Alpha Coverage: Bounding the Interconnection Gap for Vehicular Internet Access

Zizhan Zheng† Prasun Sinha† Santosh Kumar§
† 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 α-*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 α-coverage and approximation algorithms for determining a deployment of APs that will provide α-coverage. Our algorithm can also be used to supplement open WLAN APs in a region with appropriate number of additional APs that will provide worst-case guarantees on interconnection gap. We compare α-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, α-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 [1] 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 [2], 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 [6], [10], [11], [20], and in various controlled environments [6], [14], [20]–[22] have confirmed the feasibility of WiFi-based vehicular Internet accessfor non-interactive applications. The possibility and challenges to support certain interactive applications, such as Web browsing, have also been studied [6], [7].

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 α-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 α-coverage to a road network[1], if any simple path of length α 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 three questions: 1) does a given deployment provide α-coverage? 2) how to deploy a minimum number of APs to ensure α-coverage (in a new deployment)? 3) given a deployment that does not provide α-coverage (such as a random deployment, i.e., open WLAN APs), how to deploy a minimum number of additional APs to ensure α-coverage (incremental deployment)? The third problem is a generalization of the second, and is usually more difficult to solve.

This paper makes the following contributions.

- We present the first notion of coverage for mobile users with intermittent connectivity called α-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 [3] and show that our proposed algorithms perform significantly better than random deployment.

[1] See Section II for a precise definition

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 α to begin with, and over time add new APs to gradually bring down the value of α. 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.

The organization of the rest of the paper is as follows. Section II introduces the model. Section III presents the coverage verification algorithms. Section IV discusses the hardness of the optimization problem. Section V presents two approximation algorithms for incremental deployment. Section VI discusses how certain practical issues can be considered when applying our results to a real deployment, and future extensions of our work. The performance of the algorithms are presented in Section VII. Section VIII contrasts our work with related work. Finally, Section IX concludes the paper.

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 [3]. Although we are assuming an undirected graph model, most of our results can be extended to directed graphsas discussed in Section VI .

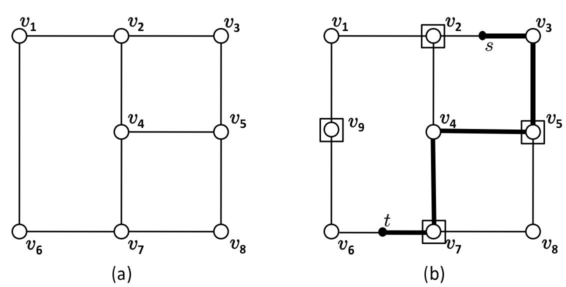

Fig. 1. (a) A graph $G_{\mathcal{R}}$ representing a small road network. Every edge except (v_1, v_6) has unit length and (v_1, v_6) has length 2. (b) A path f_{st} of length 4 is highlighted, where s is the middle point of (v_2, v_3) and t is the middle point of (v_6, v_7). $\text{dist}(s,t) = 3$. An 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 inSection VI. 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 α-coverage.

Definition II.2. α-coverage: *A deployment of APs provides α-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 α-coverage for $\alpha = 2$. Although α-coverage closely models our intuition, it is impossible to determine in polynomial time whether a deployment provides α-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 α-covered is equivalent to determining whether there is a Hamiltonian path in the graph, and the latter is NP-complete even for planar graphs [19]. Since even verifying whether a graph is α-covered is NP-complete, we propose two new metrics to approximate α-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 dist(a, b), is the length of the shortest path between a and b in $G_{\mathcal{R}} \vee \{a, b\}$.*

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

Note that if a deployment provides α-coverage, it also provides α_N-coverage. The converse is not true. For instance, the deployment in Figure 1(b) also provides α_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, α_N-coverage is satisfied without deploying any APs. However, an empty deployment does not provide α-

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 α_N**-Cover**. We show in Section III that it can be verified in polynomial time whether a deployment provides α_N-coverage. However, the decision version of α_N-Cover is NP-complete as proved in Section IV. We provide a $O(\log n)$ factor approximation algorithm in Section V, where n is the number of vertices in the graph.

Notice that, α_N-coverage requires that all paths between two points that are α 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 [15]. For instance, people usually follow close to shortest path from their source to the destination. Our third metric, α_P-coverage, captures this observation. Although determining a deployment to achieve α_P-coverage is still NP-Complete, we are able to find a more efficient approximation algorithm for it (see Section V). 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. α_P**-coverage:** *A deployment of APs provides **Path Coverage** of distance α (α_P-coverage for short) to $\mathcal{R}$, if every path f_{ab} in F_{ab} with 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} as discussed in Section VI. Also note that a deployment that provides α_N-coverage also provides α_P-coverage, but not vice versa.

