Change Awareness in Opportunistic Networks

Xiang Fa Guo
School of Computing
National University of Singapore
Email: xiangfa@comp.nus.edu.sg

Mun Choon Chan
School of Computing
National University of Singapore
Email: chanmc@comp.nus.edu.sg

***Abstract*—Information latency and information reachability are widely used metrics for measuring information flow in Opportunistic Networks. We present an alternative measure that looks at the amount of information updates or changes from a given sender, which is an important and yet relatively unexplored metric for characterizing opportunistic networks.**

In this paper, we propose a novel concept to measure information freshness — how much the latest information received differs from the most up-to-date information on the sender. Based on information freshness, awareness on how information propagated may have changed, or *change awareness*, can be computed. Change awareness can be exploited to design efficient algorithms, in particular, data aggregation algorithms. Evaluation on real world traces shows that change awareness based solutions can achieve similar results as data aggregation algorithm with a connection oracle.

***Index Terms*—Opportunistic Network; message freshness; data aggregation.**

I. Introduction

Opportunistic networks formed by mobile devices with short range wireless communication, e.g., Pocket Switch Network (PSN) [1] and Vehicle Ad hoc Networks (VANET) [2], have emerged and gained popularity in the past decade. The most distinctive feature of opportunistic networks is the intermittency of connections. The intermittent connections enable opportunistic communication, meanwhile, they also complicate many problems that have much simpler solutions in static networks with persistent connections, e.g., the computation of the distance between two nodes.

The behavior of opportunistic networks are often measured by information latency [3][4] and reachability [5]. We propose a novel measure with a focus on the *freshness* of the information available. In particular, we measure information freshness by capturing how much the latest information received by a receiver differs from the most up-to-date information being propagated by a sender. Such differences are due to the changes caused by recent connections. This *information freshness* measure is independent from transmission latency. It instead depends on how much global information may have changed from the local view. We illustrate the usefulness of information freshness by the following two examples.

(1) In a dynamic network, many applications require nodes to collect data. It is often impossible to collect real-time topology information due to the intermittent connectivity of short range communication. Knowledge on how much the topology has changed provides a way to view this information update and can be used to design efficient routing and/or data collection schemes.

(2) Mobile phones or vehicles can be used to sense the environment. Efficient data aggregation for the sensor data collection is important. In static wireless sensor networks, data aggregation has been extensively researched. Most of them focus on selecting cluster headers or good information flow paths. However, we do not see general solutions for data aggregation in opportunistic networks. How much a node can know about other nodes' updates can undoubtedly help select uploaders, i.e., cluster headers for opportunistic networks.

In this paper, the following contributions are made.

1) We propose a measure to quantify information freshness, called **change awareness**, in opportunistic networks and how it can be computed.
2) We propose the concept of change awareness coverage and show how a small number of informative nodes can be used to construct a fresh (or real-time) snapshot of network states. Our trace based evaluation shows that depending on the application scenarios, the information collected from 0.3% to 50% of the nodes can form a fresh snapshot on topology states.
3) Finally, we present the application of change awareness in data aggregation. Extensive evaluation shows that change awareness based solution can achieve similar performance as the solution with connection oracle.

This paper is organized as follows. In Section II, we present related work. We present the concept of information paths in Section III, change awareness in Section IV and the coverage of informative nodes in Section V. Application of change awareness is presented in Section VI and we conclude the paper in Section VII.

II. Related Work

A. The Measurement on Opportunistic Networks

As a natural model for opportunistic networks, Time-Varying Graphs (TVG) [4] have attracted a plethora of research interest. In a TVG, edges, i.e., communication chances, appear and disappear and hence, a node's information on other nodes is usually delayed. Research on TVGs has evolved from the study on the common feature with static graphs to a focus on temporal features. Initially, the research on TVGs concentrated on connectivity issues, such as the k-failure invulnerability problem [6], the computation of the shortest paths [3], and

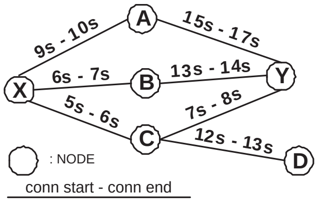


Fig. 1: An example of temporal graph.

broadcast properties [7]. More recent work focus on 'temporal features', e.g., information latencies and reachability in TVG. As an extension for the work of investigating information flow in Online Social Network (OSN) [8], message latencies in DTNs with arbitrarily long contacts was examined in [9]. Research in [4] formalized temporal related features that are common for dynamic networks and proposed techniques for investigating the evolution of network properties. The work in [5] formalized reachability graphs and set up a theoretical framework for computing reachability graphs. Along with these theoretical analysis, many other research examined information latencies, information reachability in OSN traces [10], as OSNs can provide convenient avenues of collecting scalable traces.

These existing work investigates information transmission by examining latencies and reachability. Our work instead examines the changes observable in the network and how these changes affect information collection. In fact, our work is similar in spirit to the seminar work [11] in that we attempt to find a meaningful fresh snapshot of the network state.

B. Data Aggregation on Opportunistic Networks

We briefly discuss the work on data aggregation relative to our evaluations. For a detailed survey, readers might read [12]. Our solution in data aggregation is close to cluster based solutions [13] and network-flow based solutions [14]. Existing solutions are often designed for static wireless networks and are unsuitable for opportunistic networks due to connectivity dynamics. Most solutions for opportunistic networks exploit the location information of vehicles and road side infrastructures [15]. Our approach does not rely on location information.

Recent theoretic work [16] investigates the possibility of collecting data via short range connections to a subset of nodes to reduce the use of long range connections. They proved theoretical bounds and show that no aggregation algorithm can achieve better performance than the optimal solution with connection oracle. Our solution in evaluation has the same goal of reducing the use of long range communications. We investigate how much can change awareness help minimize the number of uploaders, or cluster headers as in cluster based aggregation.

III. Temporal Graph Models and Information Paths

We use Figure 1 to illustrate three different notions of information paths used in this work. The interval indicated on each link is the duration in which the link exists.

