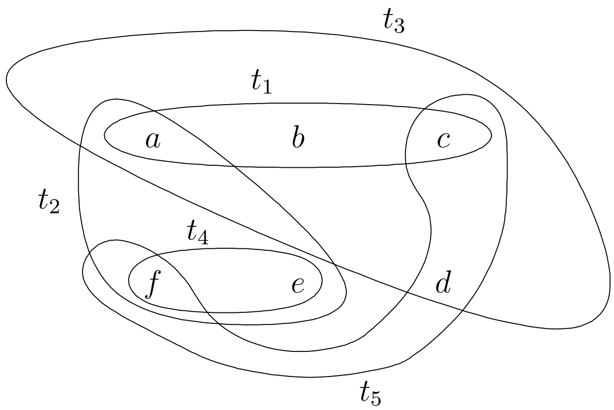

Fig. 2. Trail hypergraph for the example of Figure 1. Each vertex a through f represents an intersection and each hyperedge (closed curve) t_1 through t_5 represents a trail that visits its incident vertices. The intersections and trails are labeled as in Figure 1.

An instance of a min-cost connecting dead drops (MCDD) problem on a set of trails may now be mapped to connectedness of such a hypergraph. Although it is possible to state the equivalent problem directly on the trail hypergraph, it is more convenient to use its dual, defined below.

Intuitively, the dual of a hypergraph is obtained by exchanging the roles of vertices and hyperedges while preserving the incidence relationship. Formally, we define the dual of a hypergraph $\mathcal{H} = (V, H)$ as follows: Order the elements of V and H arbitrarily and consider the $|V| \times |H|$ binary matrix $M_{\mathcal{H}} = (m_{ij})$, called the *incidence matrix*, with $m_{ij} = 1$ if and only if the i'th vertex in V is incident on the j'th hyperedge in H. The dual of $\mathcal{H}$ is the hypergraph defined by the transpose of $M_{\mathcal{H}}$. Figure 3 depicts the dual of the trail hypergraph of Figure 2.

In the dual of a trail hypergraph, hyperedges that are incident on only one vertex correspond to intersections that are visited by only one trail. A dead drop at such an intersection does not serve to connect the trail to any other. Therefore, all such hyperedges may be safely ignored and we shall henceforth assume, without loss of generality, that every hyperedge in the dual graph is incident on at least two vertices.

We may now restate the the min-cost connecting dead drops (MCDD) problem (Section II-B) as the following equivalent problem on the dual of the trail hypergraph:

> *Min-Weight Spanning Sub-Hypergraph (MSSH)*: Given a hypergraph $\mathcal{H} = (V, H)$ and a weight function $w : H \to \mathbb{R}^+$, find a minimum-weight subset $H' \subseteq H$ such that the sub-hypergraph $\mathcal{H}' = (V, H')$ is connected, where the weight of H' is $\sum_{h \in H'} w(h)$.

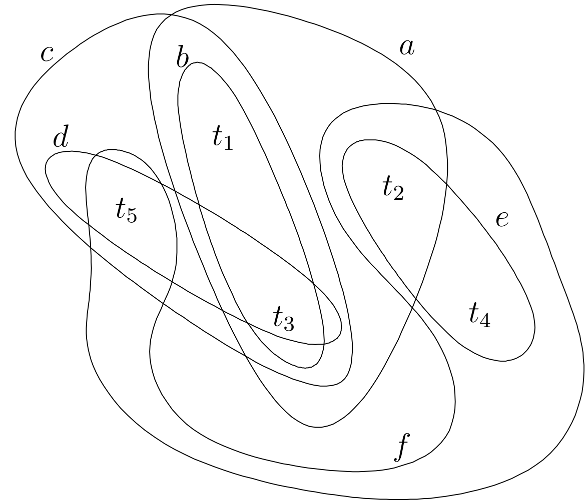

Fig. 3. The dual of the trail hypergraph of Figure 2. The vertices $t_1, \ldots, t_5$ represent trails and the hyperedges a through f represent intersections incident on the trails that visit them. The intersection and trails are labeled as in Figures 1 and 2.

The requirement that the sub-hypergraph be connected is important for our purposes. A disconnected sub-hypergraph corresponds to a distribution of dead drops that leaves at least one pair of trails with no mechanism to exchange data.