III. α-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 α, does A_0 provide α_N-coverage or α_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 α_N and α_P-coverage. To simplify the discussion, we make the following assumption in the rest of the paper:

Assumption III.1. *For any pair of vertices (u, v) of G, dist$(u, v) \neq \alpha$.*

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

A. The Verification of α_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 *α-pair* to refer to a pair of points that are a distance of α apart. We observe that *a deployment provides α_N-coverage iff the coverage distance of each α-pair is at least 1*. Notice that, according to Assumption III.1, in each α-pair, at least one point is not a vertex of G. Further, if (a, b) is an α-pair and a is a vertex, there always exists another α-pair (c, d) such that neither c or d is a vertex and $\mathrm{c}(c, d) \leq \mathrm{c}(a, b)$. Therefore, it suffices to only consider α-pairs consisting of non-vertex points. Although there are infinite such α-pairs, they can be divided into equivalent classes as follows.

Definition III.3. Equivalent α_N-pairs: *Two α-pairs (a, b) and (c, d) are α_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 α-pairs in the same α_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, α_N-coverage can be determined by checking the coverage distance of α-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, see Algorithm III.1. Suppose a is on edge (u_1, u_2) and b is on edge (v_1, v_2) and a and b is an α-pair, then $\mathrm{c}(a, b) = \min(\mathrm{c}(u_1, v_1), \mathrm{c}(u_1, v_2), \mathrm{c}(u_2, v_1), \mathrm{c}(u_2, v_2))$.

Algorithm III.1 All-pair-vertices coverage weights

procedure VCWeight(G) ▷ G: a graph with n weighted vertices.
  for i = 1 to n **do**
    for j = 1 to n **do**
      $\mathrm{c}(v_i, v_j, 0) \leftarrow \infty$;
  for i = 1 to n **do**
    $\mathrm{c}(v_i, v_i, 0) \leftarrow \mathrm{w}(v_i)$;
  for all $(v_i, v_j) \in E(G)$ **do**
    $\mathrm{c}(v_i, v_j, 0) \leftarrow \mathrm{w}(v_i) + \mathrm{w}(v_j)$;
  for k = 1 to n **do**
    for i = 1 to n **do**
      for j = 1 to n **do**
        $\mathrm{c}(v_i, v_j, k) \leftarrow \min(\mathrm{c}(v_i, v_k, k-1) +$
          $\mathrm{c}(v_k, v_j, k-1) - \mathrm{w}(v_k),$
          $\mathrm{c}(v_i, v_j, k-1))$

B. The Verification of α_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 α_P-pairs: *Two pairs (a, b) and (c, d) are α_P-equivalent if 1) dist$(a, b) \geq \alpha$ and 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 α_P-coverage can then be derived by noticing that 1) if (a, b) and (c, d) are equivalent α_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. The Hardness of α-coverage

We name the decision version of the α-coverage (resp. α_N-coverage and α_P-coverage) optimization problem α-COVER (resp. α_N-COVER and α_P-COVER). In this section, we show that α-COVER is NP-hard and α_N- and α_P-COVER are NP-complete. By the existence of verification algorithms just presented, it suffices to show that there is an NP-complete problem that can be reduced to α-COVER (resp. α_N-COVER and α_P-COVER) in polynomial time. It is known that VERTEX COVER is NP-complete when restricted to triangle-free graphs [2] without degree 1 vertices, since it remains NP-complete when restricted to triangle-free, 3-connected, cubic [3] planar graphs [23]. We will first reduce this subproblem of VERTEX COVER to a subproblem of α_N-COVER with $\alpha = 2$ and $|A_0| = 0$, that is, there are no previously deployed APs, which is then extended to the other two cases. In this section, α is fixed to 2.

Lemma IV.1. *If G is a triangle-free graph having no degree 1 vertices where each edge has unit length, a set of vertices form a vertex cover of G iff it provides α_N-coverage to G.*

Proof: Suppose that a set of vertices, A, provides vertex cover to G, but does not provide α_N-coverage. Then there is a path of length 2 that is disjoint from any vertices in A, which contains at least one edge not covered by any vertices in A, a contradiction. Conversely, suppose a set of vertices, A, provides α_N-coverage to G, but it is not a vertex cover, then there must be an uncovered edge (u, v). Since G has no degree 1 vertices and is triangle free, there must exist edges (u_1, u) and (v, v_1) with middle points a and b, respectively, such that $f_{ab} = auvb$ is a path disjoint from A and dist$(a, b) = 2$, a contradiction. ∎

[2]A graph is triangle free if it has no cycles of length three

[3]A cubic graph is a graph where each vertex is incident to exactly three edges.