Symbol	Interpretation
$\mathcal{G}(V, E, \mathcal{T}, \rho)$ or $\mathcal{G}$	a Time-Varying Graph
$\varepsilon(u, v, t)$	a connection between u and v ending at t.
$Time(\varepsilon(u, v, t))$	the end time of connection :ε, i.e., t.
$I_p(u, v, t_s, t_e)$, $I_l(u, v, t_s, t_e)$, $I_f(u, v, t_s, t_e)$	information path, the latest information path, fresh information path from u to v starting at t_s and ending at t_e
$\mathbb{I}_p(u, v, t)$, $\mathbb{I}_l(u, v, t)$, $\mathbb{I}_f(u, v, t)$	the set of information paths (the latest information paths, fresh information paths) from u and reachable to v no later than t
$F(\mathcal{G}, u, v, t)$	pairwise change awareness: the change awareness measured by v regarding to u at t.
$F(\mathcal{G}, v, t)$	vertex change awareness: the centrality of v on receiving changes from all other vertices at t.
$F(\mathcal{G}, t)$	graph change awareness: the average change awareness of vertices in $\mathcal{G}$ at t
$\mathbb{C}_m$	the smallest number of out-of-band connections required to achieve fresh snapshot
$\mathcal{P}$	the set of smallest panorama informative nodes

TABLE I: Table for symbols interpretation.

First, an information path is one where information can flow from source to destination, e.g., $X \to C \to Y$. However, when considering whether the information is the latest information the receiver can get, we can see that at different time points, the latest information are transmitted via different information paths. For example, at the time 14s, the latest information by Y comes from the path $X \to B \to Y$. Finally, when we are concerned with whether the latest information received is the most up-to-date, i.e., the same as the sender's view, then the information is up-to-date or absolute fresh only after the time 15s, since the earlier information received by Y through the paths $X \to C \to Y$ and $X \to B \to Y$ is already outdated (via a later contacts occurred to X) by the time the information reached Y.

These three different concepts of information path are utilized in our work. In the rest of this section, we define them formally using a model based on Time Varying Graph.

A. Temporal graphs and connections

We use Time-Varying Graphs (TVGs) to model dynamic networks, as TVGs can naturally exhibit dynamic properties. Our notations for TVGs are based on those used in [4][5].

Definition 1 (Temporal graph)**.** *Let V be the set of vertices*[1]*, and $E \subseteq V \times V$ be the set of edges. $\mathcal{T}$ denotes* time span*, which starts at $\mathcal{T}_s$ and ends at $\mathcal{T}_e$, where $\mathcal{T}_s, \mathcal{T}_e \in R_+$ and $\mathcal{T}_s \leq \mathcal{T}_e$. $\rho(e, t)$: $E, R \to \{0, 1\}$ is edge presence function, indicating whether an edge exists at time t. $\rho(e, t_-)$ indicates the presence of e before or at t; $\rho(e, t_+)$ indicates the presence of e after t. We call $\mathcal{G}(V, E, \mathcal{T}, \rho)$,or $\mathcal{G}$ for short, as a TVG .*

One occurrence of an edge corresponds to a connection in dynamic networks. The occurrence of the edge $\varepsilon(u, v, t)$ denotes a connection between u and v that ends at t. We consider the time instance of a connection as its connection end time. The set of universal connections in $\mathcal{G}$ is denoted by $\mathcal{E}_{\mathcal{G}}$. We use $\mathcal{E}_{\mathcal{G}}(v, t_-)$ and $\mathcal{E}_{\mathcal{G}}(v, t_+)$ denote v's connections before and after t respectively. The set $\mathcal{E}_{\mathcal{G}}(t)$ contains all connections in $\mathcal{G}$ at t, i.e., the connections in the snapshot of $\mathcal{G}$ at t. For

[1] 'vertex' and 'node' are interchangeably used.

information communication, we assume that a connection has sufficient bandwidth to allow sufficient message exchange.

B. Information Paths

An information path, also known as information flow, consists of a series of connections that can relay information from the first node to the last node.

Definition 2 (Information path)**.** $\forall u, v \in V$, $I_p(u,v,t_s,t_e)$ *designates an information path from* u *to* v*.* $I_p(u,v,t_s,t_e)$ *has the format of* $(u,v_1,t_1) \rightsquigarrow (v_1,v_2,t_2) \rightsquigarrow \cdots \rightsquigarrow (v_{k-1},v,t_k)$*, where* $(t_s = t_1 \le t_2 \le \cdots \le t_k = t_e)$*. Also,* $v_0 = u, v_k = v$*,* $\rho(e_i,t_i) = 1$*, where* $e_i : \langle v_i, v_{i+1}\rangle$ *and* $i = 0,\cdots,k-1$*.* $I_p^i(u,v,t_s,t_e)$ *indicates the* i^{th} *connection in* $I_p(u,v,t_s,t_e)$*, i.e.,* $\varepsilon_i = \langle v_{i-1}, v_i, t_i\rangle$*;* $I_p^{last}(u,v,t_s,t_e)$ *denotes the last connection in* $I_p(u,v,t_s,t_e)$*. Let* $Time(I_p^i(u,v,t_s,t_e))$ *present the time of the* i^{th} *connection.* $\mathbb{I}_p(u,v,t)$ *symbolizes the set of information paths* $I_p(u,v,t_s,t_e)$*, where* $t_e \le t$*.*

Definition 3 (Latest information path)**.** *An information path* $I_p(u,v,t_s,t_e)$ *is the latest information path or* $I_l(u,v,t_s,t_e)$ *in* $\mathbb{I}_p(u,v,t)$ *if and only if* t_s *is the latest (largest) among all information paths in* $\mathbb{I}_p(u,v,t)$*. Hence,* $I_l(u,v,t_s,t_e)$ *carries the latest information from* u *available to* v *at* t*.* $\mathbb{I}_l(u,v,t)$ *symbolizes the set of latest information paths from* u *to* v *no later than* t*.*

Definition 4 (Fresh information path)**.** *An information path* $I_p(u,v,t_s,t_e)$ *is a fresh information path at time* t *or* $I_f(u,v,t_s,t_e)$ *if and only if* $\rho(\langle u,x\rangle, t_{s+}) = 0$ *for* $\forall x \in V - \{u\}$*. In other words,* u *has no connection with any other nodes in the interval* t_s *to* t*, and the information carried by this path is* absolute fresh*, as the* u *has no connection to update its information.* $\mathbb{I}_f(u,v,t)$ *denotes the set of all fresh information paths from* u *to* v *at time* t*.*

In Figure 1, there are three information paths: $X \rightarrow A \rightarrow Y$, $X \rightarrow B \rightarrow Y$ and $X \rightarrow C \rightarrow Y$.

