

# Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access

Zizhan Zheng<sup>†</sup>      Prasun Sinha<sup>†</sup>      Santosh Kumar<sup>§</sup>  
† The Ohio State University      § University of Memphis  
{zhengz, prasun}@cse.ohio-state.edu      santosh.kumar@memphis.edu

**Abstract**—Vehicular Internet access via open WLAN access points (APs) has been demonstrated to be a feasible solution to provide opportunistic data service to moving vehicles. Using an in situ deployment, however, such a solution does not provide worst-case performance guarantees due to unpredictable intermittent connectivity. On the other hand, a solution that tries to cover every point in an entire road network with APs (full coverage) is not very practical due to the prohibitive deployment and operational cost. In this paper, we introduce a new notion of intermittent coverage for mobile users, called  $\alpha$ -coverage, which provides worst-case guarantees on the interconnection gap while using significantly fewer APs than needed for full coverage. We propose efficient algorithms to verify whether a given deployment provides  $\alpha$ -coverage and approximation algorithms for determining a deployment of APs that will provide  $\alpha$ -coverage. We compare  $\alpha$ -coverage with opportunistic access of open WLAN APs (modeled as a random deployment) via simulations over a real-world road network and show that using the same number of APs as random deployment,  $\alpha$ -coverage bounds the interconnection gap to a much smaller distance than that in a random deployment.

## I. INTRODUCTION

The growing popularity of media enabled handhelds such as vPods and iPods, and services such as vCast from Verizon, indicate that there is an increasing demand for wireless data services for mobile users. Other applications of such services include in-vehicle entertainment, remote monitoring and tracking of shipments in trucks, and communication within a mobile workforce. Although technologies such as 3G data services and upcoming WiMAX can provide coverage over large areas, they fail to provide high data rates such as in the case of Wireless LANs (WLANs). In the shipment monitoring service used by Walmart [1], satellite connectivity from the trailers is used to update information on the status of the shipped items. However, satellite connectivity and related communication equipments are expensive.

WLANs have the potential to provide high data rate coverage to support such applications. But, the prohibitive cost of deployment and management of a large number of WLAN access points (APs) for providing full coverage, calls for smarter, more scalable solutions that can leverage intermittent connectivity provided by WLAN hotspots. Evaluation of wireless data access by mobile users using in situ Wi-Fi networks [4]–[6], [10], and in various controlled environments [4], [10], [11] have confirmed the feasibility of WiFi-based vehicular Internet access. A detailed review of related work in this area is given in [13].

Solutions based on intermittent connectivity of WLANs can provide opportunistic services without any worst case

service guarantees. In busy urban areas mobile users can hope to connect with APs more frequently than in the suburban areas, but the exact frequency may drastically vary from region to region. Thus, a user lying in a coverage hole is unable to estimate the time to the next connection. In order to address such coverage uncertainties, we introduce a new notion of intermittent coverage for mobile users, called  $\alpha$ -coverage, and study how such coverage can be attained by systematic deployment of additional APs to create an economically scalable infrastructure.

Informally, a deployment of APs provides  $\alpha$ -coverage to a road network, if any simple path of length  $\alpha$  on the road network meets with at least one AP. If the expected service from an AP is known, then the cumulative service received by a mobile user over a certain path can be estimated. For a given road network, we ask the following two questions: 1) does a given deployment provide  $\alpha$ -coverage? 2) if not, how to deploy a minimum number of additional APs to ensure  $\alpha$ -coverage?

This paper makes the following contributions.

- We present the first notion of coverage for mobile users with intermittent connectivity called  $\alpha$ -coverage. We define three coverage metrics that can be used in various scenarios.
- We present efficient algorithms for coverage verification, and factor  $O(\log n)$  approximation algorithms for determining both a new and an incremental deployment.
- We evaluate the performance of random deployment and our solutions for real road networks [2] and show that our proposed algorithms perform significantly better than random deployment.

The framework and solutions presented in this paper are immediately usable by various service provider companies for enabling WLAN based services for mobile users. This is because we provide a solution for planning incremental deployment, so service providers can install few APs for large values of  $\alpha$  to begin with, and over time add new APs to gradually bring down the value of  $\alpha$ . In addition to providing a low cost solution for supporting various existing applications mentioned earlier, making worst-case service guarantee may enable new applications based on intermittent connectivity.