Although in general, a deployment that only uses vertices may be suboptimal as shown in Section V, for the set of instances of α_N-coverage we consider in this section, restricting APs to vertices actually gives optimal solutions as given by the following lemma.

Lemma IV.2. *Let G be a triangle-free graph having no degree 1 vertices where every edge has unit length. If there is a set of k **points** in G that provides α_N-coverage to G, then there is a set of k **vertices** that also provides α_N-coverage to G.*

Proof: Let A be a set of k points in G that provides α_N-coverage to G. We will construct a set of vertices, B, such that $|B| \leq k$ and B is a vertex cover for G. The claim then follows from Lemma IV.1.

First, we note that for any vertex v, there is at most one edge incident to v that is disjoint from A. Since, otherwise the two uncovered edges incident on v, say (u, v) and (v, w), form an uncovered path (u, v, w) with dist$(u, w) = 2$, due to the fact that G is triangle-free. This contradicts the fact that A provides α_N-coverage to G.

All the points in A that are also vertices are first added to B. Consider an edge (x, y) that is disjoint from A. There must exist distinct edges (w, x) and (y, z) with $w \neq z$ and points a on (w, x) and b on (y, z) such that $a \in A$ and $b \in A$, and $|\overline{ax}| \leq 0.5$ or $|\overline{by}| \leq 0.5$. Without loss of generality, suppose $|\overline{ax}| \leq 0.5$. Add x to B. If there is an edge (v, w) incident to w that is disjoint from A, then there must be an edge (u, v) and a point c on (u, v) such that $|\overline{cv}| \leq 0.5$, add v to B. Continue this process until either y or a vertex where all the incident edges are covered by points in A is reached. Then pick another edge that is disjoint from A. Repeat the process until all such edges have been considered. Finally, for each edge that contains a point in A and has not been visited, add any one of its two ends to B. By its construction, B covers all the edges and $|B| \leq |A|$. ∎

Theorem IV.1. *α_N-COVER is NP-complete.*

Proof: Given a triangle-free graph G without degree 1 vertices, make an instance of α_N-COVER with G as the graph model of a road network where each edge has unit length, $\alpha = 2$, and $|A_0| = 0$. The theorem then follows from Lemma IV.1 and Lemma IV.2. ∎

The above argument can be applied to α_P-COVER directly. Therefore, we have

Theorem IV.2. *α_P-COVER is NP-complete.*

For α-COVER, we can prove the following results by the similar argument as above.

Lemma IV.3. *If G is a graph having no degree 1 vertices where each edge has unit length, a set of vertices form a vertex cover of G iff it provides α-coverage to G.*

Lemma IV.4. *Let G be a graph having no degree 1 vertices where every edge has unit length. If there is a set of k **points** in G that provides α-coverage to G, then there is a set of k **vertices** that also provides α-coverage to G.*

Theorem IV.3. *α-COVER is NP-hard.*

Proof: Given a graph G without degree 1 vertices, make an instance of α-COVER with G as the graph model of a road network where each edge has unit length, $\alpha = 2$, and $|A_0| = 0$. The theorem then follows from Lemma IV.3 and Lemma IV.4. ■

V. Approximation Algorithms for α-Cover

In this section, we present approximation algorithms for α_N-Cover and α_P-Cover. Formally, given graph $G_{\mathcal{R}}$ and a set of point A_0 in $G_{\mathcal{R}}$, and α, find a set of points A such that $A_0 \cup A$ provides α_N or α_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 α. Assumption III.1 is still assumed.

We present two polynomial time algorithms in this section. The first algorithm reduces α_N-coverage to the vertex multicut problem [13] and the second one reduces α_P-coverage to the set cover problem [24]. Both algorithms have an $O(\log n)$ approximation factor, where n is the number of vertices in $G_{\mathcal{R}}$. The second algorithm also works for α_N-coverage. However, only for α_P-coverage, the algorithm has polynomial time complexity. 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 α_N-coverage provides higher coverage quality than α_P-coverage.

Given α, let OPT denote the minimum $|A|$ in any deployment that provides α_N-coverage where APs can be deployed at anywhere in G, and OPT$'$ denote the minimum $|A|$ for providing α_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.