From node Y's point of view, before the time 7s, it does not have any information about node X. The latest information path from X to Y are: $X \rightarrow C \rightarrow Y$ between the time 7s and 13s, $X \rightarrow B \rightarrow Y$ between the time 13s and 15s, and $X \rightarrow A \rightarrow Y$ when the time is later than 15s.

In terms of fresh information path, there is none before the time 15s, and the path is $X \rightarrow A \rightarrow Y$ from the time 17s onwards.

The information paths defined have the following properties. First, $\mathbb{I}_f(u,v,t) \subseteq \mathbb{I}_l(u,v,t) \subseteq \mathbb{I}_p(u,v,t)$. Second, $\mathbb{I}_l(u,v,t) = \emptyset$ if and only if $\mathbb{I}_p(u,v,t) = \emptyset$. Third, based on local observation only, a node can know which path is the latest information path but it cannot judge whether a path is fresh information path or not.

IV. Change Awareness

Based on the information paths introduced, we now present a concept on how *changes* in the network can be quantified. In particular, we are interested in changes that occur as a result of communication between two nodes. Such changes can affect both metadata (e.g. connection status or buffer content), as well as application data.

Trace	No. of nodes	Interface/ Trans. ange	Context
RollerNet[17]	62	bluetooth	outdoor rollerblading
Haggle IC06[18]	98	bluetooth	conference
SF taxi[19]	500	50m,100m, 200m	city taxi
Seattle Bus[20]	1200	50m	city shuttle bus
Shanghai Taxi[21]	5000	100m	city taxi
RPGM[22]	100	10m	reference point group mobility
Random Way Point (RWP)	100	10m	random and free node movement

TABLE II: Summary of traces used in evaluation.

Change awareness is measured based on the status of information paths involved. Three closely related metrics for information awareness are thereby proposed: pairwise change awareness, vertex change awareness, and graph change awareness. These three measurement have peer metrics that have stirred a plethora of research interest: geodesic vertices distance, vertex centrality, mean geodesic distance.

A. Definition on Change Awareness

Definition 5 (Pairwise change awareness)**.** *The pairwise change awareness* $F(\mathcal{G},u,v,t)$ *is the ratio of* u*'s connections that can be known by* v *at* t*.* $F(\mathcal{G},u,v,t) = |\mathcal{E}_\mathcal{G}(u,\tau_-)|/|\mathcal{E}_\mathcal{G}(u,t_-)|$*, where* τ *is the latest connection time with* u *in* $\mathbb{I}_l(u,v,t)$*.* τ *is set to* $\mathcal{T}_s$ *if* $\mathbb{I}_l(u,v,t)$ *is* $\emptyset$*.*

$F(\mathcal{G},u,v,t)$ is computed based on the set of latest information paths and measures the portion of u's connections no later than τ and u's connections no later than t. The largest freshness value, one, occurs when a fresh information path exists. The smallest freshness value, zero, occurs when no information path exists from u to v. As an example, the pairwise change awareness of the network at 17s in Figure 1 is presented in the following table, where the row is a source (u) and the column is a destination (v).

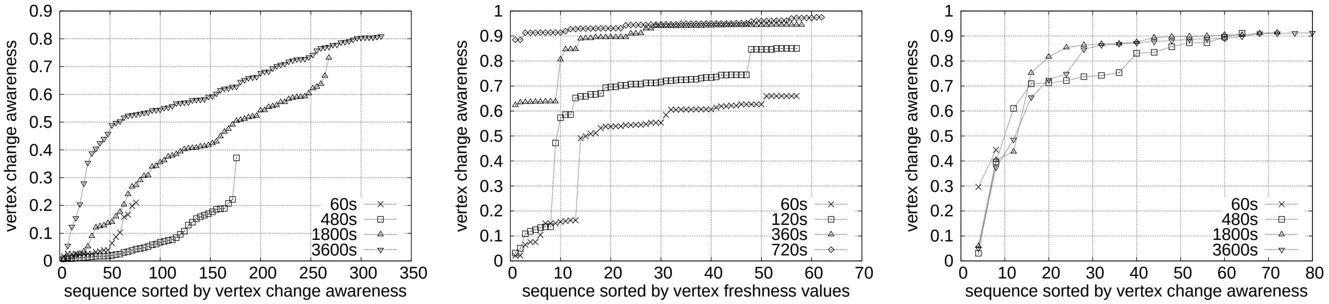
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 2/2 & 0/2 & 0/2 & 0/2 \\ B & 2/2 & 2/2 & 0/2 & 0/2 \\ C & 2/3 & 2/3 & 3/3 & 3/3 \\ D & 0/1 & 0/1 & 1/1 & 1/1 \end{array}$$

Definition 6 (Vertex change awareness)**.** *The vertex change awareness of a node* v*:* $F(\mathcal{G},v,t)$ *measures how much* v *can be aware of the changes of other nodes. It can be computed as* $\sum_{x\in V, x\neq v} F(\mathcal{G},x,v,t)/(|V|-1)$*.*

Definition 7 (Graph change awareness)**.** *The graph change awareness of* $\mathcal{G}$*:* $F(\mathcal{G},t)$ *designates the average awareness over all vertices in* $\mathcal{G}$*. It equals* $\sum_{v\in V} F(\mathcal{G},v,t)/(|V|)$ *or* $\sum_{x,y\in V} F(\mathcal{G},x,y,t)/P(|V|,2)$*.*

B. Computation of Change Awareness

Pairwise change awareness can be computed through 'last departure time': $Time(I_l^1(u,v,t))$ [23]. Vertex and graph



(a) SF taxi trace (50m).

(b) Roller Net trace.

(c) Haggle InfoCom06 trace.