In addition to permitting the above correspondence between the MCDD and MSSH problems, the dual of a trail hypergraph has a property that will prove useful in Section IV: The size of hyperedges (number of vertices incident on a hyperedge) is likely to be low. The size of a hyperedge in the dual graph is the number of trails that visit the intersection it represents. We may expect this number to be smaller than the size of a hyperedge in the primal trail graph, which is the number of intersections visited by the corresponding trail.

C. Hardness

To demonstrate the hardness of the min-weight spanning sub-hypergraph (MSSH) problem, we use the following standard problem:

> *Minimum-weight k-set cover (k-MSC) problem*: Given a collection $\mathcal{C}$ of subsets of the set $[n] = \{1, 2, \ldots, n\}$, such that each subset in $\mathcal{C}$ has at most k elements, and a weight function $w : \mathcal{C} \to \mathbb{R}^+$, find a minimum-weight sub-collection $\mathcal{S} \subseteq \mathcal{C}$ that *covers* $[n]$: $\cup_{S \in \mathcal{S}} S = [n]$.

Consider an instance of the min-weight k-set cover problem. Construct a hypergraph $\mathcal{H}$ whose hyperedges correspond to the subsets augmented with a special common node. More precisely, $\mathcal{H} = ([n+1], H)$ where $H = \{S \cup \{n+1\} \mid S \in \mathcal{C}\}$. Define the weight of a hyperedge $h \in H$ as $w'(h) = w(h \setminus \{n+1\}) + 1$. There is a one-to-one correspondence between the solutions of the MSSH instance consisting of this hypergraph $\mathcal{H}$ and the weight function w' and the solutions of the given k-MSC instance. Since k-MSC is known to be NP-hard [10], it follows that MSSH and, by Section III-B, MCDD are also NP-hard.

IV. Approximation Algorithm

A. Greedy Connected Dead Drops

The hardness result of the previous section suggests that efficient (polynomial time) algorithms that yield optimal solutions are unlikely, and we must investigate alternate strategies. We describe below a greedy approximation algorithm for MCDD. Not only does this algorithm provide a bound on the non-optimality of the solution, this bound is also the best we can expect.

Listing 1 summarizes the *Greedy Connected Dead Drops* algorithm, encapsulated by the function GCDD. The input to GCDD consists of the set I of intersections, the set T of trails, and the function c that maps an intersection to the cost of placing a dead drop there. (See the MCDD problem definition in Section II-B.) The output of GCDD is a set S of intersections such that installing dead drops at all intersections in S guarantees that every pair of trails is connected. In GCDD, we begin with an empty S and add one intersection to S in each iteration of the while loop. The intersection i^* that is added at each iteration is one that maximizes the marginal benefit per unit cost, as computed by the expression on line 4. (The function BEN is described below.) Since multiple intersections may yield the maximum value for the expression, we arbitrarily pick the one with the lowest identifier using min. Correctness does not depend on which of the multiple intersections maximizing the expression is chosen.

Listing 1 Greedy Connected Dead Drops

```
1:  function GCDD(I, T, C)
2:      S ← ∅
3:      while (BEN(S) < |T| − 1) ∧ (|S| < |I|) do
4:          i* ← { min argmax_{i∈I\S} (BEN(S ∪ {i}) − BEN(S)) / c(i) }
5:          S ← S ∪ {i*}
6:      end while
7:      return S
8:  end function

9:  function BEN(S)
10:     E_S ← ∅
11:     for all intersections X ∈ S do
12:         for all trails t_1, t_2 ∈ X such that t_1 < t_2 do
13:             E_S ← E_S ∪ {(t_1, t_2)}
14:         end for
15:     end for
16:     V_S ← {t | {(t, t'), (t', t)} ∩ E_S ≠ ∅}
17:     G ← (V_S, E_S)
18:     p ← NUM_CONN_COMP(G)
19:     return |V_S| − p
20: end function
```

The benefit of a set S of dead drops is computed by the function BEN as outlined on lines 9–20 in Listing 1. Intuitively, this benefit is the number of well-connected trails: the number of trails less the number of connected components of the trail-connection graph $G(S)$, described below. Thus, sets that include intersections incident on connected trails are preferred over those with intersections incident on isolated disconnected trails. In more detail, the nested for-loops on lines 11–15 compute the set of edges E_S for a graph (not hypergraph) $G(S)$ induced by the input set S of intersections. For each intersection (hyperedge) X in S, $\binom{|X|}{2}$ edges are added to E_S, where $|X|$ is the number of trails incident on X. These edges define a complete graph that has as vertices the trails incident on X. The vertices V_s of G are simply the vertices appearing in E_S (line 16). Line 18 invokes a function NUM_CONN_COMP(G) that returns the number of connected components in G, using standard methods [11].