Lemma V.1. *OPT $\leq$ OPT$'$ $\leq 2 \times$ OPT.*

Proof: OPT $\leq$ OPT$'$ follows directly from the definition. Let A be a set of OPT points such that $A_0 \cup A$ provides α_N-coverage. We apply the following two rules to construct a set of vertices, say A', such that $A_0 \cup A'$ also provides α_N-coverage: 1) add all the points in A that are also vertices of G to A'; 2) for each point in A that is not a vertex, add the two ends of the edge where the point is on to A'. We have $|A'| \leq 2 \times |A|$. Let F denote the set of paths required to be covered to ensure α_N-coverage. For any $f \in F$, there exists a point, say p, in $A_0 \cup A$ that covers f. If p is a vertex, f is also covered by $p \in A_0 \cup A'$. Otherwise, suppose p is on edge (u, v). Then $u \in A'$ and $v \in A'$. Since $|f| \geq \alpha \geq |(u, v)|$, f goes through at least one of u and v, and thus is covered by A'. Therefore, $A_0 \cup A'$ also provides α_N-coverage. ■

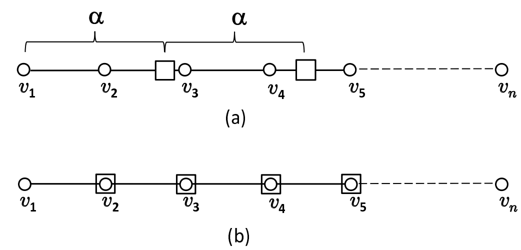

Fig. 2. A graph with a single path $(v_1, v_2, ..., v_n)$ where the squares represent APs. Each edge has unit length, and $\alpha \in (1, 2)$. (a) A minimum cover that provides α_N-coverage. (b) A suboptimal solution that uses only vertices to provide α_N-coverage.

The same result also holds for α_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.

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 α_N-coverage, an optimal solution is a set of points uniformly spaced with α 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 α_N-coverage. Therefore, OPT $= \lceil (n-1)/\alpha \rceil - 1$, OPT$' = n - 2$, and $\lim_{n \to \infty}$OPT$'$/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. In general, if G is a single path with n vertices and each edge has unit length, the factor 2 can be achieved when $n = 2k+2, \alpha \in [k+0.5, k+1)$ for any integer $k \geq 1$.

A. α_N-Cover via Vertex MultiCut

Assuming only the vertices of G are used to construct A, the α_N-Cover optimization problem can be reduced to the minimum vertex multicut problem [13] defined as follows. Given a connected undirected graph $G(V, E)$ with positive costs on its vertices, let $\{(s_1, t_1), ...(s_k, t_k)\}$ 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 V.1.

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 dist$(a, b) = \alpha$, which can be done in constant time as follows. Suppose $e_1 = (u_1, u_2), e_2 = (v_1, v_2)$. Let $t_1 = |au_1|/|e_1|$, $t_2 = |bv_1|/|e_2|$. Let $d_{ij}(a, b) =$ dist$(u_i, v_j) + |au_i| + |bv_j|, i, j \in \{1, 2\}$. Then $d_{ij}(t_1, t_2)$ is a linear function of $t_1 \in [0, 1]$ and $t_2 \in [0, 1]$ for $i, j \in \{1, 2\}$. For

Algorithm V.1 α_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 α

if A_0 provides α_N-coverage to G **then** return;
$T \leftarrow \emptyset$;
for each pair of edges (e_1, e_2) of G **do**
  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**
    $T \leftarrow T \cup (m_1, m_2)$; ▷ m_i: the midpoint of e_i
return $\text{GVY}(G, T, A_0)$; ▷ Apply the GVY algorithm [24] to find a vertex multicut with A_0 as a subset that separates each pair of midpoints in T.

given $i, j, i', j' \in \{1, 2\}$ and $(i, j) \neq (i', j')$, the solutions to the equation $d_{ij} - d_{i'j'} = 0$ divide $[0, 1] \times [0, 1]$ into two subregions. All such equations then give a partition of $[0, 1] \times [0, 1]$ composed of convex subregions. Using the partition, the minimum and maximum distance between any pair of points on the two edges can be determined by only studying the vertices of the convex regions. There is an α-pair iff α is between the two extremal values. 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 [24] to find a minimum vertex multicut with A_0 as a subset that separates each pair of midpoints inserted.

The algorithm stated in [24] actually solves the minimum *edge* multicut problem [12] 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 [13].

Let $S = \{s_1, s_2 ..., s_k\}$ denote the set of sources and $T = \{t_1, t_2, ..., t_k\}$ denote the set of destinations in the terminal pairs. For each vertex $v \in V \backslash (S \cup T)$, there is a non-negative variable d_v called distance label. For each vertex $v \in V$ and each terminal pair $(s_i, t_i), i = 1, ..., k$, there is a variable $y_{v,i}$, which is the shortest distance (in terms of the distance labels) from s_i to v. When applied to our scenario, d_v corresponds to the coverage weight of v and $y_{v,i}$ corresponds to $c(s_i, v)$, the coverage distance from s_i to v. The linear program (after LP-relaxation) of the minimum vertex multicut problem is as follows, where c_v is the cost of vertex v and is set to 1 for each v in our scenario.