Fig. 2: Vertex change awareness with different time span lengths.

```
Data: $\mathcal{G}(V, E, \mathcal{T}, \rho, \xi)$
Output: $F(\mathcal{G}, v, t)$
init.: $last(x) \leftarrow \mathcal{T}_s, \forall x \in V$, $known(\varepsilon) \leftarrow false, \forall \varepsilon \in \mathcal{E}$,
$W \leftarrow \{v\}$, $last(v) \leftarrow t$
while ($W$ is not empty) do
    select $u$ where $\max(last(u), u \in W)$
    for each $\varepsilon : \varepsilon(u, y, \tau)$ or $\varepsilon : \varepsilon(y, u, \tau)$ do
        if $known(\varepsilon) = false$ then
            if $\tau \leq last(u)$ then
                $last(y) \leftarrow \max(last(y), \tau)$
                if $y \notin W$ then
                    add $y$ to $W$
                end
                $known(\varepsilon) \leftarrow true$
            end
        end
    end
    remove $u$ from $W$
end
for $x \in V$ do
    $F(\mathcal{G}, x, v, t) \leftarrow \frac{\text{No. of known conn. involving } x}{\text{No. of conn. involving } x}$
end
$F(\mathcal{G}, v, t) \leftarrow \sum_{x \in V, x \neq v} F(\mathcal{G}, x, v, t)/(|V| - 1)$
```

Algorithm 1: Vertex change awareness computation.

change awareness can be computed via repetitively running pairwise freshness, but this can be expensive. They however can be computed by a more efficient algorithm that adapts dijkstra algorithm for computing shortest pathes from one vertex. This is shown in Algorithm 1 with the time complexity of $\Theta(|\mathcal{E}_{\mathcal{G}}| + |V| \log |V|)$ [24].

C. Change Awareness in Traces

We investigate the three types of change awareness on traces in Table II. Due to space limitation, we only present the results of vertex change awareness on SF taxi, Roller net, and Haggle InfoCom06 traces, whose contexts respectively represents vehicle traffic, outdoor human mobility, and indoor human mobility. We investigate change awareness with respect to four factors: (1) length of the time spans, (2) connection density (by varying transmission range), (3) node density (by removing random nodes), and (4) edge density (by removing random edges to simulate duty cycling) [25].

For measuring the effects of the length of time spans ($\mathcal{D}$), we varies the length from 60s to 3600s. Results in Figure 2 reveal two points. First, some nodes are much more aware of changes in the network than others; Second, pairwise change awareness usually increases as the time span becomes longer. For example, in the SF taxi trace, vertex change awareness varies from 0 to 0.8 when the time span is 3600s. For a short time span of 60s, the awareness varies from 0 to 0.38.

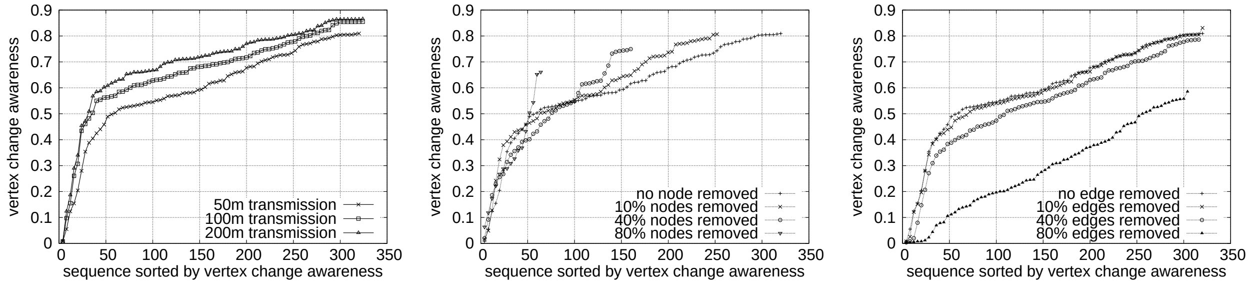
For testing the effects of contacts density, we set the transmission ranges of 50m, 100m and 200m, which respectively has the density of 16.8, 28.1, 48.7 contacts per hour per node for SF taxi trace. Figure 3a shows the results for time span of 3600s. Increase in node density have two opposite effects. One effect is to make information transmitted by v stale faster, which decreases change awareness. Another effect is to speed up the message transmission, which increases change awareness. The results indicate that the positive effects slightly outperform the negative effects as the transmission range increases. In an extreme case where the contact density is so large that end-to-end paths always exist, the change awareness approaches one.

We test effects on the loss in change awareness by performing duty cycling. The results in Fig 3c show that the vertex change awareness decreases slowly with edge removals. With up to 60%, the decrease is insignificant. Striking decreases in change awareness are observed when a node only awakes 20% of time. The results indicate that nodes can have duty cycling to save energy without significant loss in change awareness.

We also test the effects on change awareness by removing nodes from a network. The results in Figure 3b show no dramatically effects on vertex change awareness till that up to 40% of the nodes are removed. The reason could be that the removal of nodes reduces the chance of updates meanwhile it slows information dissemination. In the extreme case where there are only two nodes, change awareness is always one.

V. Informative Node

Next, we will investigate how a node's change awareness can be exploited. In particular, we focus on nodes that have the most information about changes in the network. We call such nodes *informative nodes*. An informative node is the last node in a fresh information path, i.e., v in $I_f(u, v, t_s, t_e)$, or v in $F(\mathcal{G}, u, v, t)$ when $F(\mathcal{G}, u, v, t)$ equals one. In terms of change awareness, an informative node must have a pairwise change awareness of one with respect to at least one other node.



(a) Transmission range change.

(b) Nodes removal.

(c) Edges removal by duty cycling.

Fig. 3: Vertex change awareness in different situations.

We will show that a set of informative nodes can be combined to provide a *fresh* or 'real-time' snapshot of a dynamic network and we aim to find the minimum number of informative nodes required to obtain a *fresh snapshot*.

A. Node Fresh Coverage

In an ideal case, a node v has fresh information paths from all vertices, which guarantees that v can access global fresh information. This ideal case however only occurs to some special topologies, e.g., complete bipartite. To measure how much absolute fresh information a node can collect, we introduce the concept of change awareness coverage.