## II. THE PROBLEM FORMULATION

We model a road network  $\mathcal{R}$  as a connected *undirected* geometric graph  $G_{\mathcal{R}}$ , where vertices represent the points where the road centerline segments and the road intersections meet, and edges represent the road centerline segments connecting

the road intersections. For a curved road segment, we introduce artificial road intersections, so that each edge represents a straight line segment. Let  $V_{G_{\mathcal{R}}}$  and  $E_{G_{\mathcal{R}}}$  denote the vertex set and edge set, respectively. Each edge  $e$  has a length, denoted as  $|e|$ , which is the length of the corresponding road segment. This model has been used by some publicly available road network databases, such as [2]. Although we are assuming an undirected graph model, most of our results can be extended to directed graphs [13].

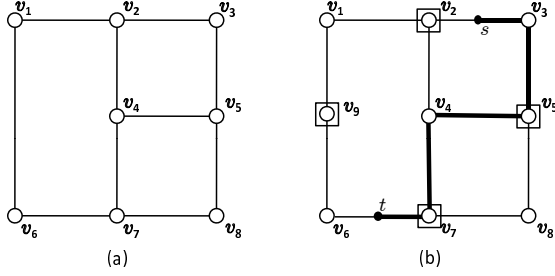


Fig. 1: (a) A graph  $G_{\mathcal{R}}$  representing a small road network. Every edge except  $(v_1, v_6)$  has unit length and  $(v_1, v_6)$  has length 2. (b) A path  $f_{st}$  of length 4 is highlighted, where  $s$  is the middle point of  $(v_2, v_3)$  and  $t$  is the middle point of  $(v_6, v_7)$ .  $\text{dist}(s, t) = 3$ . An optimal deployment that provides 2-coverage is achieved by placing APs at vertices  $v_2, v_5, v_7$ , and point  $v_9$ , the middle point of  $(v_1, v_6)$ .

For a point  $p$  on an edge  $(u, v) \in E_{G_{\mathcal{R}}}$ , if  $p$  is not a vertex, we can make it a vertex by adding  $p$  to  $V_{G_{\mathcal{R}}}$  and by subdividing edge  $(u, v)$  into two edges  $(p, u)$  and  $(p, v)$  in  $E_{G_{\mathcal{R}}}$ , with  $|(p, u)|$  and  $|(p, v)|$  equal to the length of the corresponding road segments starting at  $p$ . The resulting graph is denoted as  $G_{\mathcal{R}} \vee \{p\}$ . For instance, by inserting the middle point of edge  $(v_1, v_6)$  to the graph in Figure 1(a), we can get a graph where all the edges have the same length as shown in Figure 1(b).  $G_{\mathcal{R}} \vee \{p\} = G_{\mathcal{R}}$  if  $p$  is a vertex in  $G_{\mathcal{R}}$ .

Consider a set of APs deployed at roadside. To achieve the maximum possible coverage, it is reasonable to assume that the APs are deployed as close to the road centerline as possible. Therefore, each AP is represented by the point in  $G_{\mathcal{R}}$  closest to it. The feasibility of such modeling is further discussed in [13]. Using the operation just defined, we can make these points as vertices of the graph when needed. The trajectory of a moving vehicle is modeled as a set of consecutive *general* paths on the graph defined as follows.

**Definition II.1. A general path on a graph:** A general path  $f_{ab}$  in a graph  $G_{\mathcal{R}}$  between  $a$  and  $b$ , both of which are points on some edges of  $G_{\mathcal{R}}$ , is a (simple) path in  $G \vee \{a, b\}$ . The length of  $f_{ab}$ , denoted as  $|f_{ab}|$ , equals to the sum of the lengths of the edges composing the path in  $G_{\mathcal{R}} \vee \{a, b\}$ .

A general path will be simply called a path when there is no ambiguity. For instance, Figure 1(b) highlights a path of length 4. We are now ready to formally define  $\alpha$ -coverage.

**Definition II.2.  $\alpha$ -coverage:** A deployment of APs provides  $\alpha$ -coverage to  $\mathcal{R}$ , if every path  $f_{ab}$  in  $G_{\mathcal{R}}$  with  $|f_{ab}| \geq \alpha$  is covered by at least one AP.

