Frequency-Based Relay Placement in Mobile Networks

George Dean Bissias, Brian Neil Levine, and Ramesh Sitaraman

June 7, 2010

Abstract

Stationary relay nodes have the potential to boost routing efficiency in mobile networks at little additional cost. Unlike mesh nodes, relay nodes can be placed non-contiguously and require no wired infrastructure. However, ensuring that relay placement results in maximum performance is a challenging problem with concrete results being few. We explore the quality of a simple frequency-based placement heuristic using traces from a vehicular network testbed, and we present a novel bounding algorithm that validates this heuristic. In contrast to prior work, we consider how placement affects both network throughput and packet delivery delay.

Our experimental testbed consists of approximately 40 buses operating for 58 days, over 150 square miles, and outfitted with 802.11 radios. Our analysis reveals that placing relays at the most frequently visited sites produces high-quality results. On 40% of the days, and assuming RAPID routing, the throughput achieved for a given average delivery delay is within 40% of the optimal placement. Moreover, these results are found using location data from 30 day old traces, which are tested on the later 28 days. Indeed, overall, the most popular 20 locations are chosen 80% of the time. We also searched for the ideal relay quantity. We observed no significant improvement in throughput or delay for quantities greater than 50 relays.

1 Introduction

For mobile nodes to receive uninterrupted wireless coverage over a large geographic area, they require the support of a large infrastructure of cell towers or access points. Cellular data coverage is currently expensive for carriers to deploy and has a recurring cost of about USD$50 per device per month per user. WiMAX towers are tens of thousands of dollars and require allocation of a band, and therefore similarly cannot be easily deployed by common users wishing to support mobile networking. More accessible are WiFi APs, but such nodes rely on an underlying high speed backbone, such as provided by an institution or carrier. Studies have shown that WiFi networks typically provide intermittent access [13], even when deployed in an unbroken mesh [3].

A flexible alternative is offered by *relay nodes*, which are stationary devices with a radio and storage. Relays hold packets left by one mobile node that are later picked up by another mobile node. Because relays require no wired or fiber backbone, they are much cheaper to deploy, and relays can cover larger areas because they are not contiguously deployed. Relays support only disruption tolerant networking (DTN), which trades lower performance for cheap coverage of large areas. When solar-powered, they are also independent of the power grid [4]. Most previous work on relays for supporting mobile networks has focused on the design of the relay, including energy efficiency [4], asymptotic routing performance for large numbers of boxes [15], and performance comparison against other infrastructure [6].

Our focus is on the problem of relay placement: *given x relays, where should they be geographically placed to most improve the performance of a particular mobile network?* We call this problem the **Relay Placement Problem**, denoted Π. The problem of optimally placing stationary relay nodes is complicated by its entanglement with optimal routing. Zhao et al [26] have shown this joint problem of optimizing routing and placement is NP-Hard, even if the schedule of meetings between mobile nodes and relays is known *a priori*. Balasubramanian et al. [1] have shown that optimizing routing alone is NP-Hard and that the problem even resists close approximation. Indeed, according to these results, it is NP-hard just to evaluate the quality of a given solution to Π, which takes the Relay Placement Problem out of NP entirely. This entanglement motivates the development of a new problem, the **Decoupled Placement Problem**, Π^D, which is optimal with respect to a given (non-optimal) routing strategy.

In this paper, we evaluate a simple heuristic as a suboptimal solution to Π^D. Our simple placement strategy is to choose relay locations that are *most frequently visited*. This strategy, denoted the *Frequency-based Placement* (f for short), lacks knowledge of demand, link capacity, and other critical factors that an optimal solution would have as input. Despite its simplicity, we are able to formally validate this strategy by comparing it to the optimal solution for Π^D. This comparison is indirect since Π^D itself is NP-hard and difficult to approximate. The comparison is achieved through a relaxation of Π^D to Π^R, or the **Relaxed Placement Problem**, which has two important properties. First, it is simple enough to be solved to within high accuracy. Second, for any desired relay throughput, the optimal solution for Π^R will experience delay no greater than the optimal solution for Π^D. This allows us to show that the Frequency Based Placement performs well compared to the *optimal* solution to Π^D. Figure 1 illustrates the relationships between these four problems. As a result of our analysis, we show that the performance of f and Π^R are close, and therefore so is f and Π^D.

In sum, our contributions are as follows.

- We develop a dynamic program that solves Π^R to within high accuracy.
- We apply this solution to quantify the performance of frequency-based relay placement in a trace-driven simulation. We show that on most days, using a well-performing DTN routing protocol, the throughput achieved for a given average delivery delay is within 40% of the optimal solution to Π^D on 40% of the trace days. Better results are achieved when constraints on average delivery delay are relaxed.
- We explore the saturation point for relays, that is the point where adding more relays does not increase throughput or decrease average delivery delay. Our results show that 50 relays are sufficient to saturate the DTN used for our experiments.

We also searched for the ideal quantity of relays. We observed no significant improvement in throughput or delay for quantities greater than about 50 relays in our network. Our traces are based on our long-running vehicular network, DieselNet, which is comprised of about 40 public transit busses sparsely covering a 150 square mile area. We use 58 days of actual bus-to-bus contacts to show how placement of relays would perform in a real setting.

Complex problems routinely yield to simple heuristics for specific, but common, classes of inputs seen in practice. We hypothesize that this is the case for the class DTNs containing network hubs, which accordingly manifest a highly skewed spatial distribution of mobile nodes. In effect, this renders a small number of locations multiple orders of magnitude more popular than the remaining locations. Conceptually, our problem is the compliment of degrading a network by removing nodes.

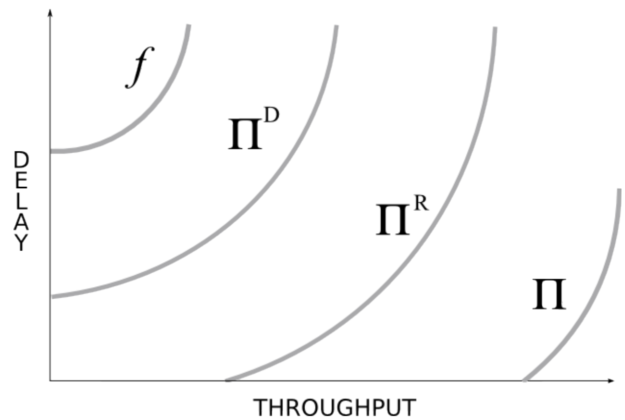


Figure 1: For all placement problems, delay increases as more throughput is satisfied. Plotted are the Frequency-based Placement, f, and the the optimal solutions to Problems Π^D, Π^R, and Π. For a given DTN and desired throughput, the placement offered by f will always incur greater delay than that found by the optimal solution to Π^D, which will itself incur more delay than the placement offered by the optimal solution to Π^R

And it has been shown empirically by authors such as Faloutsas et al. [14] and Barabasi et al. [7] and mathematically by Ballobas [8] that graphs with similarly skewed degree distributions are most vulnerable to the removal of vertices of highest degree. In our previous work, we attacked DTNs specifically by removing first nodes with the highest number of contacts. With these results as a foundation, it is reasonable to hypothesize that assigning relays to the most frequently visited locations in a DTN is a good heuristic for solving the placement problem.