For two vertices, u and v, we say u is covered by v if u has a fresh information path to v. The coverage of v is defined as the set of nodes that have at least one fresh information path to v. The node v is thus an informative node. An informative node can cover multiple vertices and a vertex can be covered by multiple informative nodes. A set of informative nodes is called the *panorama informative node set* if the union of their coverage equals the set of whole nodes. The set of all vertices obviously form a panorama informative node set, as each node can cover itself. We are interested in computing the smallest panorama informative node set. The smallest panorama informative set has potential applications in opportunistic networks, such as the one we will discuss in Section VI

```
Input: $\mathcal{G}$
Output: $\langle v, C(v)\rangle$, $v \in V$.
initialize coverage association sets,
$M \leftarrow \{\langle v, C(v)\rangle\}, v \in V, C(v) \leftarrow \{v\}$
while $\mathcal{E}_{\mathcal{G}}$ is not empty do
    remove the earliest connection $\varepsilon$ : $\langle u, v, t\rangle$ in $\mathcal{E}$
    if only one node (e.g., $u$) in $\varepsilon$ is isolated [2] then
        add $C(u)$ to $C(v)$
        remove $\langle u, C(u)\rangle$ from $M$
    end
    if both $v$ and $u$ are isolated nodes then
        randomly select a node (e.g., $u$)
        add $C(u)$ to $C(v)$
        remove $\langle u, C(u)\rangle$ from $M$
    end
end
return $M$
```

Algorithm 2: The computation for smallest panorama informative node set.

The smallest panorama informative node set can be computed by Algorithm 2 with the time complexity of $O(|\mathcal{E}|)$. The complexity of sorting edges is $O(|\mathcal{E}|\log|\mathcal{E}|)$. For example, Algorithm 2 outputs the coverage association $\{\langle C(or\ D), \{C, D\}\rangle, \langle A(or\ Y), \{A, B, X, Y\}\rangle\}$ for the network in Figure 1. We write such an association in the format of $\langle v, C(v)\rangle$, where v is an informative node and $C(v)$ is the coverage set of v. We name the set of informative nodes computed by Algorithm 2 as $\mathcal{P}$. Next, we would like to state the following lemmas. The proofs are not included in this paper due to space constraints.

Lemma 1. *The coverage sets of nodes in $\mathcal{P}$ forms a partition of V. That is, for any two informative nodes $p_i, p_j \in \mathcal{P}$, $C(p_i) \bigcap C(p_j) = \emptyset$ if $p_i \neq p_j$, and $\bigcup_{p\in\mathcal{P}} C(p) = V$.*

Lemma 2. *$F(\mathcal{G}, u, p, t) = 1$, for $u \in C(p)$, $p \in \mathcal{P}$. That is, a fresh information path exist at t.*

Lemma 3. *Algorithm 2 generates the smallest panorama informative node set.*

Based on these three lemmas, the following theorem holds.

Theorem 4. *A fresh snapshot of a dynamic network at time t can be obtained by assembling information from all informative nodes in $\mathcal{P}$, and $\mathcal{P}$ is such node set with the minimum size.*

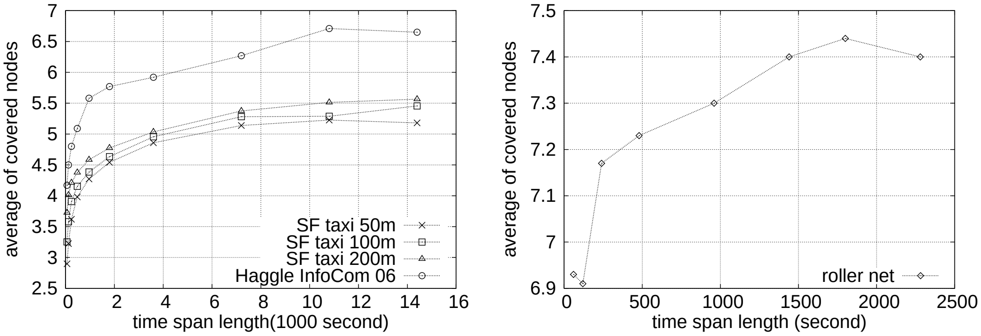
Note that while $\mathcal{P}$ is not unique, the size of different $\mathcal{P}$ is the same.

B. Evaluation using Traces

We study how $\mathcal{P}$ varies with the parameter of time spans and transmission ranges. We divide one day's trace into segments of different length $(t_s, t_s + D)$ with $t_s = 0, 1, \cdots, 86400 - D$ and compute the average size of $\mathcal{P}$ for each D. Instead of looking at the size of $\mathcal{P}$, we measure the average number of nodes that can be covered by an informative node. Average size of $\mathcal{P}$ is simply can be computed by dividing number of nodes by the average coverage. The results are shown in Figure 4 for the SF taxi trace.

The result shows that the average coverage increases slowly when the transmission range grows from 50m to 200m. For a fixed transmission range, as time spans increases, the average

[2] A node is isolated if it has no connections with any other node.



(a) SF taxi and Haggle InfoCom06 traces.

(b) Roller net trace.

Fig. 4: Informative nodes coverage in traces. It shows y: the average number of nodes that an informative node can cover. Thus, $1/y$ is the ratio of uploaders for gaining real-time global snapshot.

coverage increases quickly but stabilizes beyond 3600s. Interestingly, both the two human mobility traces have average coverage that varied between five to seven when the time span is larger than five minutes.

An important indication by the results is that it provides an lower bounder for the number of required nodes to get network states or some types of application data. This has interesting application in data aggregation, as discussed in the following section.

VI. Change Awareness Applications in Data Aggregation

In this section, we apply change awareness to data aggregation. In our evaluation model, each node in opportunistic networks has short range wireless communication, e.g. WiFi, and long range wireless communication, e.g., 3G and LTE. A server regularly collects sensor readings from all nodes. We target to minimize the number of long range communication connections, i.e., uploaders, for collecting application data. The reduced use of uploaders has two main advantages. Firstly, it can reduce the size of data to be uploaded after local aggregation. The amount of reduction depends on concrete applications. Secondly, it can save power. WiFi communication is much more energy efficient than cellular communication [26]. The transmission of large size data is more energy efficient than the transmission of small size data, especially in LTE [27].

A. Types of Sensor Data in Opportunistic Networks

We discuss the aggregation for the following three types of sensor data in opportunistic networks.