$$\begin{array}{ll} \text{minimize} & \sum_{v \in V \backslash (S \cup T)} c_v d_v \\ \text{subject to} & y_{v,i} \leq y_{u,i} + d_v, \forall (u, v) \in E, \forall i = 1, ..., k. \\ & y_{s_i,i} = 0, \forall i = 1, ..., k, \\ & y_{t_i,i} \geq 1, \forall i = 1, ..., k, \\ & d_v \geq 0, \forall v \in V \backslash (S \cup T). \\ & d_v = 0, \forall v \in S \cup T. \end{array}$$

The first constraint says that $y_{v,i}$ satisfies triangle inequality. The second constraint says that the shortest distance from s_i to itself is 0. The third constraint requires that the shortest distance between each terminal pair is at least 1, which ensures that the set of vertices with positive distance labels form a multicut. The third and fourth constraints together ensure that the distance label of each nonterminal vertex is between 0 and 1, and the last constraint says that the distance label of each terminal is 0. The dual of the above program models the vertex version of the well-known maximum multicommodity flow problem. The GVY algorithm first solves the above program to get a set of distance labels for all the vertices. Then from those vertices with positive distance labels, a subset of them are selected to form a cut (LP-rounding). This is the best result known for the minimum vertex multicut problem for a general graph. 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 solved by fixing the cost and the distance label of each vertex in A_0 to be 0 and 1, respectively. The analysis in [24] can still be applied to show that this modification does not impact the approximation factor.

Since the GVY algorithm has a factor $O(\log k)$ where $k = O(|E|^2)$, considering Lemma V.1, Algorithm V.1 has an $O(\log n)$ factor.

B. α_P-Cover via Set Cover

Assuming only the vertices of G are used to construct A, the α_P-coverage optimization problem can be reduced to the set covering optimization problem [24] as follows:

Algorithm V.2 α_P-Cover via Set Cover

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 α

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

The above algorithm also applies to α_N-Cover, where set $\mathcal{U}$ has exponential size. For α_P-Cover, $|\mathcal{U}| = O(|E|^2)$. Since there is a factor $O(\log |\mathcal{U}|)$ algorithm for set cover [24], considering Lemma V.1,this algorithm has an $O(\log n)$ factor.

VI. Discussion

In this section, we discuss several issues related to the modeling of road networks and APs including some real constraints ignored before, and an extension of α-coverage.

A. Modeling issues

We have considered an undirected graph model of road networks so far to simplify the discussion. However, most of our results can be extended to a directed graph that models one-way roads as well. First, we can extend the definitions of general paths and distance on graph by taking direction into account. In this case, $\text{dist}(a, b)$ and $\text{dist}(b, a)$ may be different. The definition of the three coverage metrics and the two types of equivalent classes can be extended as well.

Then both the coverage verification algorithm and the set cover based optimization algorithm can be extended to directed graphs. On the other hand, it is known that the minimum multicut problem is much harder to approximate for directed graphs than undirected graphs. There are some recent results for finding the minimum edge multicut [5], [16]. However, whether these results can be extended to vertex multicut with the same approximation ratio needs further study.

We have modeled APs as points in the graph of a road network, which can be justified as follows. First, we are considering a *planned* deployment, and therefore it makes sense to deploy APs as close to road centerlines as possible, to maximize the area on roads covered by APs. Second, we are considering a sparse deployment in this paper, and therefore, we ignore the case where the coverage regions of multiple AP overlap with each other. We will explore denser deployments where overlappings are frequent in the future work. Third, we are seeking a solution that provides the worst case guarantee to the interconnection gap, and therefore make the most conservative assumption about the contribution of each AP.

B. More about α_P-coverage

In our definition of α_P-coverage, the set of paths associated with a pair of points can be an arbitrary set of critical paths. Although we use shortest path as an example in the coverage verification and optimization algorithms, they can be extended to the general case as long as the following condition is satisfied: for each pair of edges of G, there are only a constant number of equivalent classes of α_P-pairs. Then the total number of equivalent classes is bounded by $O(|E|^2)$. Since the verification algorithm takes $O(c|V|)$ time to check each equivalent class where c is the maximum number of paths associated with a pair, the total running time is $O(c|V||E|^2)$. The set cover based optimization algorithm takes two collections $\mathcal{U}$ and $\mathcal{S}$ as the input, where $\mathcal{S} = O(|V|)$. Since each equivalent class contributes at most c paths to $\mathcal{U}$, $|\mathcal{U}| = O(c|E|^2)$. Therefore, the optimization algorithm can also be done in polynomial time.