2 Related Work

While still a new topic, a growing set of works has examined the performance gains that are possible in a sparse mobile network that results from adding infrastructure.

Banerjee et al. [4] design and deploy energy-efficieny solar-powered relays and demonstrate the network improvement from using a single relay in a DTN. While that paper proposed a longer range, hailing radio and shorter range data radio, Jun et al. [17] examine the use of the opposite case. Nain et al. present the asymptotic routing performance for large numbers of relays [15]. Banerjee et al. [5] show that incorporating additional Internet access points (that can be placed anywhere in a geographic area) can be twice as valuable as adding the same number of mesh points (that can only be placed contiguously); and APs are five times as valuable as adding the same number of relays. However, relays do not require an underlying wired or wireless infrastructure, and are therefore cheaper and easier to deploy. In the same paper, Banerjee et al. show the performance of placing boxes according to locations most visited by mobile nodes, but those experiments did not validate the quality of such an approach other than against other infrastructure.

Two papers are most relevant to ours. Zhao et al. [26], who specifically address the placement problem, introduce a heuristic that has the goal of increasing throughput; however, that analysis did not address delivery delay, and it lacked validation and evaluation based on a real network scenario. We could not extend this technique for out analysis as it became computationally intractable as packet delay constraints were added to the model. Lochert et al. [22] examine the use of stationary relays in VANETs, proposing a genetic algorithm for placement and evaluating the algorithm via simulated mobility in a large city using ns-2. They compose a bit vector with one entry for each potential relay location and conduct an entire round of simulations each time the vector is mutated. We did not consider adapting this technique as our preference was to identify the simplest approach possible, and we deferred including the technique in our analysis as doing so was not necessary for validating our solution.

Another set of works add mobile ferries to the network. For example, Jun et al. [18] examine how nodes can optimize routing by leveraging the known schedule of a ferry. And Burns et al. [9] design a system where ferries move autonomously to meet demand in a DTN.

A series of DTN routing protocols have been proposed based on historical mobility patterns [12, 2, 19, 23]. Jain et al. [16] show that routing in a DTN is not as simple as choosing among available paths, and show how to encode DTN properties into a linear program. We specifically use the RAPID protocol [1] in this paper to evaluate relays. RAPID aims to directly affect QoS metrics by tailoring routing accordingly. Routing decisions at local nodes are based on the estimated probability that a given node can satisfy the metric of interest.

A set of other work has examined coding [11, 21]. For example, Wang et al. [24] compared erasure-coding routing in DTNs to packet flooding, simple replication, and a history-based method. They observed that overall delay was best for the history-based strategy, while the coding strategy ensured the best worst-case performance. And Widmer and Le Boudec [25] show that coding with replication provides a higher delivery rate than replication alone, but at the cost of higher delay.

3 Problem Statement

We begin with a DTN $\mathcal{D} = (N, L, C)$, with node set N of size n, L, a list of l potential relay locations, and C, a contact mapping defined as

$$C(e_1, e_2, t) = \begin{cases} B & e_1 \sim e_2 \text{ at time } t \\ 0 & \text{otherwise} \end{cases},$$

where $e_1, e_2 \in N \cup L$ and $t \in \{1, \ldots, T\}$. $\mathcal{D}(K)$ is DTN $\mathcal{D}$ with relays $K \subseteq L$ added to the network.

DEFINITION 1: Let $\mathcal{S}(\mathcal{D}$,M$, \mathcal{R})$ denote the unique *simulation* on DTN $\mathcal{D}$ with end-to-end bus-bus demand, M, and routing protocol, $\mathcal{R}$.

For a given natural number m, our goal is to determine the best m relay locations given $\mathcal{D}$ so as to maximize throughput while minimizing delay.

PROBLEM 1:[Placement,] Π

Input: DTN $\mathcal{D} = (N, L, C)$, $m \in \mathbb{N}$, and throughput P^*.

Output: A set of locations $K \subseteq L, |K| = m$, such that for some simulation, throughput is less than or equal to P^* and delay is minimized.

We have been deliberately vague in our description of throughput and delay, both of which will be defined formally in the sequel.

3.1 Decoupling Routing and Placement

The optimal relay placement depends critically on the dynamics of the mobile network. In principle, the optimal placement must be coupled with optimal routing to optimize QoS metrics like throughput and delay. As such, the mobile routing problem reduces to the relay placement problem, which means that placement in a DTN is at least as difficult as routing.

Balasubramanian et al. [1] found that when n packets are in demand between mobile nodes, even with predictable movement, it is NP-hard to find the optimal routing, and unless $P = NP$, that routing generally delivers $\sqrt{n}$ more packets than the best polynomial-time approximation. They go on to show that the delivery rate of this class of *mobility-aware* algorithms is essentially unbounded compared to those with no knowledge of future movement.

In light of the intractability of optimal routing, we seek to decouple routing from placement by choosing a (suboptimal) routing strategy and proceeding under the assumption that it is optimal. Before placement, and in the absence of any link capacity constraints, the optimal routing will pass various components of demand between mobile nodes. The net flow in this network represents optimal aggregate one-hop demand. Figure 2 illustrates how end-to-end demand can be aggregated into one-hop demand.