For instance, Figure 1(b) shows a deployment that provides  $\alpha$ -coverage for  $\alpha = 2$ . Although  $\alpha$ -coverage closely models

our intuition, it is impossible to determine in polynomial time whether a deployment provides  $\alpha$ -coverage, unless  $P=NP$ . To see this, suppose no APs are deployed in a given graph where each edge has unit length and  $\alpha = |V| - 1$ , then verifying whether this graph is  $\alpha$ -covered is equivalent to determining whether there is a Hamiltonian path in the graph, and the latter is NP-complete even for planar graphs [9]. Since even verifying whether a graph is  $\alpha$ -covered is NP-complete, we propose two new metrics to approximate  $\alpha$ -coverage. First, we define the following terms:

**Definition II.3. Distance on a graph:** For any two points  $a$  and  $b$  in graph  $G_{\mathcal{R}}$ , the distance between them, denoted as  $\text{dist}(a, b)$ , is the length of the shortest path between  $a$  and  $b$  in  $G_{\mathcal{R}} \vee \{a, b\}$ .

**Definition II.4.  $\alpha_N$ -coverage:** A deployment of APs provides *Network Coverage* of distance  $\alpha$  ( $\alpha_N$ -coverage for short) to  $\mathcal{R}$ , if every path  $f_{ab}$  in  $G_{\mathcal{R}}$  with  $\text{dist}(a, b) \geq \alpha$  is covered by at least one AP.

Note that if a deployment provides  $\alpha$ -coverage, it also provides  $\alpha_N$ -coverage. The converse is not true. For instance, the deployment in Figure 1(b) also provides  $\alpha_N$ -coverage when  $\alpha = 2$ . Now suppose  $\alpha = 5$ , then since the diameter of the graph is 4, the distance between any pair of points in the graph is at most 4,  $\alpha_N$ -coverage is satisfied without deploying any APs. However, an empty deployment does not provide  $\alpha$ -coverage since the longest path in the graph has length 8.

Given  $G_{\mathcal{R}}$  – the graph model of a road network,  $A_0$  – a set of points in  $G_{\mathcal{R}}$  that models the APs previously deployed, we ask the following two questions – 1) determine if the deployment provides a desired coverage, and if not 2) find a minimum set of *points*  $A$  in  $G_{\mathcal{R}}$  so that when new APs are deployed around these points,  $A_0 \cup A$  provides the desired coverage. Notice that the second problem addresses both the new deployment and incremental deployment. We refer to this optimization problem as  $\alpha_N$ -Cover. We show in Section III that it can be verified in polynomial time whether a deployment provides  $\alpha_N$ -coverage. However, the decision version of  $\alpha_N$ -Cover is NP-complete as proved in [13]. We provide a  $O(\log n)$  factor approximation algorithm in Section IV, where  $n$  is the number of vertices in the graph.

Notice that,  $\alpha_N$ -coverage requires that all paths between two points that are  $\alpha$  distance away are covered, and the number of such paths could be exponential. In reality, however, there are a small subset of paths most frequently traveled between any two places, which can be learned from historical traffic data [8]. For instance, people usually follow close to shortest path from their source to the destination. Our third metric,  $\alpha_P$ -coverage, captures this observation. Although determining a deployment to achieve  $\alpha_P$ -coverage is still NP-Complete [13], we are able to find a more efficient approximation algorithm for it (see Section IV). Let  $F_{ab}$  denote the set of paths between  $a$  and  $b$  most frequently traveled, where  $|F_{ab}|$  is bounded by a small constant.

**Definition II.5.  $\alpha_P$ -coverage:** A deployment of APs provides *Path Coverage* of distance  $\alpha$  ( $\alpha_P$ -coverage for short) to  $\mathcal{R}$ ,

if every path  $f_{ab}$  in  $F_{ab}$  with  $\text{dist}(a, b) \geq \alpha$  is covered by at least one AP.

To simplify the presentation, we assume  $F_{ab}$  is the set of shortest paths. The solution can be easily extended to arbitrary  $F_{ab}$  [13]. Also note that a deployment that provides  $\alpha_N$ -coverage also provides  $\alpha_P$ -coverage, but not vice versa.