C. From α-coverage to (α, β)-coverage

We now consider an interesting extension of α-coverage, which guarantees the quality of data access in the worst case. We have not found complete solutions to this extension and will continue to work on it. In particular, we say that *a deployment of APs provides (α, β)-coverage to a road network, if for any path of length $\geq \alpha$ that a vehicle travels, it is guaranteed that the vehicle has access to at least β bits of data*. This can be viewed as an extension of α-coverage. The similar extensions can be applied to α_N- and α_P-coverage as well. We again have the following two problems to solve: 1) determine whether a deployment provides (α, β)-coverage, and if not 2) find an optimal incremental deployment to provide (α, β)-coverage. We may again want to minimize the number of new APs used. However, other optimization goals are also possible. For instance, we may consider a heterogenous deployment and try to minimize the total cost of the new APs.

For this new coverage metric, we can not simply model APs as points in the graph since the amount of data that a particular vehicle can access from an AP is approximately proportional to the amount of time that vehicle is associated with the AP, which is in turn determined by both the location of the AP and its transmission range. Notice that, by ensuring α-coverage, we can achieve (α, u)-coverage where u is the minimum amount of data that a vehicle may access from an AP within the range. Then one way to approximate (α, β)-coverage is to guarantee that for any path of length $\geq \alpha$ that a vehicle travels, the vehicle can contact with has at least $m = \lceil \beta/u \rceil$ non overlapping APs. On the other hand, if a vehicle has m antennas installed so that m APs in the range can be simultaneously accessed, (α, β)-coverage can be approximated by first finding a set of points that provides α-coverage, then deploying m APs at each point found.

VII. Evaluation

In this section, we present the simulation results where we compare α-coverage against a random deployment (as a good model for the distribution of open APs). We evaluate the distribution of the interconnection gap provided by our algorithm and that by three random deployment techniques, by using data from a real-world road network. We show that using the same number of APs, the interconnection gap under α-coverage is bounded to a much smaller distance and has a much smaller standard deviation than that in a random deployment. Now we discuss the details of our simulations.

A. Simulation settings

a) Generating Road Networks:: We obtain road network data from the 2007 Tiger/Line shapefiles [3]. We only use the All Lines shapefile since it contains all the information about a road network we need. The database is organized by counties. Our simulations are based on the road network of the Franklin county in State of Ohio. The database does not contain information about one-way roads, so we only consider the undirected graph model of road networks. In the database, 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 to facilitate the generation of movement files and ns-2 based simulations. Although the projection may distort the distance, the simulation results are still valid since we compare our approach with random deployment on the same road network. In the Tiger/Line database, the roads in U.S. are classified into different types. We consider two cases: 1) a constant speed limit (55 miles/hour) is assigned to all the roads; 2) different types of roads are assigned different speed limits to reflect the real traffic, from 25 miles/hour to 65 miles/hour.

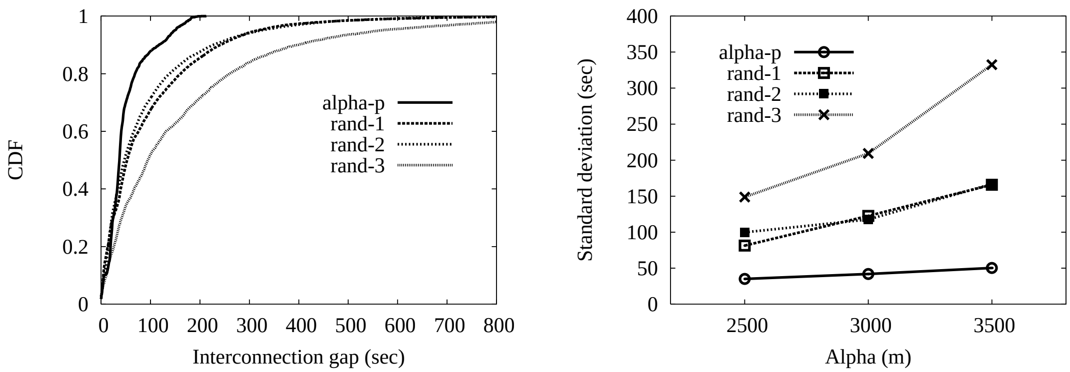

Fig. 3. Left: the cumulative distribution of the interconnection time when $\alpha = 3000$m. Right: the standard deviation of the interconnection time for various α. The speed limit is fixed to 55 miles/hour. The maximum gap for α_P-coverage is about 214 sec, bounded by the time spent on two adjacent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

b) Generating Moving Scenarios:: A restricted random waypoint mobility model [4] is used to generate ns-2 movement files. The model accepts the graph of a road network as input. We didn't start simulations at steady-state [9] since it is too time consuming for large graphs. Each mobile node starts at one end of a randomly selected road segment. The node then moves towards another randomly selected intersection by following a shortest path. Once the destination is reached, the node pauses shortly, then starts another move and so on. We add one more constraint such that the source and destination of each move must have a distance at least α since we are interested in providing performance guarantee to these moves. The movements of different mobile nodes are independent. The speed of a mobile node on a road segment is set to the speed limit of that segment with a leeway of 5 miles/hour. The mean of the pause time is set to 5 sec. The number of mobile nodes are fixed to be 5. Each simulation lasts for 1 hour.