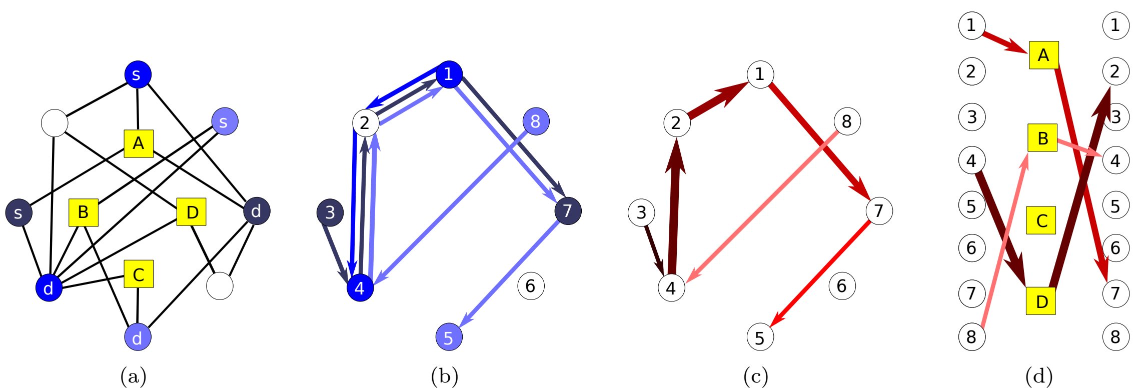


Figure 2: (a) DTN $\mathcal{D}$: at fixed time, with demand M for commodities indicated by color and labeled s for source and d for destination. (b) Stage 1: uncapacitated demand satisfaction without relays (c) M^1, one-hop demand, formed from uncapacitated solution. (d) Stage 2: relays partially accommodating demand M^1 while respecting capacity.

DEFINITION 2: For some $\mathcal{S}(\mathcal{D}, M, \mathcal{R})$, let $M^1(\mathcal{S})$ denote the *one-hop demand* between busses that results from evaluating simulation $\mathcal{S}$ on parameters $\mathcal{D}, M, \mathcal{R}$.

Below, we refer to M^1 without parameter $\mathcal{S}$ when its identity are clear from its context. Demand has been simplified, but once relays are placed, we still require a two-hop routing scheme. This routing pass data from source bus to a relay and then on to the destination bus.

3.2 Metrics

The objective criteria include maximization of a measure of throughput and minimization of a measure of delay. Decoupling routing from placement allows us to assume that finding a two-hop routing that maximizes the portion of M^1 satisfied is equivalent to maximizing packet delivery. Accordingly, we define throughput as,

$$P \equiv \textit{total number of bytes passing through relays}.$$

Similarly, we define average delivery delay as,

$$\overline{Y} \equiv \textit{weighted average of products } iF_i,$$

where F_i is the amount of flow passing through relays at time i. We keep Y as the symbol for *total delay* which is simply the sum of terms iF_i.

We will use functional notation when either P or $\overline{Y}$ are to be placed in a particular context. For example, $P(\mathcal{D}, M^1, \mathcal{R}^2)$ denotes throughput through relays in DTN $\mathcal{D}$ given one-hop demand M^1 and two-hop routing $\mathcal{R}^2$ between busses.

3.3 Decoupled Problem Formulation

We are now in a position to formally define the decoupled placement problem. Intuitively, given a DTN and one-hop demand, we seek a set of locations K to place relays and a two-hop routing $\mathcal{R}^2$, which tells us how to pass data from source bus to destination bus in the resulting network.

PROBLEM 2:[Decoupled Placement,] Π^D

> **Input:** $\mathcal{D}$, M^1, $m \in \mathbb{N}$, and P^*.
>
> **Output:** $\mathcal{R}^2$ and $K \subseteq L, |K| = m$,
> such that $P(\mathcal{D}(K), M^1, \mathcal{R}^2) = P^*$ and $\overline{Y}(\mathcal{D}(K), M^1, \mathcal{R}^2)$ is minimized.

THEOREM 1: Problem Π^D is NP-hard.

PROOF: We reduce the Maximum Coverage Problem, MC, to Problem Π^D. This makes Π^D NP-hard since MC has been shown to be NP-complete [10].

Let $\mathcal{U}$ be a universe of items and χ be any collection of subsets of $\mathcal{U}$. To solve MC, we wish to find the largest number of items in $\mathcal{U}$ that can be covered by some k sets in χ. Construct instance $I = (\mathcal{D}, M^1, m, P^*)$ of Π^D as follows. Assume that there is a single time step in $\mathcal{D}$ and let $m = k$. For each $i \in \mathcal{U}$, create busses b_{i1} and b_{i2} in $\mathcal{D}$ and assign a single unit of one-hop demand between them. Map each set $S \in \chi$ to a relay location $\lambda \in L$, and let both b_{i1} and b_{i2} be adjacent to λ iff $i \in S$.

Consider the output K and $\mathcal{R}^2$ from Π^D. Exactly u items in $\mathcal{U}$ can be covered by k sets in χ iff there exists finite average delay when $P(\mathcal{D}(K), M^1, \mathcal{R}^2) = P^*$. We can find the maximum value for u by testing a logarithmic number of values for P^* using binary search. This constitutes a polynomial-time reduction and completes our proof.

□

It is not currently known how well Problem Π^D can be approximated. The remainder of our analysis relaxes Π^D to a form that can be easily approximated. The next section introduces a dynamic program to achieve this end.

4 Placement Relaxation

We seek to relax Π^D as a simpler problem, Π^R, which consistently provides optimistic estimates of average delivery delay. This motivates two assumptions.

- **Ample Demand.** If two relays are each capable of satisfying a unit of demand, then there is enough demand to accommodate both.
- **No Co-location.** Busses arrive at relays at different times, so that each bus has full access to the bandwidth available a relay.

4.1 Availability

Algorithm FRONT_LOAD, below, will calculate, *maximal egress*, $E(x, t)$, for each relay x and each time unit t. This is the greatest aggregate flow achievable through the given relay during the given time interval. As its name implies, FRONT_LOAD also ensures that all flow is reported available as early as possible. The idea is to update the *store* of flow at each location, and to pass that flow as quickly as possible (since this will afford the lowest delay). The following definitions help with exposition. $M^1(b_1, b_2; t)$ is the maximum flow between busses b_1 and b_2 at time t. $A(i, t)$ is the set of busses adjacent relay i at time t. $B(x, t)$ are the number of bytes of capacity at location x and time t.