B. Example

Figure 4 outlines the execution of GCDD on the trails of our running example, suggested by Figure 1. Each block of the table represents one iteration of the while loop in function GCDD (lines 3–6 of Listing 1). Within each block, each row summarizes the evaluation of the fraction on line 4 of Listing 1 for the intersection i noted in the first column. The second column lists the cost of placing a dead drop at that intersection. The third column is key: It lists the edges E_S in the graph G used by function BEN in Listing 1. The fourth column lists the number of nodes in G while the fifth lists the number of connected components (p in function BEN) in G; their difference, listed in the sixth column, is the benefit of $S \cup \{i\}$. The last two columns list the incremental benefit of adding i to S and the incremental benefit per unit cost. The intersection with the highest value in the last column (the first such one in case of ties) is added to S at the end of each block to proceed to the next.

All graphs G in the first block ($S = \emptyset$) are connected ($p = 1$). In fact, they are all complete graphs. The situation is more interesting in the second block ($S = \{b\}$). In the row for intersection d the graph G for $S \cup \{d\}$ is connected but not complete, as edge $(1, 5)$ is missing. The graphs for intersection e is not only not complete, it is also not connected, having two connected components composed of a single edge each. Similarly, the graph for intersection f also has two connected components: one composed of a single edge and the other a fully connected graph on nodes 2, 4, and 5.

C. Analysis

In each iteration, the choice of the intersection i^* that is added to the set S based on the expression on line 4 of Listing 1 is intuitively appealing because it is one that provides the highest marginal benefit per unit cost. However, that expression has a more interesting property: The set of hyperedges I and the function BEN form a polymatroid

i	$c(i)$	$E_{S\cup\{i\}}$	$\lvert V_{S\cup\{i\}}\rvert$	p	$\text{BEN}(S\cup\{i\})$	ΔBEN	$\Delta\text{BEN}(i)/c(i)$
$S=\emptyset$; $\text{BEN}(S)=0$							
a	4	$(1,2),(1,3),(2,3)$	3	1	2	2	0.5
b	1	$(1,3)$	2	1	1	1	1.0
c	4	$(1,3),(1,5),(3,5)$	3	1	2	2	0.5
d	4	$(3,5)$	2	1	1	1	0.25
e	1	$(2,4)$	2	1	1	1	1.0
f	4	$(2,4),(2,5),(4,5)$	3	1	2	2	0.5
$S=\{b\}$; $\text{BEN}(S)=1$							
a	4	$(1,2),(1,3),(2,3)$	3	1	2	1	0.25
c	4	$(1,3),(1,5),(3,5)$	3	1	2	1	0.25
d	4	$(1,3),(3,5)$	3	1	2	1	0.25
e	1	$(1,3),(2,4)$	4	2	2	1	1.0
f	4	$(1,3),(2,4),(2,5),(4,5)$	5	2	3	2	0.5
$S=\{b,e\}$; $\text{BEN}(S)=2$							
a	4	$(1,2),(1,3),(2,3),(2,4)$	4	1	3	2	0.5
c	4	$(1,3),(1,5),(3,5),(2,4)$	5	2	3	2	0.5
d	4	$(1,3),(1,5),(3,5),(2,4)$	5	2	3	2	0.5
f	4	$(1,3),(2,4),(2,5),(4,5)$	5	2	3	2	0.5
$S=\{b,e,a\}$; $\text{BEN}(S)=3$							
c	4	$(1,3),(1,2),(1,5),(2,3),(2,4),(3,5)$	5	1	4	1	0.25
d	4	$(1,3),(1,2),(1,5),(3,5),(2,4)$	5	1	4	1	0.25
f	4	$(1,2),(1,3),(2,4),(2,5),(4,5)$	5	1	4	1	0.25
$S=\{b,e,a,c\}$; $\text{BEN}(S)=4$							

Fig. 4. A summary of the greedy algorithm GCDD of Listing 1 operating on the example of Figure 1.

(I,b), thereby providing the following approximation guarantee [12]:

> If D is the output of GCDD (Listing 1) and D_o is the optimal solution to the given instance of MCDD (Section II-B) then
>
> $$c(D) \leq c(D_o)\cdot H(t-1)$$
>
> where t is the maximum number of trails crossing at an intersection and $H(k)=\sum_{i=1}^{k} 1/i$ is the k'th harmonic number. That is, Listing 1 is a $H(t-1)$-approximation algorithm for MCDD.