c) Generating Deployments:: We compare α_P-coverage with the following three random deployment techniques that use the same number of APs as our approach: 1) each AP is deployed on a randomly selected vertex of G; 2) each AP is deployed on a randomly selected point of a randomly selected edge of G; 3) the region spanned by G is first divided into 50m$\times$50m 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. We name the three random schemes as rand-1, rand-2, and rand-3, respectively. Given a deployment output by our algorithm, 10 random deployments are generated for rand-1, 2, and 3, respectively. For a given α, 10 movement files with constant speed limit and 10 movement files with variable speed limit are generated. The simulations are carried over a 4000m $\times$ 4000m region around the center of the Franklin county including about 1000 intersections and 1300 road segments, and the graph diameter is about 7300m. The transmission range of each AP is set to be 100m.

B. Simulation results

Figure 3(left) shows the cdf of the interconnection time of α_P-coverage and three random deployments, where α 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 α_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, the maximum gap for α_P-coverage is larger than α divided by the speed limit, which is about 122 sec. This is because of the moving pattern we use. Since each move follows a shortest path of length at least α, α_P-coverage guarantees that at least one AP will be touched within a move. However, an interconnection gap may span two adjacent moves, and therefore in the worst case, the gap can be as large as the time spent on two adjacent moves. Figure 3(right) shows the standard deviation of the interconnection time under various α and the constant speed limit, where the number of APs used for $\alpha = 2500$m, 3000m, 3500m is 28, 21, and 15, respectively. We can clearly see that the standard deviation for α_P-coverage is much smaller than that for a random deployment.

In Figure 4, the cdf of the connection duration of different deployments are shown. We can see that $rand$-3 performs slightly worse than others since the APs are not as close to the roads as in other deployments.

Figure 5 shows the cdf of the interconnection time under variable speed limit and $\alpha = 3000$m. All the deployments have larger maximum interconnection gaps compared with the case of constant speed limit, but α_P-coverage still performs best.

VIII. Related Work

The idea of Drive-thru Internet was first introduced in the seminal paper [21], which shows that a single moving vehicle connected via 802.11b with an AP located at roadside of an empty street can access several megabytes of TCP or UDP traffic, even when the velocity is as high as 180km/h. Thereafter, evaluations in various controlled environments [6],

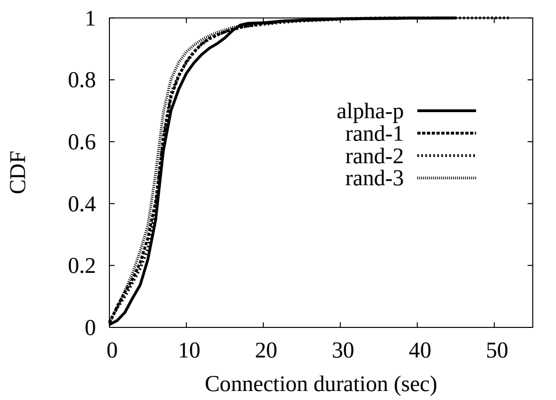

Fig. 4. $\alpha = 3000$m, speed limit = 55 miles/hour

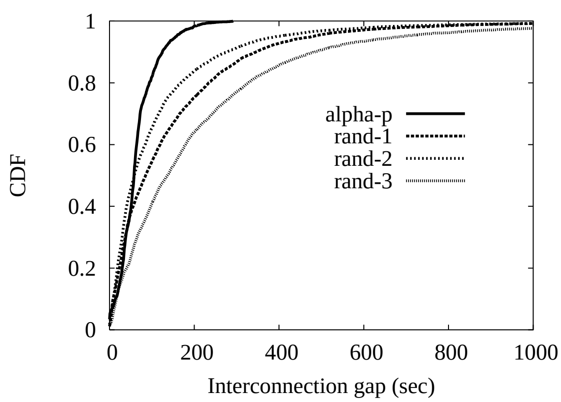

Fig. 5. $\alpha = 3000$m, variable speed limit

[14], [20], [22] and in situ Wi-Fi networks [6], [10], [11], [20] were conducted, which further confirm the feasibility of WiFi-based Vehicular Internet Access for non-interactive applications. The possibility and challenges to support certain interactive applications, such as Web browsing, have also been studied [6], [7].