### III. $\alpha$ -COVERAGE VERIFICATION

In this section, the following problem is considered: given a graph  $G_{\mathcal{R}}$ , a set of points  $A_0$  in  $G_{\mathcal{R}}$ , and  $\alpha$ , does  $A_0$  provide  $\alpha_N$ -coverage or  $\alpha_P$ -coverage to  $\mathcal{R}$ ? First, a new graph  $G(V, E) = G_{\mathcal{R}} \vee A_0$  is obtained. That is, we make each point in  $A_0$  a vertex. Each vertex  $v$  of  $G$  is then assigned a weight, denoted as  $w(v)$ , which equals to 1 if  $v$  represents an AP, and is 0 otherwise. We then give polynomial time algorithms for verifying  $\alpha_N$  and  $\alpha_P$ -coverage. To simplify the discussion, we make the following assumption in the rest of the paper: For any pair of vertices  $(u, v)$  of  $G$ ,  $\text{dist}(u, v) \neq \alpha$ .

If a given  $\alpha$  does not satisfy this assumption, we can choose a small  $\epsilon$  such that  $\alpha + \epsilon$  satisfies this condition. By making  $\epsilon$  small enough,  $(\alpha + \epsilon)$ -coverage can be viewed as equivalent to  $\alpha$ -coverage in any real settings.

#### A. The Verification of $\alpha_N$ -coverage

**Definition III.1. Coverage weight:** The coverage weight of a path  $f$ , denoted as  $c(f)$ , equals to the sum of weights of the vertices on  $f$ .

**Definition III.2. Coverage distance:** The coverage distance of a pair of points  $(a, b)$  in  $G$ , denoted as  $c(a, b)$ , equals to the minimum coverage weight of all paths between  $a$  and  $b$ .

We use the term  $\alpha$ -pair to refer to a pair of points that are a distance of  $\alpha$  apart. We observe that a deployment provides  $\alpha_N$ -coverage iff the coverage distance of each  $\alpha$ -pair is at least 1. Notice that, according to the assumption made above, in each  $\alpha$ -pair, at least one point is not a vertex of  $G$ . Further, if  $(a, b)$  is an  $\alpha$ -pair and  $a$  is a vertex, there always exists another  $\alpha$ -pair  $(c, d)$  such that neither  $c$  or  $d$  is a vertex and  $c(c, d) \leq c(a, b)$ . Therefore, it suffices to only consider  $\alpha$ -pairs consisting of non-vertex points. Although there are infinite such  $\alpha$ -pairs, they can be divided into equivalent classes as follows.

**Definition III.3. Equivalent  $\alpha_N$ -pairs:** Two  $\alpha$ -pairs  $(a, b)$  and  $(c, d)$  are  $\alpha_N$ -equivalent if  $a$  and  $c$  are on the same edge, and  $b$  and  $d$  are on the same edge.

By this definition, all the  $\alpha$ -pairs in the same  $\alpha_N$ -equivalent class have the same coverage distance. Furthermore, the number of equivalent classes is bounded by  $O(|E|^2)$  since for any pair of edges, there is at most one equivalent class. Therefore, once all the equivalent classes are identified,  $\alpha_N$ -coverage can be determined by checking the coverage distance of  $\alpha$ -pairs in each class one by one, which can be done as follows. First, we note that the coverage distance of every pair of vertices in  $G$  can be computed by extending the Floyd's all-pairs shortest paths algorithm [13]. Suppose  $a$  is on edge  $(u_1, u_2)$  and  $b$  is on edge  $(v_1, v_2)$  and  $a$  and  $b$  is an  $\alpha$ -pair, then  $c(a, b) = \min(c(u_1, v_1), c(u_1, v_2), c(u_2, v_1), c(u_2, v_2))$ .

#### B. The Verification of $\alpha_P$ -coverage

**Definition III.4. The core of a general path:** The core of a general path is its longest subpath ending at vertices of  $G$ .

For instance, in Figure 1, the core of  $f_{st}$  is the path  $(v_3, v_5, v_4, v_7)$ .

**Definition III.5. The core of a set of paths:** Let  $F$  be a set of paths. The core of  $F$  is a set of paths, where each of them is the core of a path in  $F$ .