4.2 Allocation

It only remains for us to devise a scheme to select relays in such a way so as to minimize average delivery delay for a fixed throughput, P^*, and number of relays, m. By construction, $E(x, t)$ is an $l \times T$ matrix. We aim to find the m rows of E that, when constrained to an operating window of time, will deliver a total of P^* bytes and incur minimum average delivery delay. Since earlier flow is better, it's always a good idea to choose the earliest possible time window for each relay. Accordingly, we mandate each relay to send flow only for a certain period of time beginning at time $t = 1$ and ending at some *termination time* $t = t^*$. To summarize, with matrix $E(x, t)$, we must choose m relay locations and m corresponding termination times for each relay. Also, notice that, since P^* is fixed, it's sufficient to minimize the total delivery delay, Y, instead of its average. Formally, we wish to solve,

PROBLEM 3:[Relaxed Placement Problem, Π^R]

Input: $E(x, t)$, $m \in \mathbb{N}$, P^*.

Output: m row indices of $E(x, t)$, I, and m termination times, X, which achieve P^* and minimize Y.

Algorithm 1 FRONT_LOAD(F, B)

```
∀t_o, x,  E(x, t_o) ← 0                                      Initialize Egress for each out-time
∀t, b_i, b_o,  d(t, b_i, b_o) ← M^1(b_i, b_o; t)             Initialize Demand between input and
output busses
for x ∈ {1, ..., L} do
  ∀b_i,  I(b_i) ← 0                                          Initialize Ingress for the given input bus
  for t ∈ {1, ..., T} do
    ∀b_i ∈ A(x, t),  I(b_i) ← I(b_i) + B(x, t)               Add bytes to each relay, for each
    adjacent bus at time t
    for b_o ∈ A(x, t);  t_i ∈ {1, ..., t};  b_i ∈ A(x, t_i) do
      δ ← max{B(x, t), min{I(b_i), d(t_i, b_i, b_o)}}        Pass as many bytes as Egress without
      exceeding B
      E(x, t) ← E(x, t) + δ
      I(b_i) ← I(b_i) − δ
      d(t_i, b_i, b_o) ← d(t_i, b_i, b_o) − δ
    end for
  end for
end for
```

Problem Allocation is analogous to a Knapsack Problem that is confined to carry m items and allows fractional assignment (corresponding to termination times) of each. In the analogy, potential relay locations map to items. The weight of an item maps to the throughput through the given relay, which varies linearly and proportional to the relay's termination time. The value of an item is analogous to average delivery delay, and is a (generally) nonlinear function of its termination time. Most instances of Knapsack yield to dynamic programming solutions, so we take the same approach in solving Problem Allocation. To that end, divide throughput P^* into a number of intervals, $F = \{f_1, \dots, f_p\}$, so that $\sum f_i = P^*$. Let $Y(x, f_i)$ give the minimum total delivery delay for relay location x and flow f_i. We have,

$$Y(x, f_i) = \sum_{t \le t^*} tE(x, t),$$

where

$$t^* = \arg\min_{t'} \sum_{t \le t'} E(x, t) \ge p^*.$$

Algorithm ALLOCATE, whose construction is deferred to Appendix A, constructs a table of total delay values, δ. Each entry $\delta(m, L, f_i)$ of the table gives the smallest total delivery delay achievable by sending as many as f_i units of flow through at most m relays chosen from the set L. In contrast to table δ, let $\Delta(k, L, f)$ be a *continuous* function of flow f giving the smallest total delivery delay achievable by sending as many as f units of flow through at most k relays chosen from the set L. The largest gap between the total delivery delay reported for $\delta(m, L, f_i)$ and that reported by $\Delta(m, L, f)$ with $f_{i-1} \le f \le f_i$ represents the approximation error of Algorithm ALLOCATE and the optimal solution to the Allocation Problem.

THEOREM 2: For any choice of k and L, $\delta(k, L, f_i) \le TU + \Delta(k, L, f)$ when $f_i \le f$.

PROOF:

For a given f_i, the functions $\Delta(k, L, f_i)$ and $\delta(k, L, f_i)$ can be broken down as

$$\delta(k, L, f_i) = \min_{\sum F_j = f_i} \sum_j \delta(1, L, F_j)$$

and

$$\Delta(k, L, f_i) = \min_{\sum x_j = f_i} \sum_j \Delta(1, L, x_j)$$

where F_j is the discrete flow and x_j the continuous flow assigned to relay j, respectively.

Because the function δ can assume only discrete values of flow, it's possible that the true minimum total delay is less than what is rendered. However, since the function Δ is non-increasing with x and $\delta(1, L, f_i)$ is accurate at discrete flows, f_i, we have

$$\begin{aligned}
\Delta(k, L, f_{i-1}) &= \min_{\sum x_j = f_{i-1}} \sum_j \Delta(1, L, x_j) \\
&= \min_{\sum F_j = f_{i-1}} \sum_j \delta(1, L, F_j) \\
&\leq \min_{\sum F_j = f_i} \sum_j \delta(1, L, F_j) \\
&= \delta(k, L, f_i) \\
&= \min_{\sum x_j = f_i} \sum_j \Delta(1, L, x_j) \\
&= \Delta(k, L, f_{i-1}),
\end{aligned} \tag{1}$$

or $\Delta(k, L, f_{i-1}) \leq \delta(k, L, f_i) \leq \Delta(k, L, f_{i-1})$. Going from $\Delta(k, L, f_{i-1})$ to $\Delta(k, L, f_i)$, we will pass no more than one additional unit (U bytes) of flow through all boxes. This additional flow cannot increase the total delivery delay by more than a factor of T. Hence,

$$\Delta(k, L, f_i) - \Delta(k, L, f_{i-1}) \leq TU.$$

□

The preceding analysis shows that we can make this gap as small as we like by continuing to refine the set F. On the other hand, this makes ALLOCATE less efficient computationally.

5 Testbed

All mobility traces came from DieselNet [20], which is a network of busses serving a large area of western Massachusetts. The area of operation is a sparse covering of about 150 square miles. Traces came from a three month period spanning the beginning of February 2007 to the beginning of May 2007. All weekends and holidays were dropped leaving a total of 58 days. On each day, we

Figure 3: Bus routes.

eliminated busses that made no contact with other busses. The result was an average of more than 15 busses on the road each day. Figure 3 shows a map of the area of operation that is overlayed with lines dipicting the routes upon which the DTN is active.

Each bus was outfitted with a small computer comprising an 802.11 wireless transmitter/receiver and hard drive. During the course of the day, busses recorded their location at 10 second intervals as well as any contact events with other busses. During contact events, busses attempted to send as much data as possible via wireless transmission in duplex mode. The average transmission size was 1.7MB with a standard deviation of 1MB.