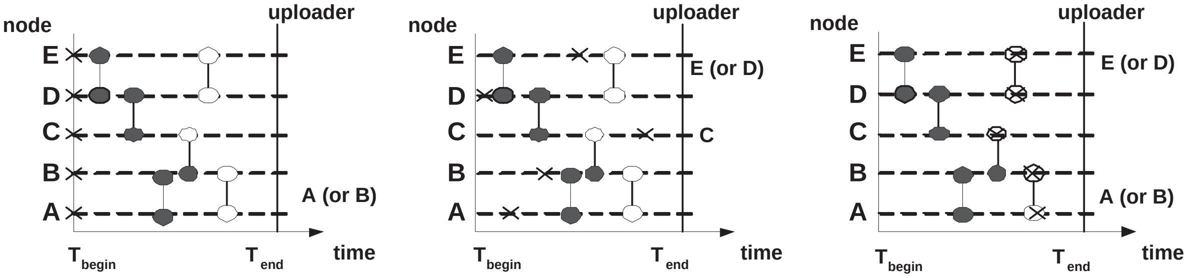
1 **Synchronous Sensor Reading (SSR).** In SSR sensors values on all nodes are read at a synchronized time. One example application is to build pollution or temperature maps for a city. Suppose that all taxis in a city have synchronized time and Global Positioning System (GPS). They hourly sense air temperatures or pollution. The sever can collect all the sensor reading on temperature/pollution with GPS and form a temperature/pollution map with one hour delay. In SSR, if the data is required to be collected at the sensing time, then no data aggregation algorithm based on short range communication can help reduce the number of uploaders. Thus, delays are usually allowed.

2 **Asynchronous Sensor Reading (ASR).** In this type of applications, each sensor logs its sensor readings once it gets a new reading. The log of sensor readings is in an asynchronous manner. One example is the monitor of noise pollution level. The pollution level changes every 20dB. The pollution is at level 0 if the noise decibel is within 0 - 20dB; it is level 1 if the noise decibel falls into 20 - 40dB, and so on. A sensor in a taxi records the sensed noise pollution level only when the new level is different from the previous one. Another example is the collection of taxi states, e.g., does a taxi have passengers on board, or which area a taxi is in. The sensed data is aggregated and uploaded to the server at the end of an aggregation period.

3 **Connection-triggered Sensor Reading (CSR).** In this type of applications, the data changes only when the hosting node has a connection with other nodes. For example in a system where news or advertisements are disseminated over a VANET, the business operators are interested to know the distribution information of advertisement copies to charge customers and to control advertisement dissemination. The distribution of news change only when the nodes have connections.

Figure 5 illustrates the three types of applications. The server periodically collects data on all nodes through uploaders that are a subset of nodes. These uploaders can aggregate data via short range connections. For example, in Figure 5a, node E can transfer its data to D via their connections. And E and D's data can be collected by C via the connections between C and D, and so on. Finally, either only node A or B needs to upload. However, in Figure 5b, D's data can reach C, but E's data cannot reach C. In this case, either E or D has to upload. In Figure 5c, due to the connection patterns, two nodes E (or D) and A (or B) have to upload. The lower bound for the number of uploaders can be exploited in the follow ways.

Under the assumption on the access to connection oracle, the server can compute the reachability of data. Let $t_s(v)$ be



(a) Synchronous Sensor Reading (SSR). (b) Asynchronous Sensor Reading (ASR). (c) Conn.-triggered Sensor Reading (CSR).

Fig. 5: The illustration of different types of applications. A line between two nodes indicates a connection. An empty circle is the last connection of a relative node. A solid circle indicate a connection that is not the last one of the relative node. The crossing symbolize the point when the sensor data generate.

the sensor data generation time on v. If the last departure time from v to u is later (larger) than $t_s(v)$, then v's sensor data can be uploaded by u. We call the set of nodes whose sensor data can be uploaded by u as $K(u)$. With the uploader-ship, $\langle u, K(u) \rangle, u \in \mathcal{N}$ ($\mathcal{N}$ is the set of nodes.), the minimization of uploader set equals the minimum Set Covering Problem (SCP), which is NP-hard. Greedy algorithms can be exploited with proved approximation ratio, as in our benchmark solution (COMPLETE-Greedy) that we will discuss later.

For these three types of applications, we give a summary on minimum number of uploaders in Table III.

App	Uploader No.	Note
SSR	$[0, \lvert\mathcal{P}\rvert]$	depending on reachability of data
ASR	$[0, \lvert\mathcal{N}\rvert]$	the node whose data are generated after its last connection need to upload the data itself.
CSR	$= \lvert\mathcal{P}\rvert$	as shown by Theorem 4.

TABLE III: The lower bound of the number of uploaders. ($\mathcal{N}$ is the total number of nodes)

Please note that only CSR tightly fits the definition of change awareness. However, we also show how change awareness based solutions help the data aggregation for SSR, ASR, and CSR.

B. The Change Awareness Based Data Aggregation Algorithm (CA-Base, CA-Plus)

In change awareness based solution, for each data aggregation period, the server needs to collect each nodes' last connection to compute informative nodes by Algorithm 2. Because the last connection information is of constant small data size (several bytes), it can be transferred using the control or low bit rate channel of the cellular network, e.g., Short Message Service (SMS), which is power efficient. Once the informative nodes are identified, they upload collected sensor data via the cellular data channel.

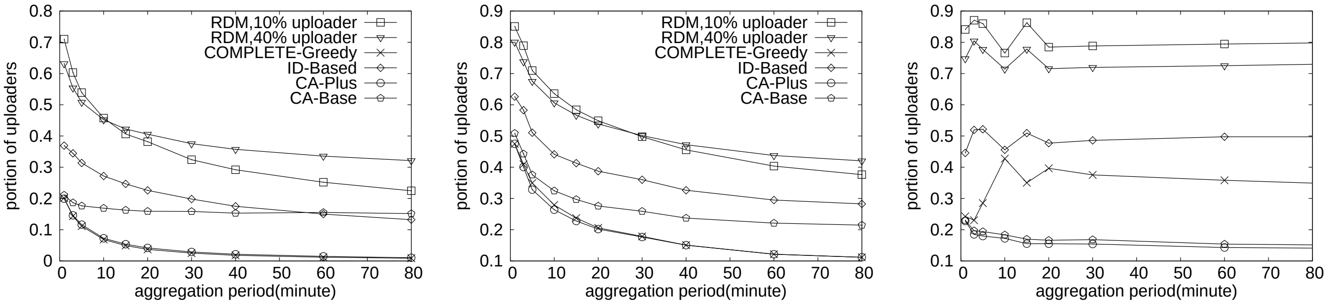
We tried two different change awareness based algorithms: CA-Base and CA-Plus. In CA-Base, each node uploads its last connection information to the server, and then the server can compute a set of coverage associations $\langle v, C(v) \rangle$ by Algorithm 2. An informative node uploads the sensor readings of nodes under its coverage. By this method, the server can collect all data that are generated before their hosting nodes' last connection. In ASR, the nodes whose sensor readings are generated after their last connection upload their own sensor reading by themselves. Actually, no algorithm can help aggregate these data as no short range connection occurs to the hosting node after the generation of the data.