Vehicular communication through Wi-Fi infrastructure is characterized by short-lived and intermittent connections, which challenges both the 802.11 MAC and the transport protocols. It was shown in [21] that the channel quality when a vehicle moving through an AP can be roughly divided into three phases: the entry phase when connectivity is weak and loss and delay are both high, the production phase when effective communication can take place, and the exit phase when loss and delay are high again. The initial setting of parameters and the default behavior of 802.11 and TCP can lead to performance loss in all the three phases as shown in [18]. In [10], the time spent on each stage of connection setup was measured, and a simple IP caching approach was proposed to reduce the delay induced by DHCP. It was reported in [11] that the mean connection setup time can be reduced from over 10s to 400ms by using a streamlined client-side connection setup process, which also increases the number of usable short connections significantly. Transport protocols that hide the wireless losses from the wired side, and the temporary unavailability of connections from the client were proposed in [11], [22]. Interesting ideas have been proposed regarding the scenarios where a moving vehicle is in the transmission ranges of multiple APs [6], [20] and vice verse, multiple vehicles are associated with a single AP [17]. In particular, [20] shows that the use of directional antennas at vehicle side and beaming steering techniques can improve the performance of 802.11 link via carefully designed handoff algorithms. In [6], a lightweight coordination protocol was designed that allows a vehicle to communicate with multiple APs simultaneously to reduce disruption in connectivity. It was noted in [17] that multiple vehicles associated with the same AP may choose different transmission rates and therefore suffer from the 802.11 performance anomaly, that is, the date rates of all these vehicles will eventually be slowed down to the lowest one. A medium access protocol that grants the channel to the vehicle with best SNR was suggested in [17].

It should be noted that the deployment issues with respect to WiFi-based Vehicular Internet Access have not been carefully studied so far. Instead, an unplanned deployment of APs is commonly assumed in most previous works. A simple non-uniform strategy that places more stationary nodes in the network core was considered in a recent work [8]. However, it was completely based on intuition without providing any performance guarantees.

IX. Conclusion

In this paper, we propose α-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 α-coverage and if not, to find the optimal places to deploy new APs so that α-coverage can be ensured. Both the networking protocols and applications may explore such guarantees to optimize their behaviors. Furthermore, α-coverage is the first step towards a scalable deployment that guarantees the worst case data service that a mobile user can expect.

X. 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] WiMAX Technology. http://www.wimaxforum.org/technology/.

[2] VeriWise Asset Intelligence. http://www.ge.com/equipmentservices/assetintelligence/.

[3] U.S. Census Bureau - TIGER/Line http://www.census.gov/geo/www/tiger/.

[4] The Random Trip Mobility Model. http://lrcwww.epfl.ch/RandomTrip/.

[5] A. Agarwal, N. Alon, and M. S. Charikar. Improved Approximation for Directed Cut Problems. In *Proc. of STOC*, 2007.

[6] 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.

[7] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani. Web Search From a Bus. In *Proc. of CHANTS*, 2007.

[8] N. Banerjee, M. Corner, D. Towsley, and B. Levine. Relays, Base Stations and Meshes: Enhancing Mobile Networks with Infrastructure. In *Proc. of ACM MOBICOM*, Sept. 2008.

[9] J.-Y. L. Boudec and M. Vojnović. Perfect Simulation and Stationarity of a Class of Mobility Models. In *Proc. of INFOCOM*, 2005.

[10] 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.

[11] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: A WiFi-Based Vehicular Content Delivery Network. In *Proc. of ACM MOBICOM*, Sept. 2008.

[12] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications. *SIAM J. Comput.*, 25(2):235–251, 1996.

[13] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *Journal of Algorithms*, 50(1):49–61, 2004.

[14] R. Gass, J. Scott, and C. Diot. Measurements of In-Motion 802.11 Networking. In *Proc. of WMCSA*, Apr. 2006.

[15] 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.

[16] A. Gupta. Improved Results for Directed Multicut. In *Proc. of SODA*, 2003.

[17] D. Hadaller, S. Keshav, and T. Brecht. MV-MAX: Improving Wireless Infrastructure Access for MultiVehicular Communication. In *Proc. of CHANTS*, 2006.

[18] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *Proc. of MOBISYS*, 2007.

[19] M.R. Garey, D.S. Johnson, and R.E. Tarjan. The Planar Hamiltonian Circuit Problem is NP-Complete. *SIAM J. Comput*, 5(4), 1976.

[20] 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.

[21] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for "Automobile" Users. In *Proc. of INFOCOM*, Mar. 2004.

[22] J. Ott and D. Kutscher. A Disconnection-Tolerant Transport for Drive-Thru Internet Environments. In *Proc. of INFOCOM*, Mar. 2005.

[23] Uehara, R. NP-complete problems on a 3-connected cubic planar graph and their applications. Technical Report TWCU-M-0004, Tokyo Woman's Christian University, 1996.

[24] V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.