5.1 Simulation

We used the RAPID routing protocol [1] for all simulations. RAPID is a history-based routing protocol that maintains information at each node regarding past contact events with other nodes, and in our case, other relays. When a pair of nodes or relays establish contact they combine meta information concerning past contact events. Each half of the contact pair then prioritizes his own packets according to how likely it is that the other node will be able to deliver them according to a certain service metric. In our simulations we elected to have RAPID attempt to minimize average delivery delay. Each simulation injected 2000 packets, each 1K in size, into the network at the beginning of the day at source busses chosen uniformly and independently at random. The corresponding destination busses were also chosen uniformly and independently at random. Any location registered by the GPS unit on a bus was considered a candidate for relay placement.

5.2 Assumption and Simplifications

We were able to use real transmission values for connection events between busses, but naturally had no transfer data for GPS events. In order to estimate transmission values between busses and relays, we used the mean transfer value over all connection events for the given day. We further assumed that all relays had a radio range of 140 meters, and tessellated the region of operation

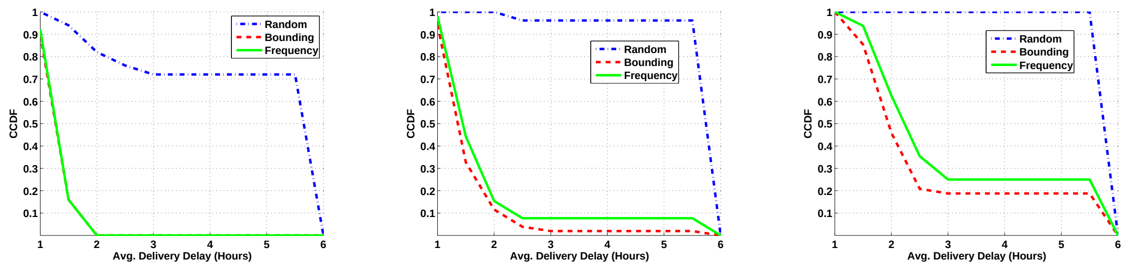


Figure 4: Average delay of (a) 5MB (b) 15MB and (c) 30MB flow passing through 50 relays is plotted for Strategies Random, Frequency, and Bounding.

with squares this same length on every side. Each square was allowed to hold no more than one relay at a time.

6 Evaluation

We tested the Frequency-based Placement Strategy against the Bounded Solution and in trace driven simulations. Our analysis serves four distinct purposes.

1. Demonstrates the quality of the Frequency-based Placement, f, relative to the optimal solution for Π^D

2. Illustrates the application of our bounding algorithm, an approximation to the optimal solution for Π^R

3. Shows that choosing locations based on long-term frequency (that is, over many weeks) will not only provide good locations for future days, it actually outperforms the basic Frequency-based Placement

4. Identifies the saturation point for relays in the DieselNet testbed— that is, the number of relays after which adding more relays does not improve throughput or delay

For simplicity, we shall hence forth refer to the optimal solutions to f, Π^D, and Π^R as the *Frequency*, *Target*, and *Bounding* solutions, respectively. We also introduce an improvement to the Frequency solution called *Trained*, which chooses locations from the list of locations most frequently visited in the first 30 days of traces.

6.1 Bounding Analysis

Comparing the Frequency solution to the Target solution is accomplished via proxy, the Bounding solution. Algorithm ALLOCATE constructs a table, δ, of delay values. Each entry, $\delta(m, L, f_i)$, gives a close estimate of the lowest total delivery delay, Y, for throughput $P = f_i$ and m relays chosen

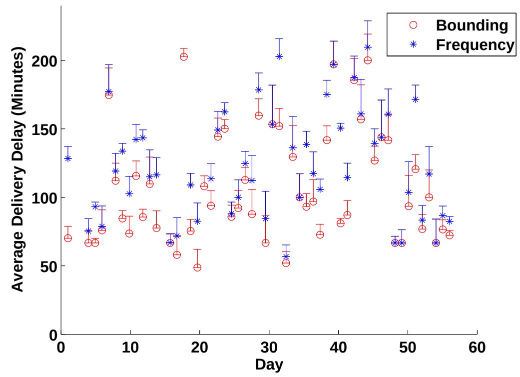


Figure 5: Daily minimum average delay required to achieve 15MB throughput with error shown due to granularity in dynamic program.

from the set of location L. The reported value for Y bounds, from below, the total delivery delay for *all* two-hop routing strategies for the given problem. The two-hop routing is that which satisfies one-hop demand, M^1, by passing flow from source bus to a relay and then on the destination bus. It's not to be confused with the end-to-end routing, which satisfies overall demand M and always routes via RAPID.

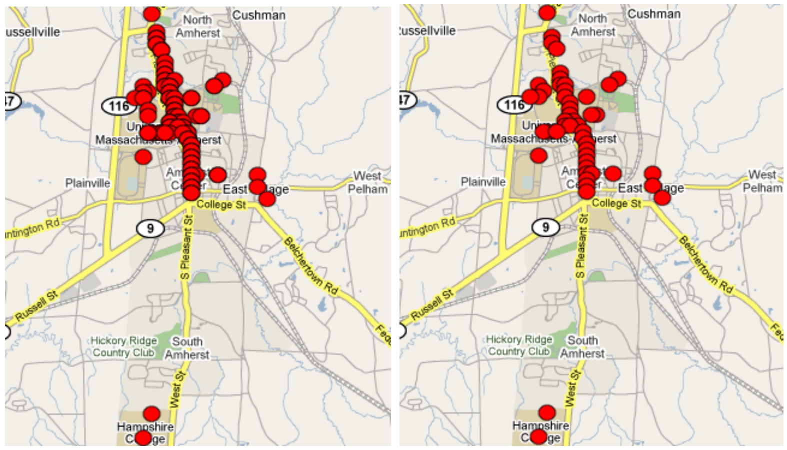
While this process is responsible for *generating* the Bounding Solution, it can also be used to provide a lower bound on the optimal two-hop routing for the Frequency Placement. This is accomplished by limiting L to the m most frequented locations. This allows us to test various scenarios quickly before committing greater resources to full-blown simulation. The comparison is accurate to the extent that a) RAPID is close to optimal and b) our relaxation Π^R closely follows Π^D. In addition to Frequency and Bounding Placements, we also tested Random Placement, which chose m locations uniformly at random from all those that were at some point occupied by busses.