Recall, from the end of Section III-B, that t is likely to be small, yielding a good approximation factor. For example, $t=4$ (at most 4 trails visiting an intersection) yields a 1.8-approximation, while $t=10$ yields a 2.83-approximation. Further, recall the reduction from k-set cover to MSSH (and thus MCDD) outlined in Section III-C. It is known that set cover cannot be approximated to better than $\ln n$ in polynomial time [13]. Since $H(n)$ differs from $\ln n$ by only an additive constant, it follows that no polynomial time algorithm can improve on the approximation to MCDD provided by GCDD. Our greedy algorithm for MCDD is thus optimal in this sense.

Let m and n denote, respectively, the number of intersections and the number of trails in the input. If at most t trails cross at any intersection, then line 13 is executed at most $k(k-1)/2$ times for each intersection in S, for a time complexity of $O(|S|\cdot k^2\cdot \log k)$ for an invocation $\text{BEN}(S)$. Since the graph G contains at most n vertices and n^2 edges, the number of connected components of G on line 18 is computed in time $O(n^2)$. Since all other operations in **ben** require only constant time, the time complexity of each invocation of BEN is $O(mk^2logk+n^2)$. The while loop in GCDD invokes BEN at most m times giving $O(m^2k^2logk+mn^2)$ as the overall time complexity. Treating k as constant, we have $O(m^2+mn^2)$.

V. Related Work

Hasegawa et al. describe a reference model for inter-vehicle communications [1]. We have followed their broad interpretation of IVC, which includes road-to-vehicle communications as well. Their taxonomy of IVC applications and services provides a useful framework for work in this area. Little and Agarwal present an infrastructure-less scheme for information-propagation in VANETs [4]. Their work shares with ours the goal of minimizing infrastructure. While theirs requires truly no infrastructure, ours requires the installation of dead drops. As a result, our methods function at extremely low traffic densities while their methods are designed for significant intra-cluster and inter-cluster proximity of vehicles. The FleetNet project uses Internet-connected gateways to connect disconnected clusters of vehicular networks (scatternets) [6]. This approach is similar to ours, with one key difference: Unlike FleetNet gateways, our dead drops are completely stand-alone devices that have no outside network connectivity. In deployment, a combination of these two approaches, which are compatible, is likely to work well.

Wischhof, Ebner, and Rohing describe an information-dissemination method for self-organizing networks, along with a prototype implementation using 802.11 devices [5], [14]. Their method is based on a segment-based data abstraction and dissemination model. Ghosh et al. describe a variable-resolution information-dissemination method that is based on probabilistic delivery of messages in order to manage bandwidth [2]. As the distance between the data source and a recipient increases, the probability of delivery decreases. A different scheme for managing bandwidth is described by Michael [3]: An adaptive layered data structure is used to allow nodes to progressively strip off high-resolution information from messages as they propagate farther from the source.

Our focus in this paper has been on data dissemination. The related problem of routing in IVC networks has been extensively studied. For example, Füßler et al. have compared the location-based GPSR algorithm with the topology based DSR algorithm and found the former to perform better [15]. Their results also emphasize the need for using traffic in both travel directions. Location-based routing is also used by the CarTALK project [16], [17]. Wang et al. report work on a flooding-based routing protocol and its implementation, designed for small groups of vehicles moving in the same lane. Cathey and Dailey describe methods that use transit vehicles as probes for estimating travel times [18], [19]. Our methods in this paper can perform the data collection task in this context. Our greedy approximation algorithm for the min-cost connecting dead drops builds on work by Baudis et al. on minimum connected spanning sub-hypergraphs and other problems related to polymatroids [12].

We have focused on application-level methods for data-dissemination. Our methods assume appropriate support for the lower levels of the network stack. Since we focus on low traffic densities, we may be able to use popular protocols such as 802.11b even though they are not ideal for IVC. This position finds support in the recent study of WiFi networking for IVC by Goel, Imielinski, and Ozbay [20]. MAC protocols specifically designed for this environment, such as the protocol by Fujimura and Hasegawa [21] are also applicable here. Finally, wireless dead drops have apparently also been used in a very different domain (espionage) with pedestrians instead of vehicles [7].