The second approach (CA-Plus) enhances CA-Base by controlling different uploaders. Each node also needs to upload its last connection to the server. Firstly, when a node generates data after its last connection (as in ASR), the data is only knowable to itself. Such nodes upload their own sensor data and the collected data to the server. Then, the server computes the informative nodes and their coverage, and greedily selects the next uploader from informative nodes that has the most uncollected sensor data. This step is repeated until the server has collected data from all nodes.

C. Data Aggregation Algorithms for Comparison

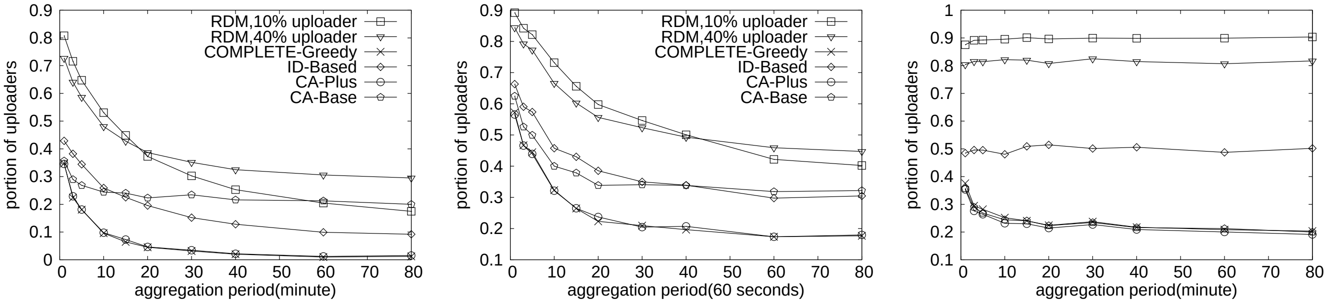
1) The Benchmark Algorithm (COMPLETE-Greedy): The benchmark is gained in the case when a server collects all short range connection information and thus can compute the complete reachability of all information flows. The benchmark algorithm COMPLETE-Greedy uses the greedy algorithm [28] to select the next uploader as the one having the most information to upload. COMPLETE-Greedy can compute the minimum number of uploaders by constrained approximation ratios, but it can cause high overhead as all nodes need to upload all its connection information to the server.

2) Random algorithm and ID based algorithm: Beside the benchmark algorithm, we also compare our solution with two other algorithms: RDM and ID-Based. RDM adapts from LEACH [13]. RDM randomly selects $x\%$ of nodes as fixed uploaders. A non-uploader forwards its data to an encountered uploader. At the aggregation time point, both the fixed uploaders and non-uploaders that have never encountered any fixed uploader upload data to the server. We compare with RDM with different percentage of fixed uploaders. ID-Based chooses uploaders by nodes' IDs. When two nodes encounter, the one with smaller identity dump collected sensor data to the one with larger identity. At aggregation time, all the nodes having data are uploaders.



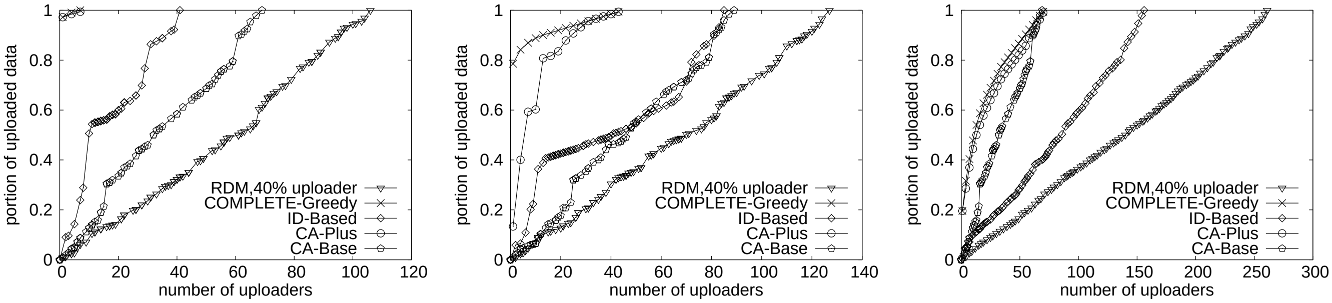
(a) Synchronous Sensor Reading. (b) Asynchronous Sensor Reading. (c) Connection-triggered Sensing Reading.

Fig. 6: Data aggregation uploader ratio on ShangHai taxi trace



(a) Synchronous Sensor Reading. (b) Asynchronous Sensor Reading. (c) Connection-triggered Sensing Reading.

Fig. 7: Data aggregation uploader ratio on San Francisco taxi trace



(a) Synchronous Sensor Reading. (b) Asynchronous Sensor Reading. (c) Connection-triggered sensor Reading.

Fig. 8: The insight into the uploading process on San Francisco taxi trace

D. Evaluation Settings and Results

We present evaluation results on two vehicle traces, Shanghai taxi trace and SF taxi trace, where Shanghai Taxi traces has ten times more taxies than SF taxi trace. The results are shown in Figure 6 and 7. The ratio of uploaders ranges from 0.003(for SSR, COMPLETE-Greedy and CA-Plus) to 0.9 (for CSR, RDM). For CSR, CA-Base can achieve similar performance as CA-Plus and COMPLETE-Greedy. CA-Plus and COMPLETE-Greedy achieve similar results for SSR and ASR. In most settings, the three algorithms require significantly less uploaders than RDM and ID-Based algorithms.

These results further indicate the following points. First, for CSR, CA-Base provides a lower bound for the number of uploaders required for fresh snapshot, as shown by Theorem 4. Second, in many cases in SSR and ASR, not all informative nodes need to upload data. The number of nodes required to upload all data depends on the data reachability constrained by intermittent connections. This can be computed based on connections, as is done in COMPLETE-Greedy.

The results on more traces are listed in Table IV. These results show that CA-Plus can gain similar performance as COMPLETE-Greedy, which are the best among the five tested algorithms. However, CA-Pluse can save much communication overhead ranging from six times to 15 times.