Using Algorithm ALLOCATE, we generated three bounding tables for placements of up to 50 boxes. We made one table each for Random, Frequency, and Bounding. This was accomplished by restricting the location sets, L, as follows:

- **Random:** 50 locations chosen uniformly at random
- **Frequency:** 50 most popular locations
- **Bounding:** all locations

Each table was reported along with a bound on rounding error provided implicitly by Theorem 2. In the case of Bounding, we also generated the list of locations predicted to achieve the reported total deliver delay. For a given throughput P^*, the *average* delivery delay, $\overline{Y}$, was calculated from each table value by dividing it by P^*. Figure 4(a)–(c) show how $\overline{Y}$ compares between each routing strategy and for three different values for P^*. The Random strategy performs far worse than the other two for all values of P^*. For this reason, we explore Random Placement no further.

When just 5MB of data pass through the relays, the Frequency and Bounding solutions are predicted to perform nearly identically, which means that, up to our relaxed solution, there should



(a) (b)

Figure 6: The 50 most commonly chosen locations over the first 30 days for (a) bounding placement (b) Frequency-based placement.

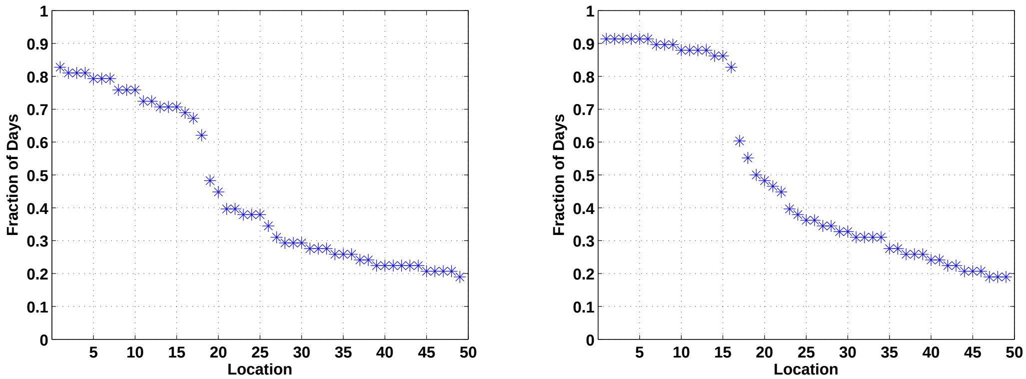
be no difference between the Frequency Solution and the Target Solution. A small gap appears for $P^* = 15$MB and $P^* = 30$MB, indicating a slight advantage for the Target Solution over the Frequency Solution as throughput increases. Figure 4(c) also provides a practical upper bound for P^* because the average delivery delay, with median value at approximately 2 hours for even the Bounding Placement, becomes untenable when $P^* = 30$MB.

This causes us to focus for the remainder of our analysis on the case where $P^* = 15$MB. In other words, we restrict the aggregate flow through the relays to 15MB. Figure 5 is a more detailed rendering of the information in Figure 4(b). It shows the differences between $\overline{Y}$ for the Frequency and Bounding Solutions on each of the 58 trace days. Additionally, error bars show the rounding error predicted by Theorem 2. To keep each value a lower bound on $\overline{Y}$, we consider the lower value to be the true average delivery delay. Though there is significant deviation from day to day, there are many days where Frequency and Bounding Solutions are nearly identical. This implies that, for those days, the Frequency Solution will be very close in performance to the Target Solution.

6.2 Visualizing Placement

The placement strategies, Frequency and Bounding, operate with vastly different sets of information and depth of analysis. While the Frequency Solution relies entirely on GPS data, the Bounding Solution additionally utilizes complete knowledge of one-hop demand, and the former is much less computationally expensive than the later. In light of these differences, the similarity of the two placements is striking. Figures 6(a)–(b) show the 50 most commonly chosen relay locations for Bounding and Frequency Solutions over the first 30 days of traces. Comparing the placements to the bus routes depicted in Figure 3, we see that the relays span a length of road running through Amherst Center and various adjacent side streets.

For the Bounding and Frequency Placements, respectively, Figures 7(a)-(b) show the fraction of days that utilize each of the 50 most popular locations. As we alluded in Section 1, there are



(a)

(b)

Figure 7: Redundancy of top 50 location choices for the first 30 days for (a) Bounding Placement (b) Frequency-based Placement.

small number of locations (roughly 20 according to both strategies) that are chosen on the majority of days. These locations constitute hubs, whose existence provide support for the rationality of a frequency-based placement scheme. The relatively high frequency of the other 30 locations makes a case for broadening the scope of our frequency analysis. To test this idea, we create a new placement strategy that trains on the first 30 days by simply observing the popularity of each location on each day, then tests on the later 28 days. We call this the *Trained* Placement Strategy. The Trained Strategy is particularly appealing because it has the potential to provide a good relay placement without prior knowledge of bus activity.

6.3 Differences in Relay Quantity

Thus far we have only considered how the network behaves when 50 relays are placed. The question remains how the average delivery delay changes when placing other quantities of relays. Figure 8 shows how average delivery delay changes when 15MB of data is passed through 10, 20, 30, 40, and 50 relays. It indicates that there is a significant reduction in average delivery delay when moving from 10 to 20 relays. However, moving from 40 to 50 relays produces no discernible improvement. This suggests that for $P^* = 15$MB, 50 relays saturates network.

6.4 Simulation Results

The previous three sections looked at lower bounds on average delivery delay for Frequency and Bounding Placement for the best two-hop routing strategy. This analysis allowed us to choose appropriate network parameters and offered insight into the physical layout of each placement. Unfortunately, even if we could develop an algorithm for optimal two-hop routing, such an algorithm would not necessarily translate to an optimal end-to-end routing strategy. In the remainder of