**Definition III.6. Equivalent  $\alpha_P$ -pairs:** Two pairs  $(a, b)$  and  $(c, d)$  are  $\alpha_P$ -equivalent if 1)  $\text{dist}(a, b) \geq \alpha$  and  $\text{dist}(c, d) \geq \alpha$ ; 2)  $a$  and  $c$  are on the same edge, and  $b$  and  $d$  are on the same edge; 3) the set of shortest paths between  $a$  and  $b$  and that between  $c$  and  $d$  have the same core.

A polynomial time algorithm for verifying  $\alpha_P$ -coverage can then be derived by noticing that 1) if  $(a, b)$  and  $(c, d)$  are equivalent  $\alpha_P$ -pairs, then all the shortest paths between  $a$  and  $b$  are covered iff all the shortest paths between  $c$  and  $d$  are covered; 2) each equivalent class can be verified in polynomial time since there are only constant number of paths in the core shared by all the pairs in the same class; 3) the number of equivalent classes is bounded by  $O(|E|^2)$ .

### IV. APPROXIMATION ALGORITHMS FOR $\alpha$ -COVER

In this section, we present approximation algorithms for  $\alpha_N$ -Cover and  $\alpha_P$ -Cover. Formally, given graph  $G_{\mathcal{R}}$  and a set of point  $A_0$  in  $G_{\mathcal{R}}$ , and  $\alpha$ , find a set of points  $A$  such that  $A_0 \cup A$  provides  $\alpha_N$  or  $\alpha_P$ -coverage to  $\mathcal{R}$ . We would like to minimize  $|A|$ . Let  $G = G_{\mathcal{R}} \vee A_0$ . For any edge  $e$  of  $G$ , if  $|e| > \alpha$ ,  $e$  is chopped into  $\lceil |e|/\alpha \rceil$  pieces of equal length. In the following discussion, we will assume that the length of any edge of  $G$  is no more than  $\alpha$ .

We present two polynomial time algorithms in this section. The first algorithm reduces  $\alpha_N$ -coverage to the vertex multicut problem [7] and the second one reduces  $\alpha_P$ -coverage to the set cover problem [12]. Both algorithms have an  $O(\log n)$  approximation factor, where  $n$  is the number of vertices in  $G_{\mathcal{R}}$ . It should be noted that for a given road network, the first algorithm is much more time consuming than the second one, which is expected since  $\alpha_N$ -coverage provides higher coverage quality than  $\alpha_P$ -coverage.

Given  $\alpha$ , let  $\text{OPT}$  denote the minimum  $|A|$  in any deployment that provides  $\alpha_N$ -coverage where APs can be deployed at anywhere in  $G$ , and  $\text{OPT}'$  denote the minimum  $|A|$  for providing  $\alpha_N$ -coverage when APs can only be deployed at the vertices of  $G$ . Both of our algorithms use only the vertices of  $G$  to construct  $A$ , which avoids an infinite search space so that approximation solutions can be found. The following lemma states that such a deployment decision doubles the number of APs used in the worst case. See [13] for a proof.

**Lemma IV.1.**  $\text{OPT} \leq \text{OPT}' \leq 2 \times \text{OPT}$ .

The same result also holds for  $\alpha_P$ -coverage. Notice that, an important advantage of restricting APs to vertices is that it increases the chance of data access since vehicles may stop or slow down around road intersections.

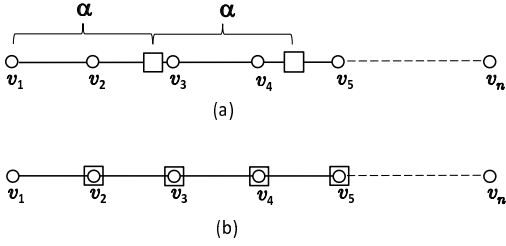


Fig. 2: A graph with a single path  $(v_1, v_2, \dots, v_n)$  where the squares represent APs. Each edge has unit length, and  $\alpha \in (1, 2)$ . (a) A minimum cover that provides  $\alpha_N$ -coverage. (b) A suboptimal solution that uses only vertices to provide  $\alpha_N$ -coverage.