Wang and Wu describe a method for information-gathering in a mobile sensor network [22]. They model an environment with two kinds of nodes: low-powered wearable sensor nodes and higher-powered sink nodes, such as mobile phones or personal digital assistants with sensor interfaces. The latter receive data from one or more sensors and forward them to a network backbone composed of access points. In their environment, limited battery and buffer capacities of sensors are prime concerns, whereas they are not limiting factors in our model of vehicular networks. In general, work on delay-tolerant networks addresses environments in which networks hosts are occasionally connected and the network is frequently partitioned [23], [24]. Methods that cope with very high latencies, as may be encountered in interplanetary networks, are useful in the environment proposed in this paper as well. Our methods for the design of the minimal infrastructure can be used in conjunction with some of these protocols.

In order to conserve energy at sensor nodes, the Data MULE method uses a mobile node to collect data from stationary sensors [25]. The mobile ferrying method [26] is based on the idea of introducing some movement of mobile nodes in order to facilitate data delivery. In our environment, an analogous strategy is the use of vehicles plying on select routes in order to improve connectivity.

Musolesi, Heiles and Mascolo propose a context-aware routing algorithm for asynchronous communication in partially connected mobile ad hoc networks [27]. The MOVE algorithm uses vehicle velocity information to guide forwarding decisions in a network that uses mobile nodes to form a transit network [28]. In effect, the roles of vehicles and stationary nodes are reversed: Instead of using stationary nodes to provide inter-vehicle communication, vehicles are used to improve connectivity of stationary nodes. Such methods are complementary to ours and it seems promising to devise protocols that use them in conjunction.

VI. Conclusion

We motivated the need for data-dissemination methods that can cope with very low densities of equipped vehicles and that require minimal extra-vehicular infrastructure. We proposed a data-dissemination method that uses strategically placed wireless dead drops (dead letter boxes) as intermediaries to facilitate vehicle-to-vehicle data transfer when the traffic densities do not permit protocols that require sizeable clusters of vehicles. An important feature of this method is that the envisioned dead drops can be implemented using small and inexpensive commodity off-the-shelf hardware and readily available open-source software, facilitating deployment.

We focused on the problem of determining a set of intersections at which dead drops can be deployed to satisfy connectivity constraints at minimum cost. We formalized connectivity constraints using a flexible specification of interesting travel routes (trails) and formulated the min-cost connecting dead drops (MCDD) problem. We mapped

MCDD to the min-weight spanning sub-hypergraph (MSSH) problem, yielding a proof of the hardness of MCDD. We presented an efficient greedy approximation algorithm for MCDD that is guaranteed to produce an optimal approximation in the sense that no polynomial-time algorithm with a better approximation is likely [10].

The informal optimization problem described in Section II-A admits several formulations, of which we have studied only one in this paper. In continuing work, we are exploring some of these alternative formulations. Instead of minimizing cost subject to the constraint that every pair of trails must be connected in any admissible placement of dead boxes, we may ask for a placement that maximizes connectivity subject to a maximum-cost constraint. In this paper, we have focused on data dissemination, treating tasks such as the processing, aggregation, and expiry of data as orthogonal to dissemination. In general, it may be useful to devise methods that combine these aspects in a dead drop setting, as has been done in some other IVC environments. Given the modest requirements of our target applications (e.g., traveler information systems) we have also ignored capacity constraints on the data storage available on board vehicles and at the dead drops. However, it should be interesting to extend our method to more data-intensive applications, which will require addressing capacity constraints. We also plan to evaluate our method experimentally using both simulation studies and limited deployment.

REFERENCES

[1] Takaaki Hasegawa, Kiyoshi Mizui, Haruki Fujii, and Kaoru Seki, "A concept reference model for inter-vehicle communications (report 2)," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 810–815.

[2] Arpita Ghosh, Dan Greene, Qingfeng Huang, and Juan Liu, "Variable-resolution information dissemination," in *Proceedings of the IEEE Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, California, Sept. 2005.

[3] Lachlan B. Michael, "Adaptive layered data structure for inter-vehicle communciation in ad-hoc communcation networks," in *Proceedings of the 8th World Congress on Intelligent Transportation Systems*, Sydney, Australia, Oct. 2001.

[4] Thomas D. C. Little and Ashish Agarwal, "An information propagation scheme for VANETs," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Vienna, Austria, Sept. 2005, pp. 155–160.

[5] Lars Wischhof, André Ebner, and Hermann Rohling, "Information dissemination in self-organizing intervehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 90–101, Mar. 2005.