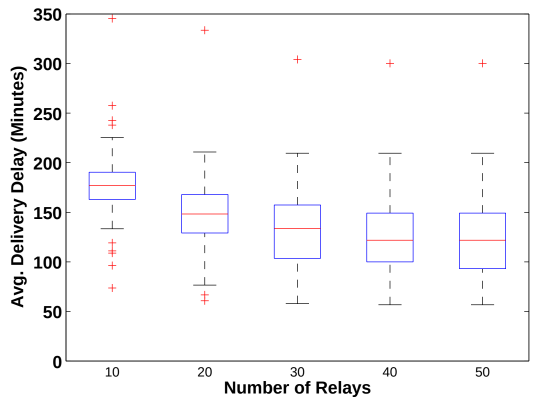


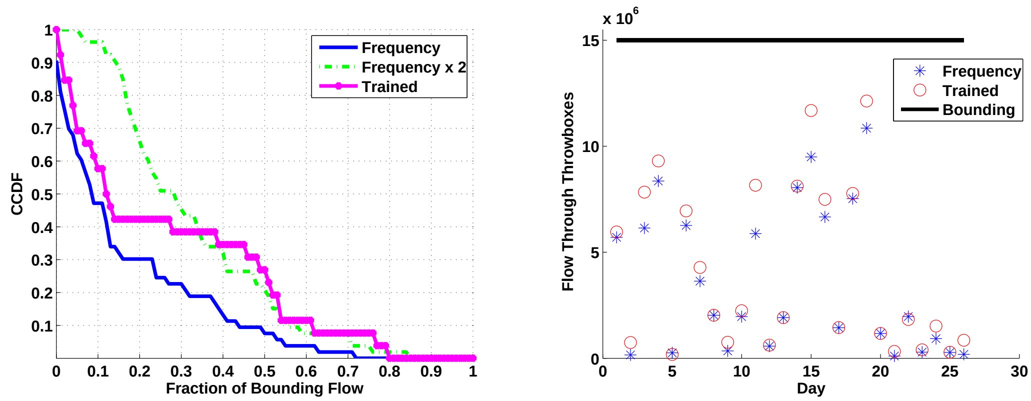
Figure 8: Average Delay for 15MB throughput given various number of relays.

this document, we turn our attention to validation via trace driven simulation. This validation is important because it allows us to draw conclusions about the absolute difference between the Frequency and Target Solutions.

Several different trace-driven simulations were run in broad accordance with the stipulations of Section 5.1 and with additional refinements concerning relay placement and packet TTL (time-to-live). Specifically, we implemented five versions of the simulator

- **Basic:** No relays and no TTL
- **Bounding:** Relays and TTL from Bounding Solution
- **Frequency:** Relays from Frequency Solution and TTL from Bounding Solution
- **None:** No relays and TTL from Bounding Solution
- **Trained:** Relays from the Trained Solution and TTL from Bounding Solution

The Basic Simulation was run first for each trace day to provide one-hop demand, M^1, for the Bounding Solution. We next ran Algorithm ALLOCATE using M^1 and with parameters $m = 50$ and $P^* = 15$MB. This gave the Bounding Solution and TTL values for packets. Finally, each of the four simulators Bounding, Frequency, None, and Trained were run on each day of traces. For Frequency, None, and Trained, the TTL for packets in each simulation was set to the average delivery delay found in the Bounding Solution for 15MB of throughput. For Frequency×2 we doubled the TTL used by the others. We then compared the aggregate throughput through relays, P. Since the Bounding Solution is optimistic, we expect all placements (except possible for Frequency×2) to path something less than 15MB before all packets reach TTL. Figure 9(a) shows a CCDF for the fraction of total possible flow (15MB) actually achieved using each placement strategy. The difference between Frequency and Frequency×2 Simulations is expected. Frequency×2 has far more days achieving modest P values than Frequency, but the higher P becomes, the closer the two get in terms of number of days. This indicates that there is another factor, besides TTL, that keeps each



(a) (b)

Figure 9: Aggregate flow by day through relays in simulator where average delivery delay is fixed by Bounding Solution on that day and plotted as (a) CCDF over all days (28 for Trained, 58 for others) (b) a point for each day and line indicating upper bound on throughput.

from ever achieving total capacity. The Trained Simulation, however, produces startling results. Although it is operating on data from 30 days prior, the aggregation of bus location information helps it to perform consistently better than the more short-sighted Frequency Simulation. Trained also outperforms Frequency×2 for large values of P. One possible reason for this behavior might be that Trained allows RAPID to distribute one-hop demand more equitably than it can for the other strategies. Figure 9(b) compares just Frequency and Trained Simulations to the throughput bound on each of the 28 days on which Trained was tested. Here we see Trained consistently outperforming Frequency in terms of throughput, often by a significant margin.

Our analysis has sought to optimize throughput and average delivery delay for the two-hop routing problem. In essence, this just the throughput and delay of flow through relays. Our reasoning was that RAPID is a good routing algorithm that produces reasonable one-hop demand used in our analysis. It's important to validate this assumption by looking at the end-to-end delivery rate and average delivery delay. Figures 10(a)–(b) show the packet delivery rate and average delivery delay, respectively, for each of our placement strategies. For contrast, we added the results from the None Simulation, which enforces the same TTL as the others (with the exception of Frequency×2), but uses no relays. Not surprisingly, None does not perform well, it has the lowest delivery rate and the second worst average delivery delay. Mirroring the results shown in Figure 9(a), Frequency×2 performs the best in terms of delivery rate except that it had no days achieving more than 85% of their packets. Because of its construction, Frequency×2 has far higher average delivery delay than all the other strategies. Trained consistently delivers more packets than Frequency, which was also predicted by Figure 9. Figure 10(b) reveals that there is no penalty in terms of average delivery delay for Trained either.

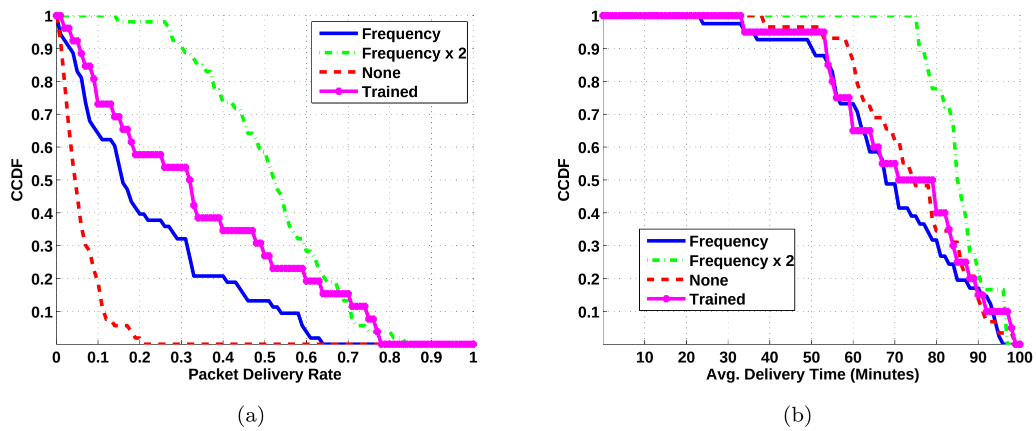


Figure 10: End-to-end performance in Simulations Frequency, Frequency×2, None, and Trained for (a) packet delivery rate and (b) average delivery delay.