Figure 2 gives an example where using only vertices gives a suboptimal solution. In the figure,  $|A_0| = 0$ ,  $G$  is a single path with  $n$  vertices, where each edge has unit length, and  $\alpha \in (1, 2)$ . To achieve  $\alpha_N$ -coverage, an optimal solution is a set of points uniformly spaced with  $\alpha$  distance along the path. On the other hand, if only the vertices are allowed to be used, then all the vertices except the two ends of the path have to be used to ensure  $\alpha_N$ -coverage. Therefore,  $\text{OPT} = \lceil (n-1)/\alpha \rceil - 1$ ,  $\text{OPT}' = n - 2$ , and  $\lim_{n \rightarrow \infty} \text{OPT}'/\text{OPT} = \alpha$ . In particular, when  $n = 4$ , and  $\alpha \in [1.5, 2)$ , a minimum cover contains only one point at the center of the path, while two vertices are needed to ensure the coverage.

#### A. $\alpha_N$ -Cover via Vertex MultiCut

Assuming only the vertices of  $G$  are used to construct  $A$ , the  $\alpha_N$ -Cover optimization problem can be reduced to the minimum vertex multicut problem [7] defined as follows. Given a connected undirected graph  $G(V, E)$  with positive costs on its vertices, let  $\{(s_1, t_1), \dots, (s_k, t_k)\}$  be a set of pairs of vertices, named as terminals, where each pair is distinct, but vertices in different pairs are not required to be distinct. A *vertex multicut* is a set of non-terminal vertices whose removal separates each pair. The problem is to find a vertex multicut of minimum cost. We assume that all vertices have the same cost in this section. The main steps of our algorithm are summarized in Algorithm IV.1.

---

#### Algorithm IV.1 $\alpha_N$ -Cover via Vertex Multicut

---

Input:  $G$ , the graph model of a road network  $\mathcal{R}$ ,  $A_0$ , the set of vertices of  $G$  that represents APs previously deployed, and  $\alpha$

- 1: **if**  $A_0$  provides  $\alpha_N$ -coverage to  $G$  **then** return;
  - 2:  $T \leftarrow \emptyset$ ;
  - 3: **for** each pair of edges  $(e_1, e_2)$  of  $G$  **do**
  - 4:   **if** there are points  $a$  on  $e_1$  and  $b$  on  $e_2$  such that  $\text{dist}(a, b) = \alpha$  and  $c(a, b) = 0$  **then**
  - 5:      $T \leftarrow T \cup (m_1, m_2)$ ;      $\triangleright m_i$ : the midpoint of  $e_i$
  - 6: **return**  $\text{GVY}(G, T, A_0)$ ;      $\triangleright$  Apply the GVY algorithm [12] to find a vertex multicut with  $A_0$  as a subset that separates each pair of midpoints in  $T$ .
- 

For each pair of edges  $(e_1, e_2)$ , the algorithm determines whether there are points  $a$  in  $e_1$  and  $b$  in  $e_2$  such that  $\text{dist}(a, b) = \alpha$ , which can be done in constant time [13]. If there is such a pair and the coverage distance of the pair is 0, the middle points of  $e_1$  and  $e_2$  are inserted to  $T$ . The algorithm

then resorts to the well known GVY algorithm [12] to find a minimum vertex multicut with  $A_0$  as a subset that separates each pair of midpoints inserted.

The algorithm stated in [12] actually solves the minimum *edge* multicut problem defined similarly and achieves an  $O(\log k)$  approximation factor where  $k$  is the number of pairs. However, it can be extended to the vertex version while preserving the approximation factor [7]. In addition, the standard GVY algorithm does not consider the case where a set of vertices are forbidden to be chosen as cut nodes. This can be easily solved as shown in [13]. Since the GVY algorithm has a factor  $O(\log k)$  where  $k = O(|E|^2)$ , considering Lemma IV.1, Algorithm IV.1 has an  $O(\log n)$  factor.