[6] Kendy Kutzner, Jean-Jaques Tchouto, Marc Bechler, Lars Wolf, Bernd Bochow, and Thomas Luckenbach, "Connecting vehicle scatternets by Internet-connected gateways," in *Proceedings of Multiradio Multimedia Communications (MMC)—Communication Technology for Vehicles*, Dortmund, Germany, Feb. 2003.

[7] Paul Reynolds, "Old spying lives on in new ways," BBC News. http://news.bbc.co.uk/go/pr/fr/-/2/hi/europe/4639758.stm, Jan. 2006.

[8] "OpenWRT," http://openwrt.org/, 2004.

[9] Claude Berge, *Graphs and Hypergraphs*. Amsterdam, Netherlands: North-Holland, 1973.

[10] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman Company, Nov. 1990.

[11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, July 2001.

[12] Gregor Baudis, Clemens Gröpl, Stefan Hougardy, Till Nierhoff, and Hans Jürgen Prömel, "Approximating minimum spanning sets in hypergraphs and polymatroids," in *Proceedings of the 27th International Colloquium on Automata, Languages, and Programming (ICALP)*, Geneva, Switzerland, July 2000.

[13] Uriel Feige, "A threshold of $\ln n$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[14] Lars Wischhof, André Ebner, and Hermann Rohling, "Self-organizing traffic information system based on car-to-car communication: Prototype implementation," in *Proceedings of the 1st International Workshop on Intelligent Transportation*, Hamburg, Germany, Mar. 2004, pp. 49–53.

[15] Holger Füßler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, and Dieter Vollmer, "Location-based routing for vehicular ad-hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 1, pp. 47–49, Jan. 2003.

[16] Jing Tian and Luca Coletti, "Routing approach in CarTALK 2000 project," in *Proceedings of the 12th IST Mobile and Wireless Communications Summit*, Aveiro, Portugal, June 2003.

[17] Jing Tian, Kurt Rothermel, and Lu Han, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Proceedings of the 6th IEEE International Conference on Intelligent Transportaion Systems (ITSC)*, Shanghai, China, Oct. 2003.

[18] Fredrick W. Cathey and Daniel J. Dailey, "Implementation of traffic management using transit probes," in *Proceedings of the IEEE 8th International Conference on Intelligent Transportation Systems*, Vienna, Austria, Sept. 2005, pp. 13–16.

[19] ——, "Estimating corridor travel time by using transit vehicles as probes," *Transportation Research Record*, pp. 60–65, 2003.

[20] Samir Goel, Tomasz Imielinski, and Kaan Ozbay, "Ascertaining viability of WiFi based vehicle-to-vehicle network for traffic information dissemination," in *Proceedings of the 7th IEEE International Conference on Intelligent Transportaion Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 1086–1091.

[21] Kaichi Fujimura and Takaaki Hasegawa, "A collaborative MAC protocol for inter-vehicle and road to vehicle communications," in *Proceedings of the 7th IEEE International Conference on Intelligent Transportaion Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 816–821.

[22] Yu Wang and Hongyi Wu, "DFT-MSN: The delay fault tolerant mobile sensor network for pervasive information gathering," in *Proceedings of the 25th IEEE Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, Apr. 2006.

[23] Vinton G. Cerf, Scott C. Burleigh, Robert C. Durst, Kevin Fall, Adrian J. Hooke, Keith L. Scott, Leigh Torgerson, and Howard S. Weiss, "Delay-tolerant network architecture," IETF DTN Research Group. Internet Draft. http://www.ietf.org/, Mar. 2006.

[24] Kevin Fall, "A delay-tolerant network architecture for challenged internets,," in *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[25] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the First International Workshop on Sensor Network Protocols and Applications*, Anchorage, Alaska, May 2003, pp. 30–41.

[26] Wenrui Zhao, Mostafa H. Ammar, and Ellen W. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Roppongi Hills, Tokyo, Japan, May 2004, pp. 187–198.

[27] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proceedings of the IEEE 6th International Symposium on a World of Wireless Mobile and Multimedia Networks (WOWMOM)*, Taormina, Italy, 2005, pp. 1–7.

[28] Jason LeBrun, Chen-Nee Chuah, Dipak Ghosal, and Michael Zhang, "Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of IEEE 62nd Semiannual Vehicular Technology Conference (VTC), Fall*, Dallas, Texas, 2005, pp. 1–5.