6.5 Discussion

The Trained Placement Strategy emerges as the clear choice for the DieselNet deployment, but our analysis also suggests that other DTNs with similar node hubs may benefit from the same strategy. This development is particularly welcome because Trained Placement does not require prior knowledge of bus locations, nor is it difficult to compute. If time had permitted, we would have liked to further investigate network dynamics with changes in relay quantity. It is possible that different quantities would have revealed slightly different properties of each of the placement strategies. On the other hand, we believe that the strong performance of Frequency and Trained is fundamental to the DTN itself. Therefore, we would not expect vastly different results for different numbers of relays. Indeed, this same reasoning applies to changes in other network parameters, such as the number of unique packets.

7 Conclusion

Our work has focussed on the Relay Placement Problem. Mobile computing is becoming more pervasive but the most popular pieces of infrastructure, such as cellular networks and wired backbones, do not scale well and are costly to operate and deploy. DTNs, in contrast, arise from the mobile nodes themselves, so they will always offer a strong supplement to other technologies. Relay nodes have the potential to further augment DTNs by providing cheap additional infrastructure. Prior to our work, very little was known about the best relay placement strategy. The problem is unusually difficult: it does not even appear to be in NP since a witness placement can only be verified using the optimal routing strategy, which itself is not in P. Our major contributions are in devising a scheme to validate potential placement strategies and actually demonstrating the efficacy of several strategies.

In particular, we first decoupled the placement problem from the routing problem by assuming routing was conducted using the RAPID algorithm. This allowed us to show that choosing those locations that are most frequently visited is a highly effective strategy in DTNs with node hubs. For example, assuming RAPID routing and choosing the 50 most frequented locations over 30 days of traces yields a placement that, when tested on the remaining 28 days of traces, achieves at least 40% of the optimal throughput 40% of the time. The Trained Placement Strategy emerges as the best strategy for DTNs with node hubs similar to ours.

References

[1] Aruna Balasubramanian, Brian Neil Levine, and Arun Venkataramani. Dtn routing as a resource allocation problem. *ACM Special Interest Group on Data Communication*, 2007.

[2] Aruna Balasubramanian, Brian Neil Levine, and Arun Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM SIGCOMM*, August 2007.

[3] Aruna Balasubramanian, Ratul Mahajan, Arun Venkataramani, Brian Neil Levine, and John Zahorjan. Interactive wifi connectivity for moving vehicles. *ACM Special Interest Group on Data Communication*.

[4] Nilanjan Banerjee, Mark D. Corner, and Brian Neil Levine. An Energy-Efficient Architecture for DTN Throwboxes. In *Proc. IEEE Infocom*, May 2007.

[5] Nilanjan Banerjee, Mark D. Corner, Don Towsley, and Brian N. Levine. Relays, base stations, and meshes: Enhancing mobile networks with infrastructure. *International Conference on Mobile Computing and Networking*, 2008.

[6] Nilanjan Banerjee, Mark D. Corner, Don Towsley, and Brian Neil Levine. Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure. *Proceedings of ACM MobiCom*, 2008.

[7] Albert-laszlo Barabasi, Reka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. 2000.

[8] Bela Bollobas. Mathematical results on scale-free random graphs. 2003.

[9] Brendan Burns, Oliver Brock, and Brian Neil Levine. Mv routing and capacity building in disruption tolerant networks. *The IEEE Conference on Computer Communications*, 2005.

[10] Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. *Approximation, Randomization, and Combinatorial Optimization*, 2004.

[11] Ling-Jyh Chen, Chen-Hung Yu, Tony Sun, Yung-Chih Chen, and Hao-hua Chu. A hybrid routing approach for opportunistic networks. *ACM Special Interest Group on Data Communication*, 2006.

[12] James Davis, Andy Fagg, and Brian Neil Levine. Wearable Computers and Packet Transport Mechanisms in Highly Partitioned Ad hoc Networks. In *Proc. IEEE Intl. Symp on Wearable Computers (ISWC)*, pages 141–148, October 2001.

[13] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: Vehicular Content Delivery Using WiFi. In *Proceedings of ACM MobiCom*, San Francisco, CA, September 2008.

[14] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. *Special Interest Group on Data Communications*, 1999.

[15] Mouhamad Ibrahim, Ahmad Al Hanbali, and Philippe Nain. Delay and Resource Analysis in MANETs in Presence of Throwboxes. In *Proc. Performance 2007*, October 2007.

[16] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. *ACM Special Interest Group on Data Communication*, 2004.

[17] H. Jun, M. H. Ammar, M. D. Corner, and E. Zegura. Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization. In *Proc. ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, September 2006.

[18] Hyewon Jun, Wenrui Zhao, Mostafa Ammar, Ellen Zegura, and Chungki Lee. Trading latency for energy in densely deployed wireless ad hoc networks using message ferrying. *Elsevier Journal of Ad Hoc Networks*, 2006. To Appear.

[19] Jeremie Leguay, Timur Friedman, and Vania Conan. Dtn routing in a mobility pattern space. *ACM Special Interest Group on Data Communication*, 2005.

[20] Aruna Balasubramanianm Brian Neil Levine and Arun Venkataramani. Enabling Interactive Applications for Hybrid Networks. *The Annual International Conference on Mobile Computing and Networking*, 2008.

[21] Yong Liao, Kun Tan, Zhensheng Zhang, and Lixin Gao. Estimation based erasure-coding routing in delay tolerant networks. *The International Wireless Communications and Mobile Computing Conference*, 2006.

[22] Christian Locher, Bjorn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. Data aggregation and roadside unit placement for a vanet traffic information system. *ACM International Workshop on VehiculAr Inter-NETworking*, 2008.

[23] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 2005.

[24] Yong Wang, Sushant Jain, Margaret Martonosi, and Kevin Fall. Erasure-coding based routing for opportunistic networks. *ACM Special Interest Group on Data Communication*, 2005.

[25] Jorg Widmer and Jean-Yves Le Boudec. Network coding for efficient communication in extreme networks. *ACM SIGCOMM workshop on Delay-tolerant networking*, 2005.

[26] Wenrui Zhao, Yang Chen, Mostafa Ammar, Mark D. Corner, Brian Neil Levine, and Ellen Zegura. Capacity Enhancement using Throwboxes in DTNs. *Proceedings of IEEE International Conf on Mobile Ad hoc and Sensor Systems (MASS)*, 2006.