#### B. $\alpha_P$ -Cover via Set Cover

Assuming only the vertices of  $G$  are used to construct  $A$ , the  $\alpha_P$ -coverage optimization problem can be reduced to the set covering optimization problem [12] as follows:

---

#### Algorithm IV.2 $\alpha_P$ -Cover via Set Cover

---

Input:  $G$ ,  $A_0$ , and  $\alpha$  (same as in Algorithm IV.1)

- 1: **if**  $A_0$  provides  $\alpha_P$ -coverage to  $G$  **then** return;
  - 2:  $\mathcal{U} \leftarrow$  the union of cores with respect to all  $\alpha_P$ -pairs except those have been covered by  $A_0$ ;
  - 3:  $\mathcal{S} \leftarrow \{S_v : v \in V \setminus A_0\}$  where  $S_v \subset \mathcal{U}$  is the set of paths covered by vertex  $v \notin A_0$ ;
  - 4: Output a subcollection of  $\mathcal{S}$  of minimum size that covers all the elements of  $\mathcal{U}$ ;
- 

The above algorithm also applies to  $\alpha_N$ -Cover, where set  $\mathcal{U}$  has exponential size. For  $\alpha_P$ -Cover,  $|\mathcal{U}| = O(|E|^2)$ . Since there is a factor  $O(\log |\mathcal{U}|)$  algorithm for set cover [12], this algorithm has an  $O(\log n)$  factor.

## V. EVALUATION

#### A. Simulation settings

Our simulations are based on the road network of the Franklin county in State of Ohio. We obtain the road network data from the 2007 Tiger/Line shapefiles [2], where each road segment is a polyline that contains two intersections and zero or more interior shape points. We ignore the shape points and connect the two ends of a road segment by a straight line to reduce the size of the graph. We then map the road network to a 2D plane by Mercator projection. The simulations are carried over a  $4000\text{m} \times 4000\text{m}$  region around the center of the Franklin county including about 1000 intersections and 1300 road segments, and the graph diameter is about 7300m.

A restricted random waypoint mobility model [3] is used to generate ns-2 movement files, with an additional constraint that the source and destination (both are road intersections) of each move must have a distance at least  $\alpha$  since we are interested in providing performance guarantee to these moves. The speed of a mobile node is set to the speed limit of the segment where the node is moving on with a leeway of 5 miles/hour. The average pause time at each destination is 5 sec. 10 movement scenarios are generated, each of them consists of 5 mobile nodes moving independently for 1 hour.

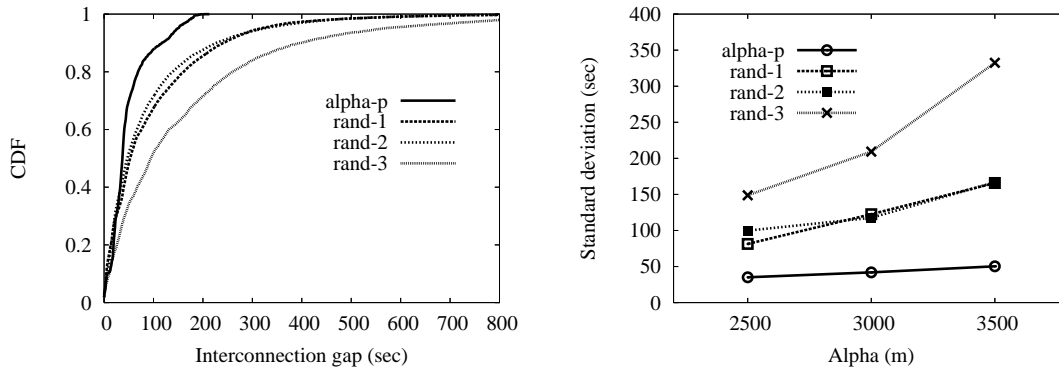


Fig. 3: Left: the cumulative distribution of the interconnection time when  $\alpha = 3000\text{m}$ . Right: the standard deviation of the interconnection time for various  $\alpha$ . The speed limit is fixed to 55 miles/hour. The maximum gap for  $\alpha_P$ -coverage is about 214 sec, bounded by the time spent on two adjacent moves (see the text for an explanation.) The maximum gap for a random deployment can be larger than 2000 sec (not shown).