We also investigate the intermediate states of data collection process, i.e., how much information is collected versus how many uploaders are utilized. This is meaningful to applications where only a portion of data needs to be collected. The result in Figure 8 shows that for a fixed number of uploaders, CA-Plus and COMPLETE-Greedy can collect much more data than other algorithms. The CA-Plus and COMPLETE-Greedy collect similar size of data for SSR and CSR. This implies that they have similar speed in information collection. For ASR, the COMPLETE-Greedy outperforms CA-Plus for the

Trace	Algorithms	SSR	ASR	CSR	Overhead
SF taxi (50m)	RDM	0.531	0.677	0.849	0
	CA-Base	0.222	0.360	0.224	269
	ID-Based	0.256	0.430	0.492	0
	CA-Plus	0.080	0.272	0.211	269
	COMPLETE-Greedy	0.075	0.267	0.229	5156
ShangHai Taxi	RDM	0.484	0.615	0.823	0
	ID-Based	0.268	0.428	0.538	0
	CA-Base	0.160	0.282	0.160	4037
	CA-Plus	0.053	0.216	0.147	4037
	COMPLETE-Greedy	0.048	0.224	0.230	266550
Haggle IC06	RDM	0.659	0.788	0.867	0
	ID-Based	0.277	0.558	0.529	0
	CA-Base	0.138	0.476	0.138	60
	CA-Plus	0.109	0.422	0.122	60
	COMPLETE-Greedy	0.116	0.430	0.190	650
Seattle Bus	RDM	0.682	0.797	0.807	0
	ID-Based	0.269	0.559	0.506	0
	CA-Base	0.241	0.454	0.241	921
	CA-Plus	0.139	0.383	0.234	921
	COMPLETE-Greedy	0.133	0.389	0.280	5870
Roller Net	RDM	0.356	0.437	0.813	0
	ID-Based	0.187	0.200	0.471	0
	CA-Base	0.179	0.212	0.179	61
	CA-Plus	0.027	0.080	0.172	61
	COMPLETE-Greedy	0.023	0.093	0.249	5870
RWP	RDM	0.515	0.684	0.817	0
	ID-Based	0.23	0.442	0.478	0
	CA-Base	0.248	0.376	0.248	100
	CA-Plus	0.03	0.256	0.246	100
	COMPLETE-Greedy	0.022	0.268	0.26	1560
RPGM	RDM	0.471	0.624	0.827	0
	ID-Based	0.216	0.365	0.506	0
	CA-Base	0.220	0.316	0.220	100
	CA-Plus	0.094	0.228	0.202	100
	COMPLETE-Greedy	0.085	0.232	0.257	2860

TABLE IV: The uploader ratio ((No. of uploaders)/(No. of total nodes)) on different traces. We take aggregation period as 1800s. For RDM, we set 40% of nodes as prefixed uploaders. The overhead is computed as the number of contacts required to be uploaded for computing uploaders.

first 22 uploaders. For example, with only 10 uploaders, the COMPLETE-Greedy can collect 85% of data, while CA-Plus only can collect 60% of data. When the number of uploaders are larger than 22, the two algorithms gain similar performance. The reason is that without complete connection information, CA-Plus is unable to always select the best uploader under the asynchronous scenario.

VII. Conclusion

This work proposes and investigates the novel concept of information freshness in dynamic networks. We have presented the interpretation and measurement of information freshness, broadly examine the factors affecting information freshness on real world traces, and evaluate its applications by proposing and testing solutions to address practical dynamic network problems. Results from our investigation attest that research on information freshness can shed light on analyzing dynamic network and therein address practical problems. As the concept of information freshness and change awareness have attracted little attention, we believe that our work presents a novel angel to deepen the understanding on dynamic network.

References

[1] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *WDTN '05*. ACM, 2005.

[2] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: Applications and related technical issues," *Communications Surveys & Tutorials, IEEE*, 2008.

[3] B. Xuan, A. Ferreira, and A. Jarry, "Computing shortest, fastest, and foremost journeys in dynamic networks," *International Journal of Foundations of Computer Science*, 2003.

[4] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, "Time-varying graphs and dynamic networks," *Ad-hoc, Mobile, and Wireless Networks*, 2011.

[5] J. Whitbeck, M. Dias de Amorim, V. Conan, and J. Guillaume, "Temporal reachability graphs," in *MobiCom'12*. ACM.

[6] K. A. Berman, "Vulnerability of scheduled networks and a generalization of menger's theorem," *Networks*, 1996.

[7] A. Casteigts, P. Flocchini, B. Mans, and N. Santoro, "Deterministic computations in time-varying graphs: Broadcasting under unstructured mobility," *Theoretical Computer Science*, 2010.

[8] G. Kossinets, J. Kleinberg, and D. Watts, "The structure of information pathways in a social communication network," in *SIGKDD'08*. ACM.

[9] A. Casteigts, P. Flocchini, B. Mans, and N. Santoro, "Measuring temporal lags in delay-tolerant networks," in *IPDPS'11*. IEEE.

[10] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *IMC '07*.

[11] K. M. Chandy and L. Lamport, "Distributed snapshots: determining global states of distributed systems," *ACM Trans. Comput. Syst.*, 1985.

[12] R. Rajagopalan and P. Varshney, "Data-aggregation techniques in sensor networks: a survey," *Communications Surveys Tutorials, IEEE*, 2006.

[13] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, MIT, 2000.

[14] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *INFOCOM'04*. IEEE.

[15] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2004.

[16] A. Cornejo, S. Gilbert, and C. Newport, "Aggregation in dynamic networks," in *PODC'12*.

[17] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *IEEE INFOCOM'09*.

[18] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/infocom2006."

[19] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering," in *COMSNETS'09*, 2009.

[20] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "CRAWDAD data set rice/"ad hoc city" (v. 2003-09-11)."

[21] Wireless and S. networks Lab (WnSN). Shanghai Jiao Tong University, "Shanghai taxi trace."

[22] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *MSWiM '99*.

[23] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The diameter of opportunistic mobile networks," in *CoNEXT'07*. ACM.

[24] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *JACM*, 1987.

[25] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *MobiCom'12*. ACM.

[26] G. Ananthanarayanan and I. Stoica, "Blue-fi: enhancing wi-fi performance using bluetooth signals," in *MobiSys'09*.

[27] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *MobiSys'12*.

[28] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, 1979.