We compare  $\alpha_P$ -coverage with three random deployment schemes: rand-1) each AP is deployed on a randomly selected vertex of  $G$ ; rand-2) each AP is deployed on a randomly selected point of a randomly selected edge of  $G$ ; rand-3) the region considered is first divided into  $50\text{m} \times 50\text{m}$  squares; each AP is then deployed at the center of a randomly selected square. At most one AP can be deployed at a vertex or within a square. For a given  $\alpha$ , an  $\alpha_P$ -coverage is first computed. 10 scenarios using the same number of APs as the  $\alpha_P$ -coverage does are then generated for each of the three random schemes. The transmission range of each AP is 100m.

### B. Simulation results

Figure 3(left) shows the cdf of the interconnection time of  $\alpha_P$ -coverage and the three random schemes, where  $\alpha$  equals to 3000m and the speed limit is 55 miles/hour. 21 APs are used in all the deployments. The data is accumulated over all the five mobile nodes and 10 movement patterns. The maximum gap for  $\alpha_P$ -coverage is about 214 sec, while that for a random deployment can be as large as 1866 sec for rand-1, 1446 sec for rand-2, and 2210 sec for rand-3. The fact that rand-3 performs worst verifies that APs should be deployed as close to road centerlines as possible to achieve the best coverage. Notice that, due to the movement pattern we use, the maximum gap for  $\alpha_P$ -coverage is larger than  $\alpha$  divided by the speed limit ( $\approx 122$  sec), since an interconnection gap may span two adjacent moves. Figure 3(right) shows the standard deviation of the interconnection time under various  $\alpha$  and the constant speed limit, where the number of APs used for  $\alpha = 2500\text{m}$ , 3000m, 3500m is 28, 21, and 15, respectively. We can clearly see that the standard deviation for  $\alpha_P$ -coverage is much smaller than that for a random deployment. The results for variable speed limit are similar to the constant speed limit case and are omitted.

## VI. CONCLUSION

In this paper, we propose  $\alpha$ -coverage, a new notion of intermittent coverage for mobile users that guarantees the interconnection gap of vehicular Internet access. We provide algorithms to verify whether a given deployment provides  $\alpha$ -coverage and if not, to find the optimal places to deploy new

APs so that  $\alpha$ -coverage can be ensured. Both the networking protocols and applications may explore such guarantees to optimize their behaviors. Furthermore,  $\alpha$ -coverage is the first step towards a scalable deployment that guarantees the worst case data service that a mobile user can expect.

## VII. ACKNOWLEDGEMENT

This material is based upon work supported by NSF grants CNS-0546630 (CAREER Award), CNS-0403342, CNS-0721983, CCF-0728928, NIH Grant U01DA023812 from National Institute for Drug Abuse (NIDA), and Fedex Institute of Technology (FIT) at the University of Memphis.

## REFERENCES

- [1] VeriWise Asset Intelligence. <http://www.ge.com/equipmentservices/assetintelligence/>.
- [2] U.S. Census Bureau - TIGER/Line <http://www.census.gov/geo/www/tiger/>.
- [3] The Random Trip Mobility Model. <http://lrcwww.epfl.ch/RandomTrip/>.
- [4] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan. Interactive WiFi Connectivity For Moving Vehicles. In *Proc. of ACM SIGCOMM*, Sept. 2008.
- [5] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *Proc. of ACM MOBICOM*, Sept. 2006.
- [6] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: A WiFi-Based Vehicular Content Delivery Network. In *Proc. of ACM MOBICOM*, Sept. 2008.
- [7] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *Journal of Algorithms*, 50(1):49–61, 2004.
- [8] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag. Adaptive Fastest Path Computation on A Road Network: A Traffic Mining Approach. In *Proc. of VLDB*, Sept. 2007.
- [9] M.R. Garey, D.S. Johnson, and R.E. Tarjan. The Planar Hamiltonian Circuit Problem is NP-Complete. *SIAM J. Comput.*, 5(4), 1976.
- [10] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-giel, and S. R. Das. MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access. In *Proc. of MOBISYS*, 2007.
- [11] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for "Automobile" Users. In *Proc. of INFOCOM*, Mar. 2004.
- [12] V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.
- [13] Z. Zheng, P. Sinha, and S. Kumar. Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access. Technical Report OSU-CISRC-1/09-TR03, <ftp://ftp.cse.ohio-state.edu/pub/tech-report/2009/TR03.pdf>, 2009.